

'Coulweb' Work Experience

12th-23rd November 2018

Daniel Wilkins



Contents

| | |
|--|-----|
| Introduction | 3 |
| Day 1 – Front-end Page Edits | 4 |
| The Sale Banner..... | 5 |
| The Description Section of the Sale Page | 6 |
| The Smaller Banner of the Sale Page | 8 |
| Making the Previously Shown Aspects of the Sale Page Responsive..... | 10 |
| Undertaking Edits for the Home Page | 30 |
| Day 2 – WordPress Tasks | 38 |
| Introduction | 38 |
| Undertaking the Task | 39 |
| Days 3, 4 and 5 – Creating Account and Login Forms | 63 |
| Introduction | 63 |
| Creating the 'Login' Form for this Task | 64 |
| Establishing the Table for the Database to Contain User Details..... | 67 |
| Creating the 'Create an Account' Form..... | 69 |
| Allowing for the 'Create an Account' Form to Function and add Data to the Database | 71 |
| Allowing for the 'Login' form to Function Properly..... | 72 |
| Checking Entered Details into the Form | 72 |
| Creating a Persistent Login Function to Allow for Logging Out of the System..... | 74 |
| Navigating the User to Another Page from the Login Function, Displaying their Data to them | 75 |
| Allowing Users to Logout of the Account Summary Page | 87 |
| Password Reset Function | 92 |
| Summary of the Tasks for this Project..... | 115 |
| Day 6 – Creating a Landing Page Utilising HTML and CSS..... | 116 |
| Introduction | 116 |
| Establishing the files and the Page | 116 |
| Creating the Page | 117 |
| The Banner Section of the Page | 117 |
| The Main Section of the Page | 121 |
| The Footer Section of the Landing Page | 135 |
| The Final Outcome | 143 |
| The Final HTML Code..... | 143 |
| The Final CSS Code | 144 |
| The Final Outcome on the Web Page (General and Whilst Hovering over Sections)..... | 148 |

'Cullweb' Work Experience Documentation – Daniel Wilkins

General Introduced Aspects..... 150
 Introduction 150
Conclusion 152
Bibliography/Reference List/Acknowledgements for the Work Experience 153

Introduction

As explained in the document for the work experience undertaken at 'The ICE Agency', I contacted various companies regarding any possible work experience/placement opportunities in the area of website development (both front-end and back-end aspects), although I was largely unsuccessful. However, a company in Basingstoke called 'Coullweb' were very kindly able to offer me two weeks of work experience of which I am very appreciative of. This was undertaken between the 12th and 23rd November 2018, excluding the Thursday of both weeks as I had to be present in my lectures on that day. I undertook the work experience on the days of Monday, Tuesday and Friday. Throughout both weeks I was able to learn of new areas through many set tasks and I am very grateful to be provided with this experience as I am sure it will improve my skillset. This document shows the areas undertaken during the two weeks with screenshots and annotations provided.

Day 1 – Front-end Page Edits

During my first day at Coullweb, I was set the tasks of editing some aspects of a current website called 'Davidson' from the front-end aspect. To begin, I read the brief typed up by the developer at 'Coullweb' which detailed each step of what it was I needed to do. These steps can be viewed below:

Davidson sale page

1.

Change sale page hero image

Add background colour

Add image to sale page hero image

```
<?php echo $this->getSkinUrl('images/'); ?>
```

Davidson banner hex - A70E2A

2.

Add description section

Use our container

```
<div class="container">
```

```
<div class="eight columns">
```

3. *Add ex display banner*

Add background colour

Add image to ex display banner

Text to be included:

As our Christmas gift to you, we are offering a special 5% discount off ?on all orders this Winter.

For one month only, any new orders placed between the ?14th November and the 12th December are viable to ?receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:

WINTERORDERS!

The Sale Banner

After reading through the brief set, I then began with changing the sale page banner section. The desired outcome on the page can be viewed below:

The Desired Outcome for this part of the Page – Invision Design



After being helped with this section of the page, the following outcome I produced can be seen below:

The HTML/PHP Code

```
<div class="saleBanner">  
  
```

As is evident here, I simply changed the 'URL' of the image to the one that was provided and this is highlighted above.

The CSS Code

```
.sale .saleBanner {  
  width: 100%;  
  background-color: #A70E2A;  
  padding: 120px 0px;  
}  
  
.sale .saleBanner img {  
  width: 501px;  
  display: block;  
  margin: auto;  
}
```

As can be seen above, the image was the text section which read 'SALE ENJOY 5% OFF ON ALL ORDERS IN OUR WINTER SALE' and the '.sale' and '.saleBanner' was included to style the background colour of this part of the page, hence the 'HEX' value '#A70E2A'. With regards to the '.sale' and '.saleBanner' styles, the 'padding' was placed here to centre the image inside vertically. The outcome of both the 'CSS' code above and the previous 'HTML'/'PHP' code can be viewed on the following page.

The Outcome of these Changes on the Web Page – This now Reflected the Invision Design file



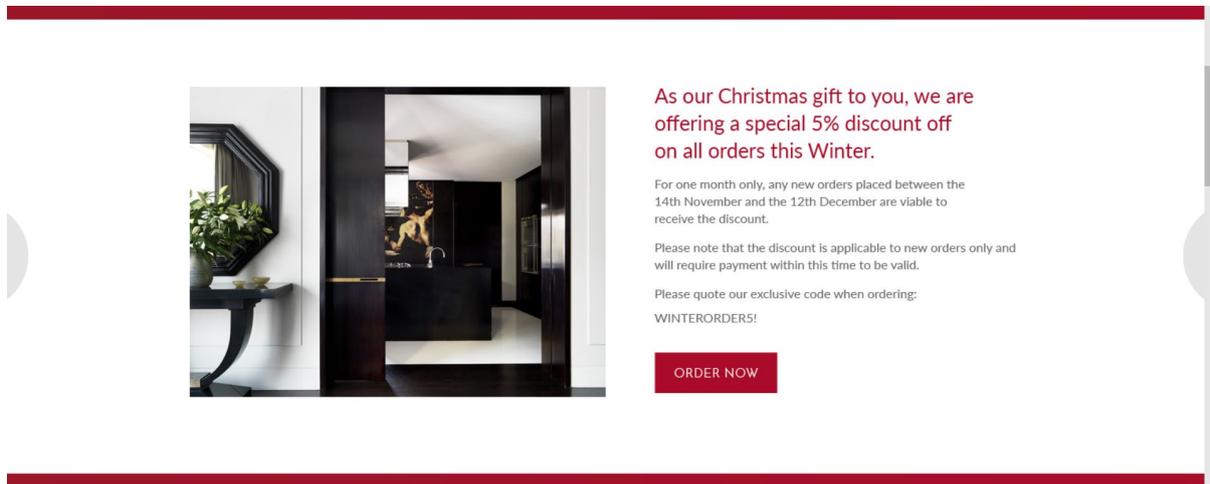
The Image Section on the Page

The Background/Container Section of the page for this section

The Description Section of the Sale Page

The next task was to alter the current description section of the page to show the desired outcome displayed below:

The Invision Design File



Therefore, to create this outcome, I was helped again and I ensured I used the container that 'Coullweb' used and created the following 'HTML' code:

The Final 'HTML'/'PHP' Code for this Section of the Page

```
<div class="container saleDescription">
  <div class="eight columns image offset-by-one">
    
  </div>
  <div class="six columns content">
    <p class="heading">As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.</p>

    <p>For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.</p>

    <p>Please note that the discount is applicable to new orders only and will require payment within this time to be valid.</p>

    <p>Please quote our exclusive code when ordering:<br>WINTERORDERS5!</p>

    <a href="#" class="order_button">ORDER NOW</a>
  </div>
</div>
```

As can be seen above, for the 'eight columns' section, the provided image was placed here and the provided text was placed inside the 'six columns' section as well as the button/link for 'ORDER NOW'. To style this, the following 'CSS' code was added which can be viewed below:

```
.saleDescription {
  padding: 80px 0px;
}

.saleDescription .image {
}

.saleDescription .content {
}

.saleDescription .content p {
  font-weight: 400;
}

.order_button {
  margin-top: 10px;
}

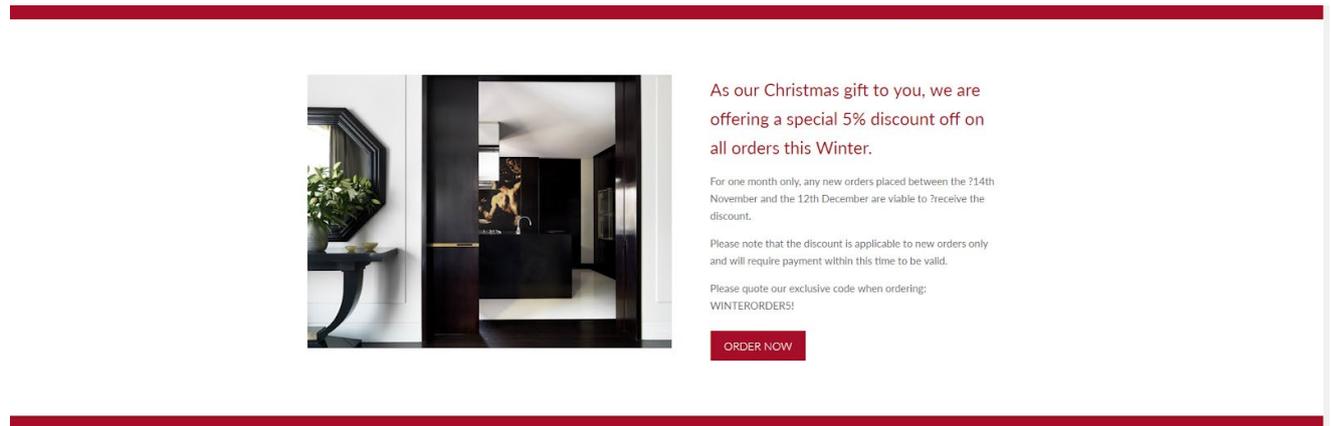
.saleDescription .heading {
  font-size: 25px;
  color: #A70E2A;
  font-weight: 700;
}

.saleDescription a {
  background-color: #a70e2a;
  color: #fff;
  padding: 10px 20px;
  display: inline-block;
  font-weight: 100;
}
```

The '.saleDescription' had 'padding' assigned to it to vertically centre each of the items inside which in this case would be both the image and the content. For the '.saleDescription .content p', 'font-weight' was added to make the text inside bolder as currently this was lighter than required. The '.order_button' 'class' related to the button/link where 'margin-top' was added to space it further from the text above itself and for both the '.saleDescription .heading' and '.saleDescription a', styles were assigned to them to style both the visual appearance of the heading section and the link which related to the button in the content section.

The outcome of these changes can be viewed below:

The Outcome on the Web Page – This now Represented the Invision Design File



The Smaller Banner of the Sale Page

After completing both of the previous sections, I was then required to change the smaller banner on the page to the following desired outcome:

The Required Outcome on the Page – InVision Design file



After being taken through this stage, I was then helped to create the following 'HTML' and 'PHP' code for this section of the web page:

```
<div class="saleBanner small">  
    
</div>
```

As is evident above a 'class' name above called 'small' was added and the image was added to the 'URL' section which are both highlighted above. The following 'CSS' code was then applied to style this section:

```
.sale .saleBanner.small {  
  width: 100%;  
  background-color: #A70E2A;  
  padding: 80px 0px;  
}  
  
.sale .saleBanner.small img {  
  width: 360px;  
  display: block;  
  margin: auto;  
}
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

As is evident on the previous page a width of '100%' was assigned to the '.small' 'class' along with the background colour and 'padding' to vertically centre the image inside. For the image itself, a width of '360px' was assigned to it to make it smaller along with displaying it as a 'block' element with a 'margin' of 'auto' to centre the image on the page.

The following outcome of the previous two changes can be seen below:

This now Created the Desired Outcome on the Web Page



Making the Previously Shown Aspects of the Sale Page Responsive

As the previous aspects had now been completed, I was then set the task of making these aspects responsive through different provided 'media queries'. Again, I was provided with material to follow with regards to building these aspects. The first aspect to implement was for the 'media query' of 1168px. The desired outcome can be seen below:

The Required Outcome on the Page for Screen Sizes of 1168px or Less

DAVIDSON Trade Log In GBP

EVOKE FURNITURE FINISHES SERVICE ABOUT US LOOKBOOK CONTACT SALE

SALE

ENJOY 5% OFF
ON ALL ORDERS IN OUR WINTER SALE

As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

SALE

Ex-Display

We strive to keep our showroom looking fresh at all times introducing new pieces and selling the old.

This page is dedicated to clients looking for something in a hurry. All pieces are from the showroom floor and available on short or immediate lead times. Requirements should be discussed with a DAVIDSON sales representative.

For further information please contact a member of our sales team.

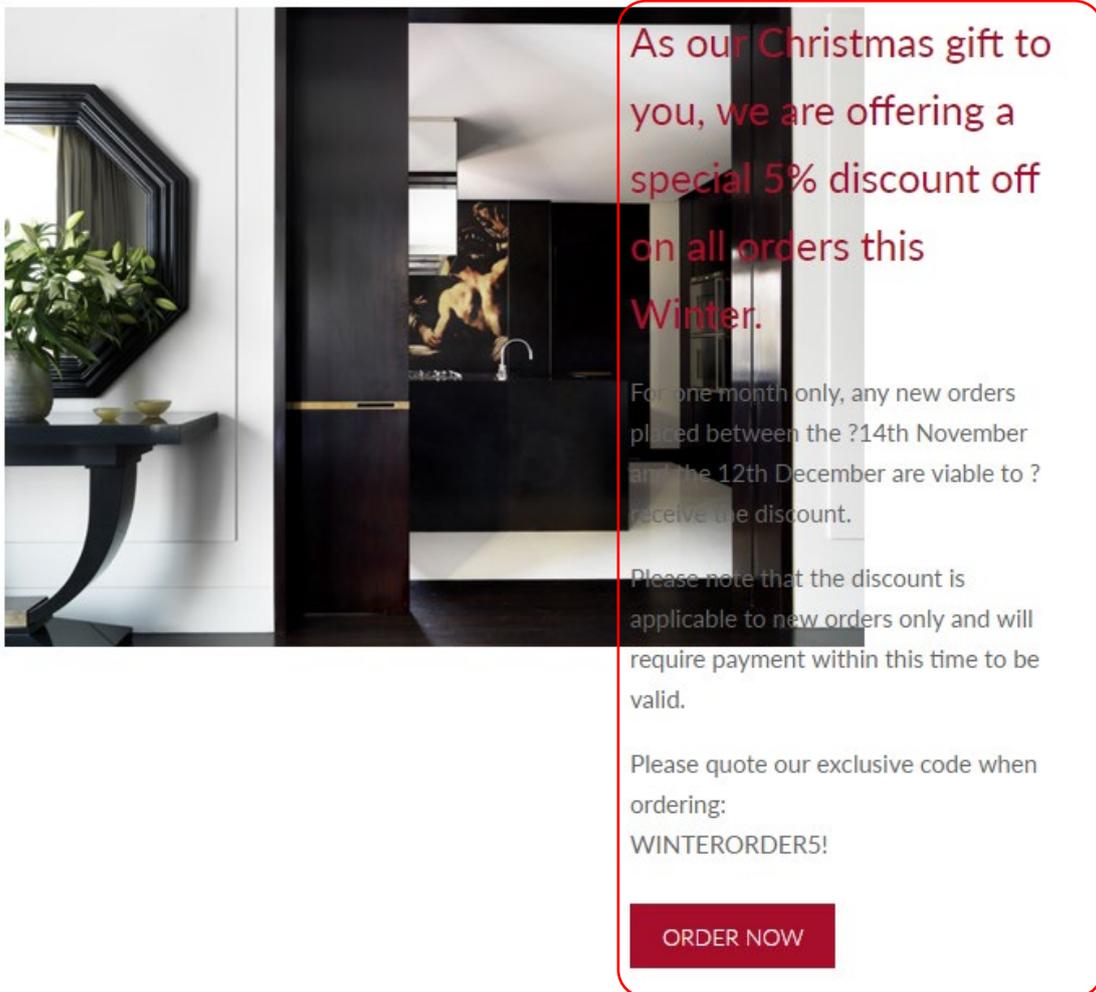
THE MONTAGU

To begin, I started to change the second highlighted aspect above but encountered a problem where setting the width of the 'image' section wouldn't work and cause the following problem instead:

Changing the Width of the Image Section in the CSS Code

```
.saleDescription |.container .image {  
  width: 100%;  
}
```

The Issue this was Causing



After seeking advice, I was then told that there didn't need to be a space between the 'saleDescription' and '.container' sections in the 'CSS' file. Therefore, I removed this space and this now worked as will be evident below:

Removing the Space between the 'class' names within the CSS file

```
.saleDescription.container .image {  
  width: 100%;  
}
```

The Outcome on the Web Page (This now Solved the Issue)



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

After being informed I could assign more 'classes' for the content section in the 'CSS' file through the use of a comma, I then added the following to add the same styles to the content section and this styled the elements in a column format on the page:

Adding the Styles for the '.content' 'class' in the CSS file through Inserting a Comma

```
.saleDescription.container .image, .saleDescription.container .content {  
  width: 100%;  
}
```

The Outcome of this on the Page (The Sections were now Displaying in Column Format)



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

However, at this stage, myself and the developer helping me realised that to style each individual section, that this would have to be completed through separate styles and so therefore changed the 'CSS' code to the following shown below:

The Changes to the CSS file

```
.saleDescription.container .image {  
  width: 100%;  
  text-align: center;  
  padding: 0px;  
}  
  
.saleDescription.container .content {  
  width: 100%;  
  margin-top: 70px;  
}
```

As can be seen above, 'text-align: center;' had to be applied to the '.image' 'class' so that it would be positioned to the centre of the page and 'margin-top' had to be applied to the '.content' class so that the text would be positioned further apart from the image on the page. The outcome of these changes can be viewed on the following page.

The Outcome on the Web Page (This was now Beginning to Represent the Required Outcome for this Section of the Page)



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

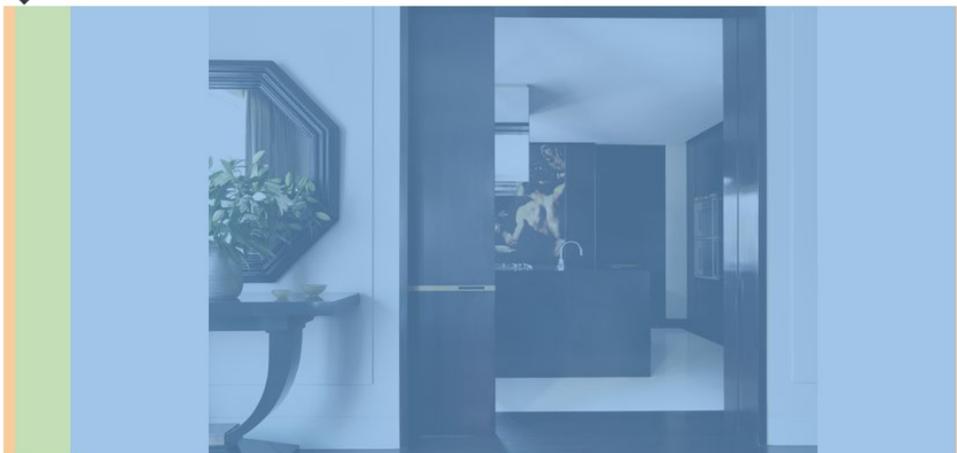
Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

As the image wasn't completely central, I then used the 'inspect' tool on 'Google Chrome' to help identify why this was occurring. I thought at first it was caused by the 'padding' displayed by the first '.container .offset-by-one' 'classes'. However, I soon realised that it was the second version of this further down on the 'inspect' tool:

Using the 'Inspect tool' to Highlight the Issue ('padding')

div.eight.columns.image.offset-by-one | 816 x 396



The 'padding' Affecting this part of the Page

```
@media only screen and (max-width: 1168px)
and (min-width: 768px)
.container .offset-by-one {      grid.css:95
  padding-left: 48px;
}

@media only screen and (max-width: 1168px)
and (min-width: 768px)
.container .column, .container  grid.css:67
.columns {
  margin-left: 10px;
  margin-right: 10px;
}

.container .offset-by-one {      grid.css:41
  padding-left: 53px;
}
```

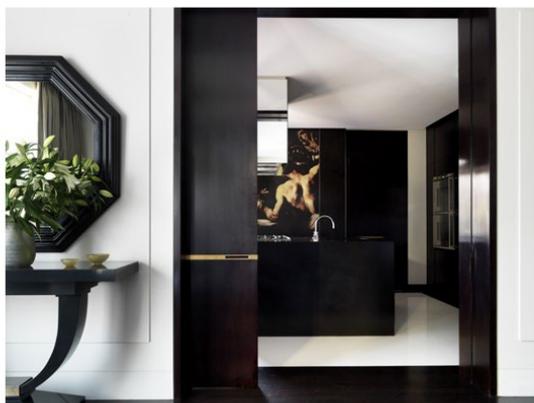
I was then advised by the developer that it would be best to assign 'padding: 0px;' to the '.image' 'class' to ensure that all padding would be removed:

Adding 'padding: 0px;' to the '.image' Class in the CSS file

```
.saleDescription.container .image {
  width: 100%;
  text-align: center;
  padding: 0px;
}
```

This now created the following and final outcome on the page for this section of the page for this breakpoint of 1168px:

The Final Outcome for this Section on the Page



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

This now meant that the final page outcome for this breakpoint appeared as follows:

The screenshot shows a website homepage with a dark red header. The navigation menu includes 'EVOKE', 'FURNITURE', 'FINISHES', 'SERVICE', 'DAVIDSON', 'ABOUT US', 'LOOKBOOK', 'CONTACT', and 'SALE'. A 'Trade Log In' link is also visible. The main content area features a large red banner with the word 'SALE' in white serif font, followed by 'ENJOY 5% OFF ON ALL ORDERS IN OUR WINTER SALE'. Below this is a white section with a photograph of a modern interior on the left and promotional text on the right. The text includes a 5% discount offer, a date range from November 14th to December 12th, and a note that the discount is for new orders only. A red 'ORDER NOW' button is positioned below the text. The bottom of the page features another red banner with 'SALE' in white serif font and 'Ex-Display' in a smaller, italicized font.

EVOKE FURNITURE FINISHES SERVICE DAVIDSON ABOUT US LOOKBOOK CONTACT SALE Trade Log In

SALE

ENJOY 5% OFF
ON ALL ORDERS IN OUR WINTER SALE



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering: WINTERORDERS!

ORDER NOW

SALE

— Ex-Display —

The next breakpoint was for devices with a screen size of 768px or less and the desired outcome can be viewed below:



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 21st November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

[ORDER NOW](#)



We strive to keep our showroom looking fresh at all times introducing new pieces and selling the old.

This page is dedicated to clients looking for something in a hurry. All pieces are from the showroom floor and available on short or immediate lead times. Requirements should be discussed with a DAVIDSON sales representative.

For further information please contact a member of our sales

'Coulweb' Work Experience Documentation – Daniel Wilkins

The first problem experienced on this breakpoint was the fact that on the description section of the page, the image wasn't completely central as will be seen below:



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

After seeking advice, I was then informed that this was because the image wasn't being contained so therefore adding 'max-width: 100%;' would work otherwise keeping the width at '100%' would inherit another value in the stylesheet which was a set width, therefore not appearing centrally:

Adding 'max-width: 100%;' to the '.image' 'class' in the Stylesheet

```
.saleDescription.container .image img {  
  max-width: 100%;  
}
```

This then resolved the issue as will be seen below:



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

After this had been solved, I then started to change the banner section at the top of the page by changing the width of the 'img' element in the 'CSS' file to '400px' of which didn't work as the width had been reduced but the height of the container had not, causing the image to not be vertically centred within the banner:

The Current Outcome on the Web Page (The Banner Image was too Large)



Changing the width of the 'img' Element in the CSS file to '400px'

```
.sale .saleBanner img {  
  width: 400px;  
}
```

The Outcome of this on the Web Page (This Created the Issue Explained before)



I then removed the height from the banner which resolved the issue and centred the image again vertically:

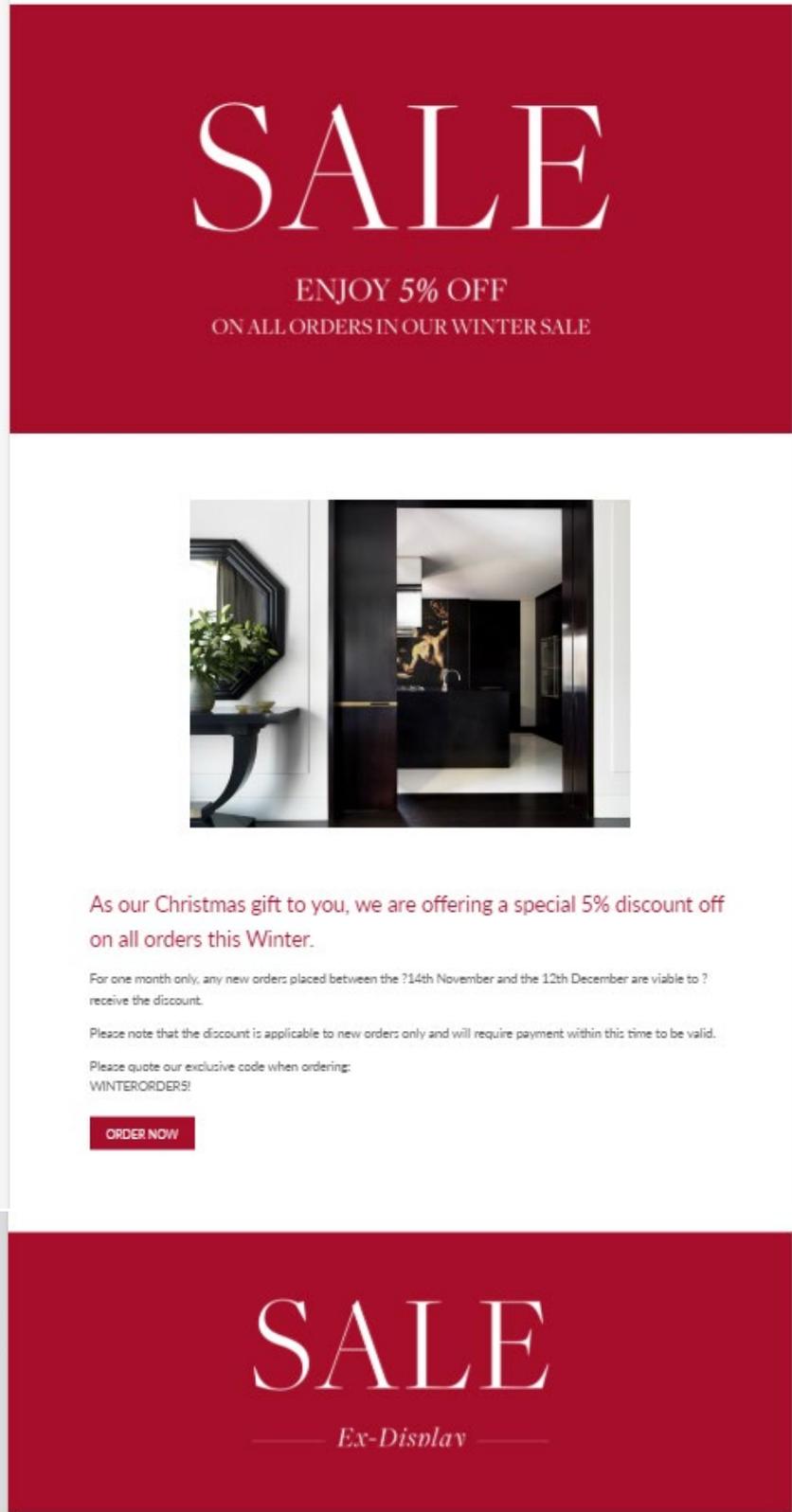
Removing the Height from the 'class' in the CSS file

```
.sale .saleBanner {  
  height: 330px;  
}
```

The Outcome of this on the Web Page (This now Resolved the Issue)



After making the previous changes to the page, the page now appeared as seen below whilst viewing it on a screen resolution of 768px or less:



SALE

ENJOY 5% OFF
ON ALL ORDERS IN OUR WINTER SALE



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

[ORDER NOW](#)

SALE
Ex-Display

After completing the previous work for the other breakpoints/'media queries' of the page, the final one to apply styling to for the sections was for devices with a screen resolution of 468px or less. The required outcome can be viewed below:



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

[ORDER NOW](#)



We strive to keep our showroom looking fresh at all times introducing new pieces and selling the old.

This page is dedicated to clients looking for something in a hurry. All pieces are from the showroom floor and available on short or immediate lead times. Requirements should be discussed with a DAVIDSON sales representative.

For further information please contact a member of our sales team.

To begin, I first of all decided to change the description section of the page which at the current time appeared as the following:



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDER5!

ORDER NOW

I then added the following 'CSS' code to alter the styling of this section:

Changing the Styling of this Section in the CSS file

```
.saleDescription.container .image {  
  width: 100%;  
  text-align: center;  
  padding: 0px;  
}  
  
.saleDescription.container .content {  
  width: 100%;  
  margin-top: 50px;  
}
```

As is evident above, I reduced the 'margin-top' as currently the space between the text and image was too large. At first, I also tried adding 'padding' as I thought the elements were quite close to the edges of the page. However, I was then informed that this was pushing the container out of place and that I had the browser window too small which meant that this was causing the page to appear as if the elements were too close to the edges when in fact they weren't.

The outcome of these changes then created the required appearance of this section of which can be viewed below:



'Coulweb' Work Experience Documentation – Daniel Wilkins

After this, I then started to work on creating the outcome for the banner section on the page, applying 'max-width: 100%;' to the page as I thought I had been advised to do this and also adding 'padding' to push the image inside the banner inwards in order to add space around the image:

Adding the Styles explained above in the CSS file

```
.sale .saleBanner img {  
  max-width: 100% !important;  
  padding: 30px;  
}
```

However, this then caused the following problem where the banner then became formatted incorrectly and worsened the situation:

The Outcome on the Web Page (This didn't Create the Required Outcome)



I was then advised that there had to be a 'width' instead of 'max-width' as this was causing the width of the image to be applied from where it had been styled before in the 'CSS' file:

Changing the CSS file as Advised

```
.sale .saleBanner img {  
  width: 100% !important;  
  padding: 0px 30px;  
  box-sizing: border-box;  
}
```

The Outcome of this on the Web Page (The Banner Section was now Complete for this Part)



This also then helped to style the other smaller banner section as well on the page:

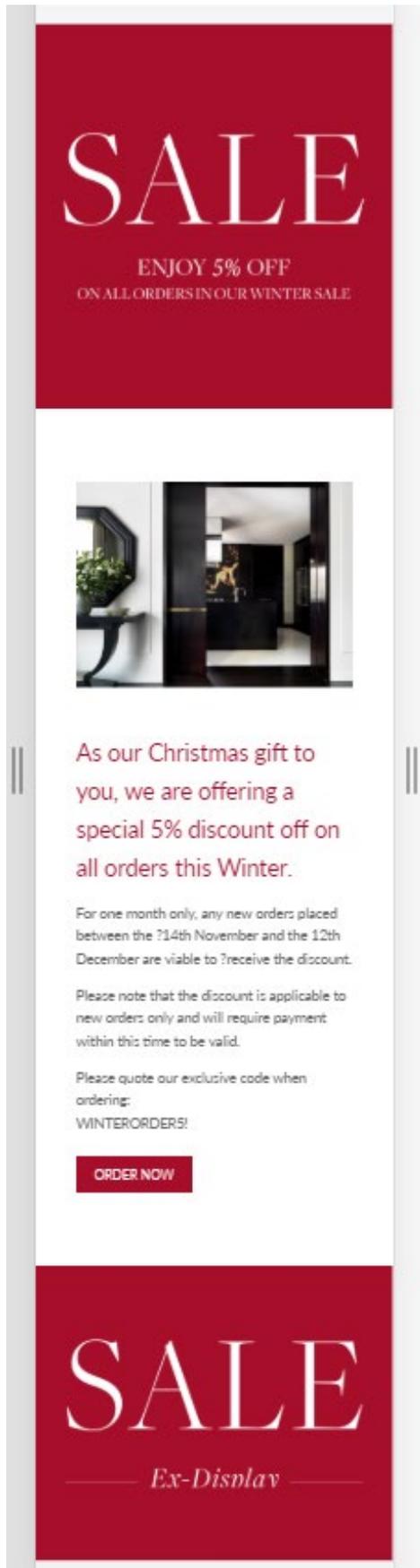
The other Banner Section – Before



The other Banner Section – After

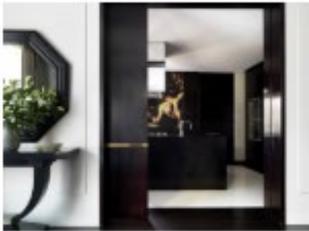


After making the previous changes, the page now appeared as seen below on screen resolutions of 468px or less:



SALE

ENJOY 5% OFF
ON ALL ORDERS IN OUR WINTER SALE



As our Christmas gift to you, we are offering a special 5% discount off on all orders this Winter.

For one month only, any new orders placed between the 14th November and the 12th December are viable to receive the discount.

Please note that the discount is applicable to new orders only and will require payment within this time to be valid.

Please quote our exclusive code when ordering:
WINTERORDERS!

ORDER NOW

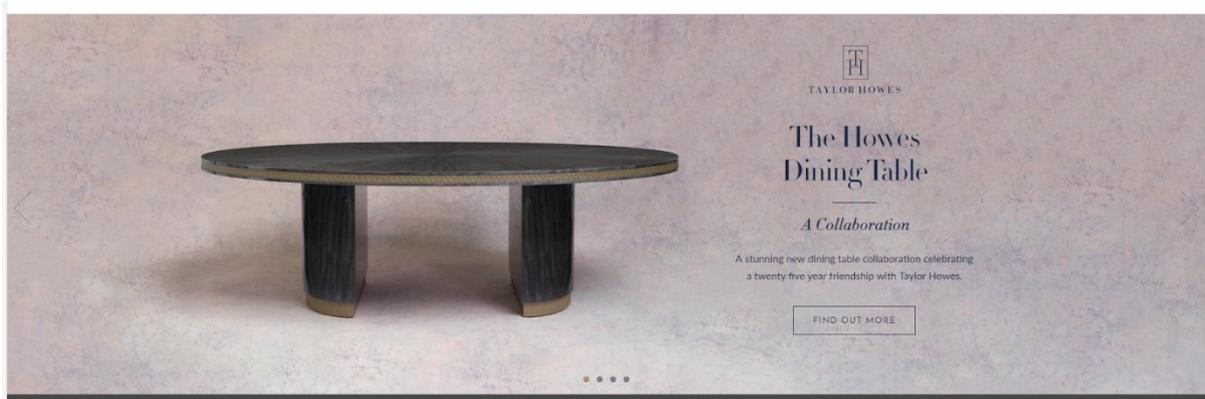
SALE

Ex-Dislay

Undertaking Edits for the Home Page

After completing the work for the sales page, I was then set the task of changing different aspects of the home page. The first task was to edit the banner section so that an extra slide could be added for the sale image section. The required outcome can be seen below:

The Current Banner Section (Without the Sale Section)



The Required Outcome (Adding the Sale Section to the Banner)



The current 'HTML/PHP' code for this section can be viewed below:

The Current HTML/PHP Code for this Aspect (Relating to the other Slides in the Banner)

```
<div class="flexslider-hero">
  <ul class="slides">
    <!-- Add homebanner here -->
    <li>
      <div class="evoke home-slide sixteen columns first taylor">
        <div class="container">
          <div class="background-image">
            
          </div>
          <div class="content">
            
            
            <p>A stunning new dining table collaboration celebrating a twenty five year friendship
            with Taylor Howes.</p>
            <a href="/taylor-howes-table">
              <button>Find out more</button>
            </a>
          </div>
        </div>
      </li>
    </ul>
  </div>
```

As would have been seen above, the developer indicated where I needed to place the 'HTML'/'PHP' code for the new slide. I was then helped to create the following 'HTML'/'PHP' code for this section:

The Final HTML/PHP for this Section

```
<div class="flexslider-hero">
  <ul class="slides">
    <li>
      <div class="sixteen columns winter_sale">
        <div class="container">
          <div class="content contentVPosition">
            
            <a href="/sale.html">
              <button>Find out more</button>
            </a>
          </div>
        </div>
      </div>
    </li>
  </ul>
</div>
```

As is evident above, the two aspects to highlight are the fact that the provided image was added into this section and a 'class' called 'contentVPosition' was added as this was a 'class' that had already been created to help position the content inside the banner centrally. The following styles were then applied to this section in the 'CSS' file with help from the developer:

The CSS Code for this Part of the Banner Section

```
.index .flexslider-hero .winter_sale {
  background-color: #A70E2A;
  height: 100%;
  position: relative;
}

.index .flexslider-hero .winter_sale .container {
  height: 100%;
}

.index .flexslider-hero .winter_sale .content {
  text-align: center;
}

.index .flexslider-hero .winter_sale .content img {
  width: 501px;
  display: block;
  margin: 0 auto;
}

.index .flexslider-hero .winter_sale .content button {
  border: 1.5px solid white;
  color: white;
  font-weight: 200;
  text-transform: uppercase;
}
```

With regards to the '.winter_sale', a background colour was added as well as 'height: 100%;' to both style the colour of the background of the banner section and also to help with the formatting. The '.container' also had 'height: 100%;' applied to itself to also help formatting and with regards to the '.content' 'class', 'text-align: center;' was added to position each element inside centrally horizontally. The styles applied to the 'img' element related to mainly changing its size and position on the banner and the appearance of the 'button' element was refined to reflect that of the required outcome with 'text-transform: uppercase;' helping to capitalise the letters inside.

After adding the 'CSS' code above, I was then helped to add the following 'CSS' code relating to this section being responsive and this can be viewed below:

The Responsive CSS Code for this Banner Section

```
.index .flexslider-hero .winter_sale .content img {  
  width: 351px;  
}  
  
.index .flexslider-hero .slides li {  
  height: 500px;  
}
```

The code above reduced the image size for the 'img' element and also the height of any content placed within the 'li' element as currently, the content was overflowing onto the slider buttons and being positioned incorrectly.

The Slider Buttons



The outcome of this section can be viewed below with each image representing a different screen resolution:

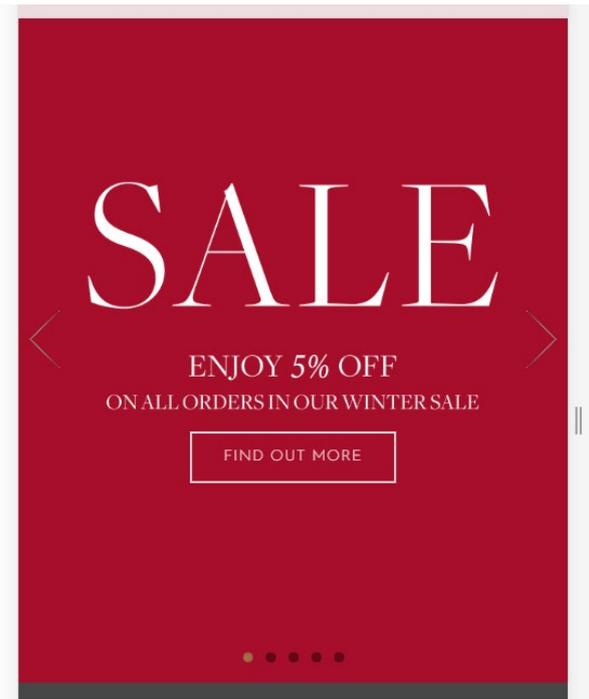
Default Screen Size



1168px or Less



768px or Less



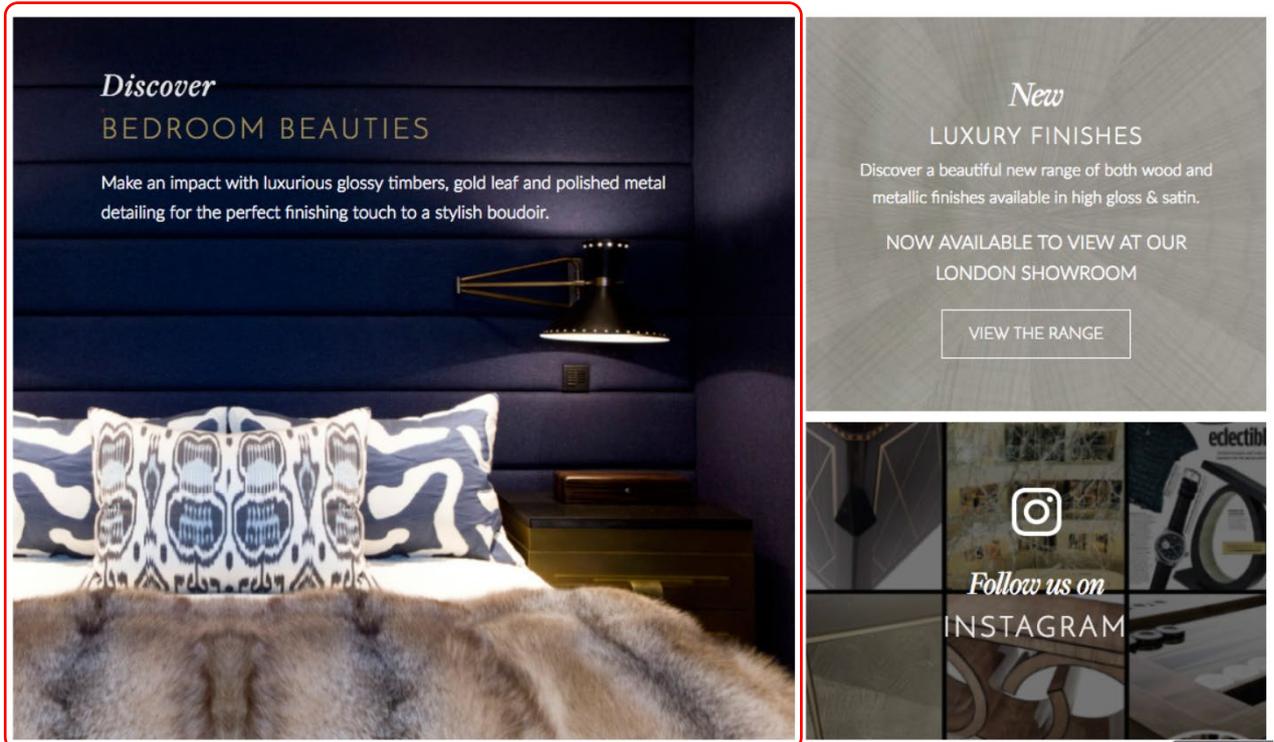
468px or Less



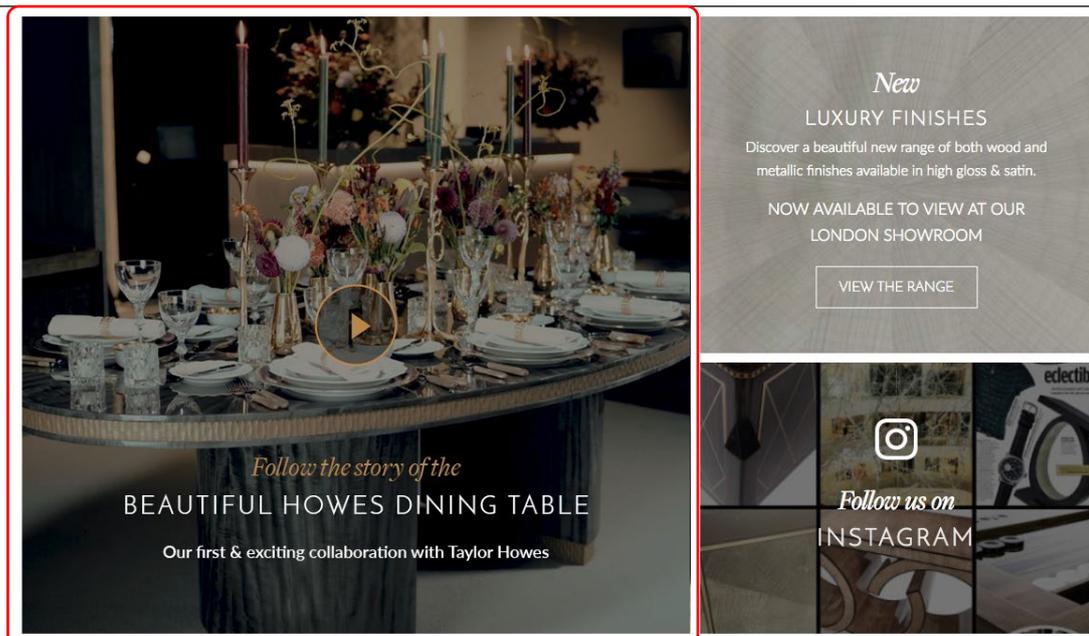
'Coulweb' Work Experience Documentation – Daniel Wilkins

The next and final section to focus on was the video section of the home page of which the current and required outcomes can be viewed below:

The Current Situation of the Video Section on the Home Page



The Required Outcome of the Video Section on the Home Page



The current 'HTML' code for this section can be viewed below:

The Current HTML Code

```
<div class="container main-upsell">

  <a href="/furniture/bedside-dressing-tables.html">
    <div class="ten columns left-upsell">
      <div class="overlay"></div>
      <h2><span>Discover</span>
      <br/>Bedroom Beauties</h2>
      <p>Make an impact with luxurious glossy timbers, gold leaf and polished metal detailing for the
      perfect finishing touch to a stylish boudoir.</p>
    </div>
  </a>

  <!--
  Play button
  <i class="fa fa-play" aria-hidden="true"></i>

  H2 Howes dining table line

  the rest will be paragraph tags

-->
```

This code was then changed to the following 'HTML' code:

The Changed HTML Code

```
<div class="container main-upsell">

  <a href="/furniture/bedside-dressing-tables.html">
    <div class="ten columns left-upsell">
      <div class="overlay"></div>
      <div class="play_button">
        <i class="fa fa-play" aria-hidden="true"></i>
      </div>
      <h2><span>Follow the story of the</span>
      <br/>Beautiful Howes Dining Table</h2>
      <p>Our first & exciting collaboration with Taylor Howes</p>
    </div>
  </a>
```

As will be evident above, I used the supplied code for both the 'play button' by the developer and the original 'HTML' code to create the final code above.

After this had been completed, I then began to style this section in the 'CSS' file of which can be seen below:

Beginning to Style this Section in the CSS file

```
.left-upsell .play_button {
  width: 80px;
  height: 80px;
  border-radius: 50%;
  background-color: rgba(0, 0, 0, 0.5);
  border: 2px solid orange;
}
```

With regards to the first stage of styling the 'play button', I thought that adding an identical width and height whilst also adding a 'border-radius' of 50% would be the best way to make a circular button of which I was informed was correct. However, I was advised that it would be best to add 'padding' instead as this would change in any circumstance as a set width and height wouldn't:

Changing the Styling of the 'play button' in the CSS file with some Help

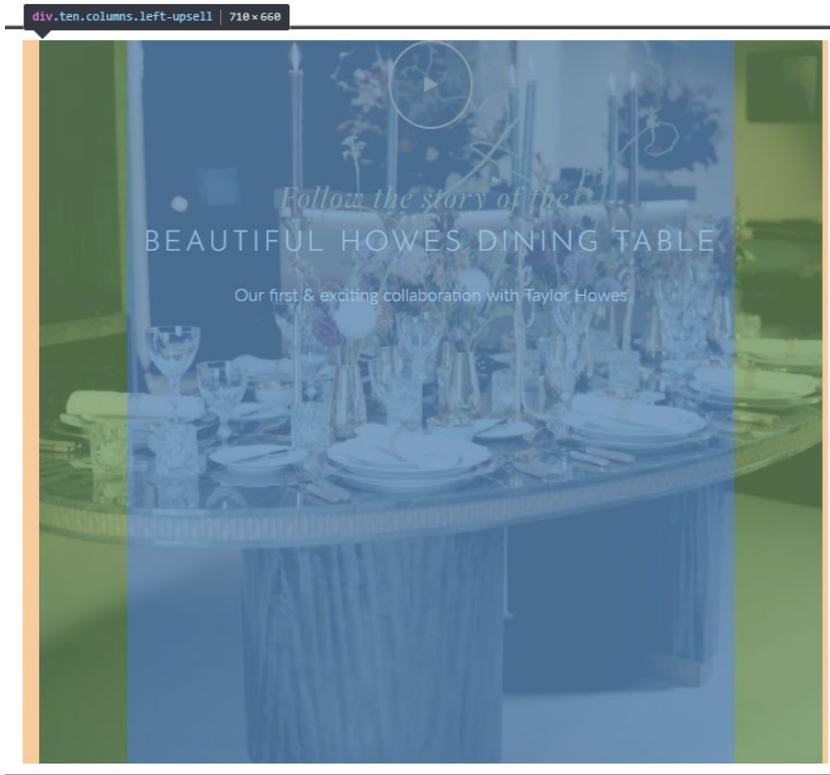
```
.left-upsell .play_button {  
  border-radius: 50%;  
  background-color: rgba(0, 0, 0, 0.5);  
  padding: 30px;  
  display: inline-block;  
  line-height: 1;  
  color: #BA884D;  
  border: 2px solid #BA884D;  
}
```

The Outcome of this on the Page at the Current time (The 'play button' appeared as Required)



After having changed some of the other styles with regards to the appearance of the text within this section, I then used the 'inspect' tool on 'Google Chrome' to identify how I could add 'padding' to the elements within this section so that they would be positioned centrally in vertical format:

Using the 'inspect' tool on 'Google Chrome'



'Coullweb' Work Experience Documentation – Daniel Wilkins

It was at this stage that the work for the first day had been completed as the developer explained to myself that the changes needed to be completed for the next day. He explained that he would complete this and explain to me the next day how. Therefore, the outcome of this section at the end of the day can be seen below. The following day, I was set a different task to complete but the first day gave me some experience of how to make 'front-end' page edits in the method of the company as well as using my own thoughts. I was also able to ask for any advice and help when needed:

The Final Appearance of this Section at the end of the day



PLEASE NOTE

The website which this work was applied to for the first day can be viewed below with the provided links to the home and sale pages of the 'Davidson' website:

<https://www.davidsonlondon.com/>

<https://www.davidsonlondon.com/sale.html>

Day 2 – WordPress Tasks

Introduction

Today, I was set the task of using 'WordPress' to explore different areas such as custom post types and custom fields to create a 'client partner page' and the brief provided by the developer can be viewed below for this day:

Client partner page - PHP

1. Create custom post type named clients - for now we don't need a separate page per client
2. Create custom fields
3. Pull custom fields to frontend using wp query

Brief:

Show a list of our clients in alphabetical order

Each client should have the following:

- Image
- Client name
- Client subtitle
- Short description
- Client tag - any of the following (industrial, educational, management, technology)

Images will be in your PHP - Day 2 folder on the desktop

Make up the client names using the logo

subtitles & descriptions can be lorem ipsum

Client tags should be random pick 2 per client

Undertaking the Task

To begin, I first of all needed to create a custom post type named 'clients' in order to progress with the other aspects of the task. After being provided with the necessary code I then had to change aspects using my own knowledge to do this:

The Provided Code (Changing the Aspects with '[name]' based on my Knowledge)

```
register_post_type( '[name]', array(
    'labels' => array(
        'name' => '[name]',
        'singular_name' => '[name]',
    ),
    'description' => '[name]',
    'public' => false, // it's not public, it shouldn't have it's own permalink, and so on
    'publicly_queryable' => true, // you should be able to query it
    'show_ui' => true, // you should be able to edit it in wp-admin
    'exclude_from_search' => true, // you should exclude it from search results
    'show_in_nav_menus' => false, // you shouldn't be able to add it to menus
    'has_archive' => false, // it shouldn't have archive page
    'rewrite' => false, // it shouldn't have rewrite rules
    'menu_position' => 20,
    'supports' => array( 'title', 'editor', 'custom-fields' )
));
```

Changing the '[name]' Aspects to 'clients' as this Related to Something else already Created

```
register_post_type( 'clients', array(
    'labels' => array(
        'name' => 'clients',
        'singular_name' => 'clients',
    ),
    'description' => 'clients',
    'public' => false, // it's not public, it shouldn't have it's own permalink, and so on
    'publicly_queryable' => true, // you should be able to query it
    'show_ui' => true, // you should be able to edit it in wp-admin
    'exclude_from_search' => true, // you should exclude it from search results
    'show_in_nav_menus' => false, // you shouldn't be able to add it to menus
    'has_archive' => false, // it shouldn't have archive page
    'rewrite' => false, // it shouldn't have rewrite rules
    'menu_position' => 20,
    'supports' => array( 'title', 'editor', 'custom-fields' )
));
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

After completing the previous task, I then began to create the custom fields so that the content could link to these to allow for changing of the content inside the 'WordPress' interface. To begin, I believed that I had to create individual custom field sets for each client that needed to be added to the page:

Beginning to Create Individual Sets of Custom Fields for each Client/Partner

| Field Order | Field Label | Field Name | Field Type |
|-------------|--|---------------------|------------|
| 1 | Client1 Image | client1_image | Image |
| 2 | Client1 Name <small>Edit Duplicate Delete</small> | client1_name | Text |
| 3 | Client1 Subtitle | client1_subtitle | Text |
| 4 | Client1 Description | client1_description | Text Area |
| 5 | Client1 Tag | client1_tag | Checkbox |

Drag and drop to reorder + Add Field

However, it was at this stage that I was advised that it would be best to create one set of custom fields for all of the clients/partners as a 'loop' would be used to search through the different clients and display the required quantity on the page. Therefore, I did this and this can be seen below:

Creating Custom Fields Relating to all of the Clients/Partners

| Field Order | Field Label | Field Name | Field Type |
|-------------|--------------------|--------------------|------------|
| 1 | Client Image | client_image | Image |
| 2 | Client Name | client_name | Text |
| 3 | Client Subtitle | client_subtitle | Text |
| 4 | Client Description | client_description | Text Area |
| 5 | Client Tag | client_tag | Checkbox |

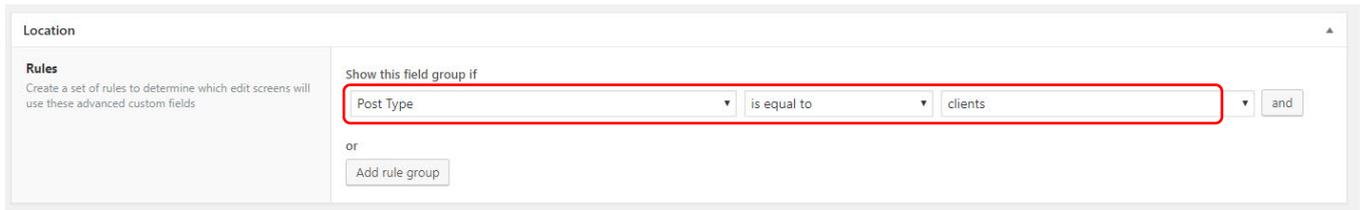
Drag and drop to reorder + Add Field

It is also worth noting that I had been applying the fields to the 'Client' page and I was advised that this needed to be set so that the 'post type' was equal to the 'clients' custom post that I had created before in the code. This was to be able to display the custom fields on the created custom post type aspect of 'WordPress'. Therefore, I changed this to the following:

The Setup of the Fields before Advice

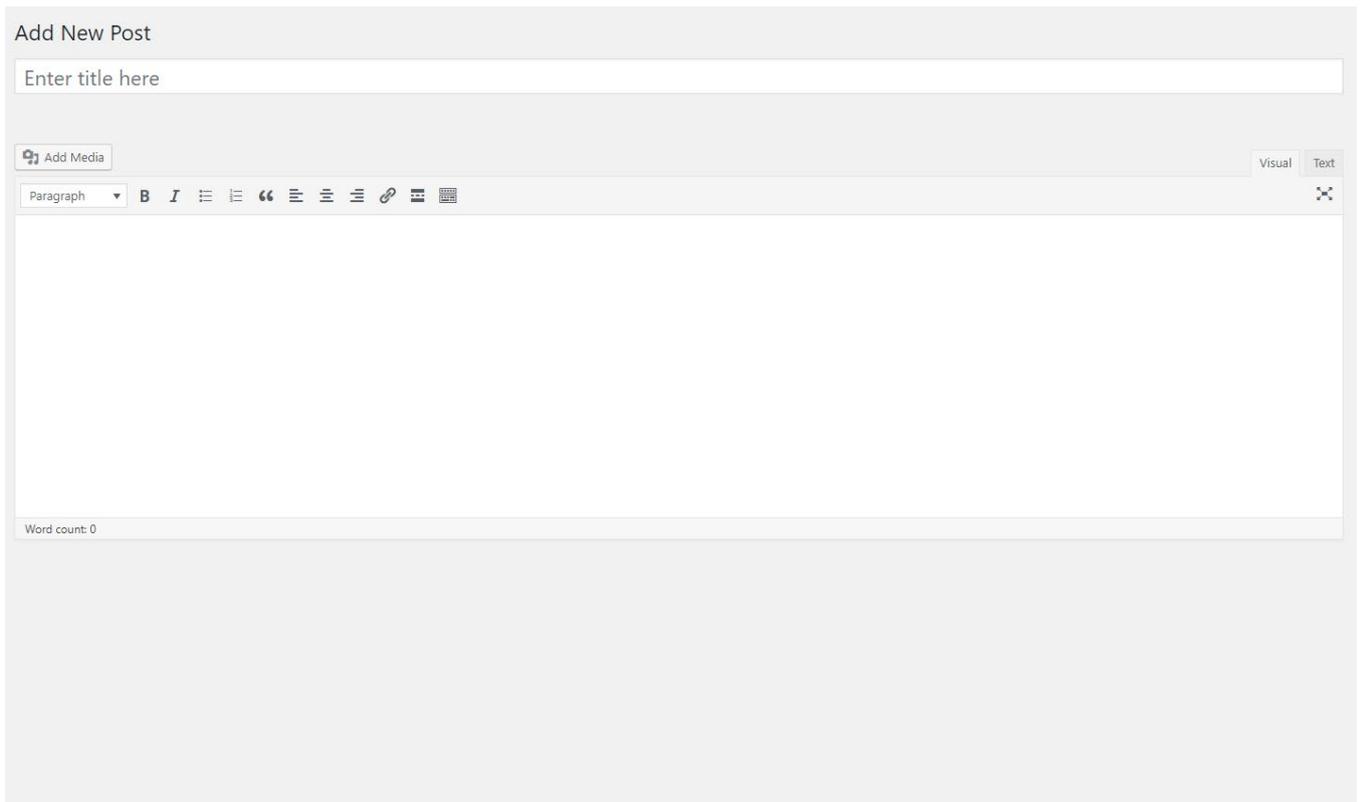
| Location | |
|--|---|
| Rules <small>Create a set of rules to determine which edit screens will use these advanced custom fields</small> | Show this field group if <div style="border: 1px solid red; padding: 2px;">Page ▼ is equal to ▼ Client ▼ and ▼</div> <p>or</p> <input type="button" value="Add rule group"/> |

The Setup of the Fields After Advice



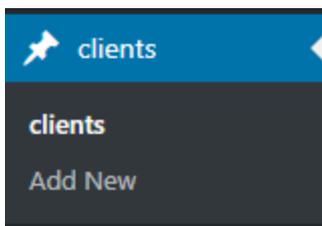
After changing the previous, I then began to start adding the content through the custom fields and I visited the 'Posts' section of 'WordPress', selected 'Add New Post' but the created custom fields weren't appearing on the page as will be evident below:

The Custom Fields weren't Appearing on the Web Page



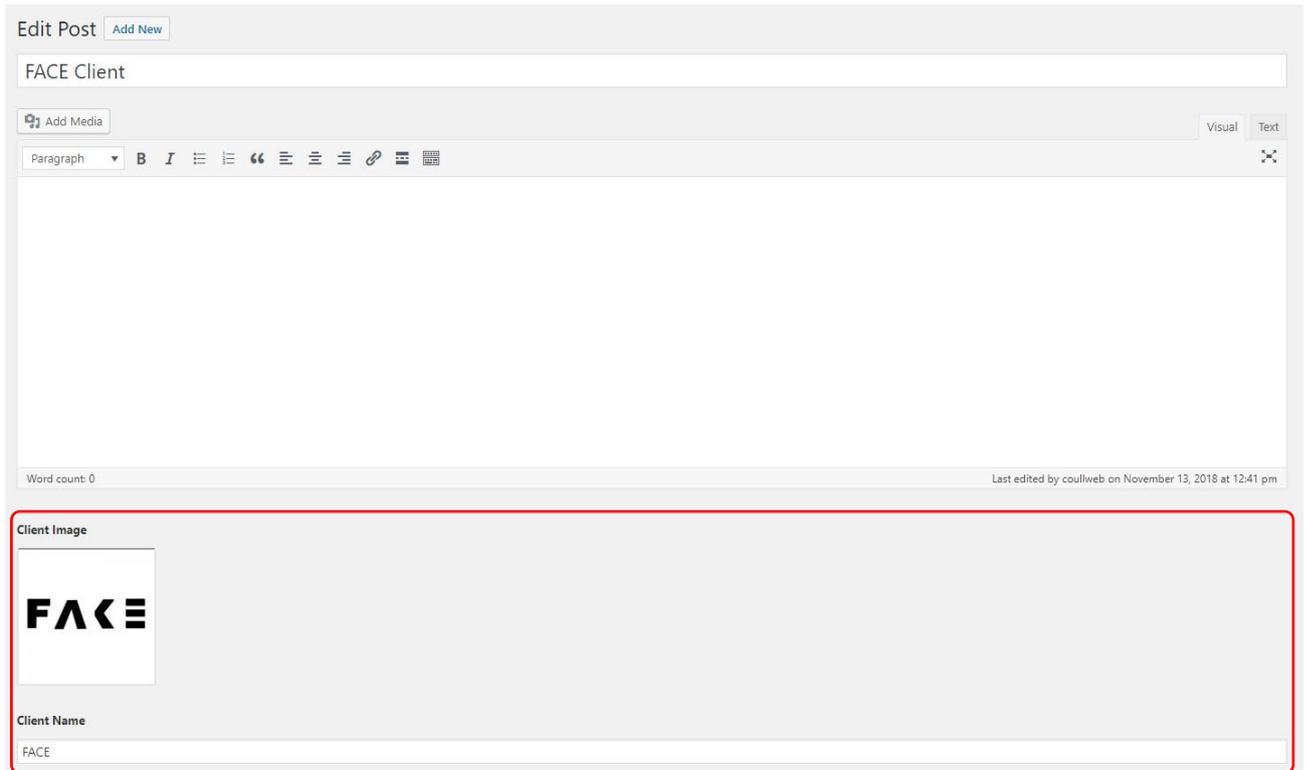
I was then advised to look at the 'clients' custom posts instead as this was the element I had created in the code earlier and now the custom fields were appearing on the page:

The 'clients' Custom Posts Created on WordPress



'Coulweb' Work Experience Documentation – Daniel Wilkins

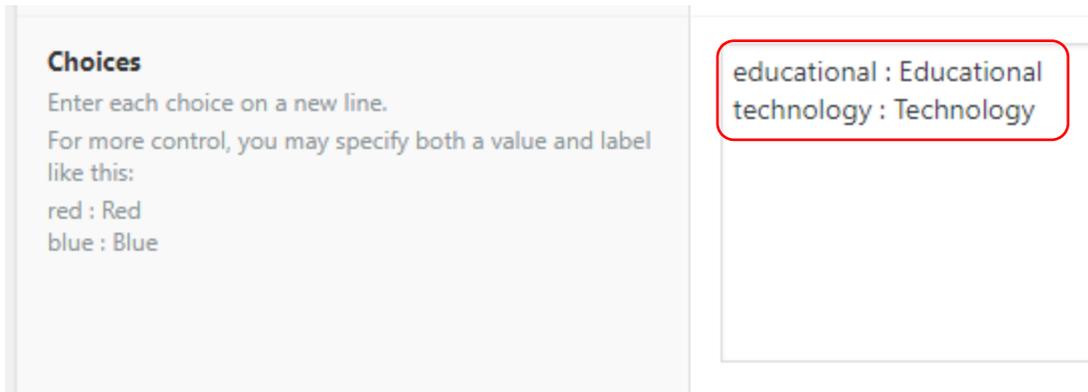
The Custom Fields were now Appearing on the Page when Selecting 'Add New Post'



The screenshot shows the WordPress 'Edit Post' interface. At the top, there is a title field containing 'FACE Client'. Below the title is a rich text editor with a toolbar and a large text area. At the bottom of the editor, there is a 'Word count: 0' and 'Last edited by coullweb on November 13, 2018 at 12:41 pm'. Below the editor, a custom field is visible, outlined in red. It is titled 'Client Image' and contains a placeholder image of the 'FACE' logo. Below the image, there is a 'Client Name' field containing the text 'FACE'.

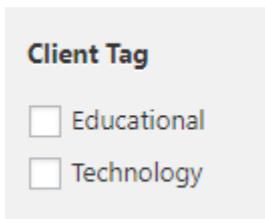
After this had been resolved, I then encountered another error where the 'tag' custom fields wasn't appearing for each of the posts to edit. However, I had noticed that I hadn't completed this section in the custom field setup so I therefore did this and this resolved the problem:

Completing the Custom Field Setup for the 'tag' Section



The screenshot shows the 'Choices' section of the WordPress Custom Fields setup. It includes instructions: 'Enter each choice on a new line. For more control, you may specify both a value and label like this: red : Red blue : Blue'. To the right, there is a text input field containing the following text, which is highlighted with a red rounded rectangle: 'educational : Educational' and 'technology : Technology'.

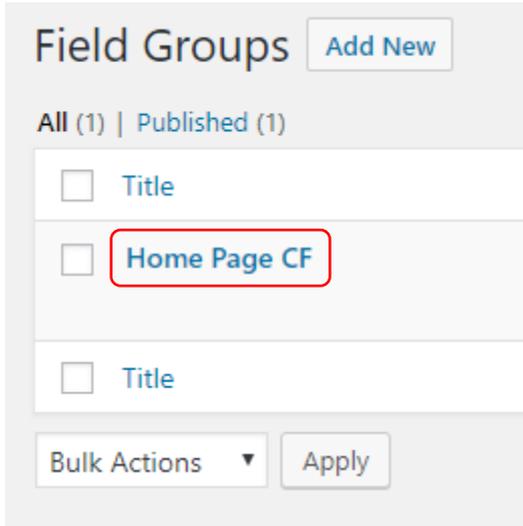
This Custom Field now Appeared on each of the Post Pages to Edit



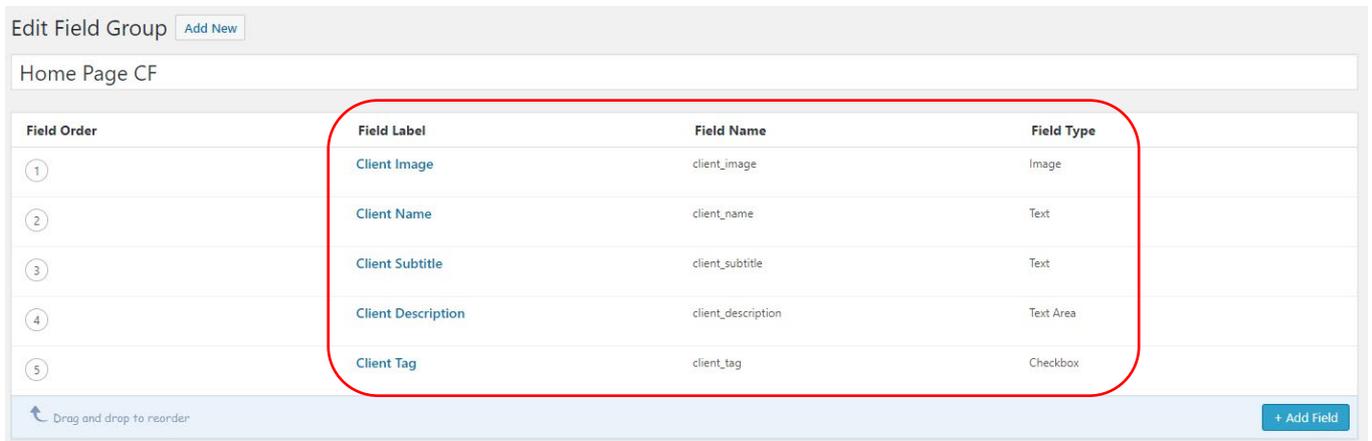
The screenshot shows a custom field titled 'Client Tag' in a post editor. It contains two radio button options: 'Educational' and 'Technology', both of which are currently unselected.

As I hadn't displayed the final custom fields for this task, this is now viewable below:

The Custom Field Group 'Home Page CF' Containing all of the Custom Fields



The Custom Fields for the 'Home Page CF' Field Group



The 'Client Image' Field

| Field Order | Field Label | Field Name | Field Type |
|---|---|--------------|------------|
| 1 | Client Image | client_image | Image |
| Field Label * This is the name which will appear on the EDIT page | <input type="text" value="Client Image"/> | | |
| Field Name * Single word, no spaces. Underscores and dashes allowed | <input type="text" value="client_image"/> | | |
| Field Type * | <input type="text" value="Image"/> | | |
| Field Instructions Instructions for authors. Shown when submitting data | <div style="border: 1px solid #ccc; height: 40px;"></div> | | |
| Required? | <input type="radio"/> Yes <input checked="" type="radio"/> No | | |
| Return Value Specify the returned value on front end | <input type="radio"/> Image Object <input checked="" type="radio"/> Image URL <input type="radio"/> Image ID | | |
| Preview Size Shown when entering data | <input checked="" type="radio"/> Thumbnail <input type="radio"/> Medium <input type="radio"/> Large <input type="radio"/> Full <input type="radio"/> Medium_large <input type="radio"/> Twentyseventeen Featured Image <input type="radio"/> Twentyseventeen Thumbnail Avatar | | |
| Library Limit the media library choice | <input checked="" type="radio"/> All <input type="radio"/> Uploaded to post | | |
| Conditional Logic | <input type="radio"/> Yes <input checked="" type="radio"/> No | | |
| | <input type="button" value="Close Field"/> | | |

As is evident with this field, this was an image field where the 'Return Value' was set to 'Image URL' as this was something that was advised by the developer and from my own knowledge, I knew this was the only option that would return the image properly. Everything else remained at their default values on the page as these didn't require changes.

The 'Client Name' Field

| Field Order | Field Label | Field Name | Field Type |
|---|---|--------------------|------------|
| 2 | Client Name | client_name | Text |
| Field Label * This is the name which will appear on the EDIT page | <input type="text" value="Client Name"/> | | |
| Field Name * Single word, no spaces. Underscores and dashes allowed | <input type="text" value="client_name"/> | | |
| Field Type * | <input type="text" value="Text"/> | | |
| Field Instructions Instructions for authors. Shown when submitting data | <div style="border: 1px solid #ccc; height: 40px;"></div> | | |
| Required? | <input type="radio"/> Yes <input checked="" type="radio"/> No | | |
| Default Value Appears when creating a new post | <input type="text"/> | | |
| Placeholder Text Appears within the input | <input type="text"/> | | |
| Prepend Appears before the input | <input type="text"/> | | |
| Append Appears after the input | <input type="text"/> | | |
| Formatting Affects value on front end | <input type="text" value="Convert HTML into tags"/> | | |
| Character Limit Leave blank for no limit | <input type="text"/> | | |
| Conditional Logic | <input type="radio"/> Yes <input checked="" type="radio"/> No | | |
| | <input type="button" value="Close Field"/> | | |
| 3 | Client Subtitle | client_subtitle | Text |
| 4 | Client Description | client_description | Text Area |
| 5 | Client Tag | client_tag | Checkbox |
| <input type="button" value="Drag and drop to reorder"/> <input type="button" value="+ Add Field"/> | | | |

'Coulweb' Work Experience Documentation – Daniel Wilkins

This was a text field as the name needed to only be text and all settings remained at their defaults as nothing required changing for this section. Please note the 'Client Subtitle' custom field has not been included as this was identical to the field shown above.

The 'Client Description' Field

The screenshot shows a configuration interface for a field named 'Client Description'. The interface is divided into two main sections: a left-hand sidebar with configuration options and a right-hand main area with input fields. The sidebar options include: Field Label (set to 'Client Description'), Field Name (set to 'client_description'), Field Type (set to 'Text Area'), Field Instructions (empty), Required? (set to 'No'), Default Value (empty), Placeholder Text (empty), Character Limit (empty), Rows (set to '8'), Formatting (set to 'Convert new lines into
 tags'), and Conditional Logic (set to 'No'). The main area shows the corresponding input fields for each option. At the bottom of the interface, there is a 'Close Field' button and a list of other fields: 'Client Tag', 'client_tag', and 'Checkbox'. A blue '+ Add Field' button is located in the bottom right corner.

With regards to this field, I set this to a 'Text Area' field as this required a larger amount of text than that of the name and subtitle and everything was left at their defaults as nothing required any changes.

The 'Client Tag' Field

The screenshot displays the configuration page for a field named 'Client Tag'. The page has a blue header with the field name and type. The configuration is organized into several sections:

- Field Label ***: This is the name which will appear on the EDIT page. Value: Client Tag
- Field Name ***: Single word, no spaces. Underscores and dashes allowed. Value: client_tag
- Field Type ***: A dropdown menu set to 'Checkbox'.
- Field Instructions**: Instructions for authors. Shown when submitting data. (Empty text area)
- Required?**: Radio buttons for 'Yes' and 'No'. 'No' is selected.
- Choices**: Enter each choice on a new line. For more control, you may specify both a value and label like this: red : Red, blue : Blue. Value: educational : Educational, technology : Technology, industrial : Industrial, management : Management
- Default Value**: Enter each default value on a new line. (Empty text area)
- Layout**: Radio buttons for 'Vertical' and 'Horizontal'. 'Vertical' is selected.
- Conditional Logic**: Radio buttons for 'Yes' and 'No'. 'No' is selected.

At the bottom, there is a 'Close Field' button and an 'Add Field' button.

As is evident with this field, the field type was set to 'Checkbox' as this was something advised by the developer due to the fact that this would act as a selection for the user. The only element changed in this was the choices as these had to be specified in order for the user to select an option from the list. The layout may have also been changed to vertical which related to the order in which the list would be displayed as this appeared to be a better option aesthetically than choosing the horizontal option.

With regards to the code for displaying the content inside the custom fields, I first of all started creating the following in the 'HTML/PHP' code:

```
<?php
/*
 * Template Name: Home
 */
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <h1>Hello World</h1>
    </div>
    <div>
      
      <h2><?php the_field('client_title'); ?></h2>
      <h3><?php the_field('client_subtitle'); ?></h3>
      <p><?php the_field('client_description'); ?></p>
      <p><?php the_field('client_tag'); ?></p>
    </div>
  </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

However, I was then advised that I had to place a query within the code as the code above wouldn't work and would possibly display all the content for each created client/partner post.

To begin adding the query, I implemented the supplied code and then was required to change aspects so that this would work for the project I was working on:

Implementing the Query into the Code

```
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <h1>Hello World</h1>
    </div>
    <?php
    <?php
      $args = array(
        'posts_per_page' => 1,
        'post_type' => '[post_name]',
        'post_status' => 'publish',
        'meta_key' => '[search_value]',
        'meta_value' => '[search_criteria]'
      );
      $query = new WP_Query($args);
      if ( $query->have_posts() ):
        while ( $query->have_posts() ):
          $query->the_post();
        ?>
      <div>
        
        <h2><?php the_field('client_title'); ?></h2>
        <h3><?php the_field('client_subtitle'); ?></h3>
        <p><?php the_field('client_description'); ?></p>
        <p><?php the_field('client_tag'); ?></p>
      </div>
    <?php
      endwhile;
    endif;
    /* Restore original Post Data */
    wp_reset_postdata();
  ?>
  </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The Current Outcome of this on the Page (This Didn't Display any Posts)



Hello World

After implementing the previous code, I then undertook research with regards to displaying posts and displaying them alphabetically as this was one of the requirements. From the research I changed the code to the following:

The Changes to the Current Code

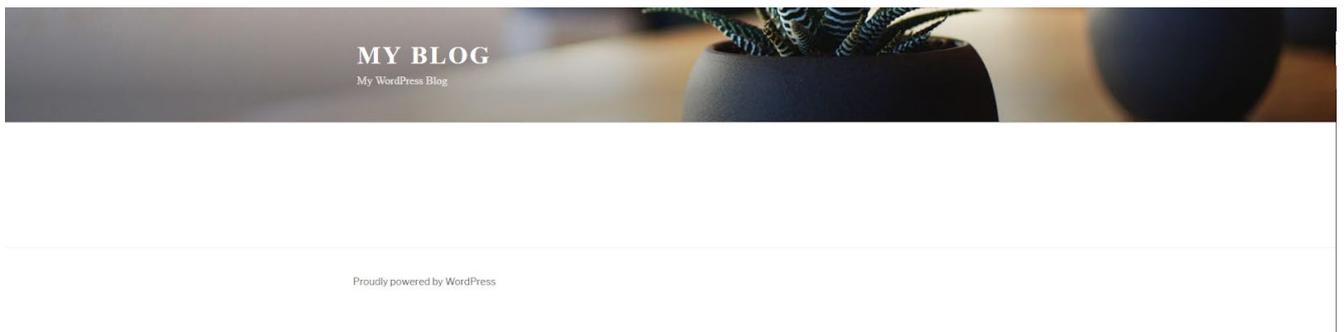
```
*/
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <?php
        $args = array(
          'posts_per_page' => 8,
          'post_type' => 'clients',
          'post_status' => 'publish',
          'orderby' => 'meta_value',
          'meta_key' => 'keyname'
        );
        $query = new WP_Query($args);
        if ( $query->have_posts() ):
          while ( $query->have_posts() ):
            $query->the_post();
          ?>
          <div>
            <?php the_field('client_title'); ?></h2>
            <h3><?php the_field('client_subtitle'); ?></h3>
            <p><?php the_field('client_description'); ?></p>
            <p><?php the_field('client_tag'); ?></p>
          </div>
          <?php
            endwhile;
          endif;
          /* Restore original Post Data */
          wp_reset_postdata();
        ?>
      </div>
    </main><!-- #main -->
  </div><!-- #primary -->

<?php
get_footer(); ?>
```

I changed the 'posts_per_page' to 8 as the page needed to display more than one post on the page. The 'orderby' and 'keyname' were added as I thought this would help display the posts alphabetically as this was something I had found whilst undertaking research. Unfortunately, this didn't work and this can be viewed below:

The Outcome of these Changes on the Page (This Didn't Work)



After struggling with this part, I then asked for some help and was helped to create the following changes to the code:

Being Helped to make the Required Changes to the Code

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <?php
        $args = array(
          'posts_per_page' => 8,
          'post_type' => 'clients',
          'post_status' => 'publish',
          'meta_key' => 'client_title',
          'orderby' => 'meta_value',
          'order' => 'ASC'
        );
        $query = new WP_Query($args);
        if ( $query->have_posts() ):
          while ( $query->have_posts() ):
            $query->the_post();
          >>
          <div>
            <?php the_field('client_title'); ?></h2>
            <h3><?php the_field('client_subtitle'); ?></h3>
            <p><?php the_field('client_description'); ?></p>
            <p><?php the_field('client_tag'); ?></p>
          </div>
        <?php
          endwhile;
        endif;
        /* Restore original Post Data */
        wp_reset_postdata();
      >>
    </div>
  </main><!-- #main -->
</div><!-- #primary -->
</?php
```

The problem was with regards to the fact that I had ordered the elements in the wrong order as well as incorrectly adding code to the highlighted section above. The 'orderby' was required and the 'ASC' was added for the 'order' as this would help the posts to be sorted in ascending order which in this case would be by name, alphabetically. However, I was still encountering the issue of the posts not displaying on the page:

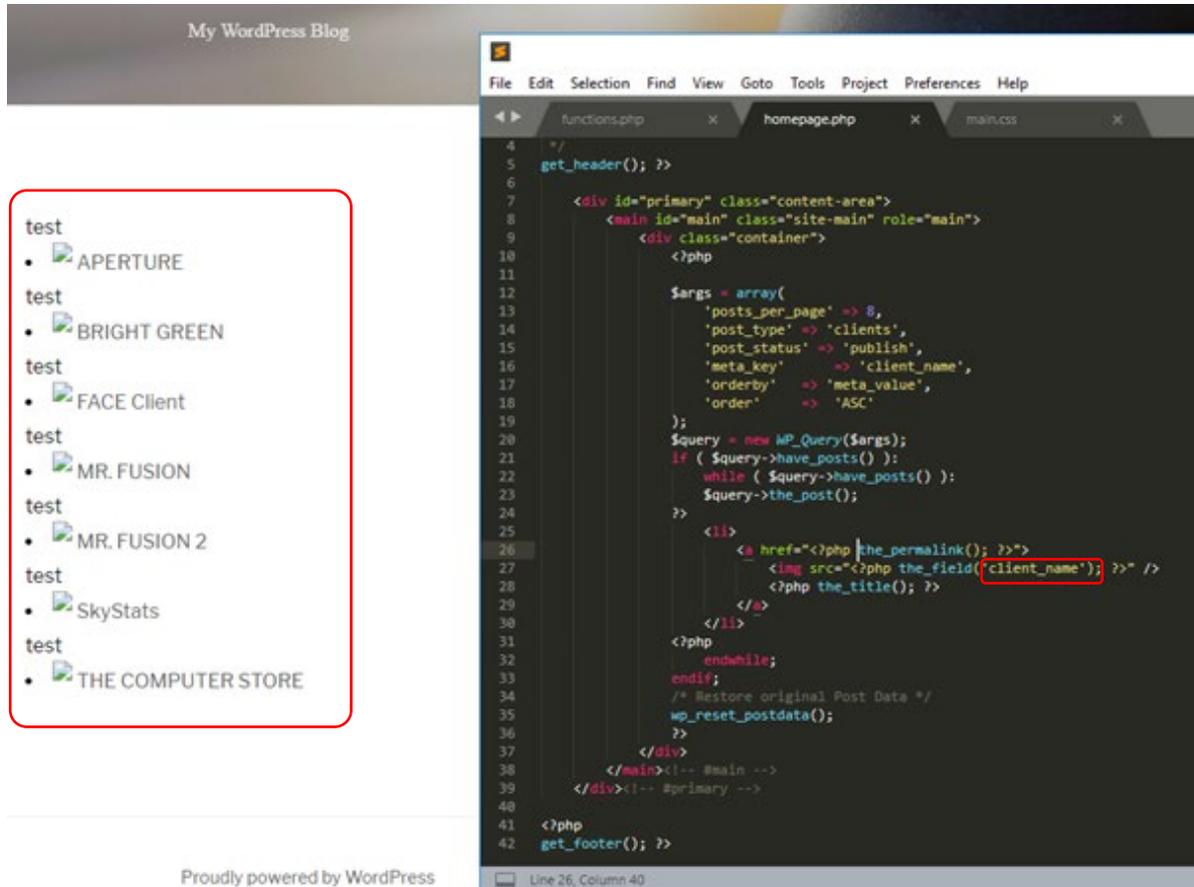
The Posts weren't Displaying on the Page Still



'Coulweb' Work Experience Documentation – Daniel Wilkins

I was then helped again, after adding new code with regards to displaying the 'permalink', 'client name' and the 'title' for testing purposes, and it was because of a simple error with regards to one of the names of the custom fields in the code which didn't match that of the one created in the 'WordPress' interface:

The Incorrect Name had been Entered for the Client's Name in the Code (Here the Name had been Changed to the Correct Name and the Content was now Appearing on the Page)

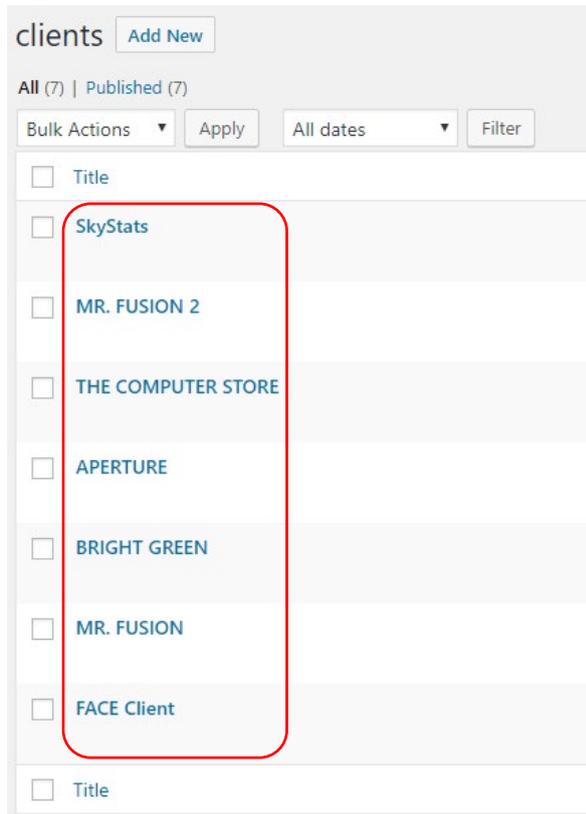


The image shows a screenshot of a WordPress blog and its source code. On the left, the blog's front-end is visible, displaying a list of posts under the heading "My WordPress Blog". Each post is titled "test" and features a client name: APERTURE, BRIGHT GREEN, FACE Client, MR. FUSION, MR. FUSION 2, SkyStats, and THE COMPUTER STORE. A red box highlights this list. On the right, the source code is shown in a dark-themed editor. The code includes a PHP query for posts of type 'clients' and an HTML loop that outputs the permalink, a client name, and the title. A red box in the code highlights the line: ``. The status bar at the bottom of the editor indicates "Line 26, Column 40".

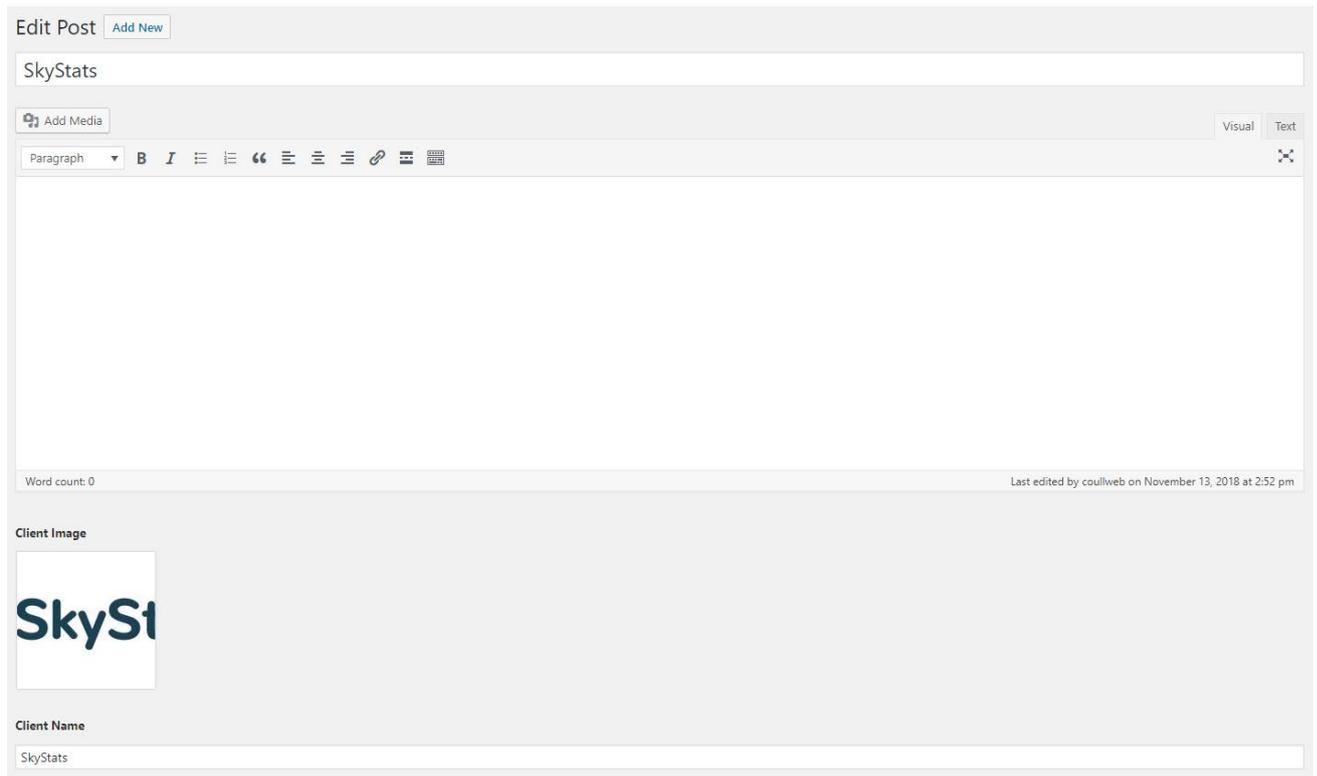
'Coulweb' Work Experience Documentation – Daniel Wilkins

With regards to entering the content for each post, this is now being shown below as this had been forgotten to be added in earlier within this document:

All of the Created Posts for the 'clients' Custom Post Type



Example of the Layout for the Client Posts in the WordPress Interface



'Coullweb' Work Experience Documentation – Daniel Wilkins

Client Subtitle

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Client Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a ex in mauris vehicula ullamcorper. Fusce accumsan nulla fringilla tortor aliquet viverra eget quis dui. Phasellus imperdiet eu odio rhoncus scelerisque. Aliquam arcu velit, ultricies vitae felis id, blandit placerat lectus. Nam pretium tempor ante sit amet cursus.

Client Tag

Educational
 Technology
 Industrial
 Management

After realising that now the posts were now being displayed on the page, I changed the code in the file so that it matched that of the original code as I had to display each of the fields on the page for each post.

Changing the Code to Display the Necessary Content for each Post

```
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <?php

      $args = array(
        'posts_per_page' => 8,
        'post_type' => 'clients',
        'post_status' => 'publish',
        'meta_key' => 'client_name',
        'orderby' => 'meta_value',
        'order' => 'ASC'
      );
      $query = new WP_Query($args);
      if ( $query->have_posts() ):
        while ( $query->have_posts() ):
          $query->the_post();
        ?>
        <div class="clients">
          
          <h2><?php the_field('client_name'); ?></h2>
          <h3><?php the_field('client_subtitle'); ?></h3>
          <p><?php the_field('client_description'); ?></p>
          <p><?php the_field('client_tag'); ?></p>
        </div>
      <?php
        endwhile;
      endif;
      /* Restore original Post Data */
      wp_reset_postdata();
    ?>
  </div>
</main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer(); ?>
```

'Coullweb' Work Experience Documentation – Daniel Wilkins

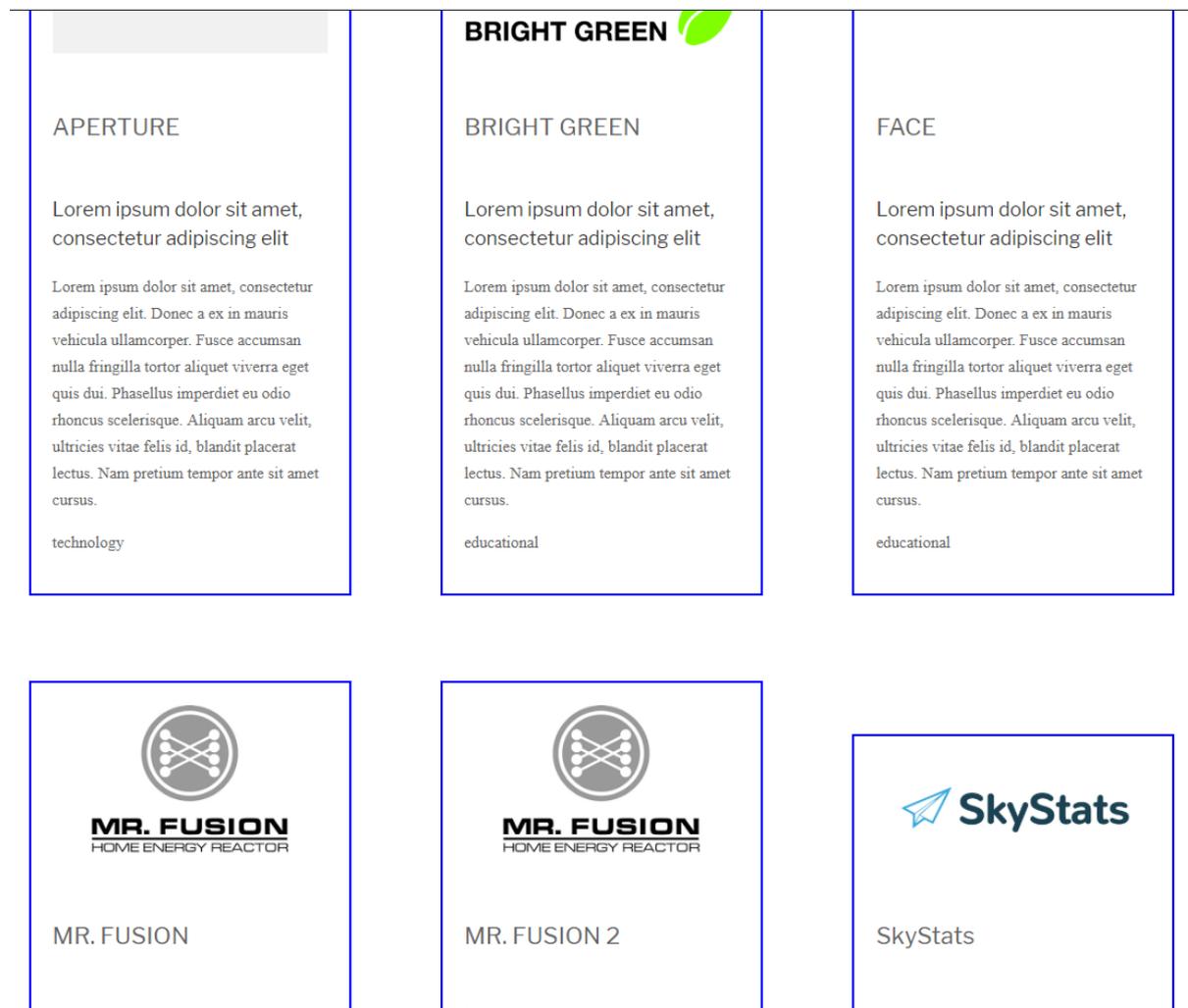
It is also worth noting that I had assigned a new container around the content of which I styled in the 'CSS' and the styles for this can be viewed below:

```
.clients {  
  width: 300px;  
  height: auto;  
  display: inline-block;  
  margin: 40px;  
  border: 2px solid blue;  
  padding: 20px;  
}
```

This would style each container for each of the posts on the page and I styled this so that each container would have a set width to allow for each post to be shown on a single row. I also added a border to separate each of the posts as well as 'margin' and 'padding' to space each container apart from each other as well as creating space between the content and the border inside each container. The 'display: inline-block' would also allow for the containers to be displayed on a single line as opposed to 'block' which would place each container on a new line.

The posts were now appearing with the added styling, although this needed to be edited so that the posts would all be displayed on a single line:

The Posts were now Displaying but not as Required



'Coulweb' Work Experience Documentation – Daniel Wilkins

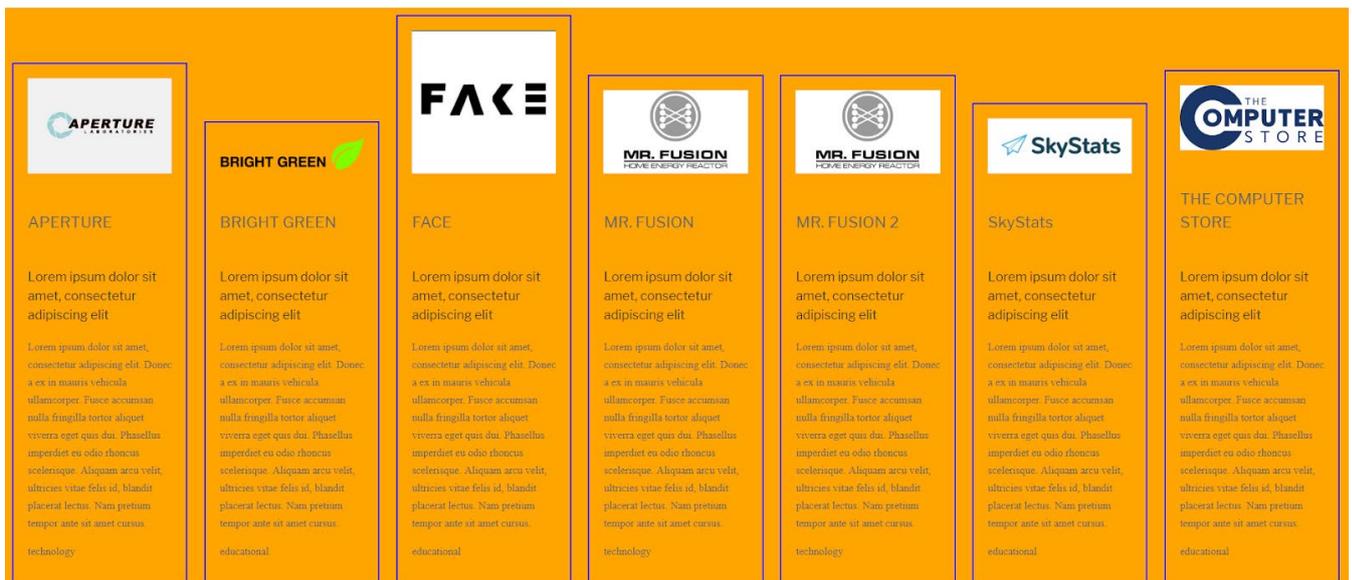
I then thought that the width of the page would be '100%' and because there were seven posts, I could divide 100 by seven to calculate the percentage each container's width would need to be. Therefore, I did this and reduced the width slightly to ensure the posts would all display on the same line. I also reduced the 'margin' and 'padding' to stop some of the posts from being pushed onto the next line. It is also worth noting that I added some styles for the '.container' 'class' surrounding the posts as I thought this would help both from a visual and coding point of view:

Altering the Styles of the '.clients' Class as well as adding some Styles to the '.container' Class Surrounding this Section of the Page

```
.container {
  width: 100%;
  height: auto;
  box-sizing: border-box;
  background-color: orange;
}

.clients {
  width: 13%;
  height: auto;
  display: inline-block;
  margin: 10px;
  border: 2px solid blue;
  padding: 20px;
}
```

The Outcome of this on the Page (This now Worked)



After completing this, I then decided to add a couple more containers inside the 'clients' 'class'; one for the images and the other for the text in case any of these styles needed to be changed which would make this easier to do:

Adding the New 'classes' to the 'clients' Container

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <?php
        $args = array(
          'posts_per_page' => 8,
          'post_type' => 'clients',
          'post_status' => 'publish',
          'meta_key' => 'client_name',
          'orderby' => 'meta_value',
          'order' => 'ASC'
        );
        $query = new WP_Query($args);
        if ( $query->have_posts() ):
          while ( $query->have_posts() ):
            $query->the_post();
          >>
          <div class="clients">
            <div class="ci_container">
              
            <div class="ct_container">
              <h2><?php the_field('client_name'); ?></h2>
              <h3><?php the_field('client_subtitle'); ?></h3>
              <p><?php the_field('client_description'); ?></p>
              <p><?php the_field('client_tag'); ?></p>
            </div>
          </div>
        <?php
          endwhile;
        endif;
        /* Restore original Post Data */
        wp_reset_postdata();
      >>
    </div>
  </main><!-- #main -->
</div><!-- #primary -->
```

After being helped, I then added and changed the following styles:

```
.container {
  width: 100%;
  height: auto;
  box-sizing: border-box;
  background-color: orange;
}

.clients {
  width: 13%;
  height: auto;
  display: inline-block;
  margin: 10px;
  border: 2px solid blue;
  padding: 20px;
  vertical-align: top;
}

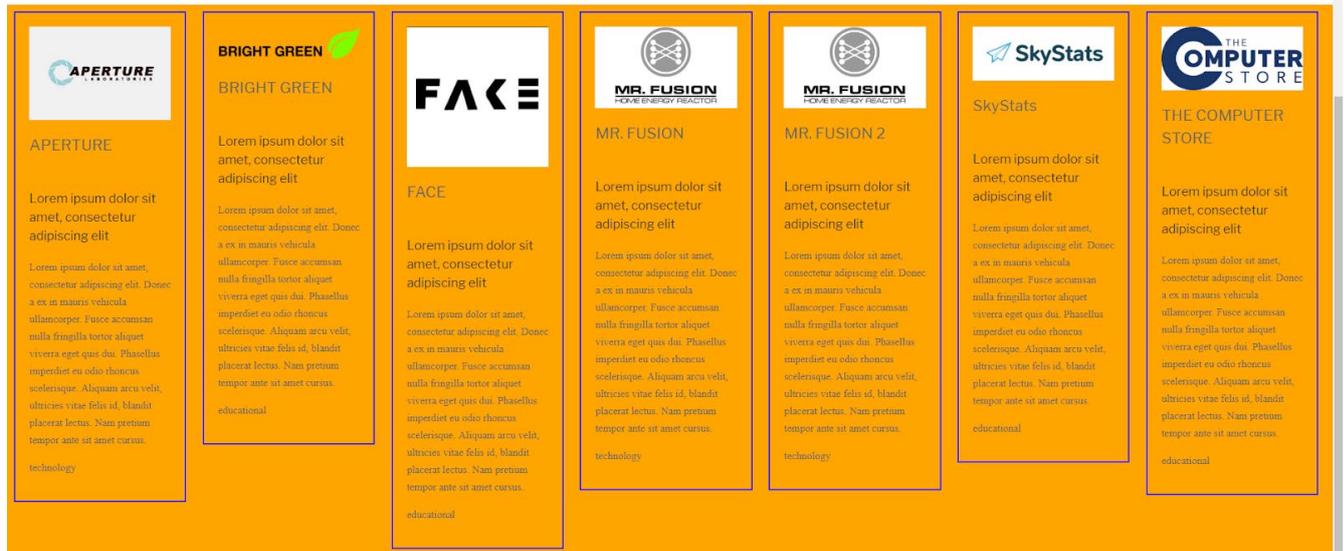
.ci_container {
  width: 100%;
  height: auto;
}

.ct_container {
  width: 100%;
  height: auto;
}
```

'Coullweb' Work Experience Documentation – Daniel Wilkins

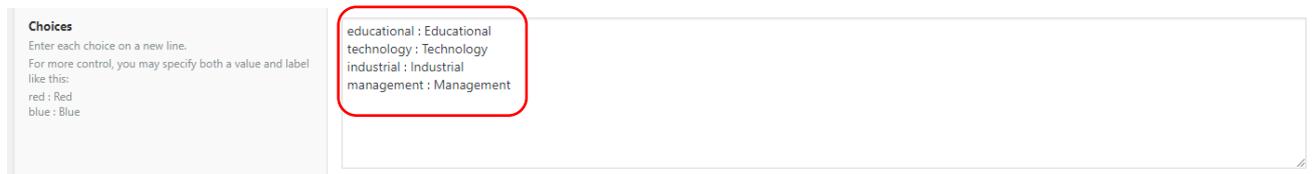
As is evident above, the main aspect to explain is the 'vertical-align: top;' as this is what the developer had advised to include in order for each post to start at the same height as each other as currently, the positioning was based on the height of each image. This then solved this issue and now this section had been styled fully to allow myself for completing the next task:

The Outcome of these Changes on the Page (The Appearance had now been Completed)

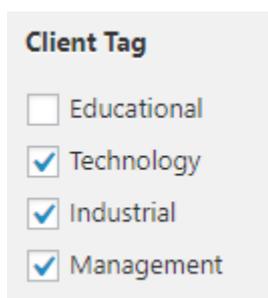


The task now set by the developer was to allow for filtering of each post so that if a category was selected, only posts with 'tags' matching that category would be displayed. To begin, I was told to add more 'tags' to allow for random selecting of categories for each post in the 'WordPress' interface:

Adding more 'tags' to the Custom Field



The Process of Randomly Selecting Categories for each Post (Example)



After completing this, I was told to create a dropdown box of which I integrated code from 'w3Schools' in order to do this:

The Code used from 'w3Schools'

HTML <select> Tag

< Previous Complete HTML Reference Next >

Example

Create a drop-down list with four options:

```
<select>
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="mercedes">Mercedes</option>
<option value="audi">Audi</option>
</select>
```

Try it Yourself >

Integrating the Code into the Task I was Working on

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <div class="dropdown_container">
        <select class="dropdown">
          <option value="">Please select...</option>
          <option value="educational">Educational</option>
          <option value="technology">Technology</option>
          <option value="industrial">Industrial</option>
          <option value="management">Management</option>
        </select>
      </div>
    </div>
  <?php
```

I also created a container for this section so that the dropdown box would be positioned separately from the posts and the styles for this can be viewed below:

The CSS Code for the '.dropdown_container' 'class'

```
.dropdown_container {
  display: block;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
  background: red;
}
```

The Outcome of these Changes on the Page (This now Displayed the Dropdown box)



'Coullweb' Work Experience Documentation – Daniel Wilkins

As this was something that I was completely new to, I therefore had to be helped quite frequently and the following and final solution to making the filter work can be viewed below:

The Final Solution

```
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <div class="dropdown_container">
        <select class="dropdown">
          <option value="">Please select...</option>

          <?php
            $filterOptions = array("educational", "technology", "industrial", "management");
            foreach ($filterOptions as $value):
              ?>
                <option value="<?php echo $value ?>" <?php if ($value == $_GET['filter']){ echo "selected"; } ?>><?php echo ucfirst ( $
                  value );| ?></option>
              <?php
                endforeach;
              ?>

          <!-- <option value="technology" <?php if ("technology" == $_GET['filter']){ echo "selected"; } ?>>Technology</option>
          <option value="industrial" <?php if ("industrial" == $_GET['filter']){ echo "selected"; } ?>>Industrial</option>
          <option value="management" <?php if ("management" == $_GET['filter']){ echo "selected"; } ?>>Management</option> -->
        </select>
      </div>
    <?php

    // selected <= HTML for selected box
    //echo $_GET['filter'];
```

```
$args = array(
  'posts_per_page' => 8,
  'post_type' => 'clients',
  'post_status' => 'publish',
  'meta_key' => 'client_name',
  'orderby' => 'meta_value',
  'order' => 'ASC',
  'meta_query' =>
    array(
      'key' => 'client_tag',
      'value' => $_GET['filter'],
      'compare' => 'LIKE'
    ),
);
$query = new WP_Query($args);
if ( $query->have_posts() ):
  while ( $query->have_posts() ):
    $query->the_post();
  ?>
  <div class="clients">
    <div class="ci_container">
      
    </div>
    <div class="ct_container">
      <h2><?php the_field('client_name'); ?></h2>
      <h3><?php the_field('client_subtitle'); ?></h3>
      <p><?php the_field('client_description'); ?></p>
      <p><?php the_field('client_tag'); ?></p>
    </div>
  </div>
<?php
  endwhile;
endif;
/* Restore original Post Data */
wp_reset_postdata();
?>
</div>
```

```
<script type="application/javascript">
  $( ".dropdown" ).change(function(){
    var selectedOption = $(this).val();
    window.location.replace("https://clientb.coullweb.com/wp/clients/?filter=" + selectedOption);
  });
</script>
</main><!-- #main -->
</div><!-- #primary -->
```

There are a few sections which will be explained below:

```
get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="container">
      <div class="dropdown_container">
        <select class="dropdown">
          <option value="">Please select...</option>

          <?php
            $filterOptions = array("educational", "technology", "industrial", "management");
            foreach ($filterOptions as $value):
          ?>
            <option value="<?php echo $value ?>" <?php if ($value == $_GET['filter']){ echo "selected"; } ?>><?php echo ucfirst ( $
              value ); ?></option>
          <?php
            endforeach;
          ?>

          <!-- <option value="technology" <?php if ("technology" == $_GET['filter']){ echo "selected"; } ?>>Technology</option>
          <option value="industrial" <?php if ("industrial" == $_GET['filter']){ echo "selected"; } ?>>Industrial</option>
          <option value="management" <?php if ("management" == $_GET['filter']){ echo "selected"; } ?>>Management</option> -->
        </select>
      </div>
    </div>
  </div>

  // selected <= HTML for selected box
  //echo $_GET['filter'];

```

The first part highlighted above created an 'array' which included all of the values for the 'tags' which in this case were 'educational', 'technology', 'industrial' and 'management'. Then the '\$filterOptions' was assigned to the '\$value' which would then allow this to be 'echoed' within the 'option value' which is highlighted in the second part. 'PHP' was then utilised to 'echo' 'selected' if the '\$value' was equal to the filter.

```
$args = array(
  'posts_per_page' => 8,
  'post_type' => 'clients',
  'post_status' => 'publish',
  'meta_key' => 'client_name',
  'orderby' => 'meta_value',
  'order' => 'ASC',
  'meta_query' =>
    array(
      'key' => 'client_tag',
      'value' => $_GET['filter'],
      'compare' => 'LIKE'
    )
);
$query = new WP_Query($args);
if ( $query->have_posts() ):
  while ( $query->have_posts() ):
    $query->the_post();
  ?>
  <div class="clients">
    <div class="ci_container">
      
    </div>
    <div class="ct_container">
      <h2><?php the_field('client_name'); ?></h2>
      <h3><?php the_field('client_subtitle'); ?></h3>
      <p><?php the_field('client_description'); ?></p>
      <p><?php the_field('client_tag'); ?></p>
    </div>
  </div>
  <?php
    endwhile;
  endif;
  /* Restore original Post Data */
  wp_reset_postdata();
  ?>
</div>

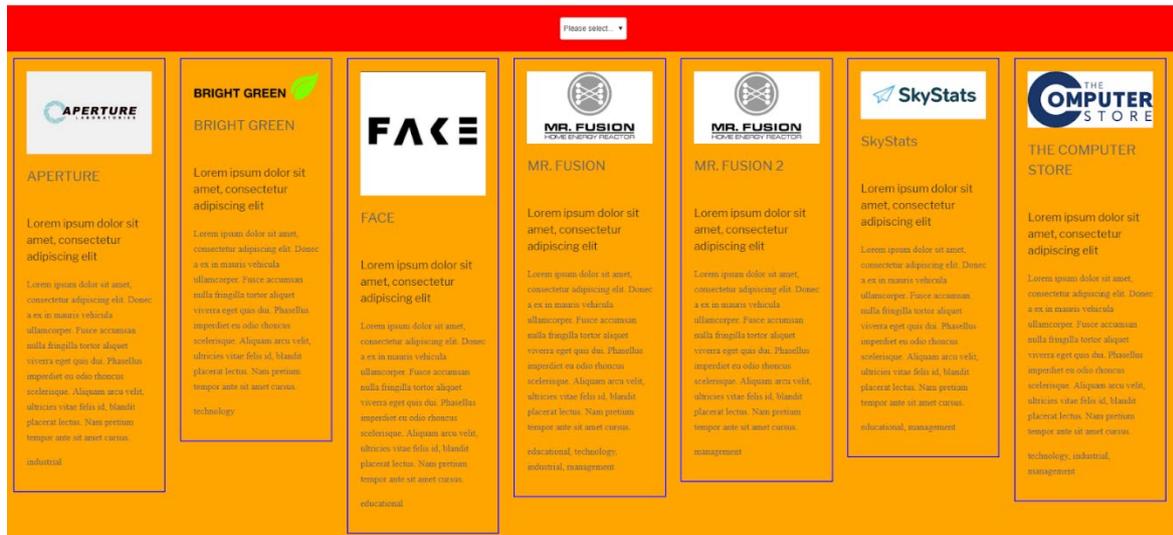
```

Within the section highlighted above, this created a 'query' where the 'client_tag' was used to check against the 'value' inside the filter which then allowed for all of this code in this section to display the posts only relating to the specific selected 'tag'.

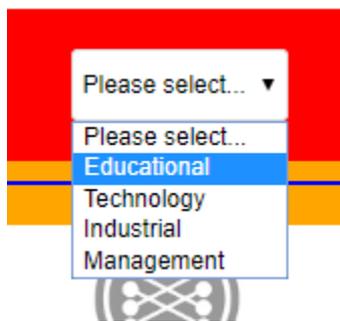
'Coullweb' Work Experience Documentation – Daniel Wilkins

The outcome of these changes can now be viewed below:

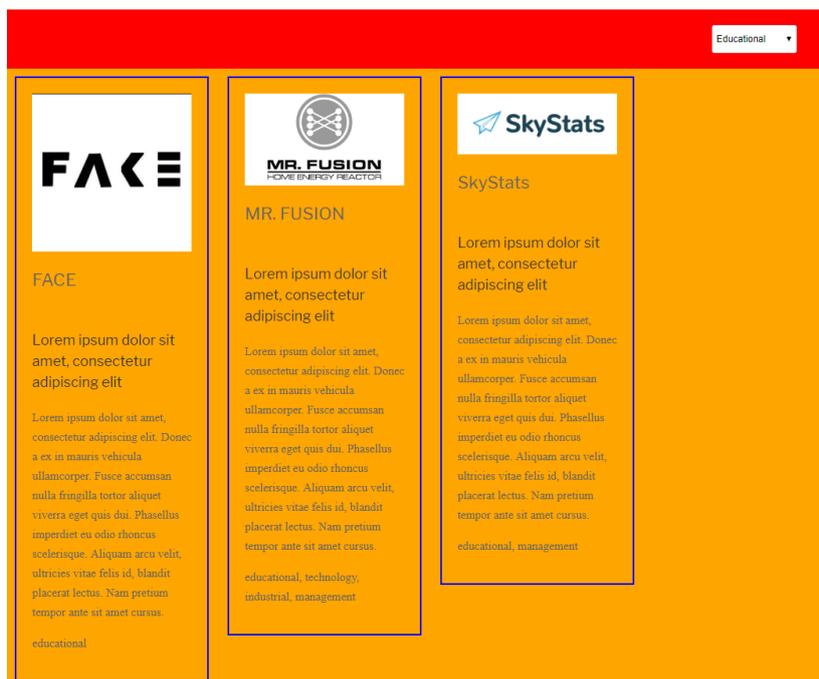
All posts being Displayed upon Loading the Page



Selecting a Category/Filter



The Result of this on the Page (Now only those with the 'Educational' tag were Displayed)



'Coullweb' Work Experience Documentation – Daniel Wilkins

This was the final task of the second day of the work experience at 'Coullweb' of which I achieved many new things in some areas that I had never explored before. I believe this to have helped improve my 'WordPress' skills, ultimately a skill which is highly demanded to have as a front-end website developer.

Days 3, 4 and 5 – Creating Account and Login Forms

Introduction

This task lasted for the longest during my work experience at 'Coullweb' and I was required to create a system where a user could create an account which would then be added to a table in a database. From completing the previous task, the user would then need to be able to log into the system where their details would be displayed as well as being able to log out of the system. Furthermore, if the user had forgotten their password, they would have the option of resetting this through a password reset form. As well as making the previously mentioned aspects function, there was a requirement of integrating security into the system to help introduce myself to this area. The main aspects of the brief can be viewed below, as written by the developer:

- Create a 'Create an Account' page
- Create a 'Login' page
- Create an 'Account' page
- Use PHP and MySQL to connect these aspects to a database in order to store details and allow users to log in to their account page

Creating the 'Login' Form for this Task

Before linking the 'login' form to the database, I first had to structure and style the form so I began by integrating a simple form from 'w3Schools' and making alterations to this:

The Simple form to Integrate from 'w3Schools'

PHP - A Simple HTML Form

The example below displays a simple HTML form with two input fields and a submit button:

Example

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Run example »

The Final HTML Code for the 'Login' Form

```
<?php
/*
 * Template Name: Login
 */

/* Login code PHP goes here */

get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="login_form_container">
      <form action="/" method="post" class="login_form">
        <h2>Login</h2>
        <hr>
        <i>Username:</i> <input type="text" name="username" class="input_fields" placeholder="Enter your username..."><br>
        <i>Password:</i> <input type="text" name="password" class="input_fields" placeholder="Enter your password..."><br>
        <input type="submit" class="submit_button">
      </form>
      <?php echo the_content(); ?>
      <?php
    ?>
    </div>
  </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The Final CSS Code for the 'Login' Form

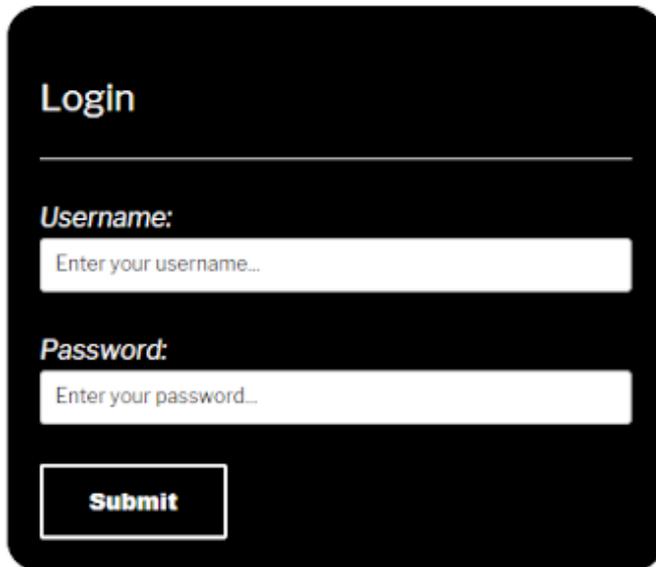
```
/* Login Form Styles */  
  
.login_form h2{  
  color: white;  
}  
  
.input_fields {  
  width: 100%;  
  height: auto;  
}  
  
.login_form_container {  
  width: 400px;  
  height: auto;  
  box-sizing: border-box;  
  background-color: black;  
  display: block;  
  margin: auto;  
  padding: 20px;  
  color: white;  
  border-radius: 20px;  
}  
  
.submit_button {  
  background-color: black!important;  
  color: white!important;  
  border: 2px solid white!important;  
  transition: 0.5s;  
}  
  
.submit_button:hover {  
  background-color: white!important;  
  color: black!important;  
  border: 2px solid white!important;  
  transition: 0.5s;  
}
```

As will be evident above, I created a 'login_form_container' which acted as a main container for the 'login_form' container in case I needed to change anything inside the main container. For this I added the main styles for the form such as the width and height as well as displaying it as a 'block' and setting the 'margin' to 'auto' to centre the form in the centre of the page. The colour of the text was also set in this container to white and 'padding' was added to space the elements inside from the outside of this container. For the 'input_fields' 'class', I set the width to '100%' as this would only ever be the width of the container they were in and therefore this would be the most beneficial way to implement this with regards to scaling down the screen resolution.

With regards to the submit button, I assigned a 'class' called 'submit_button' where the colours of the button/link were set, both for hovering over and not hovering over. It is worth noting that I had to apply 'important' in this situation due to the fact that there were other styles already created for other aspects which interfered with this and this was something advised by the developer at 'Coulweb'.

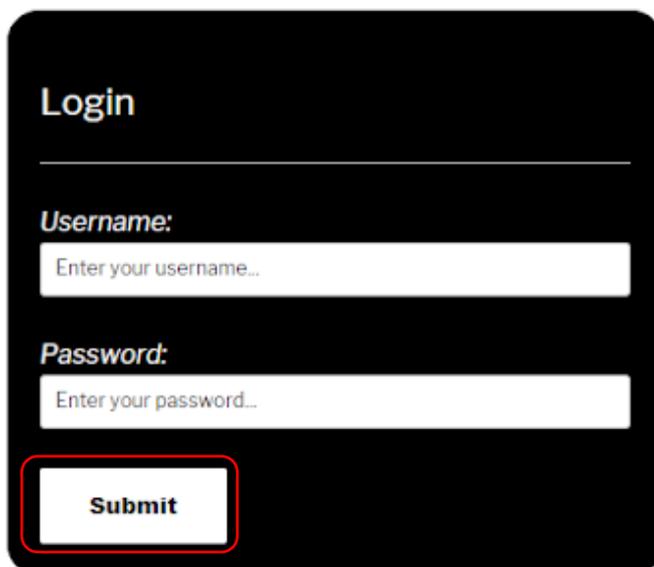
The outcome of the previous 'HTML' and 'CSS' code can be viewed below:

The 'Login' Form Outcome – Before Hovering over the 'Submit' Button/link



A screenshot of a login form on a black background. The form is titled "Login" in white text. Below the title is a horizontal line. There are two input fields: "Username:" with the placeholder text "Enter your username..." and "Password:" with the placeholder text "Enter your password...". At the bottom of the form is a white "Submit" button with a black border.

The 'Login' Form Outcome – Whilst Hovering over the 'Submit' Button/link



A screenshot of the same login form as above, but with a red border around the "Submit" button, indicating it is being hovered over.

Establishing the Table for the Database to Contain User Details

After the 'login' form had been created, the next step was to create the table in the database to store the data of each user's login details. The outcome of what I created after also being helped can be viewed below:

Table name: Add column(s)

| Name | Type | Length/Values | Default | Collation | Attributes | Null | Index | Comments | Virtuality |
|----------|---------|---------------|---------|-----------|------------|--------------------------|---------|---|------------|
| user_id | INT | | None | | | <input type="checkbox"/> | PRIMARY | PRIMARY <input checked="" type="checkbox"/> | |
| username | VARCHAR | | None | | | <input type="checkbox"/> | UNIQUE | username <input type="checkbox"/> | |
| password | VARCHAR | | None | | | <input type="checkbox"/> | INDEX | password <input type="checkbox"/> | |
| email | VARCHAR | | None | | | <input type="checkbox"/> | UNIQUE | email <input type="checkbox"/> | |

As can be seen above, the 'user_id' was set as the 'Primary Key' with an 'Auto-increment' feature which would automatically assign a user 'ID' to a new user who had recently created an account, incrementing by one each time. This was set as a 'Primary Key' as this was something that was advised by the developer. I thought that it would be beneficial to apply 'VARCHAR' to the 'username', 'database' and 'email' sections as all of these would contain a mixture of different characters. It is worth noting that I assigned the 'UNIQUE' index to the 'username' row as I thought that this would be something that wouldn't be allowed to be duplicated and so therefore would need to be unique each time. I did this for the 'email' field as well for the same reason. The reason I assigned 'INDEX' to the 'password' field was because, as a result of research, I then understood that this would need to be assigned as more than one user could have the same password as stated by the developer at 'Coullweb'. This table was created based on help and through 'Stack Overflow' research which will be credited at the end of this document.

When setting the 'index' for each field, I needed to name the 'index', selecting 'Go' once finished and an example of this can be viewed below:

Naming an 'index' for one of the fields in the Table and Selecting 'Go'

Add index

Index name:

Index choice:

+ Options

| Column | Size |
|--|----------------------|
| <input type="text" value="user_id [int]"/> | <input type="text"/> |

'Coulweb' Work Experience Documentation – Daniel Wilkins

After being helped by the developer, as the initial table shown above was incorrect in some places, the final table then appeared as shown below:

The Table for the User Details

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|--------------------------|------|-----------------------|--------------------|------------|------|---------|----------|----------------|--------|
| <input type="checkbox"/> | 1 | user_id | | | No | None | | AUTO_INCREMENT | |
| <input type="checkbox"/> | 2 | username | utf8mb4_unicode_ci | | No | None | | | |
| <input type="checkbox"/> | 3 | password | utf8mb4_unicode_ci | | No | None | | | |
| <input type="checkbox"/> | 4 | email | utf8mb4_unicode_ci | | No | None | | | |
| <input type="checkbox"/> | 5 | password_reset | utf8mb4_unicode_ci | | No | None | | | |

As would have been evident above, an extra field had been added called 'password_reset' that would allow users to reset their password if forgotten and also a character limit had been set for each of the fields apart from the 'user_id' field due to the fact that this was an area where the user wouldn't add this themselves but the database would add this automatically instead.

Creating the 'Create an Account' Form

After establishing the database, the next required aspect was to create the 'Create an Account' form of which I implemented the code from the 'Login' form and altered this where necessary:

The Final HTML Code for the 'Create an Account' Page (Altered Aspects Highlighted)

```
<?php
/*
 * Template Name: Account Setup
 */

/* Login code PHP goes here */

get_header(); ?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="login_form_container">
      <form action="/" method="post" class="login_form">
        <h2>Create your Account...</h2>
        <hr>
        <i>Username:</i> <input type="text" name="username" class="input_fields" placeholder="Enter your username..."><br>
        <i>Email:</i> <input type="text" name="username" class="input_fields" placeholder="Enter your email address..."><br>
        <i>Password:</i> <input type="text" name="password" class="input_fields" placeholder="Enter your password..."><br>
        <input type="submit" class="submit_button" value="Sign Up">
      </form>
      <?php echo the_content(); ?>
    </div>
  </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The Final CSS Code for the 'Create an Account' Page

```
/* Login Form Styles */

.login_form h2{
  color: white;
}

.input_fields {
  width: 100%;
  height: auto;
}

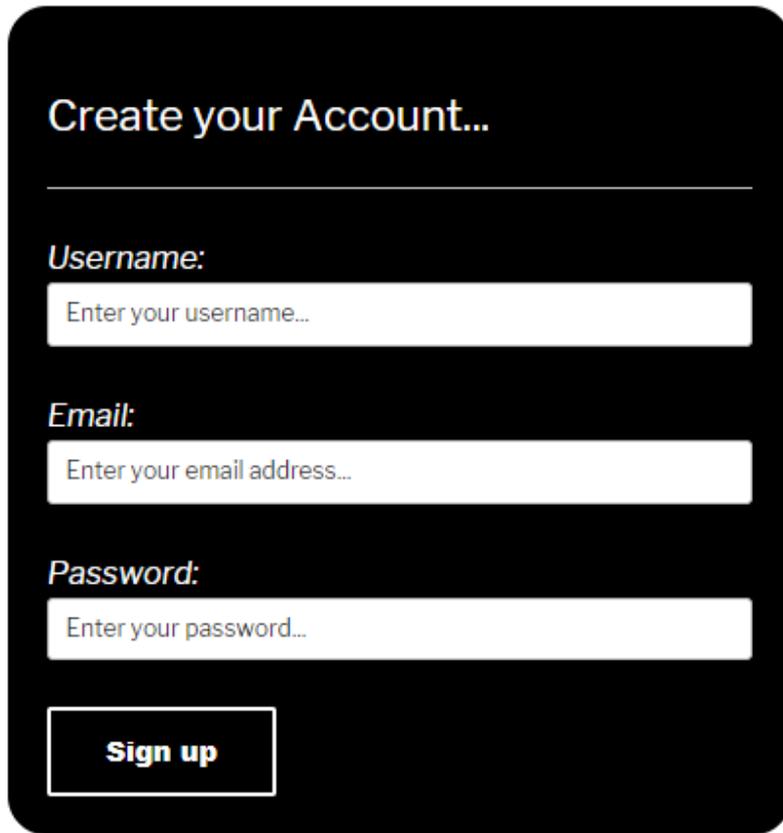
.login_form_container {
  width: 400px;
  height: auto;
  box-sizing: border-box;
  background-color: black;
  display: block;
  margin: auto;
  padding: 20px;
  color: white;
  border-radius: 20px;
}

.submit_button {
  background-color: black!important;
  color: white!important;
  border: 2px solid white!important;
  transition: 0.5s;
}

.submit_button:hover {
  background-color: white!important;
  color: black!important;
  border: 2px solid white!important;
  transition: 0.5s;
}
```

As would have been evident above, the only aspects that changed were the descriptions on the form and not any of the code itself. Due to the fact that the purpose of the task was to introduce myself to databases and 'MySQL', this was something that didn't need to be changed:

The Final Outcome of the 'Create an Account' Form



Create your Account...

Username:
Enter your username...

Email:
Enter your email address...

Password:
Enter your password...

Sign up

Allowing for the 'Create an Account' Form to Function and add Data to the Database

After creating the 'Create an Account' form, I was then helped to add 'PHP' and 'MySQL' code in order to create a connection to the database and further allow new registered users to add their details to this database:

Adding the PHP and MySQL Code into the file

```
<!-- Account Setup code PHP goes here -->
<?php
$servername = "localhost";
$username = "blank";
$password = "blank";
$dbname = "blank";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

//Create account

$username = $_POST['username'];
$email = $_POST['email'];
$password = $_POST['password'];

$sql = "INSERT INTO wp_login (username, email, password)
VALUES ('$username', '$email', '$password')";

//$sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

mysqli_close($conn);
?>
```

As can be seen above, the first highlighted piece of code created variables to link to the details of the database which would then be called into the 'mysqli_connect' function in order to create a connection to the database. If this wasn't able to be completed, then an error would appear showing 'Connection failed'. After this, the third highlighted piece of code collected the details from the form by the 'name' attribute for the '\$_POST' section and then the 'INSERT INTO' allowed this data to be inserted into the database table which in this case was 'wp_login'. For the 'VALUES' section, I was advised to place '\$' symbols in front of each aspect such as 'username' as before this was placing the data into the database as 'strings' as these values. The result of the code above can be viewed below:

The Data was now being Inserted into the Database when Creating an Account

+ Options

| | user_id | username | password | email | password_reset |
|---|---------|----------|--------------|--------------|----------------|
| <input type="checkbox"/> Edit Copy Delete | 1 | username | password | email | |
| <input type="checkbox"/> Edit Copy Delete | 2 | test | ssssss | fffff | |
| <input type="checkbox"/> Edit Copy Delete | 3 | testing | testtesttest | testing@test | |
| <input type="checkbox"/> Edit Copy Delete | 4 | test | gfdgfg | tdftd | |

↑ Check all With selected: Edit Copy Delete Export

Allowing for the 'Login' form to Function Properly

Checking Entered Details into the Form

After being able to allow users to register their data in the database, I then was told that I now needed to make sure that the person logging in could only login when entering the correct data that linked to the database/table. To begin, I had attempted at integrating the following code shown below into the login form file:

```
<?php
/*
 * Template Name: Login
 */

/* Login code PHP goes here */
get_header(); ?>

<?php
$servername = "localhost";
$username = "blank";
$password = "blank";
$dbname = "blank";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT username, password FROM wp_login";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if ($sql == 'username', 'password') {

} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

As is highlighted above, I thought I could use 'MySQL' to select data from the table in the database and then integrate an 'if' statement where if the variable '\$sql' was equal to both the 'username' and 'password', then this would allow users to login but if not, then a message of '0 results' would be shown instead.

However, after checking with the developer at 'Coullweb', I was then advised to change the code to the following shown below:

Changing the Code based on Advice by the Developer

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT username, password FROM wp_login WHERE username='$username' AND password='$password'";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        var_dump($row);
    }
}
```

As can be seen above, the highlighted code selects the required data from the table 'wp_login', 'username' and 'password', and checks these details against what is posted through the '\$_POST' that is collected from the form to understand if these are correct. If they are and the number of rows is greater than 0, the data will then be shown to the user.

Creating a Persistent Login Function to Allow for Logging Out of the System

Following on from the previous aspect, I was then told that I now needed to create a persistent login function for the user when logging into the system. I was helped to produce the following code shown below:

The PHP and MySQL Code for this Section

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT username, password FROM wp_login WHERE username='$username' AND password='$password'";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    createSessionCookie($username, $conn);
}

function createSessionCookie($username, $conn) {
    $session_token = sessionTokenGenerator();
    $cookie_name = "login_token";
    setcookie($cookie_name, $session_token, time() + (86400 * 30), "/"); // 86400 = 1 day

    $sql = "UPDATE wp_login SET password_reset = '$session_token' WHERE username='$username'";

    $result = mysqli_query($conn, $sql);

    return "success";
}

function sessionTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

//if ($sql == 'username', 'password') {
//
//} else {
//    echo "0 results";
//}

mysqli_close($conn);
get_header();
?>
```

As is evident above, the first highlighted piece of code would only allow the function called 'createSessionCookie' to be executed if the number of rows in the table for the login was greater than 0. The 'createSessionCookie' would assign a function called 'sessionTokenGenerator' to the '\$session_token' variable. This function would create a random 'string' for the login session including a 'md5 hash' so that this would add better security and prevent anyone from being able to understand what the current login session was. As is also evident, the 'setCookie' function would create the 'Cookie', linking to the 'sessionTokenGenerator' function and the '\$sql' variable would update the 'password_reset' field in the table of the database with the newly created 'Session Token'.

Navigating the User to Another Page from the Login Function, Displaying their Data to them
The task was now to ensure that once entering the required details and selecting the submit button on the login form, that the user would now be navigated to an 'account summary' page where they could view their details. The current code I had at this moment in time can be viewed below:

The Current PHP, MySQL and HTML Code

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT user_id, username, password, email FROM wp_login";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    while($row = $result->fetch_assoc()) {
        echo "Your ID: " . $row["user_id"]. "<br>" "Your Username: " . $row["username"]. "Your Password: " . $row["password"]. "<br>" "Your Email: " . $row["email"].;
    }
} else {
    echo "0 results";
}

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <hr>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

As is highlighted above, the 'user_id', 'username', 'password' and 'email' fields would be selected from the table 'wp_login' in the database where if the number of rows in the table was greater than 0, then the details of the user would be outputted onto the page through the use of 'echo'. If this wasn't the case, then '0 results' would be outputted onto the page instead.

The Current CSS Code

```
/* Account Summary Page */

.account_summary_container {
    width: 100%;
    height: auto;
    background-color: black;
    color: white;
    text-align: center;
    padding: 30px;
}
```

As is evident above, I added a minimal amount of styling to the 'HTML' code on the page as the purpose of this task was to explore 'PHP' and 'MySQL'. I set the background colour to black and the text colour to white so that the information displayed would be able to be read on the page.

However, I then encountered an error which can be viewed below:

```
Parse error: syntax error, unexpected "Your Username: " (T_CONSTANT_ENCAPSED_STRING), expecting ',' or ';' in
```

```
wp/wp-content/themes/twentyseventeen/pages/my-account.php on line 24
```

As a result of experiencing this error, I then changed the previous shown code to the following displayed below after seeking advice:

Changing the Code after the Advice to Resolve the Previous Issue

```
if (mysqli_num_rows($result) > 0) {  
    while($row = $result->fetch_assoc()) {  
        echo "Your ID: " . $row["user_id"]. "<br>Your Username: " . $row["username"]. "Your Password: " . $row["  
            password"]. "<br>Your Email: " . $row["email"];  
    }  
} else {  
    echo "0 results";  
}
```

I was told that I had included a '.' at the end by the '\$row["email"]' of which wasn't needed and this was the reason why the error was appearing. After removing this, this now displayed the data:

The Data from the Table in the Database was now Displaying when Successfully Logging in

```
Your ID:1  
Your Username: usernameYour Password: password  
Your Email: emailYour ID: 2  
Your Username: testYour Password: ssssss  
Your Email: fffffYour ID: 3  
Your Username: testIngYour Password: testtesttest  
Your Email: testing@testYour ID: 5  
Your Username: Your Password:  
Your Email: Your ID: 6  
Your Username: user1Your Password: user11234  
Your Email: user1@gmail.com
```



Your Account Summary...

As can be seen above, the data was displaying for all users and not for the specific user logging in. Therefore, this was the next task to ensure that this was functioning properly. After finding an aspect on the 'w3Schools' website, I then integrated this into the code I had at the current time:

Adding the 'limit 1' Aspect to the Code

```
$sql = "SELECT user_id, username, password, email FROM wp_login limit 1"; //limit 1 will show same result for all  
//$sql = "SELECT user_id, username, email FROM wp_login";  
$result = mysqli_query($conn, $sql);
```

The Outcome of this Change on the Web Page (The Result was now Limited to 1)

```
Your ID:1  
Your Username:usernameYour Password:password  
Your Email:email
```



As is evident, only one result was now appearing. However, this would show the same result for each login with different details instead of showing the data relating to the person logging into the system.

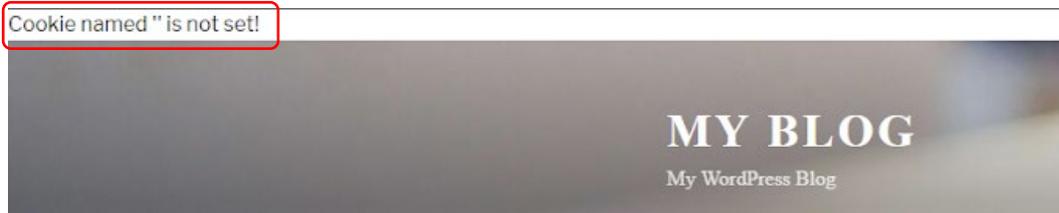
Whilst seeking advice for this, I was advised to use 'Cookies' and not 'Sessions' and as a result of this I was then helped to add the following to the code:

Adding the 'isset' Function into the Code

```
// Create connection  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
  
$sql = "SELECT user_id, username, password, email FROM wp_login limit 1"; //limit 1 will show same result for all  
// $sql = "SELECT user_id, username, email FROM wp_login";  
$result = mysqli_query($conn, $sql);  
  
if(!isset($_COOKIE[$cookie_name])) {  
    echo "Cookie named '" . $cookie_name . "' is not set!";  
} else {  
    echo "Your ID: " . $row["user_id"]. "<br>Your Username: " . $row["username"]. "Your Password: " . $row["password"].  
        "<br>Your Email: " . $row["email"];  
}  
  
mysqli_close($conn);  
  
get_header();  
?>  
  
<div id="primary" class="content-area">  
    <main id="main" class="site-main" role="main">  
        <div class="account_summary_container">  
            <h1>Your Account Summary...</h1>  
            <hr>  
        </div>  
        <div class="container">  
            <?php echo the_content(); ?>  
        </div>  
    </main><!-- #main -->  
</div><!-- #primary -->  
  
<?php  
get_footer();
```

Adding the 'isset' function meant that now if someone attempted to view details on the page without a session, created as a result of logging in, then the message would appear stating that a 'Cookie' had not been set. I then tried to test if this would work but had realised that the 'Cookie' name wasn't displaying on the page as will be evident below:

The 'Cookie' name wasn't being Displayed on the Page



I thought that the solution to this would involve placing the function for creating the 'Cookie' in the same file as the code shown previously. However, I was told that I needed to add a variable name instead that would relate to the variable '\$cookie_name' within the file. Therefore, I did this and then the fields appeared that would display the user's data when logging in:

Adding the Variable name to the Variable in the file

```
$cookie_name = "login_token";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT user_id, username, password, email FROM wp_login limit 1"; //limit 1 will show same result for all
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Your ID: " . $row["user_id"]. "<br>Your Username: " . $row["username"]. "<br>Your Password: " . $row["password"]. "<br>Your Email: " . $row["email"];
}

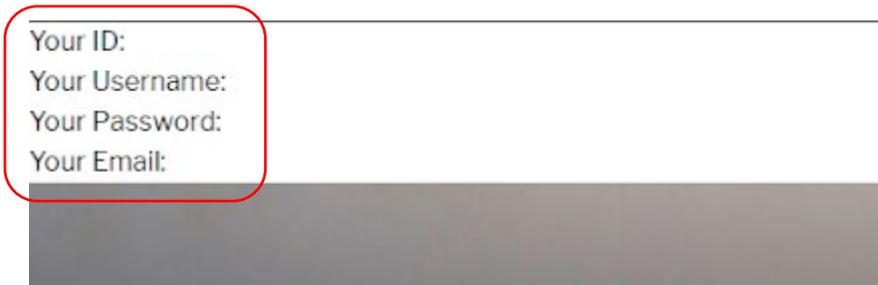
mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <hr>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The Fields were now Displaying but without the Data



I had then been advised to 'redirect' the user if they had logged in successfully to the account summary page. Whilst I began to undertake this task, I was advised and helped to change the code to the following shown below:

Changing the 'SET password_reset = '\$session_token' to 'SET session_token = '\$session_token'' in the file

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT username, password FROM wp_login WHERE username='$username' AND password='$password'";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    createSessionCookie($username, $conn);
    header("Location: your_page.php");
}

function createSessionCookie($username, $conn) {
    $session_token = sessionTokenGenerator();
    $cookie_name = "login_token";
    setcookie($cookie_name, $session_token, time() + (86400 * 30), "/"); // 86400 = 1 day

    $sql = "UPDATE wp_login SET session_token = '$session_token' WHERE username='$username'";

    //var_dump($conn);

    $result = mysqli_query($conn, $sql);

    return "success";
}

function sessionTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

//if ($sql == 'username', 'password') {
//
//} else {
//    echo "0 results";
//}

mysqli_close($conn);

get_header();
?>
```

As is evident above, I changed this aspect as I was told that currently, the session token was being generated into the password reset field in the table of the database and that this needed to be placed into the relevant field.

'Coulweb' Work Experience Documentation – Daniel Wilkins

I was then advised and helped to create a new field called 'session_token' in the table of the database and delete the current tokens placed into the 'password_reset' field:

Adding the New Field to the Table in the Database

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------------|--------------|--------------------|------------|------|---------|----------|----------------|---|
| 1 | user_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop Primary Unique Index Spatial Fulltext Distinct values |
| 2 | username | varchar(255) | utf8mb4_unicode_ci | | No | None | | | Change Drop Primary Unique Index Spatial Fulltext Distinct values |
| 3 | password | varchar(255) | utf8mb4_unicode_ci | | No | None | | | Change Drop Primary Unique Index Spatial Fulltext Distinct values |
| 4 | email | varchar(255) | utf8mb4_unicode_ci | | No | None | | | Change Drop Primary Unique Index Spatial Fulltext Distinct values |
| 5 | password_reset | varchar(255) | utf8mb4_unicode_ci | | No | None | | | Change Drop Primary Unique Index Spatial Fulltext Distinct values |
| 6 | session_token | varchar(255) | utf8mb4_unicode_ci | | No | None | | | Change Drop Primary Unique Index Spatial Fulltext Distinct values |

Deleting the Current Tokens in the 'password_reset' field

| | user_id | username | password | email | password_reset | session_token |
|---|---------|----------|--------------|-----------------|----------------------------------|---------------|
| 1 | 1 | username | password | email | | |
| 2 | 2 | test | ssssss | fffff | 8733084e8134024a6c52ad8784c01748 | |
| 3 | 3 | testing | testtesttest | testing@test | | |
| 5 | 5 | | | | 0a88558bba5120941734c404deef6d1d | |
| 6 | 6 | user1 | user11234 | user1@gmail.com | e15137f966b32b99244885b3173985d3 | |

After completing this, I then continued to attempt to redirect the user to the account summary page and changed the integrated code from a source online to suit the required link within this task:

Adding the Relevant file Link and Changing where Relevant

```
if (mysqli_num_rows($result) > 0) {  
    createSessionCookie($username, $conn);  
    header("Location: my-account.php");  
}
```

However, after expecting this to function successfully, I then encountered an error where the requested page couldn't be found:

The Appearing Error on the Page

OOPS! THAT PAGE CAN'T BE FOUND.

It looks like nothing was found at this location. Maybe try a search?

Search...

After experiencing the previous error, I was helped to understand that this was due to the fact that I had entered the wrong link and because there was a row that had empty values, meaning that the user would be logged in straight away. Therefore, I changed the link to the correct link and ensured that there were values entered into the row in the table of the database to prevent a straight redirect when navigating to the login form page:

Changing the Link to the Correct Link

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT username, password FROM wp_login WHERE username='$username' AND password='$password'";
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    createSessionCookie($username, $conn);
    header("Location: /wp/my-account");
}

function createSessionCookie($username, $conn) {
    $session_token = sessionTokenGenerator();
    $cookie_name = "login_token";
    setcookie($cookie_name, $session_token, time() + (86400 * 30), "/"); // 86400 = 1 day

    $sql = "UPDATE wp_login SET session_token = '$session_token' WHERE username='$username'";

    //var_dump($conn);

    $result = mysqli_query($conn, $sql);

    return "success";
}

function sessionTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

//if ($sql == 'username', 'password') {
//
//} else {
//    echo "0 results";
//}

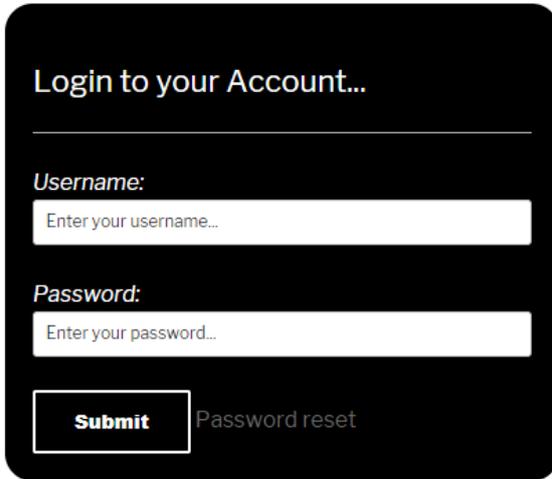
mysqli_close($conn);

get_header();
?>
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

Now when navigating to the login page, the straight redirect didn't happen and I decided to test if this redirected myself to the account summary page when logging in with details:

The Form now Appeared with no Straight Redirect



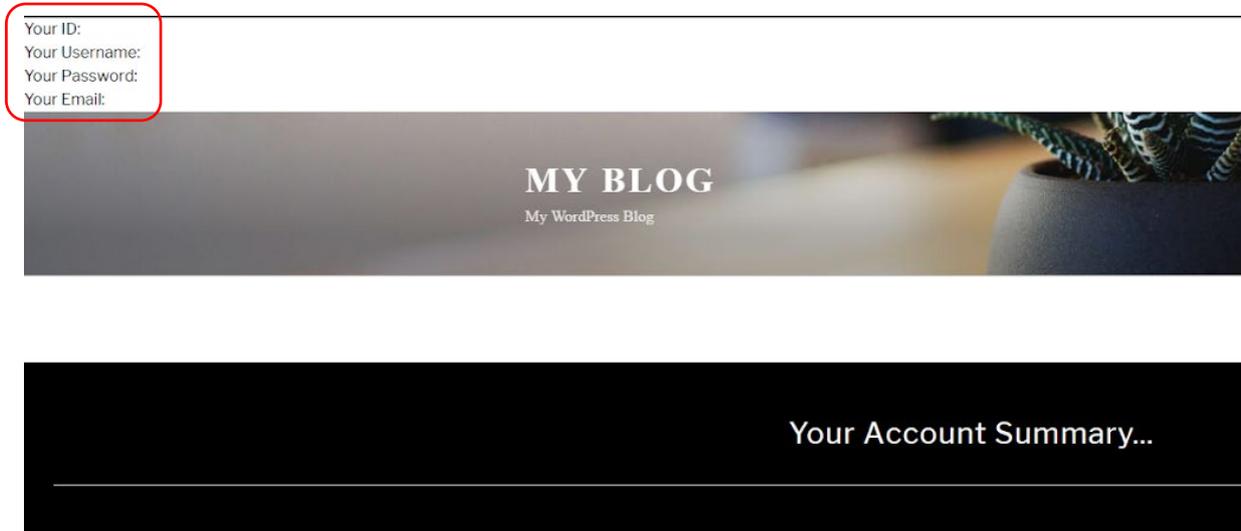
Login to your Account...

Username:
Enter your username...

Password:
Enter your password...

Submit Password reset

The Outcome after Logging in (Successful login but no Details were Appearing)



Your ID:
Your Username:
Your Password:
Your Email:

MY BLOG
My WordPress Blog

Your Account Summary...

After encountering the previous issue and after undertaking research, I then attempted at integrating the 'mysqli_fetch_row' function as I thought this may help to collect the data from the rows of the database and display them through the 'echo' function:

Implementing the 'mysqli_fetch_row' Code from 'php.net' into the file

```
$cookie_name = "login_token";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT user_id, username, password, email FROM wp_login"; //limit 1 will show same result for all
//$sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

if (!$result) {
    echo 'Could not run query: ' . mysqli_error();
    exit;
}
$row = mysqli_fetch_row($result);

echo $row[0];
echo $row[1];
echo $row[2];
echo $row[3];

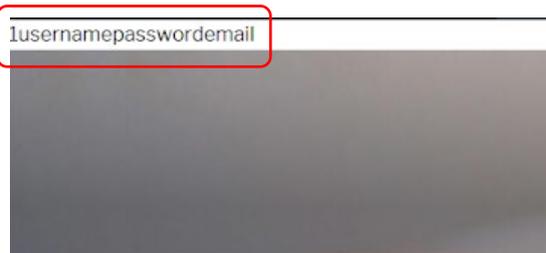
mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <hr>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The Outcome on the Page



As is evident above, this did output information from the database but related to the names of the fields and not the actual data from these fields.

As a result of understanding that this didn't work, I then thought about adding a 'Cookie' variable as I thought that this may help to distinguish between different user data and as a result display this data onto the page through the use of 'echo' for the '\$row' variable which would link to collecting the data relating to this 'Cookie' variable:

Adding the 'Cookie' Variable to the file

```
$cookie_name = "login_token";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT user_id, username, password, email FROM wp_login"; //limit 1 will show same result for all
// $sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql, $cookie_name);

if (!$result) {
    echo 'Could not run query: ' . mysqli_error();
    exit;
}
$row = mysqli_fetch_row($result);

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <p><?php echo $row[0];?></p>
            <p><?php echo $row[1];?></p>
            <p><?php echo $row[2];?></p>
            <p><?php echo $row[3];?></p>
            <hr>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

'Coullweb' Work Experience Documentation – Daniel Wilkins

However, this was unsuccessful and therefore, I asked for some advice and was helped to change the code above to the following displayed below as well as being told what was happening within this changed code:

Making Alterations to the Code through Help from the Developer at 'Coullweb'

```
$cookie_name = "login_token";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$session_cookie = $_COOKIE['login_token']; //Adding this to pull cookie into page

$sql = "SELECT user_id, username, password, email FROM wp_login WHERE session_token='$session_cookie'"; //Making the
session token for the browser match that of the database
//$sql = "SELECT user_id, username, email FROM wp_login";
$result = mysqli_query($conn, $sql);

$row = mysqli_fetch_row($result); //Moved to declare variable above

if (mysqli_num_rows($result) > 0) { //Adding this here to assign user id to relate to database
    $user_id = $row[0];
}

if (!$result) {
    echo 'Could not run query: ' . mysqli_error();
    exit;
}

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <p><?php echo "<i>Your User ID: </i>" . $row[0];?></p>
            <p><?php echo "<i>Your Username: </i>" . $row[1];?></p>
            <p><?php echo "<i>Your Password: </i>" . $row[2];?></p>
            <p><?php echo "<i>Your Email: </i>" . $row[3];?></p>
            <hr>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

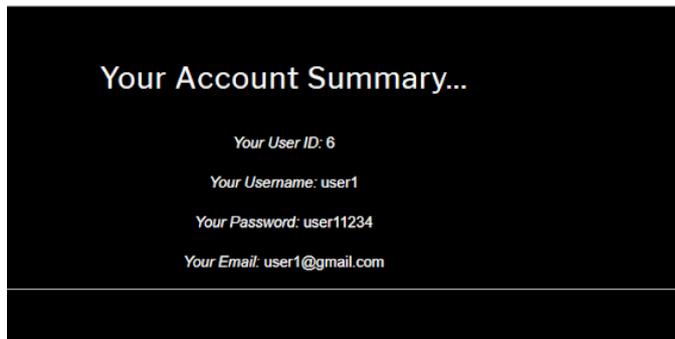
<?php
get_footer();
```

As is evident above, the first piece of highlighted code allowed for the 'Cookie' to be placed into the page through the use of the 'superglobal' variable called '\$_COOKIE' and the '\$sql' variable allowed for the data to be selected for the user where the session token of that on the page matched that of what was included in the table within the database. The third piece of code had been moved so that the variable '\$result' could be used after being declared above. The fourth piece of highlighted code related to assigning the user's 'ID' so that the data shown would be specific to them. The final piece of highlighted code relates to displaying the user's data in the form of 'paragraph' tags.

'Coulweb' Work Experience Documentation – Daniel Wilkins

I decided to then test to see if the data would now be displayed specific to the user when logging into the system and this worked as will be evident below:

The Details were now being Displayed Specific to the User



Allowing Users to Logout of the Account Summary Page

Another requirement of the account summary page was to ensure that users could logout successfully. To begin, the current code for this section on the account summary page appeared as shown below:

The Current Code with Initial Changes

```
if(isset($session_cookie)):
    setcookie("/wp/my-account/?logout=true", '', time()-3600, '/');
endif;

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="account_summary_container">
            <h1>Your Account Summary...</h1>
            <hr>
            <p><?php echo "<i>Your User ID: </i>" . $row[0];?></p>
            <p><?php echo "<i>Your Username: </i>" . $row[1];?></p>
            <p><?php echo "<i>Your Password: </i>" . $row[2];?></p>
            <p><?php echo "<i>Your Email: </i>" . $row[3];?></p>
            <!--<input type="submit" class="submit button" value="Logout"-->
            <a href="/wp/my-account/?logout=true">Logout</a>
        </div>
        <div class="container">
            <?php echo the_content(); ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

As is seen above, the current code allowed for showing of a 'GET' request in the browser and I was helped to achieve this part of the task.

The 'GET' request Showing in the 'URL' of the Browser

</wp/my-account/?logout=true>

After this process, I then tried to allow for the logout function to successfully work and therefore began by changing the 'if' statement to the following:

Making Alterations to the 'if' Statement in the Code

```
if(isset($session_cookie and $_GET == "/wp/my-account/?logout=true")) :
    setcookie("login_token", '', time()-3600, '/');
endif;
```

From my knowledge, I believed the code above would set a new 'Cookie' that would delete the existing 'Cookie' only if the '\$session_cookie' and '\$_GET' would be equal to the 'URL' shown. However, after seeking advice, I was helped to change this to the following displayed below:

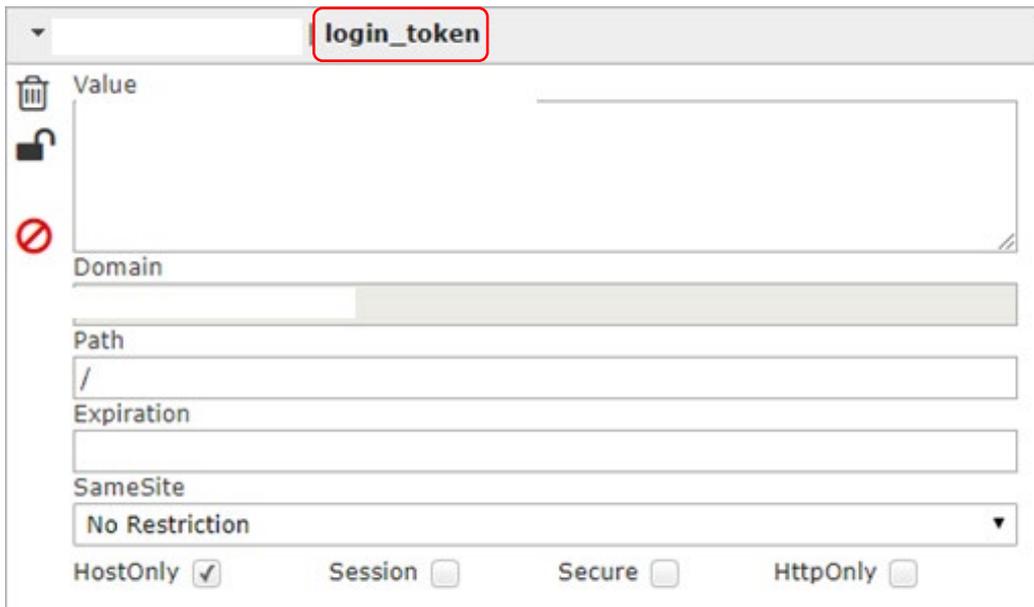
The Code following Help from the Developer at 'Coullweb'

```
if(isset($session_cookie) and $_GET['logout'] == "true") : //If value of logout is equal to being true, delete cookie
    setcookie("login_token", '', time()-3600, '/');
endif;
```

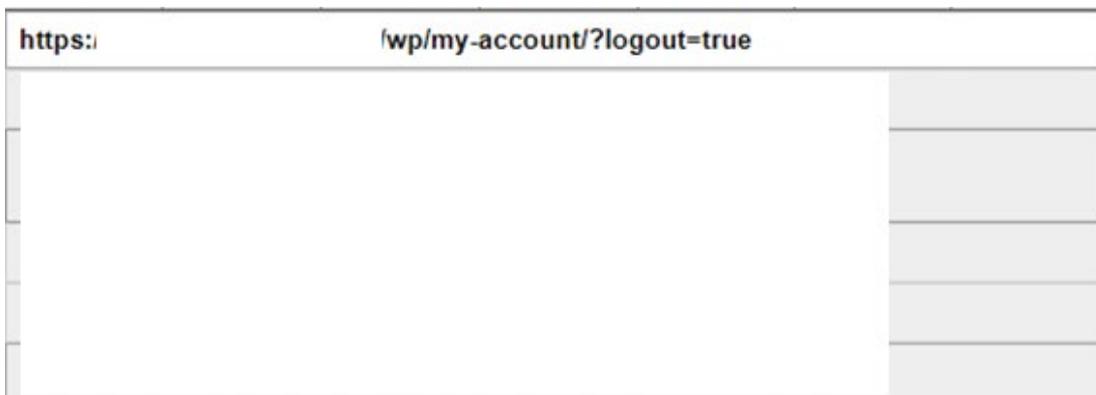
The code above would have the same concept as the code I had attempted to create through online resources but was structured in a manner which would function correctly.

After correcting this code, I then tested this to see if the 'Cookie' would disappear when selecting logout on the account summary page and this was successful as is evident below:

Logging into the System (The 'Cookie' Appeared)



Logging out of the System (The 'Cookie' had now Disappeared)



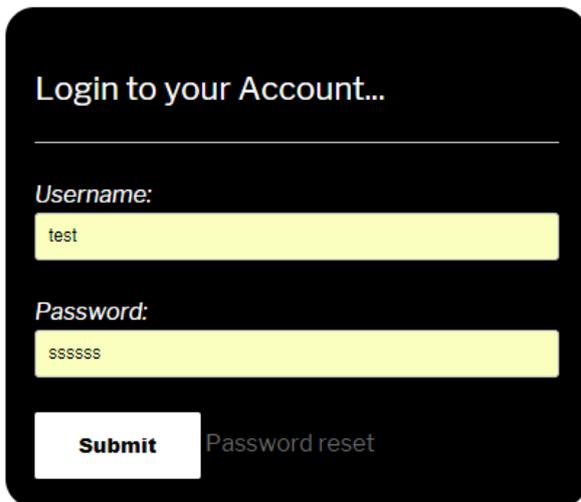
The next stage of the process was to ensure that the user would be redirected back to the login page when logging out instead of remaining on the page they were currently on as well as ensuring that they wouldn't be able to access the account summary page if not logged in. To begin I attempted at ensuring that the users would be returned to the login page when selecting logout:

Adding 'header' to the Code to Redirect Users when Selecting 'Logout'

```
if (mysqli_num_rows($result) > 0) {  
    $user_id = $row[0];  
    header("Location: /wp/login");  
}
```

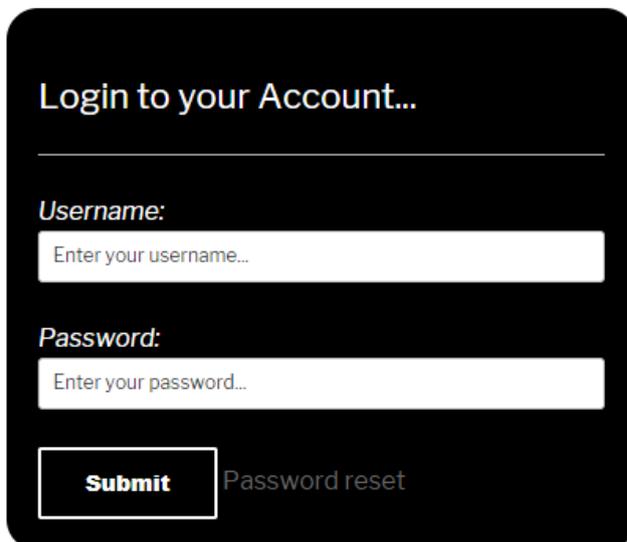
However, when testing to see if this worked, I had noticed that the user would be returned to the login page as soon as logging in which wasn't required:

The Process of Entering Details into the Login Form and Selecting Submit



A screenshot of a login form titled "Login to your Account...". The form has two input fields: "Username:" with the value "test" and "Password:" with the value "ssssss". Below the fields are two buttons: "Submit" and "Password reset".

The Process of Returning the User to the Login Page once Logging in



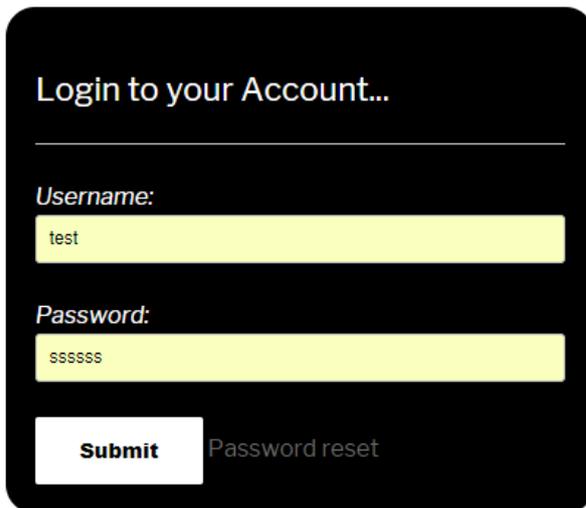
A screenshot of a login form titled "Login to your Account...". The form has two input fields: "Username:" with the placeholder text "Enter your username..." and "Password:" with the placeholder text "Enter your password...". Below the fields are two buttons: "Submit" and "Password reset".

As a result of experiencing this issue, I then added the 'header' section to the part of the code where the 'Cookie' would be deleted and this resolved the issue:

Moving the 'header' Section to the Correct Place in the Code

```
if(isset($session_cookie) and $_GET['logout'] == "true") : /
    setcookie("login token", '', time()-3600, '/');
    header("Location: /wp/login");
endif;
```

The Process of Entering Details into the Login Form and Selecting Submit

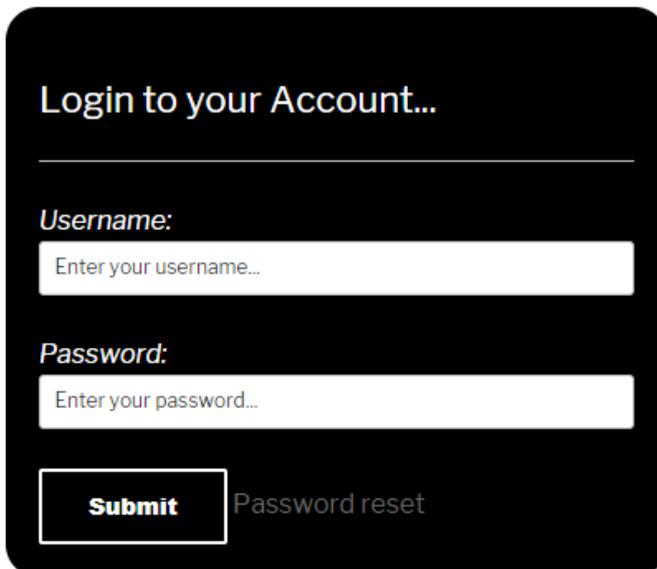


A screenshot of a login form titled "Login to your Account...". It features two input fields: "Username:" containing the text "test" and "Password:" containing "ssssss". Below the fields are two buttons: "Submit" and "Password reset".

The Process of Selecting 'Logout' on the Account Summary Page



The Process of being Logged Out Successfully



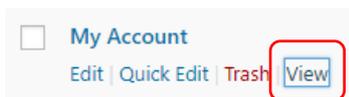
A screenshot of a login form titled "Login to your Account...". It features two input fields: "Username:" with the placeholder text "Enter your username..." and "Password:" with the placeholder text "Enter your password...". Below the fields are two buttons: "Submit" and "Password reset".

Now the logout function had been successfully created, I now moved forward and attempted to ensure that the user would be returned to the login page if trying to access the account summary page when not logged in. From my understanding, I thought that if I added an 'if' statement, stating that if a 'Cookie' hadn't been set, that the account summary page would return the user to the login page, that this would be successful. The code I implemented to ensure that this worked can be viewed below with the '!' indicating something that isn't set:

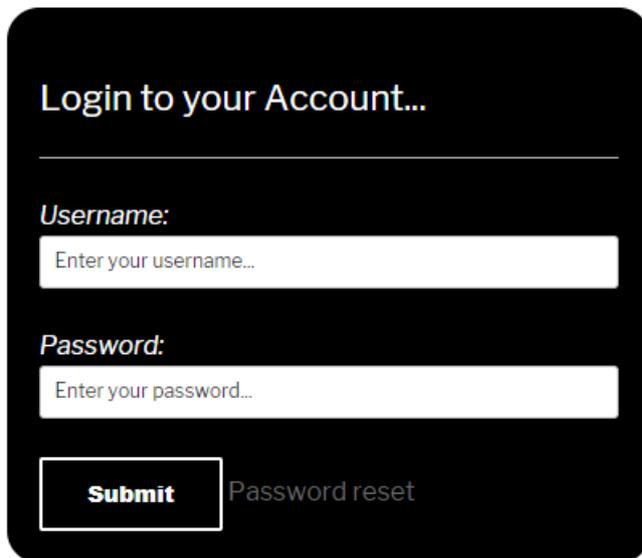
Integrating the 'if' Statement into the file

```
if (!$_COOKIE['login_token']) { //This returns the user to the login page if not logged in
    header("Location: /wp/login");
}
```

The Outcome of this on the Web Page (This was Successful) – Selecting 'View' on the Required Page



The Outcome of this on the Web Page (This was Successful) – The User had now been Returned to the Login Page



Password Reset Function

After managing to be able to make the previous sections function correctly with help, I was then told that I now needed to allow users to be able to reset their passwords if forgotten. The first step was to establish the 'Password Reset' page:

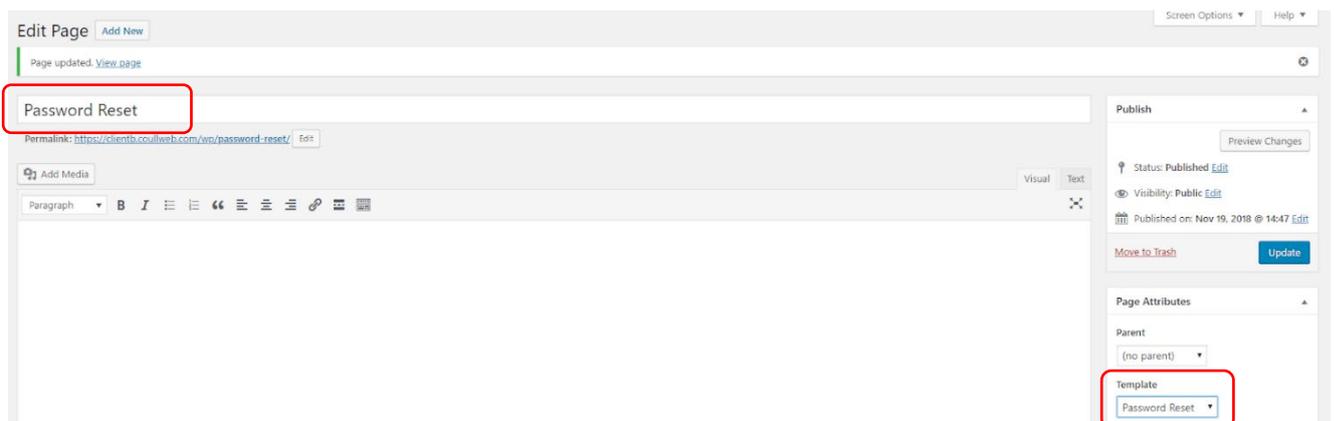
Creating a new file called 'password_reset.php'

```
password_reset.php
```

Assigning a Template name to the file to Link to WordPress

```
/*  
 * Template Name: Password Reset  
 */
```

Establishing the Page in WordPress and Assigning the Previous Template Name



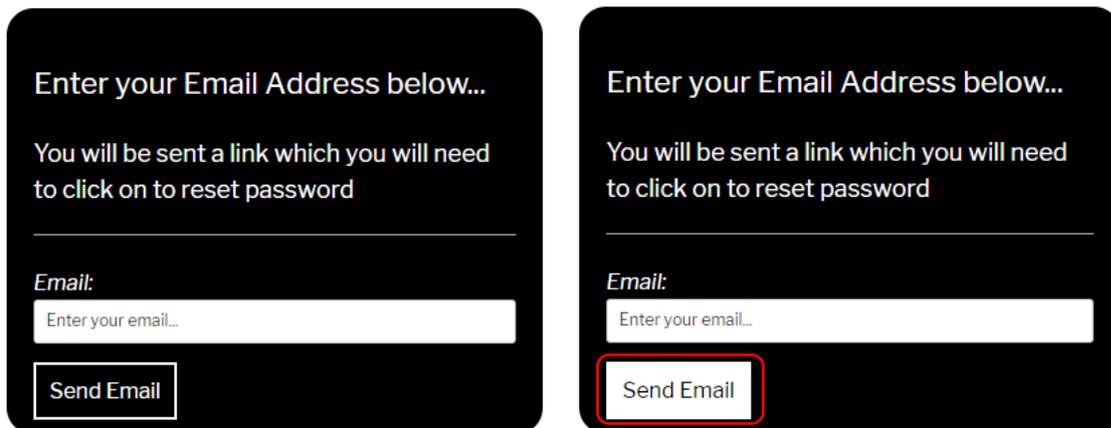
Adding the Required HTML/PHP Code to the '.php' file

```
get_header();  
?>  
  
<div id="primary" class="content-area">  
  <main id="main" class="site-main" role="main">  
    <div class="rp_form_container">  
      <form action="" method="post" class="reset_password_form">  
        <h2>Enter your Email Address below...</h2>  
        <h3>You will be sent a link which you will need to click on to reset password</h3>  
        <hr>  
        <i>Email:</i> <input type="text" name="email" class="input_fields" placeholder="Enter your email...><br>  
        <a href="#">Send Email</a>  
      </form>  
      <?php echo the_content(); ?>  
      <?php  
      ?>  
    </div>  
  </main><!-- #main -->  
</div><!-- #primary -->  
  
<?php  
get_footer();
```

Adding the Required CSS Code to the CSS file

```
/* Reset Password Page */  
  
.rp_form_container {  
  width: 400px;  
  height: auto;  
  box-sizing: border-box;  
  background-color: black;  
  display: block;  
  margin: auto;  
  padding: 20px;  
  color: white;  
  border-radius: 20px;  
}  
  
.reset_password_form h2{  
  color: white;  
}  
  
.reset_password_form h3{  
  color: white;  
  margin-top: -25px;  
}  
  
.rp_form_container a {  
  padding: 10px;  
  background-color: black;  
  border: 2px solid white;  
  color: white;  
  transition: 0.5s;  
}  
  
.rp_form_container a:hover {  
  background-color: white;  
  border: 2px solid white;  
  color: black;  
  transition: 0.5s;  
}
```

The Outcome of this Code on the Web Page (Before and Whilst Hovering over the 'Send Email' button/link)



As can be seen above, a similar layout and style was applied to the email form as the other forms shown previously. However, only one field was required as all that was needed was an email address to send the password reset link to.

Whilst beginning to add the functionality, I was advised I could either achieve this through 'PHP' or 'WordPress' functions and I chose to use 'WordPress' functions as I believed this to be a good area to explore in order to expand my skillset as a developer. Therefore, after undertaking research, I integrated the 'wp_mail' function into the code:

Integrating the 'wp_mail' Function into the file

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$to = $_POST['email']; //Changed email to to
$subject = 'Password Reset Email';
$message = 'Please click the link here'

$result = mysqli_query($conn, $sql);
wp_mail( $to, $subject, $message);
mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="rp_form_container">
            <form action="" method="post" class="reset_password_form">
                <h2>Enter your Email Address below...</h2>
                <h3>You will be sent a link which you will need to click on to reset password</h3>
                <hr>
                <i>Email:</i> <input type="text" name="email" class="input_fields" placeholder="Enter your email..."><br>
                <a href="#">Send Email</a>
            </form>
            <?php echo the_content(); ?>
            <?php
        ?>
        </div>
    </main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

As is highlighted above, first of all, variables would be created which would relate to different sections of the email. The '\$to' variable would be the email address in which the email would be sent to, the '\$subject' variable would be the subject of the email and the '\$message' variable would relate to the main content of the email. After establishing these variables, the 'wp_mail' function would use these variables to help send the email with all relevant aspects included.

After establishing the previous section, I was then helped to add the following highlighted aspects displayed below:

Adding the Highlighted Aspects to the Code

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$message = 'Please click the link <a href="#">here</a>';

$result = mysqli_query($conn, $sql);
wp_mail( $to, $subject, $message);
mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="rp_form_container">
            <form action="" method="post" class="reset_password_form">
                <h2>Enter your Email Address below...</h2>
                <h3>You will be sent a link which you will need to click on to reset password</h3>
                <hr>
                <i>Email:</i> <input type="text" name="email" class="input_fields" placeholder="Enter your email...
                "><br>
                <input type="submit" class="submit_button" value="Send Email">
                <!--<a href="">Send Email</a-->
            </form>
            <?php echo the_content(); ?>
            <?php
        ?>
    </div>
</main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

The aspects added were the link in the '\$message' variable and adding a submit button/link so that the form would function correctly.

In order to test this, I created a fictitious account for myself utilising my email address on my personal website as the other accounts created were not real. After doing this and then testing to see if I received the email, I then understood that this had worked successfully:

The Email had been Successfully Sent

Password Reset Email View Source

Date: 11/19/2018 (03:37:39 PM GMT) Save

From: WordPress

To: enquiries@danielhwilkins.co.uk

Text (1 KB)  

Please click the link here

'Coulweb' Work Experience Documentation – Daniel Wilkins

After managing to successfully get the previous aspect to work, I was then told that I needed to now ensure that resetting of passwords could only be done in certain circumstances in order to prevent anyone from being able to do this themselves. Therefore, to begin, I established the page in order to complete this task:

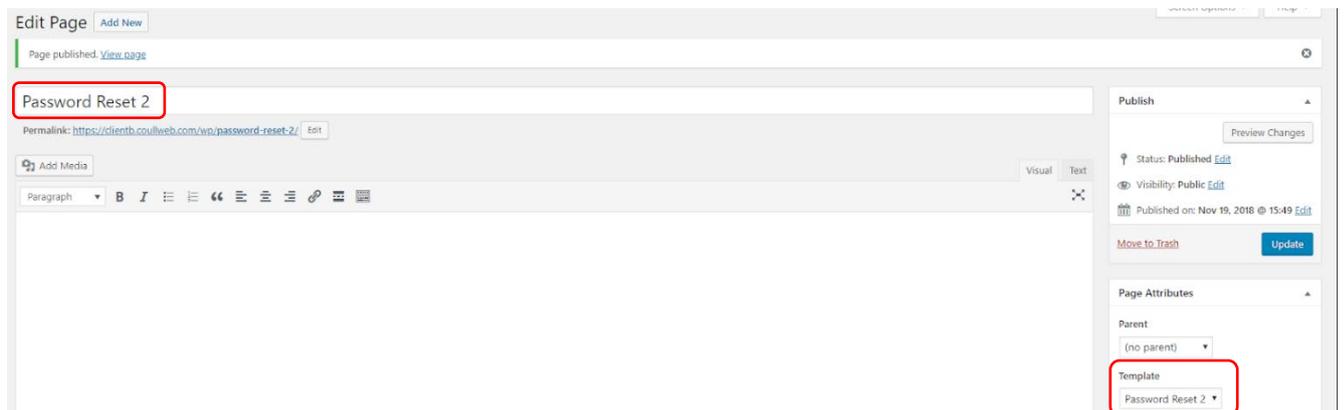
Creating the 'password_reset2.php' file

```
password_reset2.php
```

Assigning a Template name to Link to WordPress

```
<?php
/*
 * Template Name: Password Reset 2
 */
```

Establishing the Page on WordPress and Assigning the Previous Template to this



The screenshot displays the WordPress 'Edit Page' interface. The page title is 'Password Reset 2', which is highlighted with a red box. The permalink is 'https://clients.coulweb.com/wp/password-reset-2/'. The 'Page Attributes' sidebar on the right shows the 'Template' dropdown menu set to 'Password Reset 2', also highlighted with a red box. The 'Publish' section shows the page is 'Published' and was published on 'Nov 19, 2016 @ 15:49'.

After establishing all of the requirements for the new page, I then integrated the following 'PHP' and 'MySQL' code:

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];

$sql = "UPDATE password FROM wp_login WHERE email = $_GET['reset'] == 'true' ";
$result = mysqli_query($conn, $sql);

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="rp_form_container">
      <form action="" method="post" class="reset_password_form">
        <h2>Enter your new Password below...</h2>
        <hr>
        <i>New Password: </i> <input type="text" name="newpassword1" class="input_fields" placeholder="
        Enter your new password..."><br>
        <i>Confirm Password: </i> <input type="text" name="newpassword2" class="input_fields" placeholder="
        Enter your new password again..."><br>
        <a href="wp/password-reset-2/?reset=true"></a>
        <!--<input type="submit" class="submit_button" value="Reset Password">-->
        <!--<a href="">Send Email</a-->
      </form>
      <?php echo the_content(); ?>
      <?php
    ?>
  </div>
</main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

I believed that I first of all had to create two variables called '\$newpassword1' and '\$newpassword2' in order to be able to place the new password into the database where these would link to the 'name' attributes in the 'input' fields of the form. The second highlighted aspect demonstrates myself thinking that the database would need to be updated with the new values where the email was unique and also when the reset value equalled 'true', linking to the 'URL' on the 'href' aspect highlighted last.

However, after seeking advice from the developer, I was then advised and helped to change the following aspects of code:

Changing the link in the 'password_reset.php' file for Testing Purposes

```
$to = $_POST['email']; //Changed email to to
$subject = 'Password Reset Email';
$message = "Please click the link https://clientb.coullweb.com/wp/password-reset-2/?email='.$to;
```

As can be seen above, the link was changed so that this would be able to be selected in the email rather than utilising 'HTML' code as seen before.

'Coulweb' Work Experience Documentation – Daniel Wilkins

I was also advised and helped to change the code I currently had which I thought would help to collect the email and check this against the entered email into the form:

The Current PHP and MySQL Code

```
$sql = "UPDATE wp_login SET password WHERE email = '$_GET['email']'";
```

```
$newpassword1 = $_POST['newpassword1'];  
$newpassword2 = $_POST['newpassword2'];  
$email = $_GET['email'];
```

The Changed PHP and MySQL Code after Advice and Help

```
$newpassword1 = $_POST['newpassword1'];  
$newpassword2 = $_POST['newpassword2'];  
$email = $_GET['email'];  
$sql = "UPDATE wp_login SET password WHERE email = '$email'";
```

As is evident above, the only aspect which was required to be changed was the part highlighted as the email needed to check against the created variable '\$email'. However, the previous code wasn't doing this even though I thought it was because I was entering another 'GET' request and not relating to the variable itself.

After this, I was also advised to change another piece of code to the following which related to updating the 'password' field in the database of the person whose email identified them as well as if the new password and repeated new password were both identical. Originally, I hadn't included the '\$' symbol in front of '\$newpassword1' and '\$newpassword2' of which I was informed that the database wouldn't recognise these as they were in the form of 'strings' and not variables. The following code after the changes can be viewed below:

The Altered Code After Advice and Help

```
$sql = "UPDATE wp_login SET password = '$newpassword1' WHERE email = '$email' and '$newpassword1' = '$newpassword2'";
```

The next issue was the fact that I had the 'if' statement in the file below the 'SQL' statement when I was advised that this actually needed to be placed around the statement instead. Therefore, I did this and the following outcome can be viewed below:

Placing the if Statement around the SQL Statement in the file

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];
$email = $_GET['email'];

if ($newpassword1 == $newpassword2) {
    $sql = "UPDATE wp_login SET password = '$newpassword1' WHERE email = '$email'";
    $result = mysqli_query($conn, $sql);
    echo "Your Password has been Reset Successfully";
}

endif;

//if ($newpassword1 == $newpassword2) :
//    header("Location: /wp/login");
// endif;

mysqli_close($conn);

get_header();
?>

<div id="primary" class="content-area">
    <main id="main" class="site-main" role="main">
        <div class="rp_form_container">
            <form action="" method="post" class="reset_password_form">
                <h2>Enter your new Password below...</h2>
                <hr>
                <i>New Password: </i> <input type="text" name="newpassword1" class="input_fields" placeholder="
                Enter your new password..."><br>
                <i>Confirm Password: </i> <input type="text" name="newpassword2" class="input_fields" placeholder="
                Enter your new password again..."><br>
                <a href="wp/password-reset-2/?reset=true"></a>
                <!--<input type="submit" class="submit_button" value="Reset Password">-->
                <!--<a href="">Send Email</a>-->
            </form>
            <?php echo the_content(); ?>
            <?php
        ?>
    </div>
</main><!-- #main -->
</div><!-- #primary -->

<?php
get_footer();
```

I then tested to see if the password reset worked but this didn't and left the password field now blank instead:

The Password Field was now Blank within the Table in the Database



After seeking advice for this issue, I was then advised and helped to assign the '\$newpassword1' variable to the 'password_reset' field so that the new password would be placed into this field once changed:

Assigning the '\$newpassword1' Variable to the 'password_reset' field in the SQL Statement

```
if ($newpassword1 == $newpassword2) :  
$sql = "UPDATE wp_login SET password = '$newpassword1' and password_reset = '$newpassword1' WHERE email = '$email'";  
$result = mysqli_query($conn, $sql);  
echo "Your Password has been Reset Successfully";  
endif;
```

However, after testing again, I encountered an issue with regards to the fact that now the number '0' was appearing in the field instead:



I was then informed that now I needed to set up a token for the 'password_reset' field in the table so therefore I began to undertake this task and the code at this point can be viewed below:

```
$newpassword1 = $_POST['newpassword1'];  
$newpassword2 = $_POST['newpassword2'];  
$email = $_GET['email'];  
$pw_token = $_COOKIE['passwordreset_token'];  
if ($newpassword1 == $newpassword2) :  
$sql = "UPDATE wp_login SET password = '$newpassword1' WHERE password_reset='$pw_token'";  
$result = mysqli_query($conn, $sql);  
echo "Your Password has been Reset Successfully";  
endif;  
//if ($newpassword1 == $newpassword2) :  
// header("Location: /wp/login");  
// endif;  
if(isset($pw_token) and $_GET['reset'] == "true") : //If value of logout is equal to being true, delete cookie  
endif;
```

As can be seen above, at this stage I had created a new variable called '\$pw_token' and assigned the 'superglobal' variable '\$_COOKIE' whilst also passing this a value of 'passwordreset_token' into this variable. I then linked the '\$pw_token' into the 'SQL' statement so that this could link to the 'password_reset' field in the table of the database. The final highlighted piece of code related to the fact that if the token for the password reset was set and if the 'GET' request value of 'reset' was equal to 'true', then the user would be able to reset their password.

'Coullweb' Work Experience Documentation – Daniel Wilkins

Whilst starting this task, I was informed to change the 'Submit' button/link to an 'input' field rather than a set of 'anchor' tags. Whilst changing this aspect I was also informed it would be best to add a 'hidden' field into the form to understand if the password reset was working properly. Therefore, I did this as can be seen below, although at first, I had added an 'input' field with a value of 'hidden' which was incorrect as this caused the field to show the text 'hidden':

Changing the 'Submit' Button/Link to an 'input' field and Adding the 'hidden' 'input' field into the Code

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="rp_form_container">
      <form action="" method="post" class="reset_password_form">
        <h2>Enter your new Password below...</h2>
        <hr>
        <i>New Password: </i> <input type="text" name="newpassword1" class="input_fields" placeholder="
        Enter your new password..."><br>
        <i>Confirm Password: </i> <input type="text" name="newpassword2" class="input_fields" placeholder="
        Enter your new password again..."><br>
        <input type="submit" class="submit_button" value="Reset Password">
        <input type="hidden" value="hidden">
        <!--<a href="">Send Email</a>-->
        <!--<a href="wp/password-reset-2/?reset=true"></a>-->
      </form>
    </div>
  </main>
</div>
```

After progressing further with the task and adding new areas of code based on the 'session token' created before for the login form, I then had the following code at this current stage:

The Current Code at this Stage of the Work Experience

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];
$email = $_GET['email'];
$pw_token = $_COOKIE['passwordreset_token'];
$successful_pw = $_POST['successful_pw'];

if ($successful_pw == $newpassword1 and $newpassword2) : //If hidden input field is equal to both first and second
entered password, update password field with new password and echo a message saying 'Your Password has been Reset
Successfully'

$sql = "UPDATE wp_login SET password = '$successful_pw' WHERE password_reset='$pw_token'";

$result = mysqli_query($conn, $sql);

echo "Your Password has been Reset Successfully";

    endif;

//If the number of rows is greater than 0 in the database, execute the function called 'createSessionCookie' and
return user to login screen once the form has been submitted
if (mysqli_num_rows($result) > 0) {
    createSessionCookie($username, $conn);
    header("Location: /wp/login");
}

// if(isset($pw_token) and $_GET['reset'] == "true") : //If value of logout is equal to being true, delete cookie
//    endif;

function createSessionCookie($email, $conn) {
    $session_token = pwTokenGenerator();
    $cookie_name = "pw_token";
    setcookie($cookie_name, $session_token, time() + (86400 * 30), "/"); // 86400 = 1 day

    $sql = "UPDATE wp_login SET password_reset = '$session_token' WHERE email='$email'";

    //var_dump($conn);

    $result = mysqli_query($conn, $sql);

    return "success";
}

function pwTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

mysqli_close($conn);

get_header();
?>
```

As is evident above, there were a few new areas integrated into the code. The first related to a redirect for the user which would navigate them to the login page once successfully resetting their password. The other highlighted section related to the creation of a function called 'createSessionCookie' which would create a token as I thought this would help to distinguish the session of the user and help to reset their specific password. The 'pwTokenGenerator' was a function that created a random 'string' of characters to prevent anyone from interrupting the session and to also help improve the security aspect through 'md5 hash'.

'Coulweb' Work Experience Documentation – Daniel Wilkins

After testing the password reset form to see if the table in the database would be updated with the new password, I then realised that this had worked but updated all of the user passwords in the table which was poor from a security aspect:

The Password had now Reset but for All Users Instead of One

| + Options | | user_id | username | password | email | password_reset | session_token | user_privileges |
|--------------------------|------------------|---------|-----------|--------------------|--------------------------------|----------------|----------------------------------|-----------------|
| <input type="checkbox"/> | Edit Copy Delete | 1 | username | resetpasswordtest1 | email | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | test | resetpasswordtest1 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | testing | resetpasswordtest1 | testing@test | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | tre | resetpasswordtest1 | | | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 6 | user1 | resetpasswordtest1 | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 9 | testing10 | resetpasswordtest1 | enquiries@danielhwilkins.co.uk | | | 5 |

After receiving advice and help, the code for the first section was altered to the following below:

```
if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) : |
$sql = "UPDATE wp_login SET password = '$successful_pw' WHERE password_reset='$pw_token'";
$result = mysqli_query($conn, $sql);
echo "Your Password has been Reset Successfully";
endif;
```

As is evident above, the highlighted piece of code was added to attempt to help the password reset to only be applied to one specific field.

After testing again, I encountered another problem with regards to the fact that the password fields would all change to 'true_reset' as can be seen below:

| + Options | | user_id | username | password | email | password_reset | session_token | user_privileges |
|--------------------------|------------------|---------|-----------|------------|--------------------------------|----------------|----------------------------------|-----------------|
| <input type="checkbox"/> | Edit Copy Delete | 1 | username | true_reset | email | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | test | true_reset | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | testing | true_reset | testing@test | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | tre | true_reset | | | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 6 | user1 | true_reset | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 9 | testing10 | true_reset | enquiries@danielhwilkins.co.uk | | | 5 |

As a result of experiencing this issue, I then decided to change the current code to the following:

Changing One Aspect of the Code

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];
$email = $_GET['email'];
$pw_token = $_COOKIE['passwordreset_token'];
$successful_pw = $_POST['successful_pw'];

if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) :

$sql = "UPDATE wp_login SET password = '$newpassword1' WHERE password_reset='$pw_token'";

$result = mysqli_query($conn, $sql);

echo "Your Password has been Reset Successfully";

    endif;

//If the number of rows is greater than 0 in the database, execute the function called 'createSessionCookie' and
return user to login screen once the form has been submitted

if (mysqli_num_rows($result) > 0) {
    createSessionCookie($email, $conn);
    header("Location: /wp/login");
}

// if(isset($pw_token) and $_GET['reset'] == "true") : //If value of logout is equal to being true, delete cookie
//     endif;

function createSessionCookie($email, $conn) {
    $session_token = pwTokenGenerator();
    $cookie_name = "pw_token";
    setcookie($cookie_name, $session_token, time() + (86400 * 30), "/"); // 86400 = 1 day

    $sql = "UPDATE wp_login SET password_reset = '$session_token' WHERE email='$email'";

    //var_dump($conn);

    $result = mysqli_query($conn, $sql);

    return "success";
}

function pwTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

mysqli_close($conn);
get_header();
?>
```

The only aspect I changed was the variable passed into the 'createSessionCookie' function and I changed this as I thought that this may identify the user by their email and as a result only change the password for them and not everyone else as well. However, when testing again, I soon found that this didn't have an effect on solving the current issue as is evident below:

The Password Continued to Reset for all Fields

| + Options | | user_id | username | password | email | password_reset | session_token | user_privileges |
|--------------------------|--------------------|---------|-----------|--------------------|--------------------------------|----------------|----------------------------------|-----------------|
| <input type="checkbox"/> | Edit Copy Delete | 1 | username | resetpasswordtest1 | email | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | test | resetpasswordtest1 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | testing | resetpasswordtest1 | testing@test | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | tre | resetpasswordtest1 | | | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 6 | user1 | resetpasswordtest1 | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 9 | testing10 | resetpasswordtest1 | enquiries@danielhwilkins.co.uk | | | 5 |

'Coullweb' Work Experience Documentation – Daniel Wilkins

After seeking help from the developer at 'Coullweb', I was helped to produce the following code below which would create the session 'Cookie' in the file that related to the page where users would enter their email in order to be sent the password reset link:

The Created Code after Help and Advice

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$to = $_POST['email']; //Changed email to to
$reset_code = createSessionCookie($to, $conn); //Moved to allow function to be run but had to be below to variable
otherwise wouldn't work
$subject = 'Password Reset Email';
echo $message = 'Please click the link https://clientb.coullweb.com/wp/password-reset-2/?email='.$to.'&code='.$reset_code; //Changed to reset code to call code

// $result = mysqli_query($conn, $sql); //Don't need the bits of SQL here

//if (mysqli_num_rows($result) > 0) {
//    header("Location: /wp/my-account");
//}

function createSessionCookie($to, $conn) {
    $session_token = pwTokenGenerator();
    //$cookie_name = "pw_token";

    $sql = "UPDATE wp_login SET password_reset = '$session_token' WHERE email='$email'";

    //var_dump($conn);

    $result = mysqli_query($conn, $sql);

    return $session_token; //Removed success and place session token instead but not in quotes
}

function pwTokenGenerator() {
    return substr(md5(uniqid(mt_rand(), true)), 0, 64);
}

//wp_mail( $to, $subject, $message);

mysqli_close($conn);
```

As is highlighted above, the first piece of code created two variables; one called '\$to' that would identify the email that the link would be sent to and the other called '\$reset_code' which would relate to creating the session 'Cookie'. The function 'createSessionCookie' would generate a token which would create a random 'string' to prevent others from intercepting or being able to access the link which would ultimately prevent any password resets from happening that weren't required. With regards to the '\$message' variable in the first highlighted piece of code, this would contain the link that the user would be able to select that would include both the '\$to' and '\$reset_code' variables that would allow for a unique session in order to reset only the password required.

After now having tested the password reset again, I now experienced an issue shown below:

The Password was Still Resetting for all and the Token had been Generated in the Wrong Row

| + Options | | | | | | | |
|--------------------------|---------|-----------|--------------------|--------------------------------|----------------------------------|----------------------------------|-----------------|
| | user_id | username | password | email | password_reset | session_token | user_privileges |
| <input type="checkbox"/> | 1 | username | resetpasswordtest1 | email | | | 5 |
| <input type="checkbox"/> | 2 | test | resetpasswordtest1 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> | 3 | testing | resetpasswordtest1 | testing@test | | | 5 |
| <input type="checkbox"/> | 5 | tre | resetpasswordtest1 | | 850e37f05e8de5c5fd476b24d0da862c | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> | 6 | user1 | resetpasswordtest1 | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> | 9 | testing10 | resetpasswordtest1 | enquiries@danielhwilkins.co.uk | | | 5 |

'Coulweb' Work Experience Documentation – Daniel Wilkins

After asking for some guidance on this issue, I was helped to make some alterations to the current code which will be shown below:

Changing the 'WHERE email' Aspect of the Code

```
$to = $_POST['email']; //Changed email to to
$reset_code = createSessionCookie($to, $conn); //Moved to allow function to be run but had to be below to variable
otherwise wouldn't work
$subject = 'Password Reset Email';
echo $message = 'Please click the link https://clientb.coullweb.com/wp/password-reset-2/?email='.$to.'&code='.$reset_code; //Changed to reset code to call code

//$result = mysqli_query($conn, $sql); //Don't need the bits of SQL here

//if (mysqli_num_rows($result) > 0) {
//  header("Location: /wp/my-account");
//}

function createSessionCookie($to, $conn) {
  $session_token = pwTokenGenerator();
  //$cookie_name = "pw_token";

  $sql = "UPDATE wp_login SET password_reset = '$session_token' WHERE email='$to'";

  //var_dump($conn);

  $result = mysqli_query($conn, $sql);

  return $session_token; //Removed success and place session token instead but not in quotes
}
```

As is highlighted above, the variable assigned to the 'email' field within the table had been changed from '\$email' to '\$to' as in the 'createSessionCookie' this was the variable being passed into this function and not '\$email' which meant this couldn't find the original placed variable. After testing to understand if this was functioning correctly, I then found that this had resolved the issue:

The Token was now Being Generated in the Correct Row of the table

| + Options | | | | | | | | | |
|--------------------------|---------|----------|----------|-------|----------------|--------------------|----------------------------------|----------------------------------|---|
| | user_id | username | password | email | password_reset | session_token | user_privileges | | |
| <input type="checkbox"/> | Edit | Copy | Delete | 1 | username | resetpasswordtest1 | email | 5 | |
| <input type="checkbox"/> | Edit | Copy | Delete | 2 | test | resetpasswordtest1 | ffff | 5 | |
| <input type="checkbox"/> | Edit | Copy | Delete | 3 | testing | resetpasswordtest1 | testing@test | 5 | |
| <input type="checkbox"/> | Edit | Copy | Delete | 5 | tre | resetpasswordtest1 | 850e37f05e8de5c5fd476b24d0da862c | 5 | |
| <input type="checkbox"/> | Edit | Copy | Delete | 6 | user1 | resetpasswordtest1 | user1@gmail.com | bbcf77fd50e69798943735cbf969225d | 5 |
| <input type="checkbox"/> | Edit | Copy | Delete | 9 | testing10 | resetpasswordtest1 | enquiries@danielwilkins.co.uk | 5 | |

'Coulweb' Work Experience Documentation – Daniel Wilkins

After this problem had been fixed, I now needed to ensure that the reset password link only changed the password of the individual of which it related to. Therefore, in the file relating to the form where users would enter their new password, I currently changed the code to the following shown below:

Adding a 'GET' Request to Include the Unique Link Created by the Previous file

```
$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];
$email = $_GET['email'];
// $pw_token = $_COOKIE['passwordreset_token'];
$pw_token = $_GET["https://clientb.coullweb.com/wp/password-reset-2/?email=".$to."&code=".$reset_code"];
$successful_pw = $_POST['successful_pw'];

if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) :

$sql = "UPDATE wp_login SET password = '$newpassword1' WHERE password_reset = '$pw_token'";

$result = mysqli_query($conn, $sql);

echo "Your Password has been Reset Successfully";

endif;
```

I believed that if I entered a 'GET' request to include the link that was specific to the user, that this would allow for changing of only their password. However, after testing I encountered the following problem where all of the passwords would still reset apart from those that had a token in the table within the database:

The Issue Explained Previously with Regards to the Password Reset

| Options | user_id | username | password | email | password_reset | session_token | user_privileges |
|---|---------|-----------|--------------------|--------------------------------|----------------------------------|----------------------------------|-----------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | username | resetpasswordtest2 | email | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | test | resetpasswordtest2 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3 | testing | resetpasswordtest2 | testing@test | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 5 | tre | resetpasswordtest1 | | 850e37f05e8de5c5fd476b24d0da862c | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 6 | user1 | resetpasswordtest1 | user1@gmail.com | bbc77fd50e69798943735cbf969225d | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 9 | testing10 | resetpasswordtest2 | enquiries@danielhwilkins.co.uk | | | 5 |

After seeking advice from the developer, I was then helped to change the code to the following displayed below:

Changing One Aspect of the Code

```
$newpassword1 = $_POST['newpassword1'];
$newpassword2 = $_POST['newpassword2'];
$email = $_GET['email'];
// $pw_token = $_COOKIE['passwordreset_token'];
$pw_token = $_GET['code'];
$successful_pw = $_POST['successful_pw'];

if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) :

$sql = "UPDATE wp_login SET password = '$newpassword1' WHERE password_reset = '$pw_token'";

$result = mysqli_query($conn, $sql);

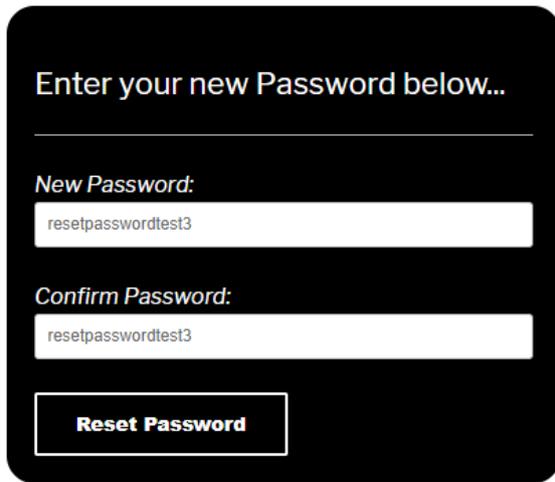
echo "Your Password has been Reset Successfully";

endif;
```

I was informed that only 'code' was needed and not the full 'URL' as the 'URL' had been placed in the previous file.

I then tested the form once more and this time this had successfully reset the password for only one user:

The Process of Entering the New Password



The Outcome of this in the Table within the Database (This had now Updated Successfully)

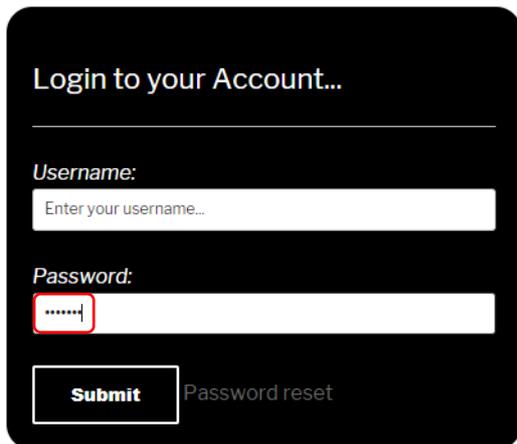
| | user_id | username | password | email | password_reset | session_token | user_privileges |
|---|---------|-----------|--------------------|-------------------------------|----------------------------------|----------------------------------|-----------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | username | resetpasswordtest2 | email | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | test | resetpasswordtest2 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3 | testing | resetpasswordtest2 | testing@test | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 5 | tre | resetpasswordtest1 | | 850e37f05e8de5c5fd476b24d0da862c | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 6 | user1 | resetpasswordtest3 | user1@gmail.com | bbc77fd50e69798943735cbf969225d | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 9 | testing10 | resetpasswordtest2 | enquiries@danielwilkins.co.uk | | | 5 |

After completing this aspect of the password reset form, I was then set further tasks. One of which was to ensure that when typing a password into any of the forms, that this would be hidden through the use of dots. To do this, I used my knowledge to remember what the developer had previously shown myself and also undertook research on the Internet as well to help. The final outcome of this can be viewed below:

Adding the 'input type' called 'password' to the Login form file

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="login_form_container">
      <form action="" method="post" class="login_form">
        <h2>Login to your Account...</h2>
        <hr>
        <i>Username:</i> <input type="text" name="username" class="input_fields" placeholder="Enter your username..."><br>
        <i>Password:</i> <input type="password" name="password" class="input_fields" placeholder="Enter your password..."><br>
        Adding password masks - https://www.w3schools.com/tags/att_input_type_password.asp | -->
        <input type="submit" class="submit_button">
        <a href="/wp/password-reset/">Password reset</a>
      </form>
      <?php echo the_content(); ?>
      <?php
    ?>
  </div>
</main><!-- #main -->
</div><!-- #primary -->
```

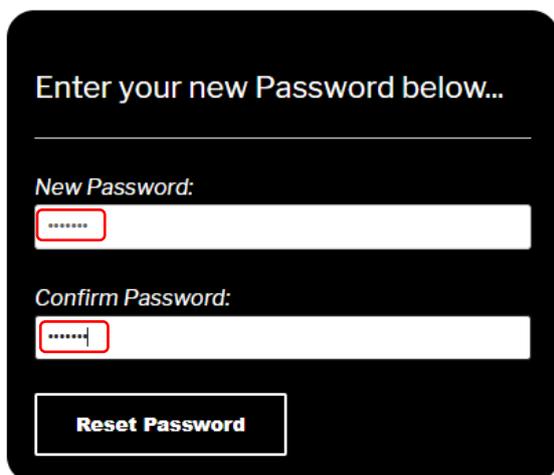
The Outcome on the Web Page (The Dots now Appeared)



Adding the 'input type' called 'password' to the Password Reset form file

```
<div id="primary" class="content-area">
  <main id="main" class="site-main" role="main">
    <div class="rp_form_container">
      <form action="" method="post" class="reset_password_form">
        <h2>Enter your new Password below...</h2>
        <hr>
        <i>New Password: </i> <input type="password" name="newpassword1" class="input_fields" placeholder="Enter your new password...">
        <br>
        <i>Confirm Password: </i> <input type="password" name="newpassword2" class="input_fields" placeholder="Enter your new password again...">
        <br>
        <input type="submit" class="submit_button" value="Reset Password">
        <input type="hidden" value="true_reset" name="successful_pw">
        <!--<a href="">Send Email</a-->
        <!--<a href="wp/password-reset-2/?reset=true"></a-->
      </form>
      <?php echo the_content(); ?>
      <?php
    ?>
  </div>
</main><!-- #main -->
</div><!-- #primary -->
```

The Outcome of this on the Web Page (The Dots now Appeared)



Another one of the set tasks was to make the password reset link expire when used as this would prevent anyone else from being able to use this if found. The final code, after help and advice can be viewed below:

The 'SQL Statement'

```
$sql = "UPDATE wp_login SET password = '$newpassword1', password_reset = '' WHERE password_reset = '$pw_token'; //  
Deletes token and stops link from being used again to change password"
```

Accessing the Link and Changing the Password for the First time

Enter your new Password below...

New Password:
[password field]

Confirm Password:
[password field]

Reset Password

This Reset the Password Successfully

| | user_id | username | password | email | password_reset | session_token | user_privileges |
|---|---------|-----------|--------------------|-------------------------------|----------------------------------|----------------------------------|-----------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | username | resetpasswordtest2 | email | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | test | resetpasswordtest2 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3 | testing | resetpasswordtest2 | testing@test | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 5 | tre | resetpasswordtest1 | | 07de6dfc823236dcfb618a076322cf32 | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 6 | user1 | testing456 | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 9 | testing10 | resetpasswordtest2 | enquiries@danielwilkins.co.uk | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 11 | AlexCoull | hacker | alex@coullweb.com | de749f2d3069fecadd80a3af4f4887b1 | 89127e0306ae9ab87218cb09091e546d | 5 |

Accessing the Link and Changing the Password for the Second Time

Enter your new Password below...

New Password:
[password field]

Confirm Password:
[password field]

Reset Password

'Coullweb' Work Experience Documentation – Daniel Wilkins

This Didn't Allow for Resetting of the Password and Also Removed the Token (This was a Success)

| + Options | | user_id | username | password | email | password_reset | session_token | user_privileges |
|--------------------------|------------------|---------|-----------|--------------------|-------------------------------|----------------------------------|----------------------------------|-----------------|
| <input type="checkbox"/> | Edit Copy Delete | 1 | username | resetpasswordtest2 | email | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | test | resetpasswordtest2 | ffff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | testing | resetpasswordtest2 | testing@test | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | tre | resetpasswordtest1 | | 07de6dfc823236dcfb618a076322cf32 | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 6 | user1 | testing456 | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 9 | testing10 | resetpasswordtest2 | enquiries@danielwilkins.co.uk | | | 5 |
| <input type="checkbox"/> | Edit Copy Delete | 11 | AlexCoull | hacker | alex@coullweb.com | de749f2d3069fecadd80a3af4f4887b1 | 89127e0306ae9ab87218cb09091e546d | 5 |

The next task I was set was to create an error message that would appear if passwords entered into the password reset form didn't match. To begin this task, I thought that adding 'else' and stating that if the '\$successful_pw' variable wasn't equal to 'true_reset' and also if the '\$newpassword1' variable wasn't equal to the '\$newpassword2' variable, that this would cause an error message to appear. This error message would be outputted onto the page through the 'echo' function:

Adding the 'else' element and Changing the other Aspects mentioned above

```
if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) :  
  
$sql = "UPDATE wp_login SET password = '$newpassword1', password_reset = '' WHERE password_reset = '$pw_token'; //  
Deletes token and stops link from being used again to change password  
  
$result = mysqli_query($conn, $sql);  
  
echo "Your Password has been Reset Successfully";  
  
else ( (!$successful_pw == 'true_reset' and $newpassword1 == $newpassword2) :  
  
    echo "Your new Passwords do not match!"  
  
endif;
```

However, this code caused an error to appear which can be viewed below:

The Error Message Appearing whilst Testing

```
Parse error: syntax error, unexpected '(', expecting ';' in wp-content/themes/twentyseventeen/pages/password_reset2.php on line 35
```

I then decided to change the 'if else' statement to an 'if else if' statement as I thought this may help to resolve the issue but this didn't have any effect:

Changing the Statement in the Code

```
if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) {  
  
$sql = "UPDATE wp_login SET password = '$newpassword1', password_reset = '' WHERE password_reset = '$pw_token'; //  
Deletes token and stops link from being used again to change password  
  
$result = mysqli_query($conn, $sql);  
  
echo "Your Password has been Reset Successfully";  
  
} else if ( !$successful_pw == 'true_reset' and $newpassword1 == $newpassword2) {  
  
    echo "Your new Passwords do not match!";  
  
}  
}
```

As I was struggling with this aspect, I therefore asked the developer for some assistance and I was helped to create the following code:

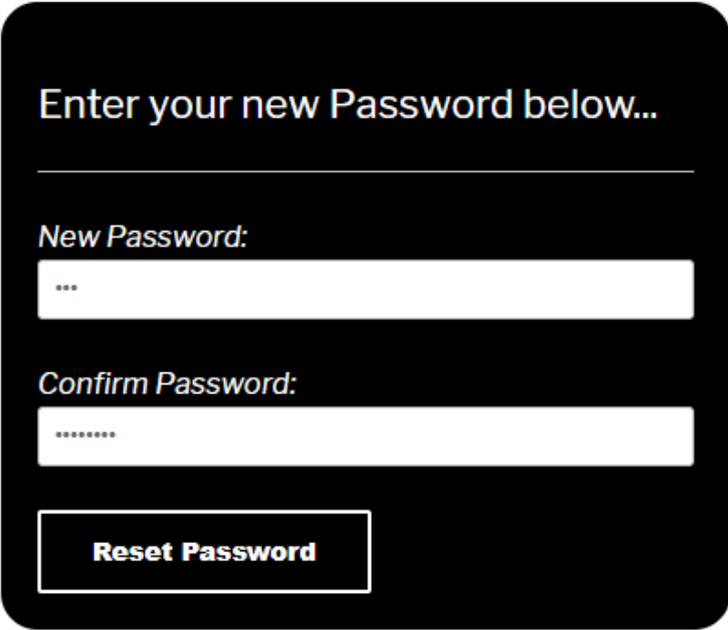
Altering Aspects of the Code after Help

```
if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) {  
  
    $sql = "UPDATE wp_login SET password = '$newpassword1', password_reset = '' WHERE password_reset = '$pw_token'"; //  
    Deletes token and stops link from being used again to change password  
  
    $result = mysqli_query($conn, $sql);  
  
    echo "Your Password has been Reset Successfully";  
  
} elseif ($successful_pw == 'true_reset' and $newpassword1 != $newpassword2){  
    echo "Your new Passwords do not match!";  
}
```

As is evident above, I was originally correct with the 'if else if' statement. However, I needed to include 'elseif' as one word and not two as well as removing the '!' from the '\$successful_pw' variable and adding '!=' between the variables called '\$newpassword1' and '\$newpassword2'. From undertaking research on the Internet, I understood that the '!=' means 'not equal to' which would compare both the '\$newpassword1' and '\$newpassword2' variables.

After testing this, I found that the error message would appear if the passwords entered didn't match. However, I also found that I could enter nothing into the form and select submit and this would cause a successful message to appear with all of the passwords being deleted for the table in the database:

The Process of Entering New Passwords which Didn't Match



The screenshot shows a dark-themed form with the following elements:

- Header: "Enter your new Password below..."
- Label: "New Password:"
- Input field: A white text box containing three asterisks (***)
- Label: "Confirm Password:"
- Input field: A white text box containing seven asterisks (*****)
- Button: A white button with the text "Reset Password"

The Error Message now Appeared at the top of the Page (This was Successful)

Your new Passwords do not match!

A screenshot of a password reset form. The form is dark-themed with white text and input fields. It has a title "Enter your new Password below...". Below the title are two input fields: "New Password:" and "Confirm Password:". Each field has a placeholder text "Enter your new password...". At the bottom of the form is a button labeled "Reset Password".

The Process of Leaving the Fields Blank and Selecting the 'Reset Password' Button/Link

A screenshot of the password reset form, similar to the one above. The "Reset Password" button at the bottom is highlighted with a white border. The input fields are empty.

This then Deleted all Passwords from the Database Apart from the Rows which Contained Tokens

| | user_id | username | password | email | password_reset | session_token | user_privileges |
|---|---------|-----------|--------------------|--------------------------------|----------------------------------|----------------------------------|-----------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | username | | email | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | test | | fiff | | d82a9096128af121cb55cd11b2c0e769 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3 | testing | | testing@test | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 5 | tre | resetpasswordtest1 | testemail1@gmail.com | c6edc15713e03bf4fc83597f71b73f05 | 56285e4311c930fa7e7e8b370ab201c6 | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 6 | user1 | | user1@gmail.com | | 6881c5b484c91d0762f13b78174b648f | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 9 | testing10 | | enquiries@danielhwilkins.co.uk | | | 5 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 11 | AlexCoul | hacker | alex@coulweb.com | de749f2d3069fecadd80a3af4f4887b1 | 89127e0306ae9ab87218cb09091e546d | 5 |

'Coullweb' Work Experience Documentation – Daniel Wilkins

After encountering this issue, I thought the following code may work as I believed this to be stating that if the hidden field ('true_reset') wasn't equal to the '\$successful_pw' variable then this wouldn't enable the user to reset their password:

Adding the '!=' to the Code

```
} elseif ($successful_pw != 'true_reset' and $newpassword1 != $newpassword2){  
    echo "Your new Passwords do not match!";  
}
```

However, this didn't work and the solution for this will be seen further below.

The final task for the password reset form was to be able to display this to users who had not reset their password before as well as not displaying the form to users who had reset their password before through the provided link. For this, I was helped to produce the final code regarding the functionality which can be viewed below:

The Final Code for this Task after Being Helped by the Developer at 'Coullweb'

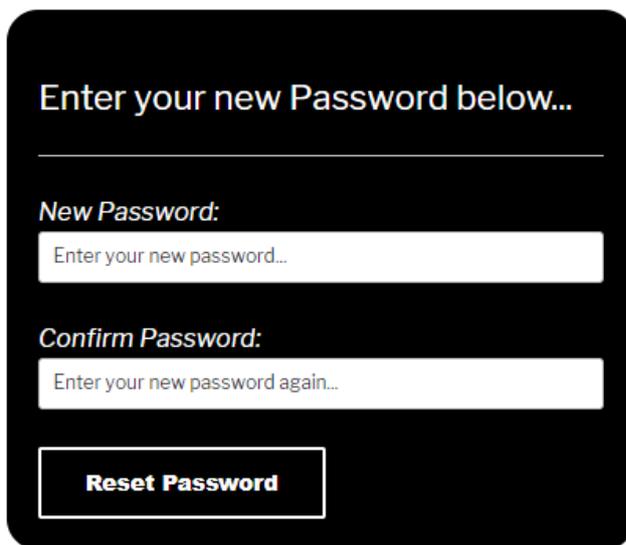
```
$newpassword1 = $_POST['newpassword1'];  
$newpassword2 = $_POST['newpassword2'];  
$email = $_GET['email'];  
// $pw_token = $_COOKIE['passwordreset_token'];  
$pw_token = $_GET['code'];  
$successful_pw = $_POST['successful_pw'];  
  
if($email != '' and $pw_token != '') {  
    $query_run = true;  
    if ($successful_pw == 'true_reset' and $newpassword1 == $newpassword2) {  
  
        $sql = "UPDATE wp_login SET password = '$newpassword1', password_reset = '' WHERE password_reset = '$pw_token'"  
            ; //Deletes token and stops link from being used again to change password  
  
        $result = mysqli_query($conn, $sql);  
  
        echo "Your Password has been Reset Successfully";  
  
    } elseif ($successful_pw == 'true_reset' and $newpassword1 != $newpassword2){  
  
        echo "Your new Passwords do not match!";  
  
    }  
}  
  
} else {  
    $query_run = false;  
}
```

```
<div id="primary" class="content-area">  
    <main id="main" class="site-main" role="main">  
        <div class="wp_form_container">  
            <?php if($query_run == true): ?>  
                <form action="" method="post" class="reset_password_form">  
                    <h2>Enter your new Password below...</h2>  
                    <hr>  
                    <i>New Password: </i> <input type="password" name="newpassword1" class="input_fields" placeholder="Enter your new password..."><br>  
                    <i>Confirm Password: </i> <input type="password" name="newpassword2" class="input_fields" placeholder="Enter your new password again..."><br>  
                    <input type="submit" class="submit_button" value="Reset Password">  
                    <input type="hidden" value="true_reset" name="successful_pw">  
                    <!--<a href="">Send Email</a-->  
                    <!--<a href="wp/password-reset-2/?reset=true"></a-->  
                </form>  
            <?php else: ?>  
                <h1>Oops, you have ran into a problem!</h1>  
            <?php endif: ?>  
            <?php echo the_content(); ?>  
            <?php  
        ?>  
    </div>  
</main><!-- #main -->  
</div><!-- #primary -->
```

As was evident on the previous page, a 'nested if' statement was included on the first screenshot so that if both the '\$email' and '\$pw_token' variables weren't equal to blank, the 'if elseif' statement would be executed inside as well as the 'query'. However, if they were equal to blank, the 'query' wouldn't be executed as well as what was included inside. The second screenshot reflects this where if the 'query' is equal to 'true', then this would show the form whereas if this wasn't, the form wouldn't be displayed and the message 'Oops, you have ran into a problem!' would be shown instead.

The Outcomes of this code can be viewed below:

The Form Appearing Whilst the 'query' was Equal to 'true'



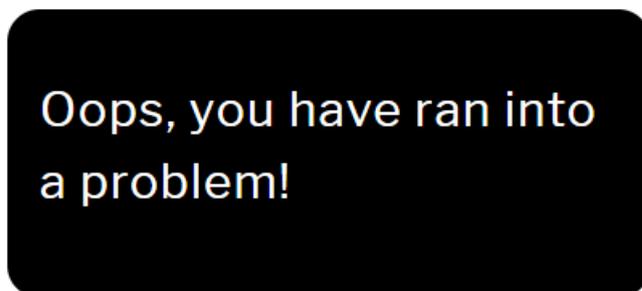
Enter your new Password below...

New Password:
Enter your new password...

Confirm Password:
Enter your new password again...

Reset Password

The Message Appearing Whilst the 'query' wasn't Equal to 'true'



**Oops, you have ran into
a problem!**

Summary of the Tasks for this Project

Throughout progressing with the tasks for the creating an account, login and password reset forms, I feel that I achieved several elements. This especially helped myself to both enhance my 'PHP', 'MySQL' and database skills of which I had a basic understanding of at the beginning. However, I now believe I can continue to develop these areas to ultimately improve my 'back-end' website development skillset as this is an area I am currently lacking in experience.

Day 6 – Creating a Landing Page Utilising HTML and CSS

Introduction

The task for the final day of my work experience at 'Coullweb' was to create a landing page based on a set design of which I am unable to show due to company rules. The landing page needed to link to different sections of a website and for each section, when hovering over them, they should have expanded by '100px' with '50px' on each side, as stated by the developer for the brief.

Establishing the files and the Page

To begin the task, I first of all established the 'HTML' file with the relevant required pieces of code and this is evident below:

Setting up of the HTML file

```
<!DOCTYPE html>
<html>
<head>
  <title>Landing Page</title>
  <link rel="stylesheet" href="styles/stylesheet.css">
  <script type="text/javascript" src="js/functions.js"></script>
  <meta charset="UTF-8">
</head>
<body>

<h1>TEST</h1>

</body>
</html>
```

As can be seen above, I included a title for the 'tab' called 'Landing Page' as well as for the actual page called 'TEST' to ensure that the file was functioning properly. As well as this, the character set 'UTF-8' was implemented as well as links to both of the other files which related to the 'CSS' and 'JavaScript' files. The intention was to utilise 'JavaScript', however, this wasn't implemented due to time constraints.

Setting up of the CSS and JavaScript files



As is evident above, I only had to name the files for this moment in time and I chose to provide them with standard names of 'stylesheet.css' and 'functions.js'.

Testing the Page to Ensure this was Functioning as it Should (This was Successful)

TEST

Creating the Page

The Banner Section of the Page

To begin, I had been focusing on the banner section of the page and the current code I had at the current stage is shown below:

The Current HTML Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Landing Page</title>
  <link rel="stylesheet" href="styles/stylesheet.css">
  <script type="text/javascript" src="js/functions.js"></script>
  <meta charset="UTF-8">
</head>
<body>

<div class="banner_section">
  
  <h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tortor turpis, tincidunt bibendum ante non, molestie pellentesque diam. In hac habitasse platea dictumst. Aliquam quis efficitur erat, ut mattis lacus. Quisque porta maximus est, at tempus ipsum volutpat eget. Nunc elementum dolor id eros faucibus rhoncus. Nullam dignissim magna sit amet sem scelerisque venenatis.</h3>
</div>

</body>
</html>
```

The Current CSS Code

```
/* Landing Page */

/* Headings */

h3 {
  font-family: Arial;
}

/* Banner Section */

.banner_section {
  background-color: orange;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
}

.banner_section h3 {
  padding-left: 100px;
  padding-right: 100px;
}

.logo_image {
  width: 300px;
  height: auto;
  border: 2px solid blue;
  text-align: center;
}
```

As is evident above, I decided to create one container which would consist of the content, the image and text in this case. With regards to the style of this container, I had styled this to have an orange background at first for testing purposes as well as positioning everything inside centrally. I also decided to add 'padding' to the 'h3' heading to position this further away from the edges of the banner section. With regards to the image, I decided to provide this with a set width and position it centrally so that everything would appear in the same position on the page.

The outcome of this code on the web page at the current time can be viewed below:

The Outcome on the Web Page



I then believed at this stage it would be more beneficial to place both the image and text in separate containers as this would make it easier to style each section separately without affecting everything inside of the banner later on:

Adding the Containers for both the Image and Text Sections of the Banner

```
<div class="banner_section">
  <div class="banner_img_container">
    
  </div>
  <div class="banner_heading_container">
    <h3>Lorem ipsum dolor sit amet, consectetur adipis
    Aliquam quis efficitur erat, ut mattis lacus. Quis
    amet sem scelerisque venenatis.</h3>
  </div>
</div>
```

At a later stage, after changing multiple aspects of the page, the banner section now appeared as can be seen below:

The Current HTML Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Landing Page</title>
  <link rel="stylesheet" href="styles/stylesheet.css">
  <script type="text/javascript" src="js/functions.js"></script>
  <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
  <meta charset="UTF-8">
</head>
<body>
  <div class="banner_section">
    <div class="banner_img_container">
      
    </div>
    <div class="banner_heading_container">
      <h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tortor turpis, tincidunt bibendum ante non, molestie pellentesque diam. In hac habitasse platea dictumst. Aliquam quis efficitur erat, ut mattis lacus. Quisque porta maximus est, at tempus ipsum volutpat eget. Nunc elementum dolor id eros faucibus rhoncus. Nullam dignissim magna sit amet sem scelerisque venenatis.</h3>
    </div>
  </div>
</body>
</html>
```

As is evident here, I hadn't made too many alterations to the 'HTML' code and one noticeable aspect is the fact that I had integrated a 'Google Font' into the file through an external stylesheet link which would allow for applying of the selected font throughout the page.

The Current CSS Code

```
div {
  box-sizing: border-box;
}

/* Landing Page */

/* Headings */

/*h3 {
  font-family: Arial;
}*/ /* Removed this to stop clashes with

/* Banner Section */
.banner_section {
  background-color: #2A283D;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
}

.banner_img_container {
  display: block;
  width: 50%;
  height: auto;
  margin: auto;
  /*background-color: red;*/
}

/* .banner_section h3 {
  padding-left: 100px;
  padding-right: 100px;
}*/

.logo_image {
  width: 300px;
  height: auto;
  /*border: 2px solid blue;*/
  text-align: center;
}

.banner_heading_container {
  display: block;
  width: 70%;
  height: auto;
  margin: auto;
  /*background-color: purple;*/
}

.banner_section h3 {
  color: #727180;
}

/* Main Section */

/*Links Section*/
```

Although not related to only the banner section, I had implemented 'box-sizing: border-box;' to all 'div' tags as advised due to the fact that this would allow 'padding' to be applied to the 'div' tags throughout without affecting the width of the page or positioning of certain elements. With regards to the actual banner section, the background colour had been changed with the 'Google Font' being applied to all elements inside through placing this in the parent container called 'banner_section'. The width of the 'banner_img_container' was set at '50%' with the use of 'display: block;' and 'margin: auto;' to help position this centrally in the banner with the same concept being applied to the 'banner_heading_container' as well. One final key aspect to note is the fact that the heading specific to the banner had been styled to be the colour highlighted above underneath the 'banner_section h3' 'class'.

'Coullweb' Work Experience Documentation – Daniel Wilkins

The outcome of this at the current time can be viewed below:

The Outcome of this on the Web Page at the Current time



This was as much as I had achieved in the final day of my work experience at Coullweb and please refer to the 'Final Outcome' Section to view the result of the whole landing page.

The Main Section of the Page

For the initial stages of this section of the page, which would contain the links to different sections, I had currently produced the following code:

The Current HTML Code

```
<div class="main_section">
  <div class="links_container1">Hello this is a test</div>
  <div class="links_container2"></div>
  <div class="links_container3"></div>
</div>
```

The Current CSS Code

```
.main_section {
  background-color: grey;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
}

.links_container1 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url(images/background_image1.jpg);
  background-size: 100% auto;
}

.links_container2 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
}

.links_container3 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
}
```

As will be seen above, I created three separate containers for each link section on the main section of the page. Within the stylesheet, I then ensured that they would all fit onto one line by providing them with widths of '31%' and by displaying them each as an 'inline-block'. The reason for providing each with a width of '31%' was because I knew that the maximum width would be '100%' and this figure divided by 3 was '33.33%'. I had to reduce the width slightly as this caused one of the containers to be placed on the next line. As is also evident, I included a background image that I had integrated from the 'PEXELS' website.

The outcome on the web page at the current time can be viewed below for the main section:

The Outcome of the Code on the Web Page



Although it is difficult to see the outcome with the provided screenshot above, it is evident that all of the containers were being displayed inline and the test text I had entered also appeared. However, the background image entered for one of the containers wasn't displaying.

At first, I believed this to be due to the fact I had forgotten to include quotation marks around the file path in the stylesheet as inspired by research. As a result, I therefore added quotation marks but this produced the same outcome:

Changing the CSS Code for the Background Image

```
.links_container1 {  
  width: 31%;  
  height: auto;  
  border: 2px solid red;  
  display: inline-block;  
  background-image: url("images/background_image1.jpg");  
  background-size: 100% auto;  
}
```

The Outcome on the Web Page (The same Issue still Remained)



However, I soon realised that the file path was incorrect which therefore meant that the image wouldn't display unless this was changed to the correct file path. As a result, I changed this and this process can be viewed below:

Changing the file path for the Background Image

```
.links_container1 {  
  width: 31%;  
  height: auto;  
  border: 2px solid red;  
  display: inline-block;  
  background-image: url("../images/background_image1.jpg");  
  background-size: 100% auto;  
}
```

The Outcome of this on the Page (The Background Image was now Showing)



'Coulweb' Work Experience Documentation – Daniel Wilkins

At a later stage, I then continued to style the main section of the landing page and the changes can be viewed below:

Adding the Logo Image to the Containers in the HTML file and Adding a 'class' called 'links_images' for the Images

```
<div class="main_section">
  <div class="links_container1"></div>
  <div class="links_container2"></div>
  <div class="links_container3"></div>
</div>
```

Progressing with the Styling of the Main Section of the Landing Page

```
.main_section {
  background-color: grey;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
}

.links_container1 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
}

.links_container2 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
}

.links_container3 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
}
```

```
.links_images {
  width: 300px;
  height: auto;
  text-align: center;
}
```

As is evident above, I added the same background image to each of the containers within this section and provided a grey background for the parent container of this section. I also styled the images placed inside the containers with the 'links_images' 'class', providing a width of '300px' with the height adjusting to the width and positioning the images centrally through the use of 'text-align: center;'.

'Coullweb' Work Experience Documentation – Daniel Wilkins

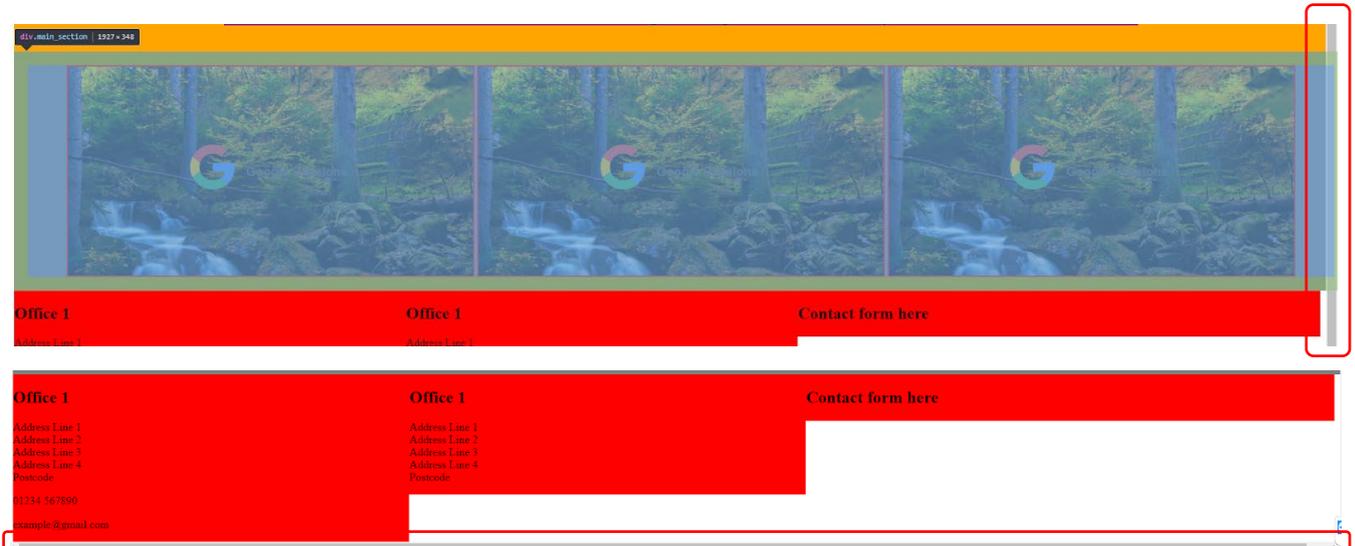
The Outcome of the changes shown on the previous page can be viewed below:

The Outcome on the Web Page (Ignore the red Section as this was the Footer Section)



Although related to the page as a whole, please note that before the screenshot above was captured, there was an issue with regards to being able to scroll horizontally which can be viewed below:

Ability to Scroll Horizontally with Additional Space on the Page (The 'Developer Tools' of 'Google Chrome' Highlighted this)



As a result of being helped, I understood that this was because of 'padding' which added more space to the page. This was due to the fact that 'box-sizing: border-box;' hadn't been applied as this would allow 'padding' to be added without affecting the page as a whole. As a result of experimenting with the 'Developer Tools' provided by 'Google Chrome', I therefore understood which areas needed their 'padding' removed and applied these changes to the actual stylesheet file:

Removing the 'padding' from the Areas Required in the CSS file

```
/* Banner Section */
.banner_section {
  background-color: orange;
  width: 100%;
  height: auto;
  text-align: center;
  /*padding: 20px;*/
}

.main_section {
  background-color: grey;
  width: 100%;
  height: auto;
  text-align: center;
  /*padding: 20px;*/ /* Removed this to stop issue with extra space */
}
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

This then corrected the issue and now no additional space existed on the page as will be shown below:

The Space Now Didn't Exist on the Page



Focusing the attention back to the main section of the page, I then realised I needed to add 'margin' in order to create space between each links section and also during the same time, I thought I could add a background colour to the image to create an overlay through the method shown below:

The Issue with each Links Section being Positioned too Close to each other



Making Alterations to the Stylesheet to add Space and Change the Background Colour

```
.links_container1 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  margin: 15px; /*Added this in to space containers apart*/
  background-color: rgba(0, 0, 0, 0.4);
}

.links_container2 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  background-color: rgba(0, 0, 0, 0.4);
}

.links_container3 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  background-repeat: no-repeat;
  background-color: rgba(0, 0, 0, 0.4);
  margin: 15px; /* Added this in to space containers apart */
}
```

As is evident above I implemented the 'background-color' style and added 'margin' to only the first and last links containers as adding this to the middle section would cause an uneven amount of space between all three containers.

The outcome of these changes can be viewed below:

The Outcome of this on the Web Page (The 'margin' had Worked but not the Background Colour Method)



As is stated above, the background colour wasn't applying and after discussing with the developer to ensure my thinking was correct, I created another container to include this background colour. This process can be viewed below:

Adding the Container called 'ms_div_bc' to each Section in the HTML file

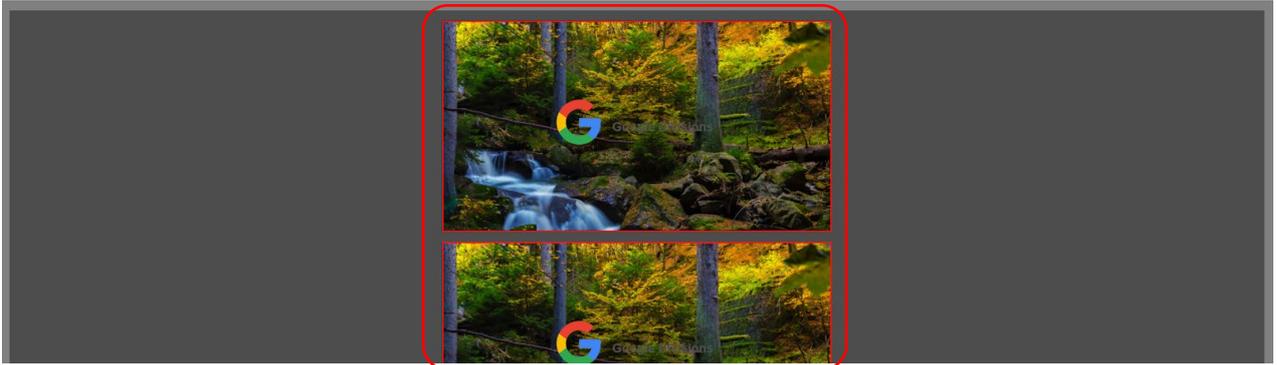
```
<div class="main_section">
  <div class="ms_div_bc">
    <div class="links_container1">
      
    </div>
  </div>
  <div class="ms_div_bc">
    <div class="links_container2">
      
    </div>
  </div>
  <div class="ms_div_bc">
    <div class="links_container3">
      
    </div>
  </div>
</div>
```

Styling the new 'class' in the CSS file to Consist of the Background Colour Required

```
.ms_div_bc {
  background-color: rgba(0, 0, 0, 0.4);
}
```

However, after testing to see if this was successful, I had realised that this hadn't been and created a problem where each container would be displayed in 'block' format:

The Outcome of this on the Web Page



After encountering the issue displayed above, I then decided to rearrange the order of containers in the 'HTML' file to the following as I thought this may help:

Positioning the 'ms_div_bc' Containers inside each 'links' Container in the HTML file

```
<div class="main_section">
.....
  <div class="links_container1">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container2">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container3">
    <div class="ms_div_bc">
      
    </div>
  </div>
</div>
.....
```

The Current CSS Code (Changing the Background Colour to display White Instead of Black)

```
.ms_div_bc {
  background-color: rgba(255, 255, 255, 0.7);
}

.links_container1 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  margin: 15px; /*Added this in to space containers apart*/
}

.links_container2 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
}

.links_container3 {
  width: 31%;
  height: auto;
  border: 2px solid red;
  display: inline-block;
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  background-repeat: no-repeat;
  margin: 15px; /* Added this in to space containers apart */
}
```

The previous changes then caused the required outcome to appear and this was now successful:

The Outcome on the Web Page (This was now Successful)



After this had been successful, I was then advised by the developer to create a new 'class' to hold the style of 'float: left;' as I was told this was good practice as well as being able to discount 'display: inline-block;' as this would reduce the original gaps between each links section. This process can be viewed below:

Adding the new 'class' Called 'ms_float_container' to each 'links' Container

```
<div class="main_section">
  <div class="links_container1 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container2 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container3 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
</div>
```

Adding the 'float: left;' Style and Removing the 'display: inline-block;' Style within the CSS file to Position each Section on One Single Line as well as Reducing the Original Gaps between each

```
.ms float container {
  float: left;
}
```

```
.links_container1 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  border-right: 2px solid white;
  /*margin: 15px; /*Added this in to space containers apart*/
}

.links_container2 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  border-right: 2px solid white;
}

.links_container3 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  background-repeat: no-repeat;
  border-right: 2px solid white;
  /*margin: 15px; /* Added this in to space containers apart */
}
}
```

As would have also been evident, I was informed that I didn't require the 'margin' for both 'links_container1' and 'links_container3' so therefore I removed these styles through comments.

At a later stage, after making other changes to both the 'HTML' and 'CSS' code, this section appeared as seen below:

The Current HTML Code

```
<div class="main_section">
  <div class="links_container1 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container2 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container3 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
</div>
```

As can be seen above, nothing had changed with regards to the structure of this section in the 'HTML' code.

The Current CSS Code

```
.main_section {
  background-color: grey;
  width: 100%;
  height: auto;
  text-align: center;
  /*padding: 20px;*/
  /*padding: 20px;*/ /* Removed this to stop issue with extra space */
}

.ms_div_bc {
  background-color: rgba(255, 255, 255, 0.7);
  transition: 0.5s;
}

.links_container1 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image1.jpg");
  background-size: 100% auto;
  border-right: 2px solid white;
  /*margin: 15px; /*Added this in to space containers apart*/
}

.links_container2 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image2.jpg");
  background-size: 100% auto;
  border-right: 2px solid white;
}

.links_container3 {
  width: 33.33%;
  height: auto;
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image3.jpg");
  background-size: 100% auto;
  background-repeat: no-repeat;
  border-right: 2px solid white;
  /*margin: 15px; /* Added this in to space containers apart */
}

.links_container1 .ms_div_bc:hover { /* Adding these for hovering effect */
  background-color: rgba(235, 65, 50, 0.9);
  transition: 0.5s;
}

.links_container2 .ms_div_bc:hover {
  background-color: rgba(251, 189, 1, 0.9);
  transition: 0.5s;
}

.links_container3 .ms_div_bc:hover {
  background-color: rgba(64, 134, 244, 0.9);
  transition: 0.5s;
}

.links_images {
  width: 300px;
  height: auto;
  text-align: center;
}

.ms_float_container {
  float: left;
}
```

As is evident above, I had changed several aspects for the styling of this section of the page of which included the widths of each 'links' container by changing these to '33.33%' each. This was because I had now applied 'box-sizing: border-box;' to the 'div' tags, as shown earlier, and needed to place each section close to each other. This was also the reason why the 'margin' had been removed from 'links_container1' and 'links_container3'. Another evident key modification was the background images for each 'links' container as I had now changed these to be different for each. I had also added a border to the right-hand side of each 'links' container which was white through the use of 'border-right: 2px solid white;'. The final key change was adding a 'hover' effect so that when hovering over each section, each one would change to another colour. This was completed by adding the ':hover' aspect to each 'ms_div_bc' container relating to the different sections. The 'transition: 0.5s' was also applied both when hovering and not hovering to create a smooth 'CSS' transition/animation.

'Coulweb' Work Experience Documentation – Daniel Wilkins

The outcome of this section of the page at the current time can be viewed below:

The Outcome of this on the Web Page

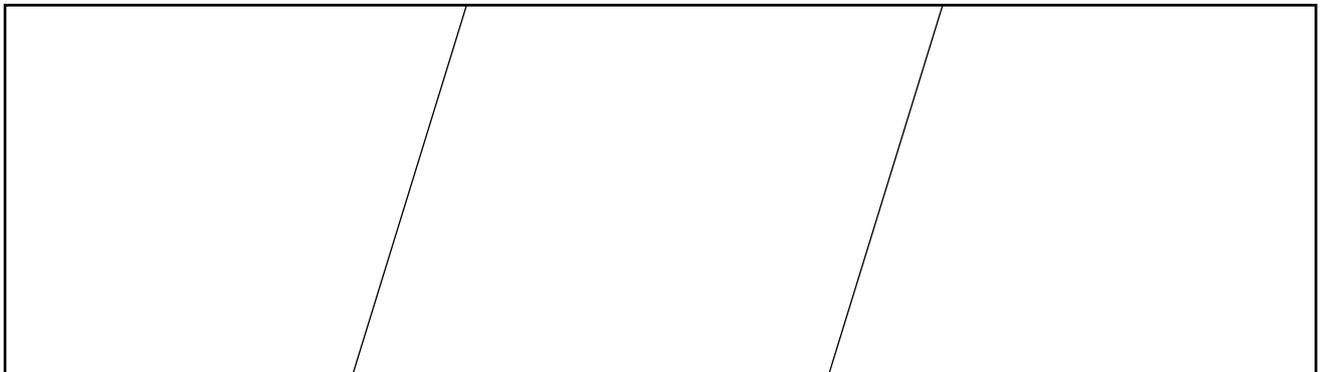


The Outcome of this on the Web Page – Hovering Example



The next task for this section was to create an angled appearance for the overlays for each of the links sections as demonstrated through the basic diagram below:

The Required Structure Outcome



Firstly, I thought that I could add a rotation to the 'links_container1' 'class' to achieve this and this process can be seen below:

Adding 'transform: rotate(20deg);' to the 'links_container1' 'class' for Testing Purposes

```
.links_container1 {  
  width: 33.33%;  
  height: auto;  
  /*border: 2px solid red;*/  
  /*display: inline-block;*/  
  background-image: url("../images/background_image1.jpg");  
  background-size: 100% auto;  
  border-right: 2px solid white;  
  transform: rotate(20deg);  
  /*margin: 15px; /*Added this in to space containers apart*/  
}
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

However, as is evident below, this didn't create the required outcome and rotated the entire links section including the image:

The Outcome of this on the Page (This wasn't Successful)



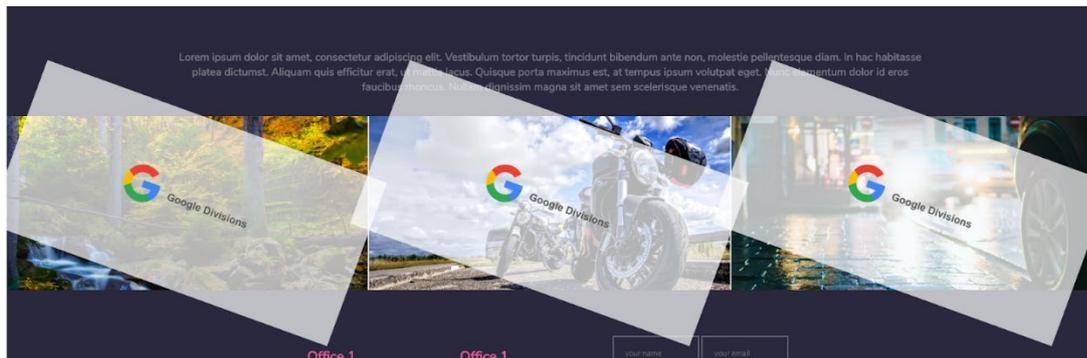
After the previously failed attempt, I then tried adding the same to the 'class' relating the overlay effect:

Adding 'transform: rotate(20deg);' to the 'ms_div_bc' 'class' in the CSS file

```
.ms_div_bc {  
  background-color: rgba(255, 255, 255, 0.7);  
  transition: 0.5s;  
  transform: rotate(20deg);  
}
```

The outcome of this was better than the first, although this time the overlay would be rotated with the image inside as seen below:

The Outcome of this on the Web Page



After struggling with this task, I asked for some help and was advised to change the code to the following:

Changing the CSS Code through Help and Advice from the Developer

```
.ms_div_bc {
  background-color: rgba(255, 255, 255, 0.7);
  transition: 0.5s;
  transform: skewY(30deg);
}

.ms_float_container {
  float: left;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  width: 33.33%;
  height: 600px;
}

.links_container1 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image1.jpg");
  border-right: 2px solid white;
  /*margin: 15px; /*Added this in to space containers apart*/
}

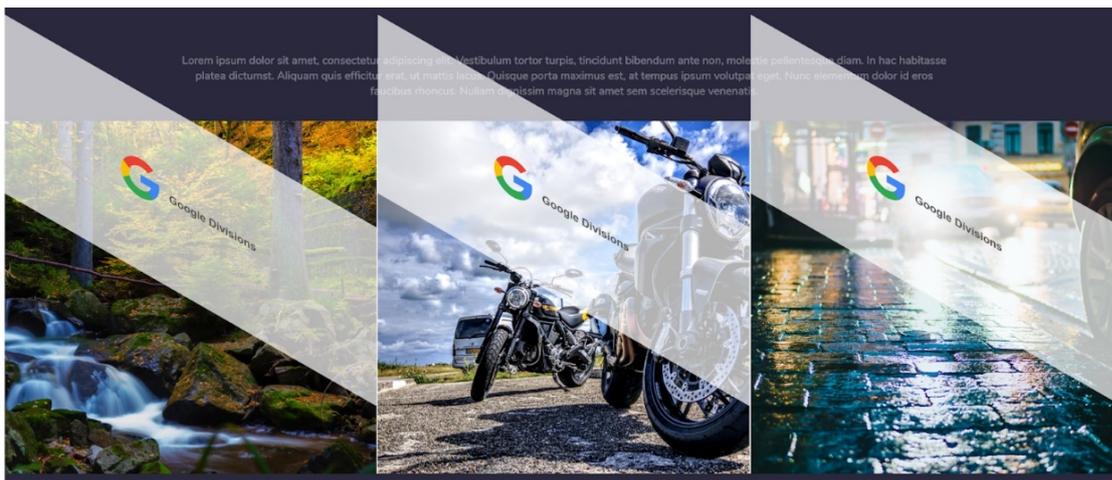
.links_container2 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image2.jpg");
  border-right: 2px solid white;
}

.links_container3 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image3.jpg");
  border-right: 2px solid white;
  /*margin: 15px; /* Added this in to space containers apart */
}
```

As is evident above, I was informed that 'skew' was the type of 'transform' required and that to keep the code efficient I should apply the styles relating to all the 'links' containers within the 'ms_float_container' in order to reduce the duplication of code. The height of each container was changed to '600px' as I was told this would be easier to create the required effect.

After these changes, the following outcome below was created:

The Outcome on the Page at this Current time



'Coullweb' Work Experience Documentation – Daniel Wilkins

Following on from this, after advice, I integrated the following changes displayed below:

Changing the 'skew' to '-30deg' in the CSS file for the 'ms_div_bc' 'class' and adding 'skew' to the Images

```
.ms_div_bc {  
  background-color: rgba(255, 255, 255, 0.7);  
  transition: 0.5s;  
  transform: skewX(-30deg);  
  width: 100%;  
  height: 600px;  
}
```

```
.links_images {  
  width: 300px;  
  height: auto;  
  text-align: center;  
  transform: skewX(30deg);  
  margin-top: 25%;  
}
```

As is highlighted above, I needed to change the 'skew' for the first 'class' shown above as currently, the positioning of the container was the opposite way to what was required. A 'skew' was added to the images to prevent them from being affected by the other style previously and 'margin-top' was also added to position the images more centrally after being affected by the other styles.

The outcome of these changes can be viewed below:



This was as much as I had achieved in the final day of my work experience at Coullweb and please refer to the 'Final Outcome' Section to view the result of the whole landing page.

The Footer Section of the Landing Page

For the initial stages of the footer section of the landing page, I currently had the following code:

The Current HTML Code at this Moment in time

```
<div class="footer_section">
  <div class="address_container1 footer_containers">
    <h2>Office 1</h2>
    <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
    <p>01234 567890</p>
    <p>example@gmail.com</p>
  </div>
  <div class="address_container2 footer_containers">
    <h2>Office 1</h2>
    <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
  </div>
  <div class="address_container3 footer_containers">
    <h2>Contact form here</h2>
  </div>
</div>
```

The Current CSS Code at this Moment in time

```
/* Footer Section */

.footer_section {
  background-color: purple;
  width: 100%;
  height: auto;
}

.footer_containers {
  float: left;
}

.address_container1 {
  width: 30%;
  height: auto;
  background-color: red;
  /*display: inline-block;*/ /* Adv
}

.address_container2 {
  width: 30%;
  height: auto;
  background-color: red;
  /*display: inline-block;*/
}

.address_container3 {
  width: 40%;
  height: auto;
  background-color: red;
  /*display: inline-block;*/
}
```

As will be seen above, to begin, I created a main container called 'footer_section' which would act as the parent container with the 'address_container1', 'address_container2', and 'address_container3' all acting as the sub-containers that consisted of different content. I ensured that the widths of each of the sub-containers totalled '100%' and I provided each of these with a background colour of red and a background colour of purple was applied to the parent container to distinguish the difference to help with testing. As will also be evident, the 'footer_containers' 'class' was applied to each of the sub-containers as advised by the developer at 'Coullweb' so that 'float: left;' could be applied. This would allow the content to be displayed on one line.

'Coulweb' Work Experience Documentation – Daniel Wilkins

The outcome of the previous code can be viewed below:

The Outcome of this on the Web Page

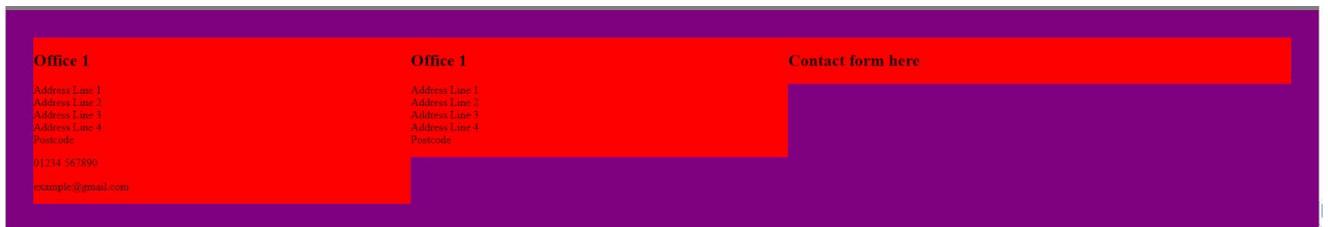


As will be seen above, there was an issue where the sub-containers wouldn't be contained within the parent container and as a result appeared as if they were floating. After seeking some advice, I was informed that I needed to add 'overflow: hidden;' to the parent container in order for this to be resolved so therefore, I did this and the outcome of this can be seen below:

Adding 'overflow: hidden;' to the Stylesheet

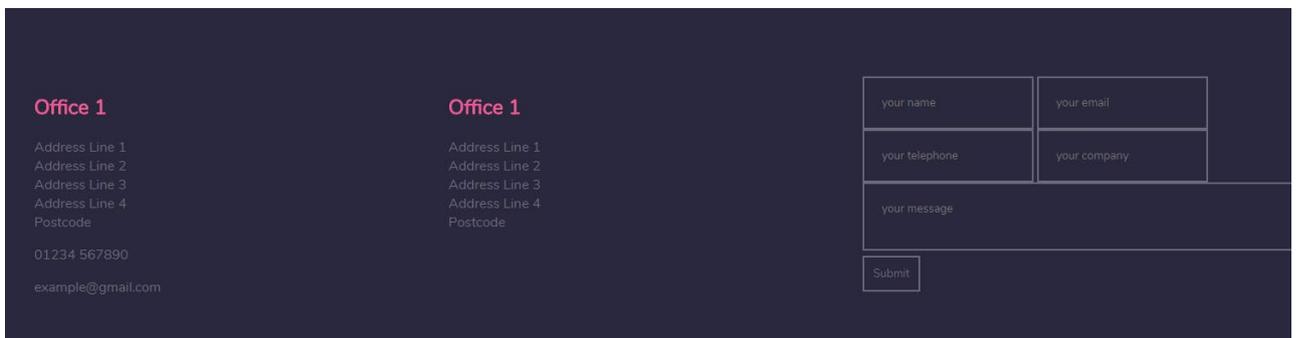
```
overflow: hidden;
```

The Outcome on the Web Page (This now Worked)



After progressing further with the footer section of the page, I had then created the following outcome at this current time:

The Outcome at this Current time for the Footer Section of the Landing Page



As was evident above, the elements inside the footer section were spaced too far apart from each other and in order to resolve this issue I undertook the following process:

The Current HTML Code at this time of the Task

```
<div class="footer_section">
  <div class="footer_content_container">
    <div class="address_container1 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
      <p>01234 567890</p>
      <p>example@gmail.com</p>
    </div>
    <div class="address_container2 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
    </div>
    <div class="address_container3 footer_containers">
      <div class="form_container1">
        <input type="text" placeholder="your name">
        <input type="text" placeholder="your email">
      </div>
      <div class="form_container2">
        <input type="text" placeholder="your telephone">
        <input type="text" placeholder="your company">
      </div>
      <div class="form_container3">
        <textarea placeholder="your message"></textarea>
      </div>
      <div class="form_container4">
        <input type="submit" placeholder="SEND" class="submit_button">
      </div>
    </div>
  </div>
</div>
```

Changing the Width of the Parent Container to '80%' and Also Changing the Width of the Sub-Containers to Suit this Change

```
.footer_content_container {
  width: 80%;
  height: auto;
  display: block;
  margin: auto;
}

.address_container1 {
  width: 20%;
  height: auto;
  /*background-color: red;*/
  padding: 20px;
  /*display: inline-block;*/
}

.address_container2 {
  width: 20%;
  height: auto;
  /*background-color: red;*/
  /*display: inline-block;*/
  padding: 20px;
}

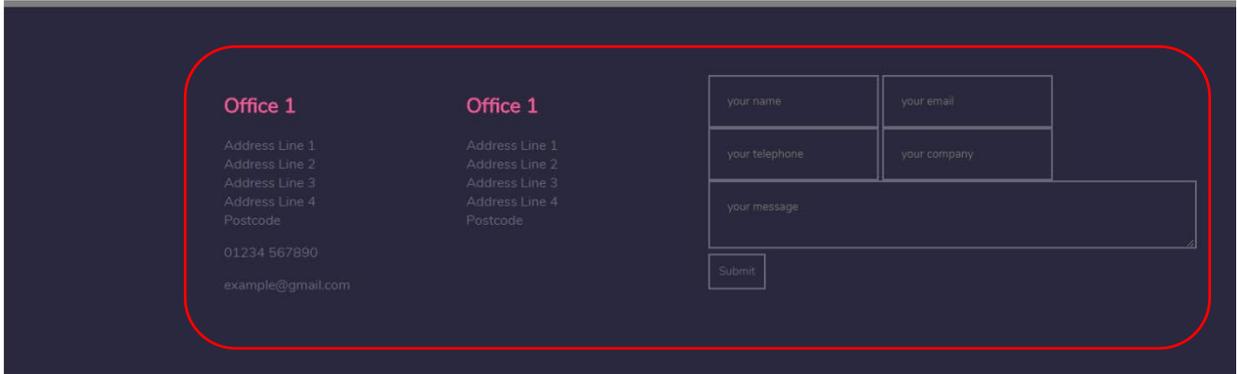
.address_container3 {
  width: 40%;
  height: auto;
  /*background-color: red;*/
  /*display: inline-block;*/
  padding: 20px;
}
```

As is evident above, I changed the width of the parent container to '80%' to reduce the width of the container and as a result, display this more centrally through the use of 'margin: auto;'. Within this, I also changed the widths of the sub-containers to suit this.

'Coullweb' Work Experience Documentation – Daniel Wilkins

The outcome of the changes shown previously can be viewed below:

The Outcome of this on the Web Page (The Elements were now Positioned Closer to each Other)



After making further changes at a later stage, this section of the page then appeared as shown below:

The HTML Code at the Current time

```
</div>
</div>
<div class="footer_section">
  <div class="footer_content_container">
    <div class="address_container1 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
      <p>01234 567890</p>
      <p>example@gmail.com</p>
    </div>
    <div class="address_container2 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
    </div>
    <div class="address_container3 footer_containers">
      <div class="form_container1">
        <input type="text" placeholder="your name">
        <input type="text" placeholder="your email" style="margin-left: 20px;">
      </div>
      <div class="form_container2">
        <input type="text" placeholder="your telephone">
        <input type="text" placeholder="your company" style="margin-left: 20px;"> <!-- Added this inline
        class in the css related to all input fields which would have affected all of these -->
      </div>
      <div class="form_container3">
        <textarea placeholder="your message"></textarea>
      </div>
      <div class="form_container4">
        <input type="submit" value="Send" class="submit_button">
      </div>
    </div>
  </div>
</div>
</body>
```

With regards to the 'HTML' file at the current moment in time, nothing had changed except for the 'value' for the 'Submit' button/link and adding of 'inline styles' regarding 'margin-left' to help position the elements highlighted above correctly.

The Current CSS Code

```
/* Footer Section */
.footer_section {
  /*background-color: purple;*/
  background-color: #2A283D;
  width: 100%;
  height: auto;
  padding: 60px;
  overflow: hidden;
  font-family: 'Nunito', sans-serif; /* Google
}

.footer_content_container {
  width: 80%;
  height: auto;
  display: block;
  margin: auto;
}

.footer_section h2 {
  color: #EB5A95;
}

.footer_section p {
  color: #6B697B
}

.footer_section input {
  color: #6B697B;
  border: 2px solid #6B697B;
  background-color: transparent;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
}

.footer_section textarea {
  color: #6B697B;
  border: 2px solid #6B697B;
  background-color: transparent; /* Realising
  width: 100%;
  height: auto;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
}

.footer_containers {
  float: left;
}

.address_container1 {
  width: 20%;
  height: auto;
  /*background-color: re
  padding: 20px;
  /*display: inline-bloc
}

.address_container2 {
  width: 20%;
  height: auto;
  /*background-color: re
  /*display: inline-bloc
  padding: 20px;
}

.address_container3 {
  width: 40%;
  height: auto;
  /*background-color: re
  /*display: inline-bloc
  padding: 20px;
}

.form_container1 {
  width: 100%;
  height: auto;
}

.form_container2 {
  width: 100%;
  height: auto;
  margin-top: 20px; /* A
}

.form_container3 {
  width: 100%;
  height: auto;
  margin-top: 20px;
}

.form_container4 {
  width: 100%;
  height: auto;
}

.submit_button {
  padding: 15px !important;
  cursor: pointer;
  transition: 0.5s;
  background-color: #EB5A95 !important;
  color: white !important;
  text-transform: uppercase;
  float: right;
  font-weight: bold;
  border: none !important;
}

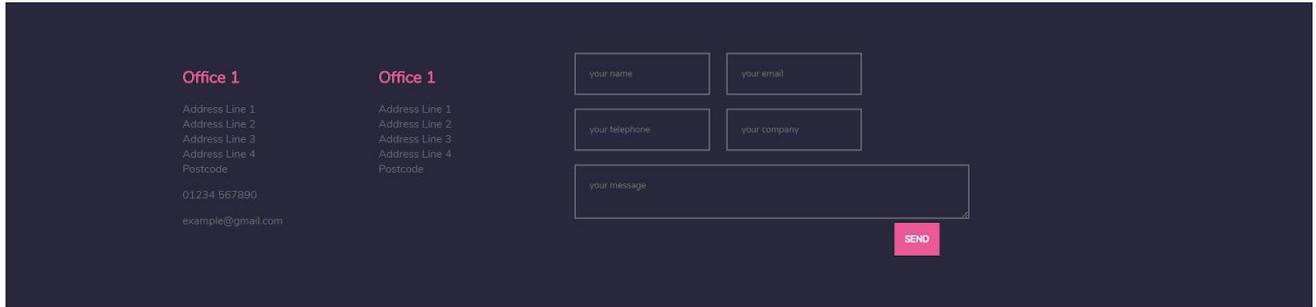
.submit_button:hover {
  background-color: white !important;
  color: #EB5A95 !important;
  transition: 0.5s;
}
```

As is highlighted above, the key aspects to notice was the fact that both the text colour of the heading and paragraph tags contained within the footer section had been set through assigning these tags to the 'footer_section' 'class'. Furthermore, a background colour of 'transparent' had been applied to both the 'input' and 'textarea' sections so that the background colour of the 'footer_section' would appear through these elements. Also, 'padding' had been applied to these areas to allow for spacing around the text inside and the 'Google Font' had been applied. Similarly to the main section of the page, the 'footer_containers' 'class' enabled all elements to be positioned on the same line through the use of 'float: left;'. The final key aspect was the fact that the 'Submit' button/link had been styled through the 'class' called 'submit_button' where 'text-transform: uppercase;' ensured for the 'value' to be placed in uppercase characters. A 'hover' aspect was also added to ensure for a smooth animation when hovering over the button/link and it is worth noting that '!important' had to be utilised to override some of the other styles included elsewhere.

'Coulweb' Work Experience Documentation – Daniel Wilkins

The outcome of these changes can be viewed below:

The Outcome on the Web Page



After this, I then began to attempt to resolve some issues. One of these issues was the fact that the 'textarea' box was much longer than that of everything else when it should have been in the same position as the button/link below. After seeking advice, I was informed that I needed to apply the 'box-sizing: border-box' to this section as this wasn't a 'div' tag. Therefore, I did this and this process can be viewed below:

Applying the 'box-sizing: border-box' to 'textarea' and 'input' Elements

```
div, textarea, input {  
  box-sizing: border-box;  
}
```

The Outcome of this on the Web Page (This Resolved the Issue)



Another issue I had experienced was the fact that the content inside the footer section wasn't completely centred and after communicating with the developer at 'Coullweb', I understood that I had to make sure the widths of each container needed to total '100%'. Before, I was setting the widths to total '80%' due to the width of the container holding these being '80%':

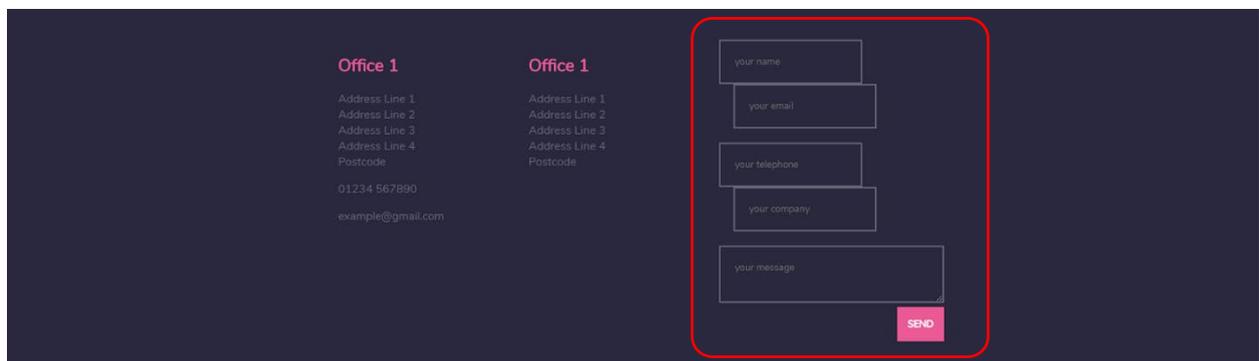
Changing the Widths of the Containers

```
.address_container1 {
  width: 30%; /* Had to change
  height: auto;
  /*background-color: red;*/
  padding: 20px;
  /*display: inline-block;*/ /
}

.address_container2 {
  width: 30%;
  height: auto;
  /*background-color: red;*/
  /*display: inline-block;*/
  padding: 20px;
}

.footer_content_container {
  width: 50%;
  height: auto;
  display: block;
  margin: auto;
}
```

The Outcome of this on the Web Page



As can be seen above, the content was now more central but I now needed to alter parts of the form to prevent the 'input' fields from positioning incorrectly as highlighted above.

After utilising the 'Developer Tools' on 'Google Chrome', I then understood I had to remove the 'margin-left' styles from the elements shown earlier in this document as well as adjusting the width of the fields to make them appear on the same line where required:

Removing the 'margin-left' Styles (Example)

```
<input type="text" placeholder="your email" style="margin-left: 20px;">
```

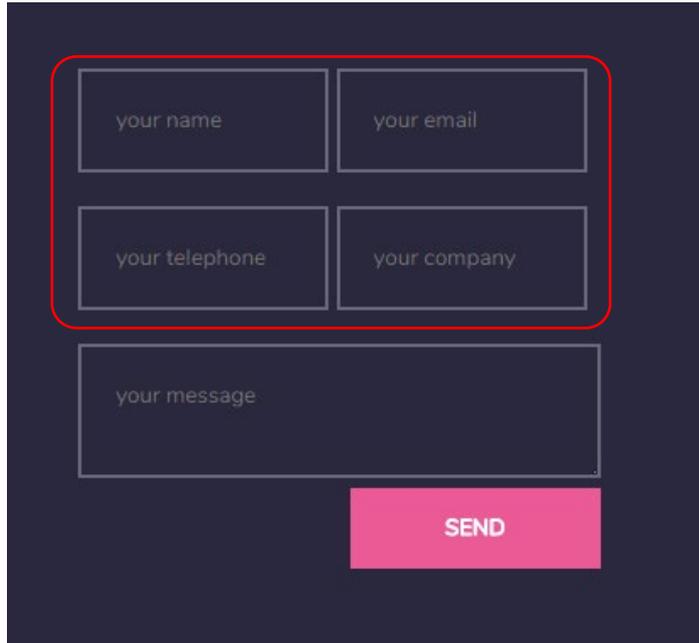
Adjusting the Widths of the fields in the CSS file

```
.footer_section input {
  color: #6B697B;
  border: 2px solid #6B697B;
  background-color: transparent;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
  width: 48%;
  height: auto;
}
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

This then helped to resolve the issue and the following outcome can be viewed below:

The Outcome of these Changes on the Web Page



A dark-themed contact form with a red border around the top four input fields. The form contains the following elements:

- Input field: your name
- Input field: your email
- Input field: your telephone
- Input field: your company
- Input field: your message
- SEND button

This was as much as I had achieved in the final day of my work experience at Coullweb and please refer to the 'Final Outcome' Section to view the result of the whole landing page.

The Final Outcome

Due to the fact that I had only spent one day working on the landing page, this wasn't fully complete but this task allowed myself to enhance my knowledge of some areas of 'CSS' such as using the transformation of 'skew' and 'float: left;'. The final outcome of both the code and the landing page itself can be seen below:

The Final HTML Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Landing Page</title>
  <link rel="stylesheet" href="styles/styleSheet.css">
  <script type="text/javascript" src="js/functions.js"></script>
  <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
  <meta charset="UTF-8">
</head>
<body>

<div class="banner_section">
  <div class="banner_img_container">
    
  </div>
  <div class="banner_heading_container">
    <h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tor
    Aliquam quis efficitur erat, ut mattis lacus. Quisque porta maximus est, at
    amet sem scelerisque venenatis.</h3>
  </div>
</div>

<div class="main_section">
  <div class="links_container1 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container2 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
  <div class="links_container3 ms_float_container">
    <div class="ms_div_bc">
      
    </div>
  </div>
</div>

<div class="footer_section">
  <div class="footer_content_container">
    <div class="address_container1 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
      <p>01234 567890</p>
      <p>example@gmail.com</p>
    </div>
    <div class="address_container2 footer_containers">
      <h2>Office 1</h2>
      <p>Address Line 1 <br> Address Line 2 <br> Address Line 3 <br> Address Line 4 <br> Postcode</p>
    </div>
    <div class="address_container3 footer_containers">
      <div class="form_container1">
        <input type="text" placeholder="your name">
        <input type="text" placeholder="your email">
      </div>
      <div class="form_container2">
        <input type="text" placeholder="your telephone">
        <input type="text" placeholder="your company"> <!-- Added this inline style due to the fact tha
        all of these -->
      </div>
      <div class="form_container3">
        <textarea placeholder="your message"></textarea>
      </div>
      <div class="form_container4">
        <input type="submit" value="Send" class="submit_button">
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

The Final CSS Code

```
div, textarea, input {
  box-sizing: border-box;
}

/* Landing Page */

/* Headings */

/*h3 {
  font-family: Arial;
}*/ /* Removed this to stop clashes with the Google Font */

/* Banner Section */
.banner_section {
  background-color: #2A283D;
  width: 100%;
  height: auto;
  text-align: center;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
}

.banner_img_container {
  display: block;
  width: 50%;
  height: auto;
  margin: auto;
  /*background-color: red;*/
}

/*.banner_section h3 {
  padding-left: 100px;
  padding-right: 100px;
}*/

.logo_image {
  width: 300px;
  height: auto;
  /*border: 2px solid blue;*/
  text-align: center;
}

.banner_heading_container {
  display: block;
  width: 70%;
  height: auto;
  margin: auto;
  /*background-color: purple;*/
}
```

```
.banner_section h3 {
  color: #727180;
}

/* Main Section */

/*Links Section*/

.main_section {
  background-color: grey;
  width: 100%;
  height: auto;
  text-align: center;
  /*padding: 20px;*/
  /*padding: 20px;*/ /* Removed this to stop issue with extra space */
}

.ms_div_bc {
  background-color: rgba(255, 255, 255, 0.7);
  transition: 0.5s;
  transform: skewX(-30deg);
  width: 100%;
  height: 600px;
}

.ms_float_container {
  float: left;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  width: 33.33%;
  height: 600px;
}

.links_container1 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  /*background-image: url("../images/background_image1.jpg");*/
  border-right: 2px solid white;
  position: relative;
  overflow: hidden;
  background: url("../images/background_image1.jpg");
  /*margin: 15px; /*Added this in to space containers apart*/
}
```

```
.links_container1:before {
  content: "";
  position: absolute;
  width: 200%;
  height: 200%;
  top: -50%;
  left: -50%;
  z-index: -1;
  -webkit-transform: rotate(30deg);
  transform: rotateY(30deg);
}

.links_container2 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image2.jpg");
  border-right: 2px solid white;
}

.links_container3 {
  /*border: 2px solid red;*/
  /*display: inline-block;*/
  background-image: url("../images/background_image3.jpg");
  border-right: 2px solid white;
  /*margin: 15px; /* Added this in to space containers apart */
}

.links_container1 .ms_div_bc:hover { /* Adding these for hovering effect on images */
  background-color: rgba(235, 65, 50, 0.9);
  transition: 0.5s;
}

.links_container2 .ms_div_bc:hover {
  background-color: rgba(251, 189, 1, 0.9);
  transition: 0.5s;
}

.links_container3 .ms_div_bc:hover {
  background-color: rgba(64, 134, 244, 0.9);
  transition: 0.5s;
}

.links_images {
  width: 300px;
  height: auto;
  text-align: center;
  transform: skewX(30deg); /* Added this in to stop google images from being skewed */
  margin-top: 25%;
}
```

```
/* Footer Section */

.footer_section {
  /*background-color: purple;*/
  background-color: #2A283D;
  width: 100%;
  height: auto;
  padding: 60px;
  overflow: hidden;
  font-family: 'Nunito', sans-serif; /* Google Fonts - https://fonts.google.com/?selection.family=Nunito */
}

.footer_content_container {
  width: 50%;
  height: auto;
  display: block;
  margin: auto;
}

.footer_section h2 {
  color: #EB5A95;
}

.footer_section p {
  color: #6B697B
}

.footer_section input {
  color: #6B697B;
  border: 2px solid #6B697B;
  background-color: transparent;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
  width: 48%;
  height: auto;
}

.footer_section textarea {
  color: #6B697B;
  border: 2px solid #6B697B;
  background-color: transparent; /* Realising that this could be used to show background of parent container */
  width: 100%;
  height: auto;
  padding: 20px;
  font-family: 'Nunito', sans-serif;
}

.footer_containers {
  float: left;
}
```

```
.address_container1 {
  width: 30%; /* Had to change these widths as I had realised that the width needed to equal 100% to be centred completely */
  height: auto;
  /*background-color: red;*/
  padding: 20px;
  /*display: inline-block;*/ /* Advised to put float left to keep elements on same line */
}

.address_container2 {
  width: 30%;
  height: auto;
  /*background-color: red;*/
  /*display: inline-block;*/
  padding: 20px;
}

.address_container3 {
  width: 40%;
  height: auto;
  /*background-color: red;*/
  /*display: inline-block;*/
  padding: 20px;
}

.form_container1 {
  width: 100%;
  height: auto;
}

.form_container2 {
  width: 100%;
  height: auto;
  margin-top: 20px; /* Added this to add space between form items */
}

.form_container3 {
  width: 100%;
  height: auto;
  margin-top: 20px;
}

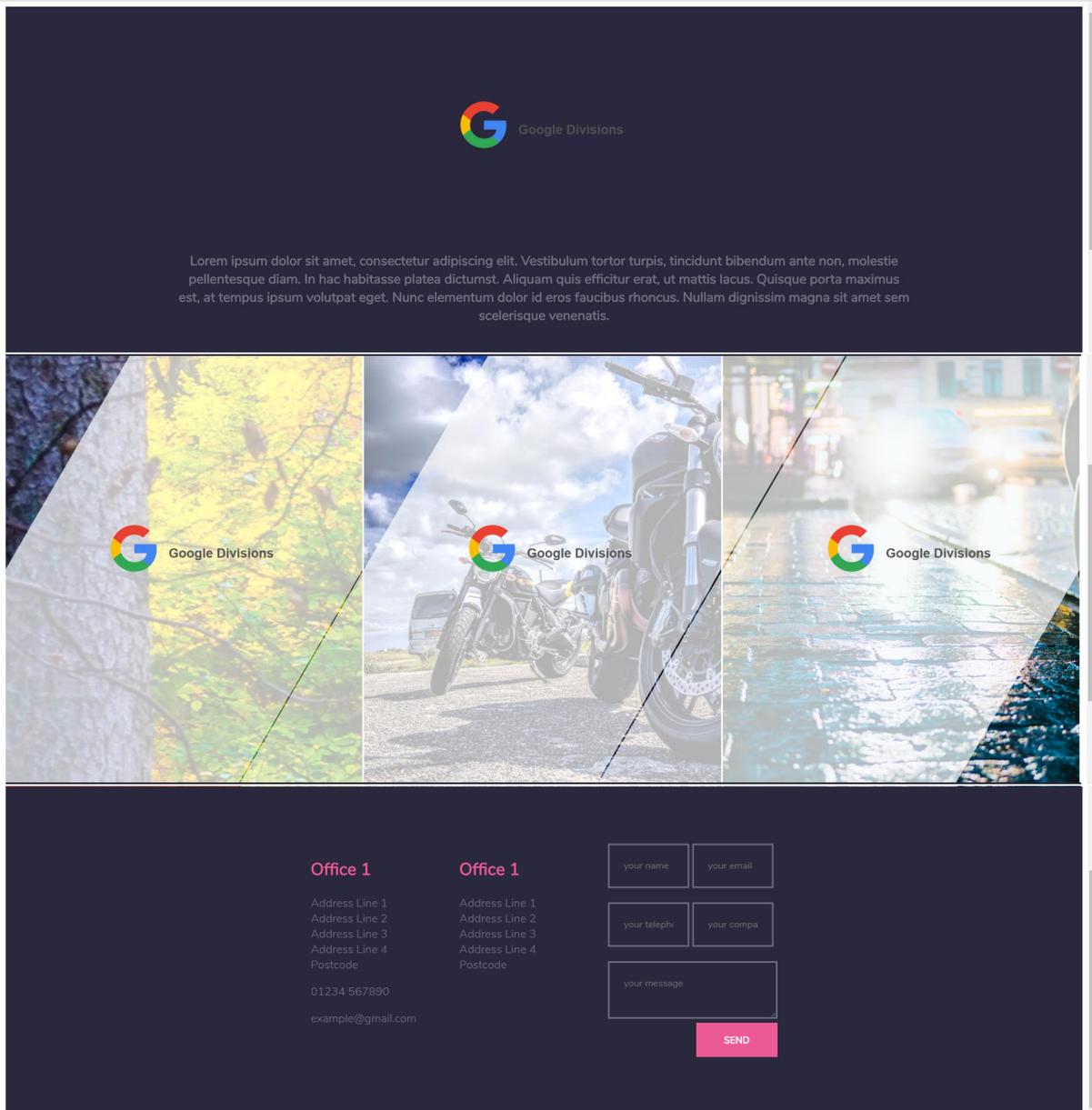
.form_container4 {
  width: 100%;
  height: auto;
}
```

```
.submit_button {
  padding: 15px!important;
  cursor: pointer;
  transition: 0.5s;
  background-color: #EB5A95!important;
  color: white!important;
  text-transform: uppercase;
  float: right;
  font-weight: bold;
  border: none!important;
}

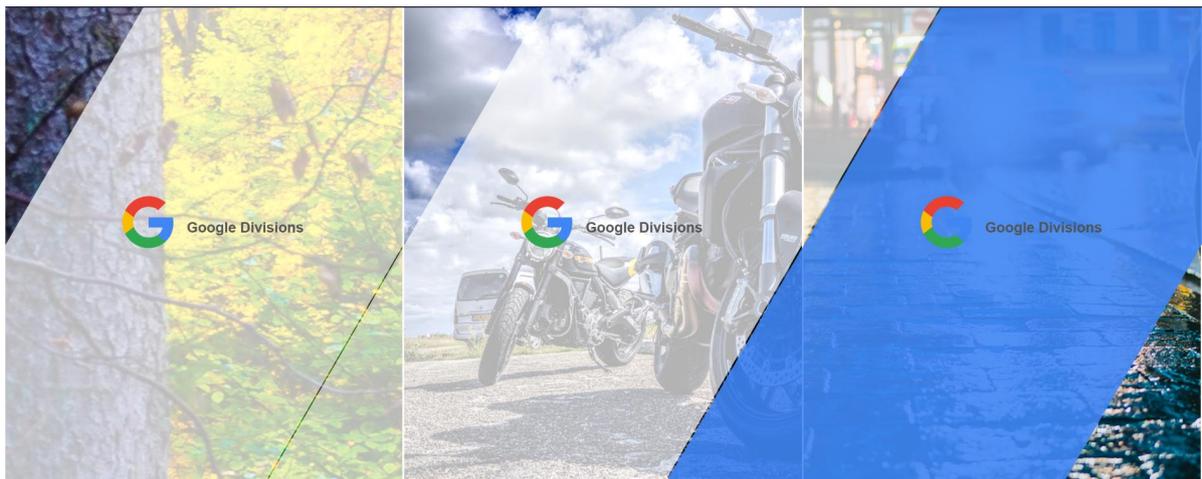
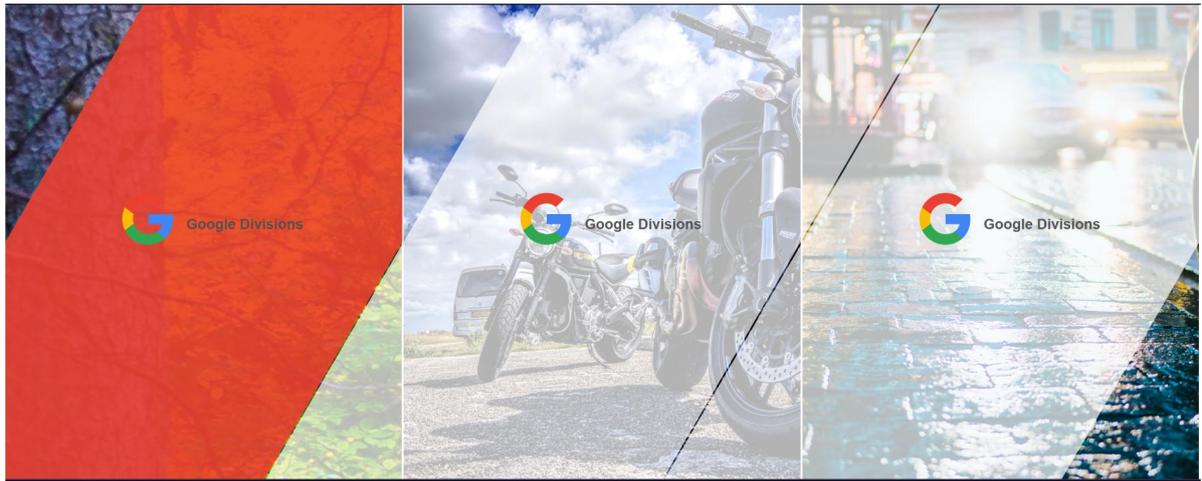
.submit_button:hover {
  background-color: white!important;
  color: #EB5A95!important;
  transition: 0.5s;
}
```

'Coulweb' Work Experience Documentation – Daniel Wilkins

The Final Outcome on the Web Page (General and Whilst Hovering over Sections)



'Coulweb' Work Experience Documentation – Daniel Wilkins



General Introduced Aspects

Introduction

As well as the tasks set above, I was also shown how to use the 'strip_tags' function. The developer showed me an example of where an error would occur and how to resolve this using 'strip_tags'.

This process can be viewed below:

The Original HTML/PHP Code of the Page

```
<a onclick="window.open('https://pinterest.com/pin/create/button?url=<?php echo urlencode($post->getAddress()); ?>&media=https://www.mayfairgallery.com/media/wysiwyg'+pictureURLBlogShare+'&description=<?php echo urlencode($post->getTitle()); ?>', 'pinterest', 'width=640,height=480,left=0,top=0,location=no,status=yes,scrollbars=yes,resizable=yes');" title="<?php echo $this->__('Pin it') ?>">
  <div class="pinterest icon">
    <i class="fa fa-pinterest"></i>
  </div>
</a>
```

The Outcome on the Web Page (Everything was as it Should have been)



Baccarat Crystal: Guide to France's
Luxury Glassmaker

Making Alterations to the Code

```
<a onclick="window.open('https://pinterest.com/pin/create/button?url=<?php echo urlencode($post->getAddress()); ?>&media=https://www.mayfairgallery.com/media/wysiwyg'+pictureURLBlogShare+'&description=<?php echo urlencode($post->getTitle()); ?> | <?php echo urlencode($post->getShortContent()); ?>', 'pinterest', 'width=640,height=480,left=0,top=0,location=no,status=yes,scrollbars=yes,resizable=yes');" title="<?php echo $this->__('Pin it') ?>">
  <div class="pinterest icon">
    <i class="fa fa-pinterest"></i>
  </div>
</a>
```

The Outcome of the Changes on the Page (The Issue Occurred of the HTML Code being Displayed – The 'p' tags)



Baccarat Crystal: Guide to France's Luxury Glassmaker | Buyers in the 19th Century were enamoured with Baccarat glassware. Our guide to the firm looks at the company's most iconic designs.

Utilising the function 'strip_tags' to Remove the HTML Code from the Page and the Outcome (The Problem had now been Resolved)

```
<a onclick="window.open('https://pinterest.com/pin/create/button/?url=?php echo urlencode($post->getAddress()); ?&media=https://www.mayfairgallery.com/media/wysiwyg'+pictureURLBlogShare+'&description=?php echo urlencode($post->getTitle()); ?> | <?php echo urlencode(strip_tags($post->getShortContent())); ?> ', 'Pinterest', 'width=640,height=480,left=0,top=0,location=no,status=yes,scrollbars=yes,resizable=yes');" title=?php echo $this->__('Pin it') ?>">
  <div class="Pinterest icon">
    <i class="fa fa-pinterest"></i>
  </div>
</a>
```



Baccarat Crystal: Guide to France's Luxury Glassmaker | Buyers in the 19th Century were enamoured with Baccarat glassware. Our guide to the firm looks at the company's most iconic designs.

Conclusion

Throughout the work experience at 'Coullweb', I managed to enhance different areas of my skillset through exploring different technologies. Although challenging at times, utilising 'PHP' and 'MySQL' helped me to better my understanding of how certain aspects of a database work and how to apply this knowledge for future work. I am very thankful to 'Coullweb' for providing myself with the opportunity to undertake this work experience and currently, I am in the process of receiving written feedback from the company.

Bibliography/Reference List/Acknowledgements for the Work Experience

ACF (2018) Documentation. Available at: <https://www.advancedcustomfields.com/resources/>. [Accessed 13 November 2018], [online].

ACF (2018) Order posts by custom fields. Available at: <https://www.advancedcustomfields.com/resources/order-posts-by-custom-fields/>. [Accessed 13 November 2018], [online].

ACF (2018) Query posts by custom fields. Available at: <https://www.advancedcustomfields.com/resources/query-posts-custom-fields/>. [Accessed 13 November 2018], [online].

Boronczyk, T. (2013) Generating One-Time Use URLs. *Php*. April 9. Available at: <https://www.sitepoint.com/generating-one-time-use-urls/>. [Accessed 20 November 2018], [online].

Buckler, C. (2018) How to Apply CSS3 Transforms to Background Images. *Advanced CSS Best Practices*. Available at: <https://www.sitepoint.com/css3-transform-background-image/>. [Accessed 23 November 2018], [online].

Google (n.d.) google logo transparent background – Google Search. Available at: https://www.google.co.uk/search?q=google+logo+transparent+background&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiui9W847XfAhVvShUIHTiBBpkQ_AUIDigB&biw=1536&bih=754. [Accessed 23 November 2018], [online].

Lorem Ipsum (n.d.) Lorem Ipsum. Available at: <https://www.lipsum.com/feed/html>. [Accessed 23 November 2018], [online].

Mahajan, M. (2015) PHP String Concatenation- How To Concatenate String And Variable. *Internet*. October 9. Available at: <https://tctrick.org/php-concatenate-variable-and-string/>. [Accessed 19 November 2018], [online].

Mike (n.d.) 1000+ Great Background Image Photos . Pexels . Free Stock Photos. Pexels. Available at: <https://www.pexels.com/photo/photography-of-street-during-rainy-day-1207951/>. [Accessed 23 November 2018], [online].

PHP (2001) elseif/else if. The PHP Group. Available at: <http://php.net/manual/en/control-structures.elseif.php>. [Accessed 20 November 2018], [online].

PHP (2001) foreach. The PHP Group. Available at: <http://php.net/manual/en/control-structures.foreach.php>. [Accessed 13 November 2018], [online].

PHP (2001) mysql_fetch_row. The PHP Group. Available at: <http://php.net/manual/en/function.mysql-fetch-row.php>. [Accessed 19 November 2018], [online].

PHP (2001) mysqli_result::fetch_assoc. The PHP Group. Available at: <http://php.net/manual/en/mysqli-result.fetch-assoc.php>. [Accessed 19 November 2018], [online].

PHP (2001) PHP: Hypertext Preprocessor. The PHP Group. Available at: <http://www.php.net/>. [Accessed 23 December 2018], [online].

PHP (2001) ucfirst. The PHP Group. Available at: <http://php.net/manual/en/function.ucfirst.php>. [Accessed 13 November 2018], [online].

PHP (2001) Variable scope. The PHP Group. Available at:

<http://php.net/manual/en/language.variables.scope.php>. [Accessed 16 November 2018], [online].

Pixabay (2016) 1000+ Great Background Image Photos . Pexels . Free Stock Photos. Pexels. Available

at: <https://www.pexels.com/photo/background-biker-clouds-daylight-207171/>. [Accessed 23 November 2018], [online].

Reitmeier, F. (2017) 1000+ Great Background Image Photos . Pexels . Free Stock Photos. Pexels.

Available at: <https://www.pexels.com/photo/timelapse-photography-of-falls-near-trees-707915/>. [Accessed 23 November 2018], [online].

Reitmeier, F. (2017) 1000+ Great Background Image Photos . Pexels . Free Stock Photos. Pexels.

Available at: <https://www.pexels.com/photo/timelapse-photography-of-falls-near-trees-707915/>. [Accessed 23 November 2018], [online].

Stack Overflow (2009) Differences between INDEX, PRIMARY, UNIQUE, FULLTEXT in MySQL?.

Available at: <https://stackoverflow.com/questions/707874/differences-between-index-primary-unique-fulltext-in-mysql>. [Accessed 16 November 2018], [online].

Stack Overflow (2009) MySQL Insert Where query. Available at:

<https://stackoverflow.com/questions/485039/mysql-insert-where-query>. [Accessed 16 November 2018], [online].

Stack Overflow (2011) MySQL Check if username and password matches in Database. Available at:

<https://stackoverflow.com/questions/5285388/mysql-check-if-username-and-password-matches-in-database>. [Accessed 16 November 2018], [online].

Stack Overflow (2012) Deleting a PHP cookie upon user logout. Available at:

<https://stackoverflow.com/questions/13862854/deleting-a-php-cookie-upon-user-logout>. [Accessed 19 November 2018], [online].

Stack Overflow (2012) How do I link a JavaScript file to a HTML file?. Available at:

<https://stackoverflow.com/questions/13739568/how-do-i-link-a-javascript-file-to-a-html-file>. [Accessed 23 November 2018], [online].

Stack Overflow (2013) Get the data of specific user in PHP. Available at:

<https://stackoverflow.com/questions/17075424/get-the-data-of-specific-user-in-php>. [Accessed 19 November 2018], [online].

Stack Overflow (2013) PHP Login Page Redirect. Available at:

<https://stackoverflow.com/questions/19782194/php-login-page-redirect>. [Accessed 19 November 2018], [online].

Stack Overflow (2014) PHP and MySQL Select a Single Value. Available at:

<https://stackoverflow.com/questions/20878089/php-and-mysql-select-a-single-value>. [Accessed 19 November 2018], [online].

w3resource (2018) MySQL not equal to (<>, !=) operator. Available at:

<https://www.w3resource.com/mysql/comparison-functions-and-operators/not-equal-operator.php>. [Accessed 3 December 2018], [online].

w3Schools (1999) CSS background-image Property. Available at:

https://www.w3schools.com/cssref/pr_background-image.asp. [Accessed 23 November 2018], [online].

w3Schools (1999) CSS transform Property. Available at:
https://www.w3schools.com/cssref/css3_pr_transform.asp. [Accessed 23 November 2018], [online].

w3Schools (1999) HTML <input type="password">. Available at:
https://www.w3schools.com/tags/att_input_type_password.asp. [Accessed 20 November 2018], [online].

w3Schools (1999) HTML <input> type Attribute. Available at:
https://www.w3schools.com/tags/att_input_type.asp. [Accessed 20 November 2018], [online].

w3Schools (1999) HTML <meta> charset Attribute. Available at:
https://www.w3schools.com/tags/att_meta_charset.asp. [Accessed 23 November 2018], [online].

w3Schools (1999) HTML <select> Tag. Available at: https://www.w3schools.com/tags/tag_select.asp. [Accessed 13 November 2018], [online].

w3Schools (1999) HTML Color Picker. Available at:
https://www.w3schools.com/colors/colors_picker.asp. [Accessed 23 November 2018], [online].

w3Schools (1999) HTML Forms. Available at: https://www.w3schools.com/html/html_forms.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) HTML Styles – CSS. Available at: https://www.w3schools.com/html/html_css.asp. [Accessed 23 November 2018], [online].

w3Schools (1999) HTTP Request Methods. Available at:
https://www.w3schools.com/tags/ref_httpmethods.asp. [Accessed 19 November 2018], [online].

w3Schools (1999) PHP 5 Cookies. Available at: https://www.w3schools.com/php/php_cookies.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP 5 Form Handling. Available at:
https://www.w3schools.com/php/php_forms.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP 5 Form Handling. Available at:
https://www.w3schools.com/php/php_forms.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP 5 if...else...elseif Statements. Available at:
https://www.w3schools.com/php/php_if_else.asp. [Accessed 20 November 2018], [online].

w3Schools (1999) PHP 5 Operators. Available at:
https://www.w3schools.com/php/php_operators.asp. [Accessed 19 November 2018], [online].

w3Schools (1999) PHP Insert Data Into MySQL. Available at:
https://www.w3schools.com/php/php_mysql_insert.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP Select Data From MySQL. Available at:
https://www.w3schools.com/php/php_mysql_select.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP Select Data From MySQL. Available at:
https://www.w3schools.com/php/php_mysql_select.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) PHP Select Data From MySQL. Available at:
https://www.w3schools.com/php/php_mysql_select.asp. [Accessed 19 November 2018], [online].

'Coullweb' Work Experience Documentation – Daniel Wilkins

w3Schools (1999) SQL AND, OR and NOT Operators. Available at:
https://www.w3schools.com/sql/sql_and_or.asp. [Accessed 16 November 2018], [online].

w3Schools (1999) SQL WHERE Clause. Available at: https://www.w3schools.com/sql/sql_where.asp. [Accessed 16 November 2018], [online].

Wikipedia (2018) Query string. Available at: https://en.wikipedia.org/wiki/Query_string. [Accessed 19 November 2018], [online].

WordPress (n.d.) Class Reference/WP Query. Available at:
https://codex.wordpress.org/Class_Reference/WP_Query. [Accessed 13 November 2018], [online].

WordPress (n.d.) wp_mail(). Available at:
https://developer.wordpress.org/reference/functions/wp_mail/. [Accessed 19 November 2018], [online].

I was also helped by the developer at 'Coullweb' to undertake all set tasks

THIS IS THE END OF THE DOCUMENT