

(FINAL) Zaeem Assignment 2 - Gaynor

by Zaeem Gulammamodo

FILE	17602_ZAEEM_GULAMMAMODO__FINAL__ZAEEM_ASSIGNMENT_2_-_GAYNOR_308223_798118385.DOCX (3.74M)		
TIME SUBMITTED	09-JAN-2021 02:40AM (UTC+0000)	WORD COUNT	4729
SUBMISSION ID	141322792	CHARACTER COUNT	26263



Image [1]

Assignment 2

Name: Zaeem

Student No.: 1902036

Course: Applied Computing (Web Development)

Module: Web and Mobile Technologies

Date: 6/12/2020

Table Of Contents

<u>Content</u>	<u>Page Number</u>
1. Introduction	3
2. Website	5
2.1. Website Theme	5
2.2. Panel Feature (Site Wide Features)	7
2.3. Transitions / Animations (Site Wide Features)	12
2.4. Single Page Application (Site Wide Features)	15
2.5. Home Page	19
2.6. Batman's Biography Page	21
2.7. Batman's Connections Page	26
2.8. Heroes Page	30
2.9. Villains Page	36
2.10. References Page	42
3. Conclusion	44
4. References	45
5. Bibliography	46

1. Introduction

In this documentation, a website has been created with features of JavaScript, jQuery, AJAX, JSON and PHP for the user to interact with the website. This website is to show how well website coding skills using HTML, CSS, JavaScript, jQuery, PHP, AJAX and JSON have been developed. When creating the website, the program Notepad++ had to be used in order to type the HTML code, CSS code, JavaScript code, jQuery code, PHP code, AJAX code and JSON code.

What is AJAX?

AJAX is a term that refers to Asynchronous JavaScript and XML which represents a collection of programming strategies used for website and web application development. The primary objective of AJAX is to asynchronously keep updating online content (the A" of AJAX), ensuring that a user's internet browser does not need to refresh the web page because it needs to modify just a small element of the content on the website. [1]

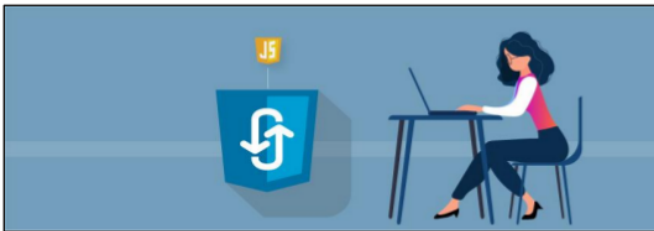


Image [3]

What is JSON?

JSON (JavaScript Object Notation) would be a widely used structured format for transferring information as text that can be transferred over a network. Most of Databases and APIs use it and it is simple to understand for both humans and computers. [2]



Image [5]

What is PHP?

PHP would be a scripting language on the server side which is used to create web pages that are interactive and can communicate with databases. It is a commonly used open source language which can be embedded within HTML and is specifically used for web application creation. [3]

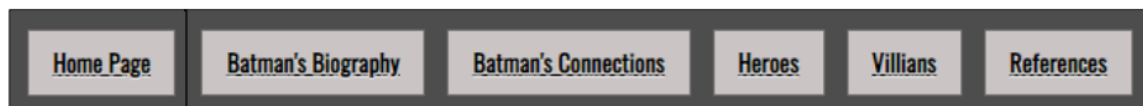


Image [3]

2. Website

The created website will be focused on DC's superhero Batman. This topic was chosen because it is a popular superhero. The Batman website will contain six different created webpages and they are called:

- Home
- Batman's Biography
- Batman's Connections
- Heroes
- Villains
- References



2.1. Website Theme

As this is Batman website, the colour scheme for this website will be based around Batman costume colours which is black, yellow, grey and dark grey. The colour scheme for this created website was used with jQuery ThemeRoller.

jQuery ThemeRoller is online tool which is used for designing fun and creative themes for created websites as the user can create a theme and download it as a CSS file for their website which saves the user's time to code a CSS file for the theme to their website.

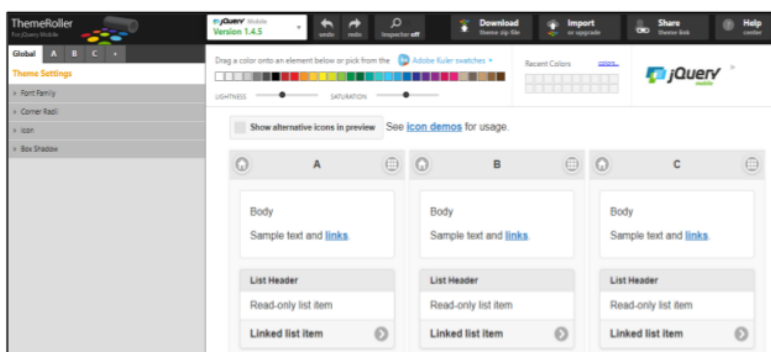
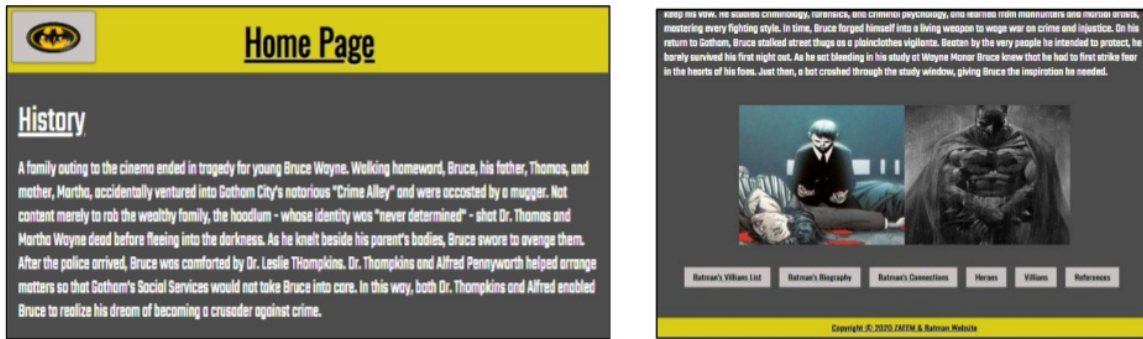


Image [7]

Link = <https://themeroller.jquerymobile.com>



Images [8]

In the images above, it shows the Home Page and theme of the Batman website. The website header and footer have the colour gold with black text, the body background is dark grey with white text and grey buttons with black text. This colour scheme has made the Batman website appealing and creative to users.

2.2. Panel Feature (Site Wide Features)

The first feature on every created web page will be a jQuery Mobile feature. This feature is called Panel. This feature will allow the user to change the interface display on the website for their specific purpose. This feature will add a widget onto the website for the user.

In the image below, it shows the website interface when the user goes on the home page.



In the image below, it shows the website interface when the user clicks Batman's Villains List button.



This feature will allow users to view a side scroll menu with data on the panel menu as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

Also, this feature has used jQuery ThemeRoller with the colour scheme as the background colour is grey and the text is black, paying tribute to the Batman costume colour.

This feature has prevented to create a separate page for the content as the user can see the famous villains from the Batman franchise on one organised widget within a jQuery feature. This feature will be available to view on every created web page.

Below will be the annotated code listing for this feature

HTML:

```
<script src="Assets/jquery-3.5.1.js"></script> <!-- this loads up the jquery file -
->

<script src="http://code.jquery.com/jquery-migrate-3.0.0.js"></script> <!-- this
loads up the jquery file -->

<script src="Assets/jquery.mobile-1.4.5/jquery.mobile-1.4.5.js"></script> <!--
this loads up the jquery file -->
```

This feature uses jQuery language so the user has to add the jQuery file on the HTML code.

```
http://code.jquery.com/mobile/1.4.5/jquery.mobile.structure-1.4.5.min.css"
/> <!-- this loads up the css file -->

<link rel="stylesheet" href="Assets/jquery-mobile-theme-152719-
0/themes/batman_theme_v4.min.css" /> <!-- this loads up the css file -->

<link rel="stylesheet" href="Assets/jquery-mobile-theme-152719-
0/themes/jquery.mobile.icons.min.css" /> <!-- this loads up the css file -->
```

This feature uses jQuery language so the user has to add the jQuery ThemeRoller CSS file on the HTML code.

```
<a href="#leftpanel" id="mybutton" class="ui-btn ui-btn-inline"> <font
face="Oswald"> <ins> Batman's Villains List </ins> </font> </a>
```

This feature uses jQuery language so the user has to add button configuration on the HTML code.

```
<!-- start of the panel bar configurations -->
  <div data-role="panel" id="leftpanel">
    <center>
      <h2>
        <font face="Teko">
          <font size="10">
            <ins> Villains List </ins>
          </font>
        </h2>
      </center>
      <font size="5">
        <ul>
          <li> Joker </li>
          <li> Harley Quinn </li>
          <li> Riddler </li>
          <li> Bane </li>
          <li> Two Face </li>
          <li> Poison Ivy </li>
          <li> Deadshot </li>
          <li> Killer Croc </li>
          <li> Scarecrow </li>
          <li> Catwomen </li>
        </ul>
      </font>
    </div>
```

Zaeem

```
<!-- end of the panel bar configurations -->
```

This feature uses jQuery and JavaScript language so the user has to configure the settings and add the content on the HTML code.

With this all of this code listing above, this feature has successfully been created and worked on the created website.

2.3. Transitions / Animations (Site Wide Features)

The second feature on every created web page will be a jQuery Mobile feature. This feature is called Transitions / Animations. This feature will allow the user to change the interface display on the website for their specific purpose. This feature will play a moving animation onto the website for the user when a new page has been clicked.

In the image below, it shows the website interface when the user goes on the home page.



In the image below, it shows the website interface when the user clicks References button.

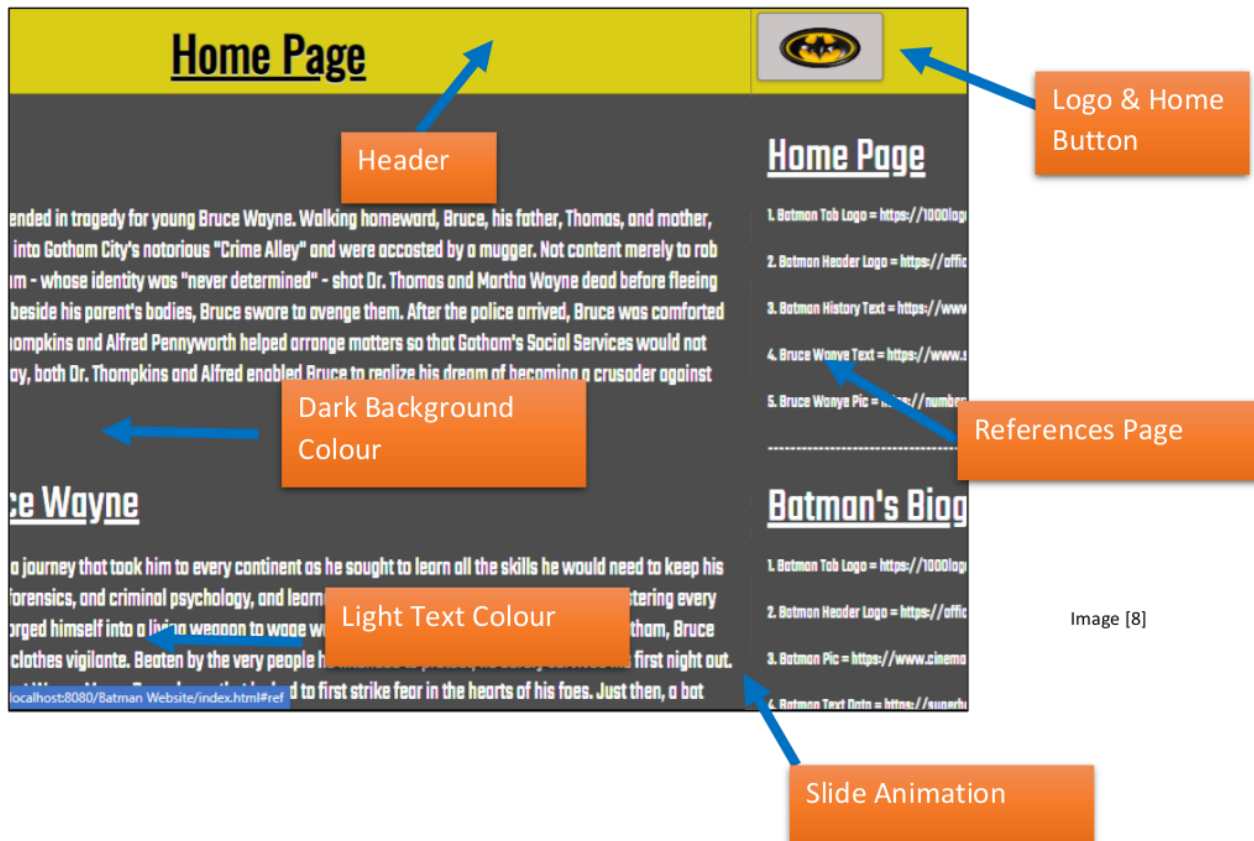


Image [8]

This feature will allow users to view an animation being played when changing to new page as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

Also, there are six different animations being played on the Batman Website and they are called Flip, Slide, Flow, Pop, Turn, Slideup and Push for the Panel Widget.

This feature has made the website more fun and creative for the content viewed by the users. This feature will be available to view on every created web page.

Below will be the annotated code listing for this feature

HTML:

```
<!-- start of the nav bar buttons configurations -->

    <center>

        <a href="#leftpanel" id="mybutton" class="ui-btn ui-btn-inline"> <font
face="Oswald"> <ins> Batman's Villains List </ins> </font> </a>

        <a href="index.html" id="mybutton" class="ui-btn ui-btn-inline"
data-ajax="false" data-transition="flip"> <font face="Oswald"> <ins> Home
Page </ins> </font> </a>

        <a href="bio.html" id="mybutton" class="ui-btn ui-btn-inline"
data-ajax="false" data-transition="flow"> <font face="Oswald"> <ins>
Batman's Biography </ins> </font> </a>

        <a href="connections.html" id="mybutton" class="ui-btn ui-btn-
inline" data-ajax="false" data-transition="pop"> <font face="Oswald"> <ins>
Batman's Connections </ins> </font> </a>

        <a href="showinfo_heroes.php" id="mybutton" class="ui-btn ui-
btn-inline" data-ajax="false" data-transition="turn"> <font face="Oswald">
<ins> Heroes </ins> </font> </a>

        <a href="showinfo_villains.php" id="mybutton" class="ui-btn ui-
btn-inline" data-ajax="false" data-transition="slideup"> <font face="Oswald">
<ins> Villains</ins> </font> </a>

<a href="#ref" id="mybutton" class="ui-btn ui-btn-inline" data-
transition="slide"> <font face="Oswald"> <ins> References </ins> </font> </a>

    </center>

<!-- end of the nav bar buttons configurations -->
```

This feature uses jQuery and JavaScript language so the user has to configure the settings and add the content on the HTML code.

With this all of this code listing above, this feature has successfully been created and worked on the created website.

2.4. Single Page Application (Site Wide Features)

The next feature on every created web page will be a jQuery Mobile feature. This feature is called Single Page Application. This feature will allow the user to change the interface display on the website for their specific purpose. This feature will go to another web page without creating or linking a new HTML file.

In the image below, it shows the website interface when the user goes on the home page.

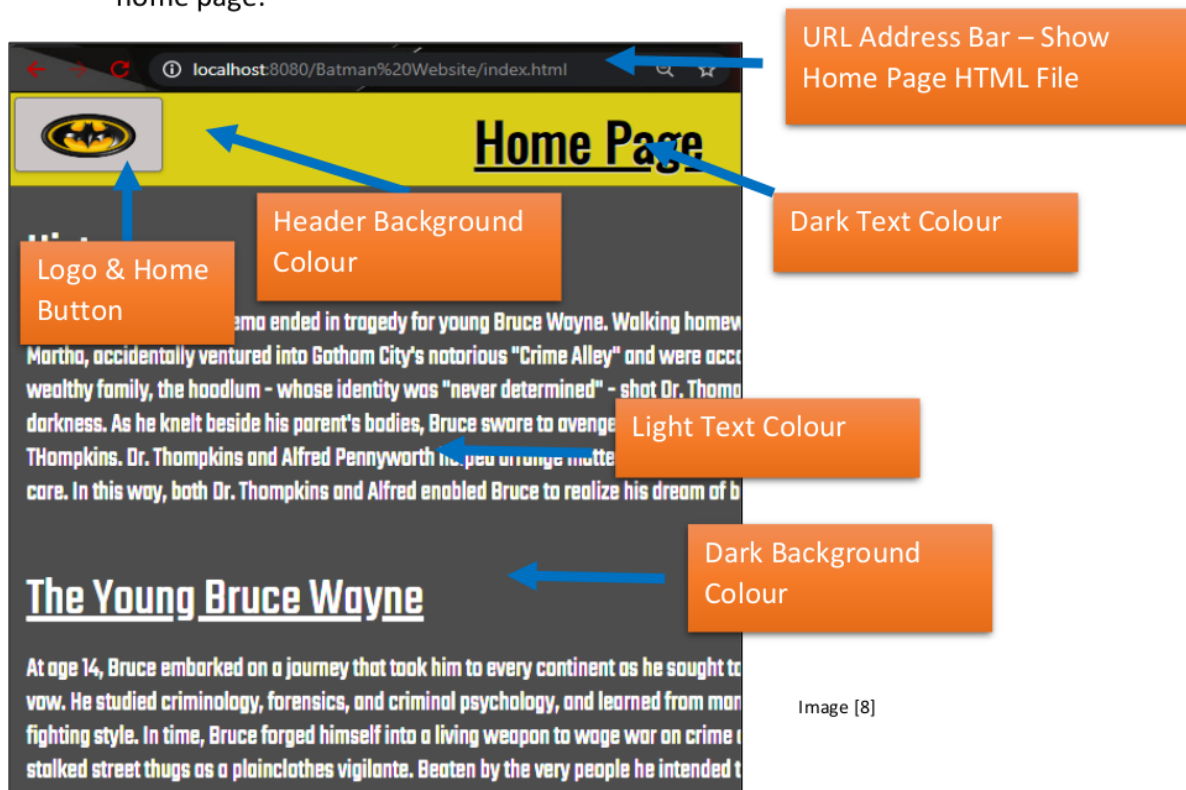


Image [8]

In the image below, it shows the website interface when the user clicks References button.



This feature will allow users to view a new page within the same HTML page the user was previously on as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

This feature has made the website more fun and creative for the content viewed by the users. Also, this feature has saved storage space and time as the user may not have rely on the internet connection to load the content for that specific page as the content will be loaded on the same page the user is on. This feature will be available to view on every created web page.

Below will be the annotated code listing for this feature

HTML:

```
<!-- this is the multiple page application configurations - start -->
<!-- this loads up home page image, button and header configurations - start -->
>
<div data-role="page" id="ref">
  <div data-role="header">
    <a href="index.html" data-ajax="false">
       </a>
    <h1>
      <font face="Oswald">
        <font size="7">
          <ins> References </ins>
        </font>
      </font>
    </h1>
  </div>
  <!-- title on page -->
  <div data-role="main" class="ui-content">
    <p>
      <font face="Teko">
        <font size="15">
          <ins> Home Page </ins>
        </font>
      </font>
    </p>
```

Zaeem

```
<!-- start of the footer configurations -->  
<div data-role="footer">  
  <h1>  
    <font face="Oswald">  
      <ins> Copyright &copy; 2020 ZAEEM & Batman Website </ins>  
    </font>  
  </h1>  
</div>  
</div>  
<!-- end of the footer configurations -->  
<!-- this is the multiple page application configurations - end -->
```

This feature uses jQuery and JavaScript language so the user has to configure the settings and add the content on the HTML code.

```
<a href="#ref" id="mybutton" class="ui-btn ui-btn-inline" data-  
transition="slide"> <font face="Oswald"> <ins> References </ins> </font> </a>
```

This feature uses jQuery language so the user has to add button configuration on the HTML code.

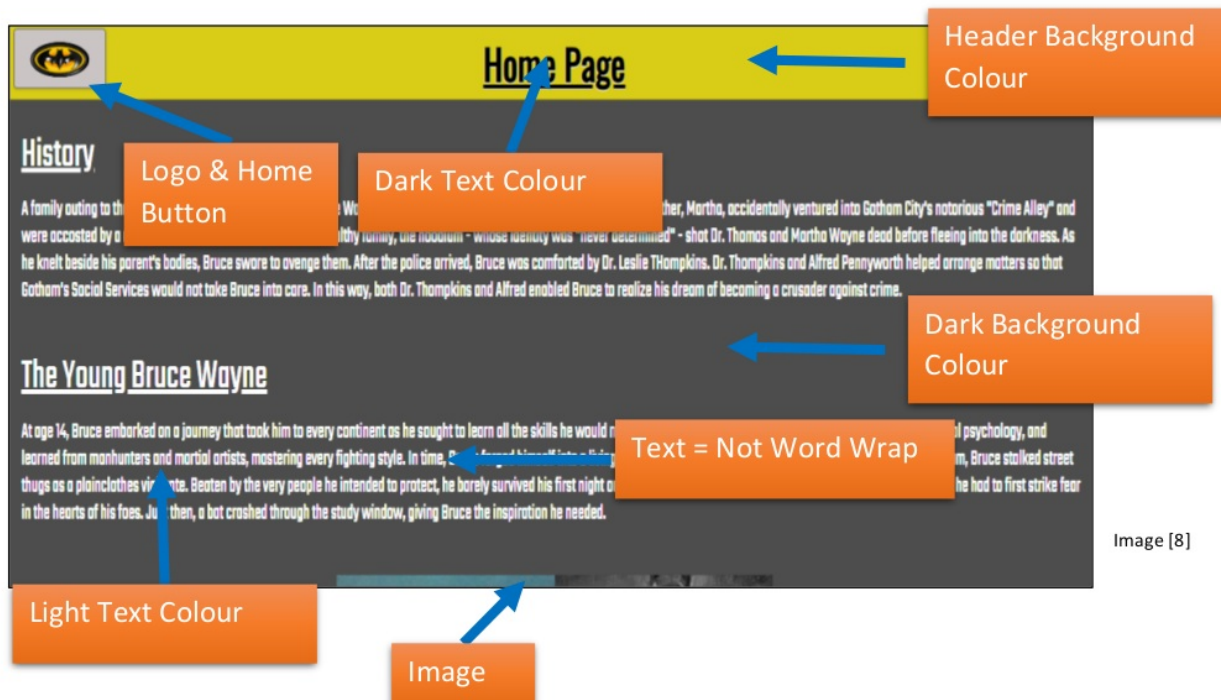
With this all of this code listing above, this feature has successfully been created and worked on the created website.

2.5. Home Page

The first web page on the website is called Home Page. This webpage will allow users to view the history and an image of Batman. No new feature will be used and shown on this webpage.

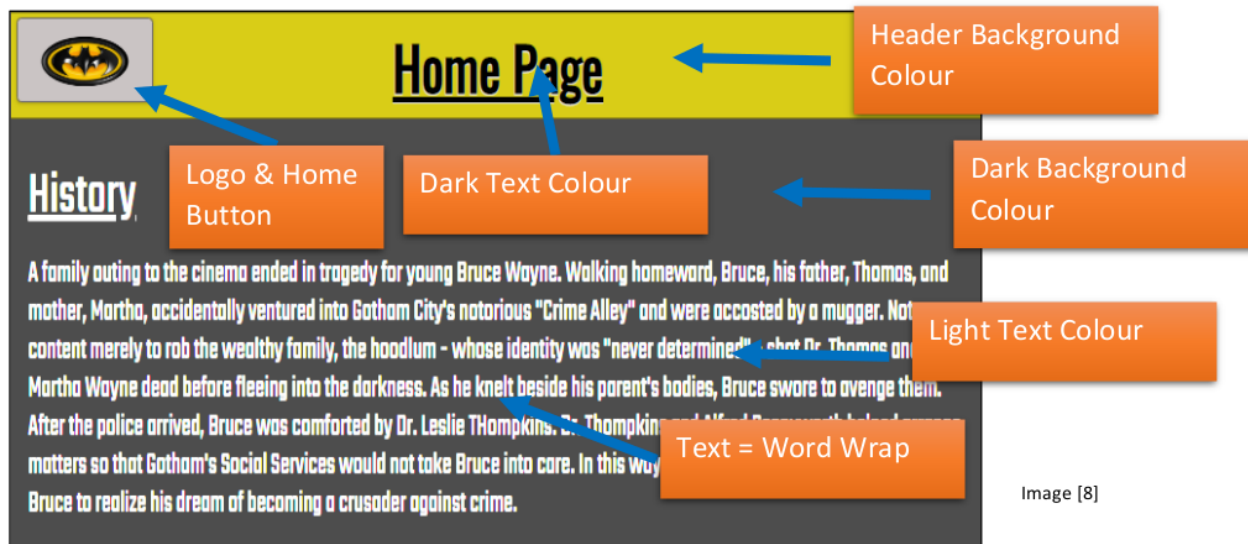
In the image below, it shows the website interface when the user goes on the home page.

Website Window – Full Maximise:



Zaeem

Website Window – Half Maximise:



As there no new feature on this web page, an annotated code listing will not be shown in this documentation.

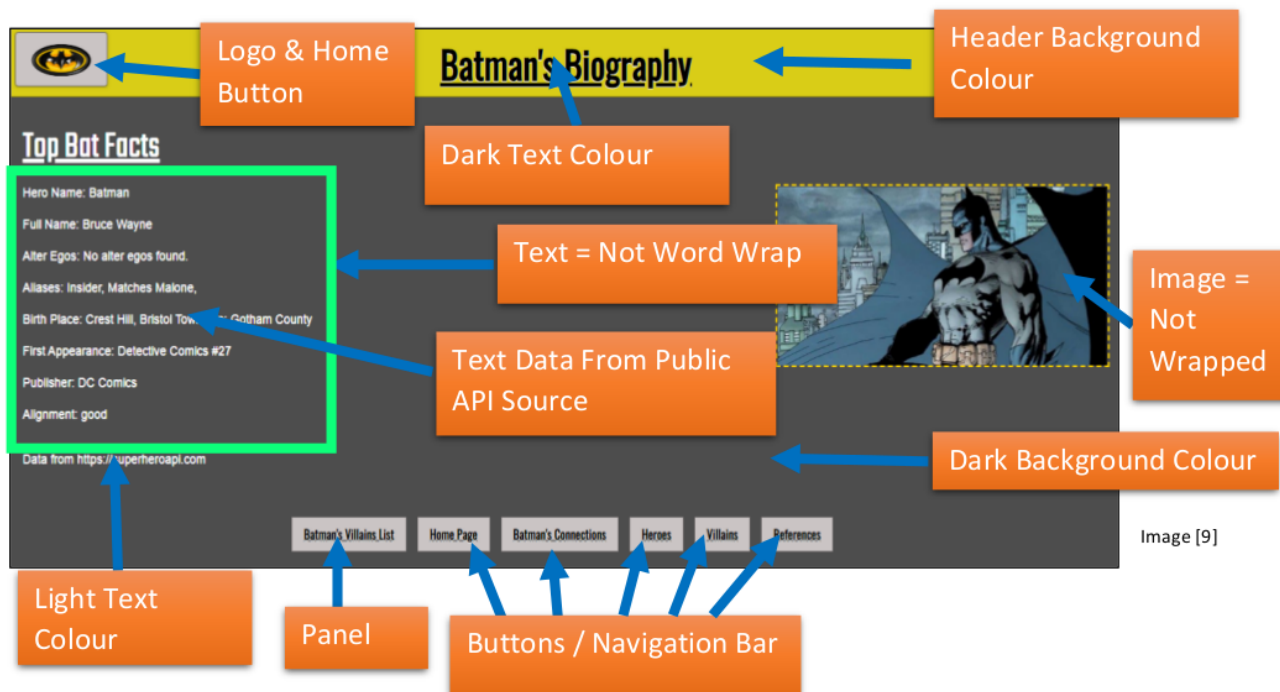
2.6. Batman's Biography Page

Public JSON API – No. 1

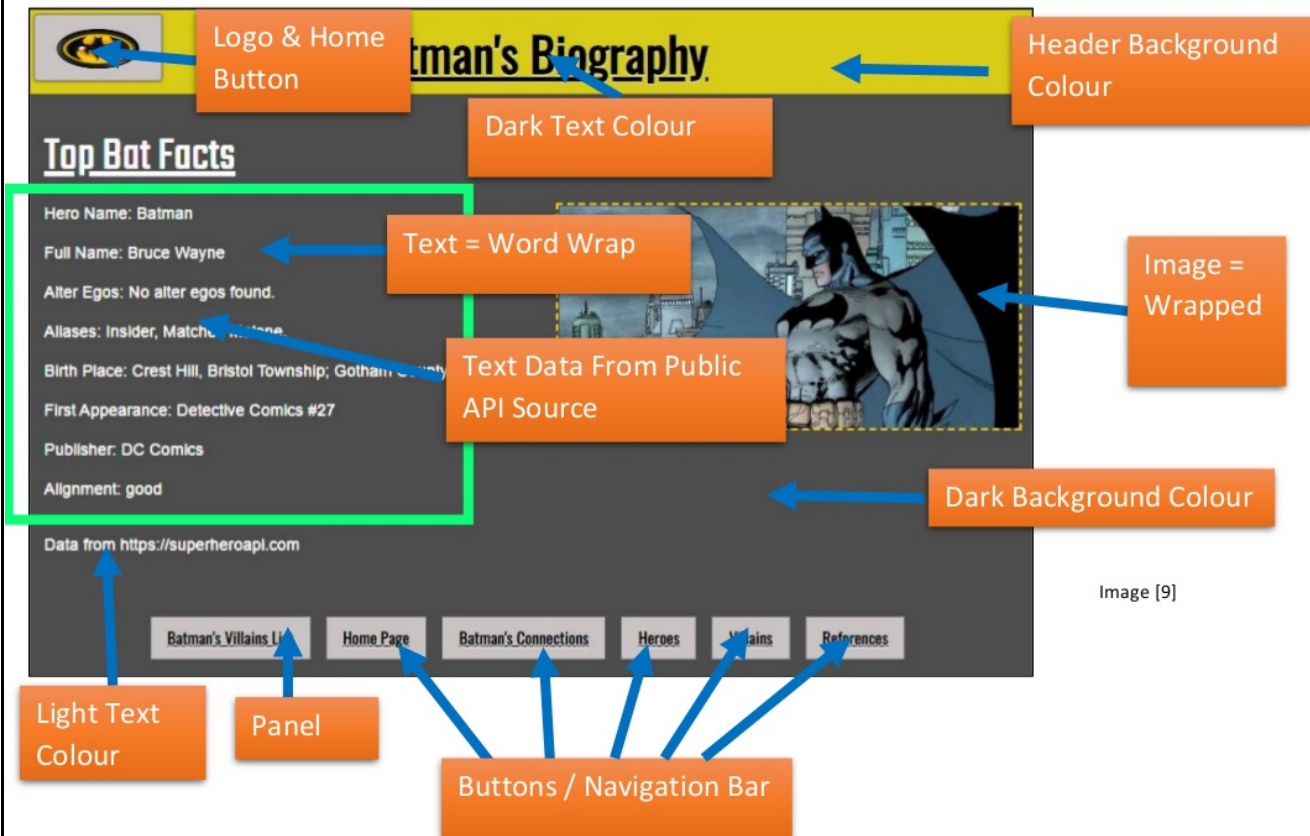
The second web page on the website is called Batman's Biography Page. This webpage will allow users to view the biography data and an image of Batman. The feature that was used on this page was a public JSON API as this is JSON and AJAX feature. This feature will allow users to view data from an online source.

In the image below, it shows the website interface when the user goes on the Batman's Biography page.

Website Window – Full Maximise:



Website Window – Half Maximise:



This feature will allow users to view Batman Biography data receiving from online text source as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

This feature has prevented typing out all of this data on the HTML file so if the data from the Public API gets updated, the HTML file may not need be updated / edited as the HTML is linked to URL of the data from the Public API source. This feature and data will be available to view on this web page only.

Below will be the annotated code listing for this feature

HTML:

```
<!-- this is the public api json configurations - start -->

<script>

var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {

    if (this.readyState == 4 && this.status == 200) {

        var myObj=JSON.parse(this.responseText);//response from our AJAX call

        mydemo=document.getElementById("demo") //connect to div tag

        //mydemo.innerHTML=JSON.stringify(myObj); //print RAW JSON to the
screen

        <!-- objId = myObj["id"]; -->

        <!-- mydemo.innerHTML += " ID: " + objId + "<br>"; -->

            //this will show the data from the public api

        objName = myObj["name"];

        mydemo.innerHTML += "Hero Name: " + objName + "<br><br>";

        objFullName = myObj["full-name"];

        mydemo.innerHTML += "Full Name: " + objFullName + "<br><br>";

        objAlter = myObj["alter-egos"];

            mydemo.innerHTML += "Alter Egos: " + objAlter + "<br><br>";

        objalias = myObj["aliases"];

        mydemo.innerHTML += "Aliases: " ;

        for(i=0; i < myObj["aliases"].length; i++)

            {

                alias = myObj["aliases"][i]

                mydemo.innerHTML += alias + ", "

            }

    }

}
```

Zaeem

```
objPlace = myObj["place-of-birth"];
mydemo.innerHTML += "<br><br>" + "Birth Place: " + objPlace + "<br><br>";
objFirst = myObj["first-appearance"];
mydemo.innerHTML += "First Appearance: " + objFirst + "<br><br>";
objPub = myObj["publisher"];
mydemo.innerHTML += "Publisher: " + objPub
+ "<br><br>";
objAlign = myObj["alignment"];
mydemo.innerHTML += "Alignment: " + objAlign;
/* for(i=0;i<9;i++)
{
cf=myObj["all"][i]["text"];
mydemo.innerHTML+=cf+"<br>"
}
*/
};

xmlhttp.open("GET",
"https://www.superheroapi.com/api.php/10157718024522703/70/biography"
, true); //this is public api link to get the data
xmlhttp.send();
</script>

<!-- this is the public api json configurations - end -->
```

This feature uses AJAX and JSON language so the user has to configure the settings and add the content on the HTML code.

```
<!-- this will show public api data on page -->  
<span id="demo">  
</span>
```

This feature uses HTML language so the user has to add show data configuration on the HTML code.

With this all of this code listing above, this feature has successfully been created and worked on the created website.

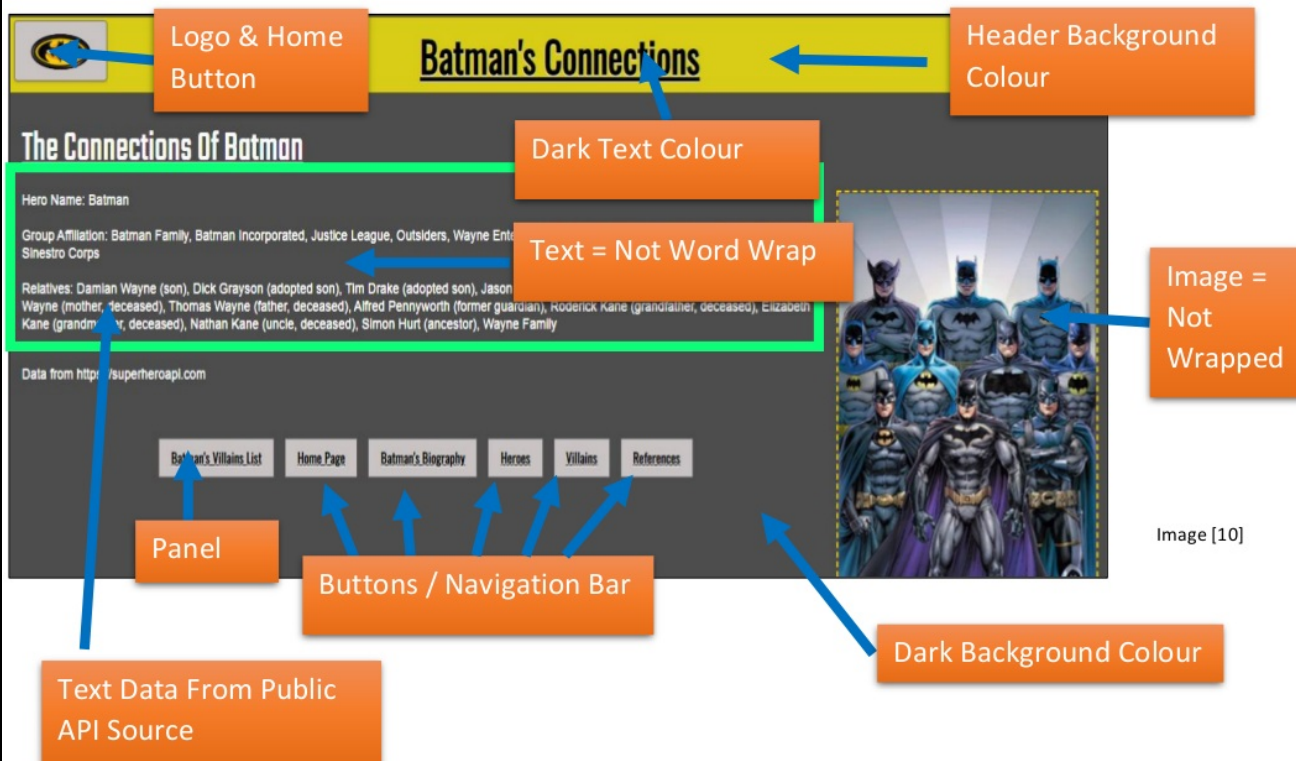
2.7. Batman's Connections Page

Public JSON API – No. 2

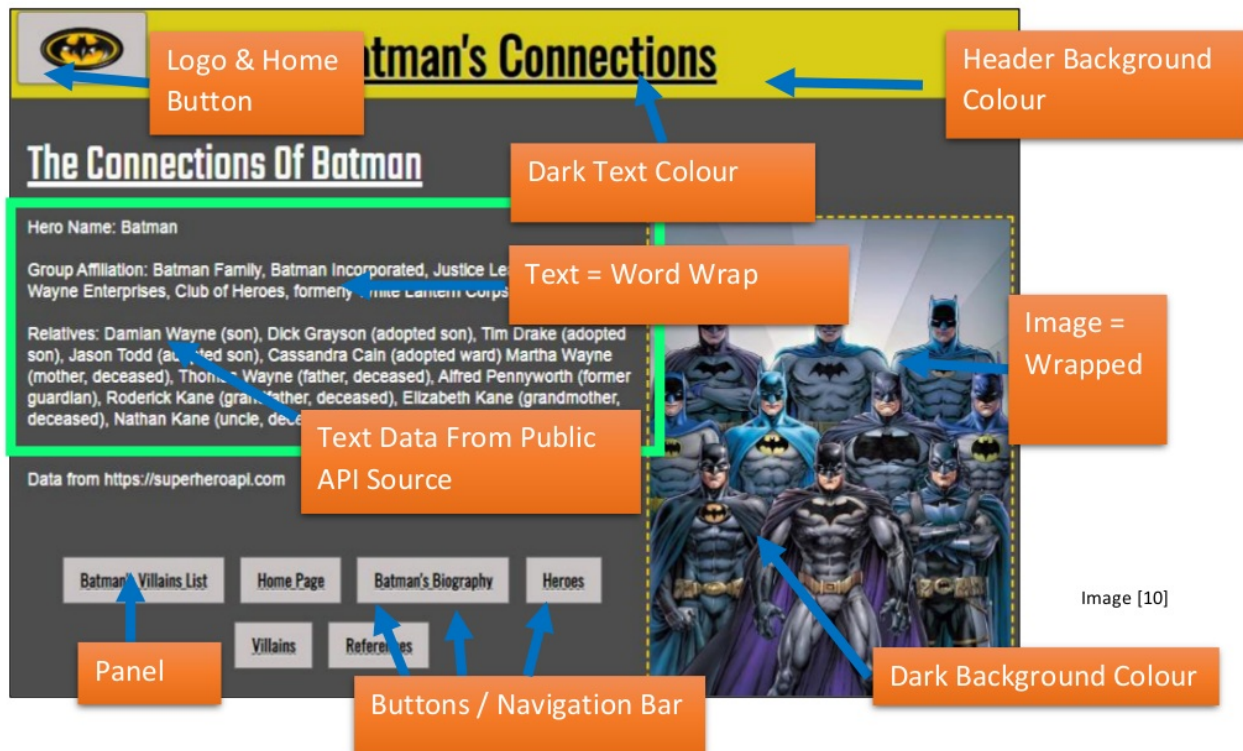
The third web page on the website is called Batman's Connections Page. This webpage will allow users to view the connections data and an image of various Batman's. The feature that was used on this page was a public JSON API as this is JSON and AJAX feature. This feature will allow users to view data from an online source.

In the image below, it shows the website interface when the user goes on the Batman's Connections page.

Website Window – Full Maximise:



Website Window – Half Maximise:



This feature will allow users to view Batman Connections data receiving from online text source as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

This feature has prevented typing out all of this data on the HTML file so if the data from the Public API gets updated, the HTML file may not need be updated / edited as the HTML is linked to URL of the data from the Public API source. This feature and data will be available to view on this web page only.

Below will be the annotated code listing for this feature

HTML:

```
<!-- this is the public api json configurations - start -->

<script>

var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var myObj=JSON.parse(this.responseText);//response from our AJAX call
        mydemo=document.getElementById("demo") //connect to div tag
        //mydemo.innerHTML=JSON.stringify(myObj); //print RAW JSON to the
screen

        <!-- objId = myObj["id"]; -->

        <!-- mydemo.innerHTML += " ID: " + objId + "<br>"; -->

            //this will show the data from the public api
objName = myObj["name"];
mydemo.innerHTML += "Hero Name: " + objName;

        objGA = myObj["group-affiliation"];
mydemo.innerHTML += "<br><br>" + "Group Affiliation: " ;
        for(i=0; i < myObj["group-affiliation"].length; i++)
        {
            ga = myObj["group-affiliation"][i]
            mydemo.innerHTML += ga
        }

        objRel = myObj["relatives"];
mydemo.innerHTML += "<br><br>" + "Relatives: " ;
        for(i=0; i < myObj["relatives"].length; i++)
        {
```

Zaeem

```
        rel = myObj["relatives"][i]
        mydemo.innerHTML += rel
    }
    /* for(i=0;i<9;i++)
    {
        cf=myObj["all"][i]["text"];
        mydemo.innerHTML+=cf+"<br>"
    }
    */
}
};

xmlhttp.open("GET",
"https://www.superheroapi.com/api.php/10157718024522703/connections", true); //this is public api link to get the data
xmlhttp.send();
</script>
<!-- this is the public api json configurations - end -->
```

This feature uses AJAX and JSON language so the user has to configure the settings and add the content on the HTML code.

```
<!-- this will show public api data on page -->
<span id="demo">
</span>
```

This feature uses HTML language so the user has to add show data configuration on the HTML code.

With this all of this code listing above, this feature has successfully been created and worked on the created website.

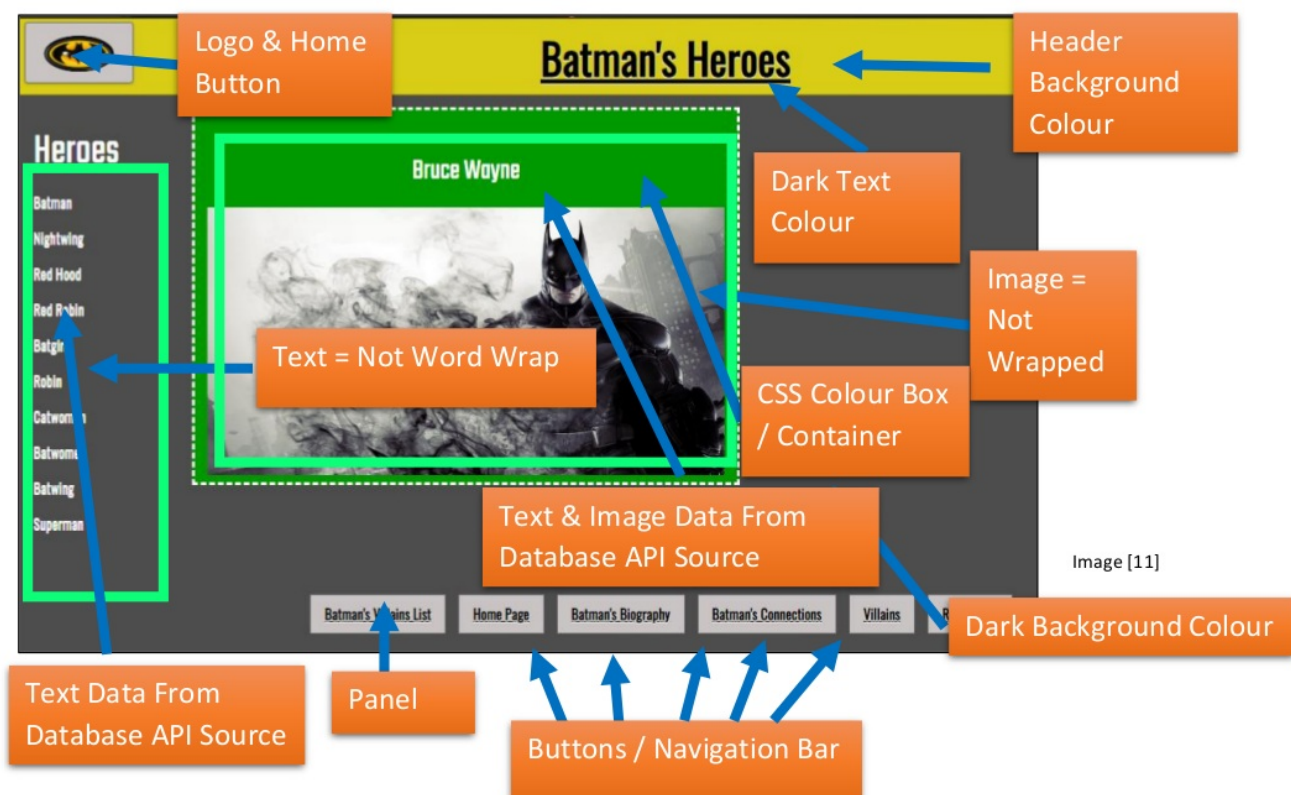
2.8. Heroes Page

Private Database JSON API – No. 1

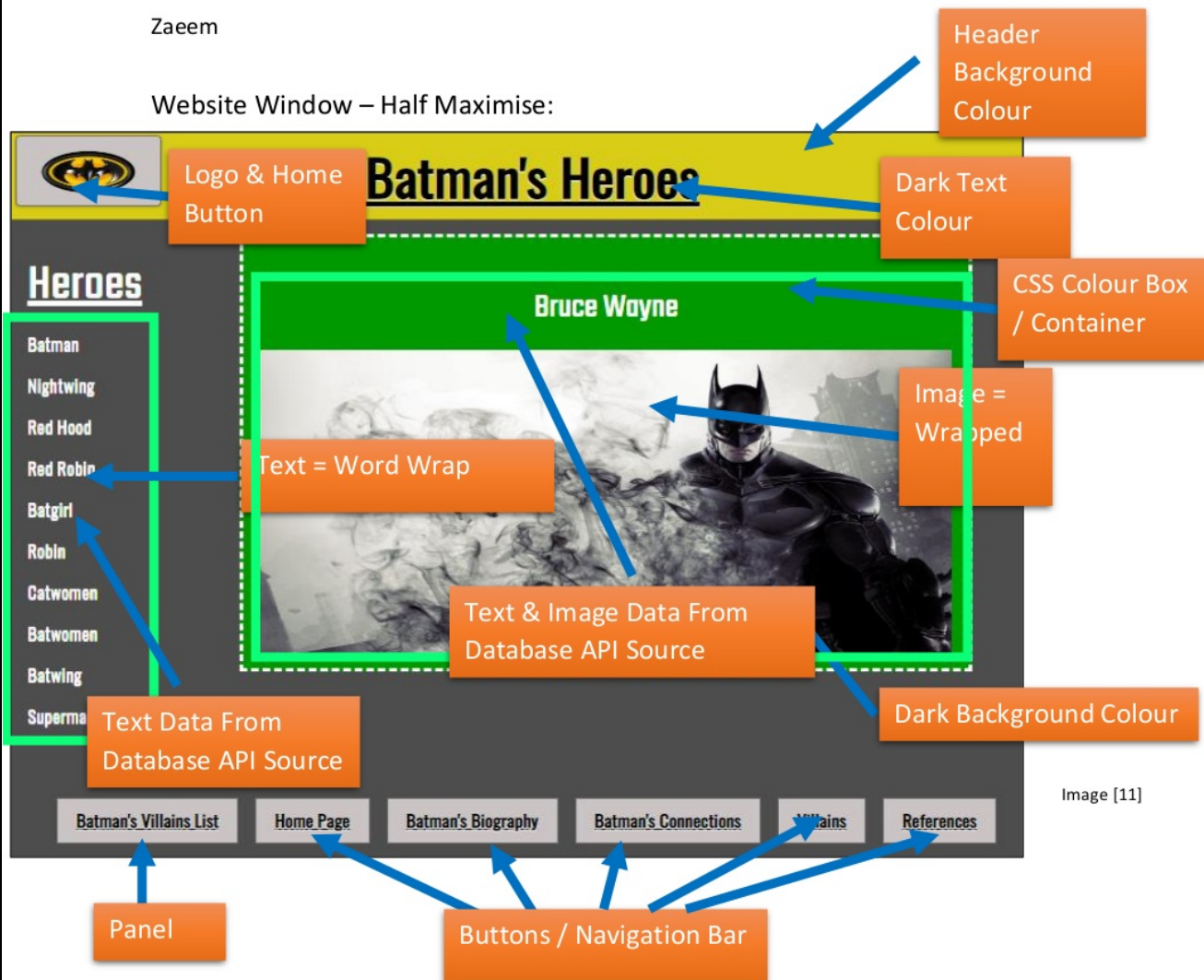
The fourth web page on the website is called Heroes Page. This webpage will allow users to view different superheroes data and images. The feature that was used on this page was a Private JSON API which used that was created on phpMyAdmin, this is JSON and AJAX feature created on a PHP file. This feature will allow users to view data from a private database.

In the image below, it shows the website interface when the user goes on the Heroes page and clicks on one of the records.

Website Window – Full Maximise:



Website Window – Half Maximise:



This feature will allow users to view the Heroes data receiving from private database source as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

This feature has prevented typing out all of this data on the HTML file so if the data from the Database API gets updated, the PHP file may not need be updated / edited as the data from the Database API source is linked within the PHP file. This feature and data will be available to view on this web page only.

Also, the web page looks fun and creative with the list of heroes, different images of the heroes and the colour box / container (made with CSS code) showing behind the name and the image.

Below will be the annotated code listing for this feature

PHP: Show Info

```
<!-- this will load up data from the coninfo file which is connected to the
database - start -->

<? include "coninfo.php";
$query="SELECT * FROM heroes ORDER BY id";
$result=mysqli_query($link,$query);

?>

<!-- this will load up data from the coninfo file which is connected to the
database - end -->
```

This feature uses AJAX and PHP language so the user has to receive the specific data from the private database on the PHP code.

```
<!-- this is the ajax call on page - start -->

<script>

function theinfo(id)
{
    fib=document.getElementById("fullinfobox")
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange=function(){
        if(this.readyState==4 && this.status==200) //4 request+response successful
        200 = OK from server
        {
            fib.innerHTML=xmlhttp.responseText
        }
    }
    xmlhttp.open("GET","getinfo_heroes.php?sendinfo="+id, true); //this is the file
    to get the data
    xmlhttp.send();
}
```

Zaeem

```
</script>
```

```
<!-- this is the ajax call on page - end -->
```

This feature uses AJAX and PHP language so the user has to configure the settings for the AJAX element on the PHP code.

```
<!-- this will show and configure the database data on page - start -->
```

```
<?while($r=mysqli_fetch_array($result))
```

```
{?>
```

```
<span onClick="javascript:theinfo(<?echo $r["id"]?>)">
```

```
<?
```

```
echo "<font size='3' face='Oswald'>";
```

```
echo $r["name"]."<br><br>";
```

```
echo "</font>";
```

```
?>
```

```
</span>
```

```
<!-- this will show and configure the database data on page - end -->
```

```
<!-- this will show database data on page -->
```

```
<span id="fullinfobox">
```

```
<font face="Teko">
```

```
<font size="5">
```

```
...
```

```
</font>
```

```
</font>
```

```
</span>
```

This feature uses AJAX, PHP and JavaScript language so the user has to configure the settings and add the content on the PHP code.

```
<!-- this will load up the coloured box behind the database data on page -->  
<style type="text/css">  
#fullinfobox  
{  
position:absolute;  
top:100px;  
left:200px;  
background-color:#009900;  
width:615px;  
border:dashed;  
padding:7px  
}  
</style>
```

This feature uses CSS language so the user has to configure the settings and add the coloured box / container on the PHP code.

PHP: Get Info

```
<?include "coninfo.php"; // load up data that is connected to database
$id=$_GET["sendinfo");//hard-code temporarily
$query="SELECT * FROM heroes WHERE id=$id"; // this select data from the
table and the field
$result=mysqli_query($link,$query);
// this will configure the data that is going on the page - start
echo "<font size='6' face='Teko'>";
$result=mysqli_fetch_array($result);
echo "<center>";
echo "<br>".$result["realname"]."<br>"; // selects the output data on page
echo "</font>";
echo "</center>";
echo "<br>"
// this will configure the data that is going on the page - end
?>
<!-- this will add images to each data shown on page -->
<center>
" width="600" height="300">
</center>
```

This feature uses AJAX and PHP language so the user has to receive and add the specific data and image from the private database on the PHP code.

With this all of this code listing above, this feature has successfully been created and worked on the created website.

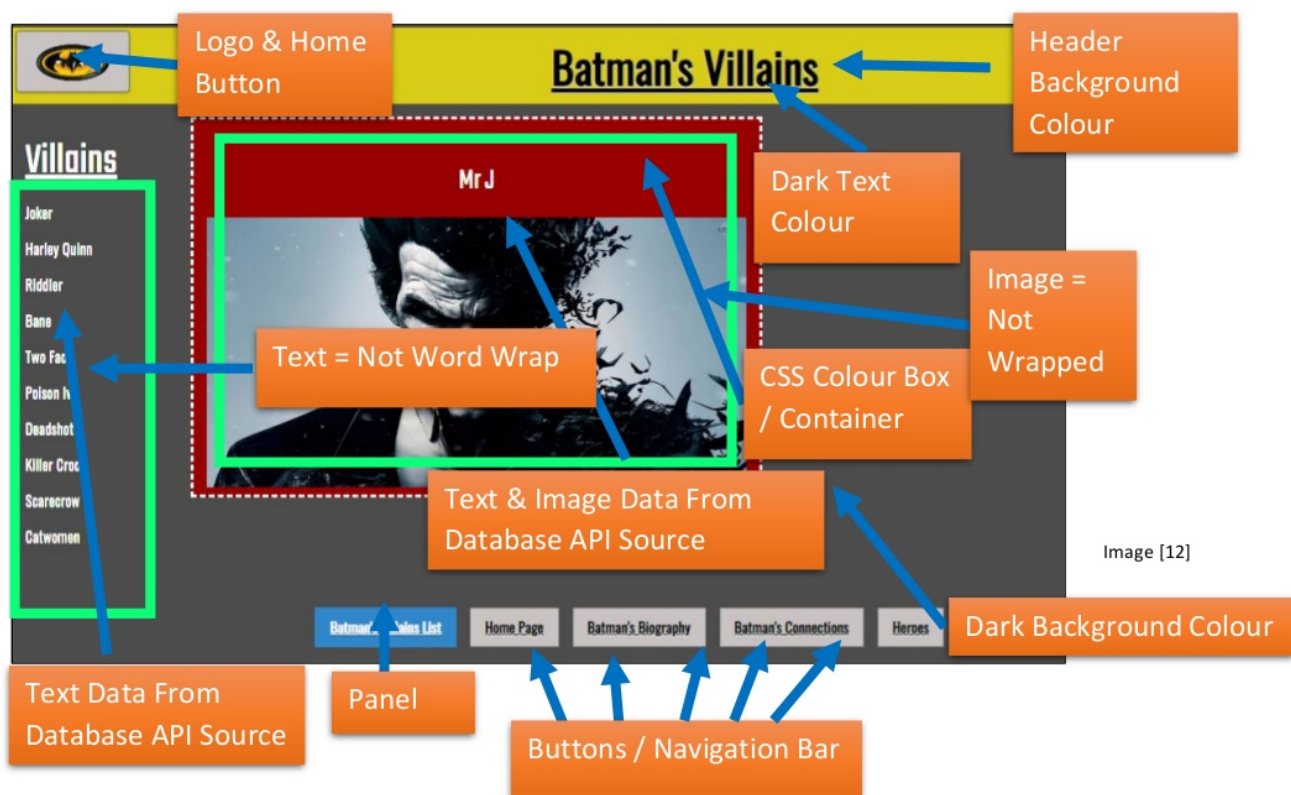
2.9. Villains Page

Private Database JSON API – No. 2

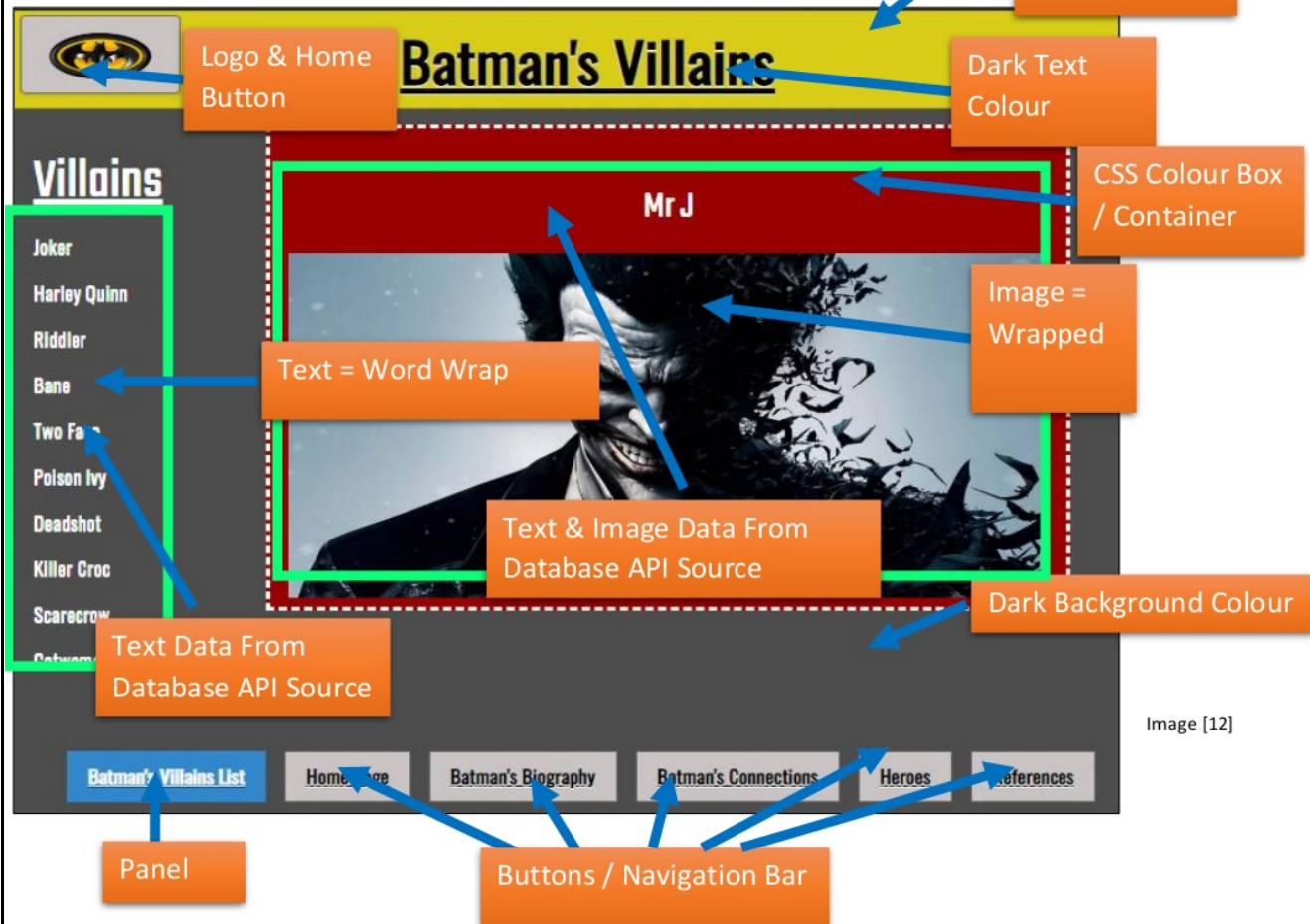
The fifth web page on the website is called Villains Page. This webpage will allow users to view different villain's data and images. The feature that was used on this page was a Private JSON API which used that was created on phpMyAdmin, this is JSON and AJAX feature created on a PHP file. This feature will allow users to view data from a private database.

In the image below, it shows the website interface when the user goes on the Villains page and clicks on one of the records.

Website Window – Full Maximise:



Website Window – Half Maximise:



This feature will allow users to view the Villains data receiving from private database source as this is modern feature used within modern technology e.g. Smartphones, Tablets, Web Browsers etc.

This feature has prevented typing out all of this data on the HTML file so if the data from the Database API gets updated, the PHP file may not need be updated / edited as the data from the Database API source is linked within the PHP file. This feature and data will be available to view on this web page only.

Also, the web page looks fun and creative with the list of villains, different images of the villains and the colour box / container (made with CSS code) showing behind the name and the image.

Below will be the annotated code listing for this feature

PHP: Show Info

```
<!-- this will load up data from the coninfo file which is connected to the
database - start -->

<? include "coninfo.php";

$query="SELECT * FROM villains ORDER BY id";

$result=mysqli_query($link,$query);

?>

<!-- this will load up data from the coninfo file which is connected to the
database - end -->
```

This feature uses AJAX and PHP language so the user has to receive the specific data from the private database on the PHP code.

```
<!-- this is the ajax call on page - start -->

<script>

function theinfo(id)
{
    fib=document.getElementById("fullinfobox")
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange=function(){
        if(this.readyState==4 && this.status==200) //4 request+response successful
        200 = OK from server
        {
            fib.innerHTML=xmlhttp.responseText
        }
    }
    xmlhttp.open("GET","getinfo_villains.php?sendinfo="+id, true); //this is the
    file to get the data
    xmlhttp.send();
}
```

Zaeem

```
</script>
```

```
<!-- this is the ajax call on page - end -->
```

This feature uses AJAX and PHP language so the user has to configure the settings for the AJAX element on the PHP code.

```
<!-- this will show and configure the database data on page - start -->
```

```
<?while($r=mysqli_fetch_array($result))
```

```
{?>
```

```
<span onClick="javascript:theinfo(<?echo $r["id"]?>)">
```

```
<?
```

```
echo "<font size='3' face='Oswald'>";
```

```
echo $r["name"]."<br><br>";
```

```
echo "</font>";
```

```
?>
```

```
</span>
```

```
<!-- this will show and configure the database data on page - end -->
```

```
<!-- this will show database data on page -->
```

```
<span id="fullinfo" >
```

```
<font face="Teko">
```

```
<font size="5">
```

```
...
```

```
</font>
```

```
</font>
```

```
</span>
```

This feature uses AJAX, PHP and JavaScript language so the user has to configure the settings and add the content on the PHP code.

```
<!-- this will load up the coloured box behind the database data on page -->  
<style type="text/css">  
#fullinfobox  
{  
position:absolute;  
top:100px;  
left:200px;  
background-color:# 990000;  
width:615px;  
border:dashed;  
padding:7px  
}  
</style>
```

This feature uses CSS language so the user has to configure the settings and add the coloured box / container on the PHP code.

PHP: Get Info

```
<?include "coninfo.php"; // load up data that is connected to database
$id=$_GET["sendinfo");//hard-code temporarily
$query="SELECT * FROM villains WHERE id=$id"; // this select data from the
table and the field
$result=mysqli_query($link,$query);
// this will configure the data that is going on the page - start
echo "<font size='6' face='Teko'>";
$r=mysqli_fetch_array($result);
echo "<center>";
echo "<br>".$r["realname"]."<br>"; // selects the output data on page
echo "</font>";
echo "</center>";
echo "<br>"
// this will configure the data that is going on the page - end
?>
<!-- this will add images to each data shown on page -->
<center>
" width="600" height="300">
</center>
```

This feature uses AJAX and PHP language so the user has to receive and add the specific data and image from the private database on the PHP code.

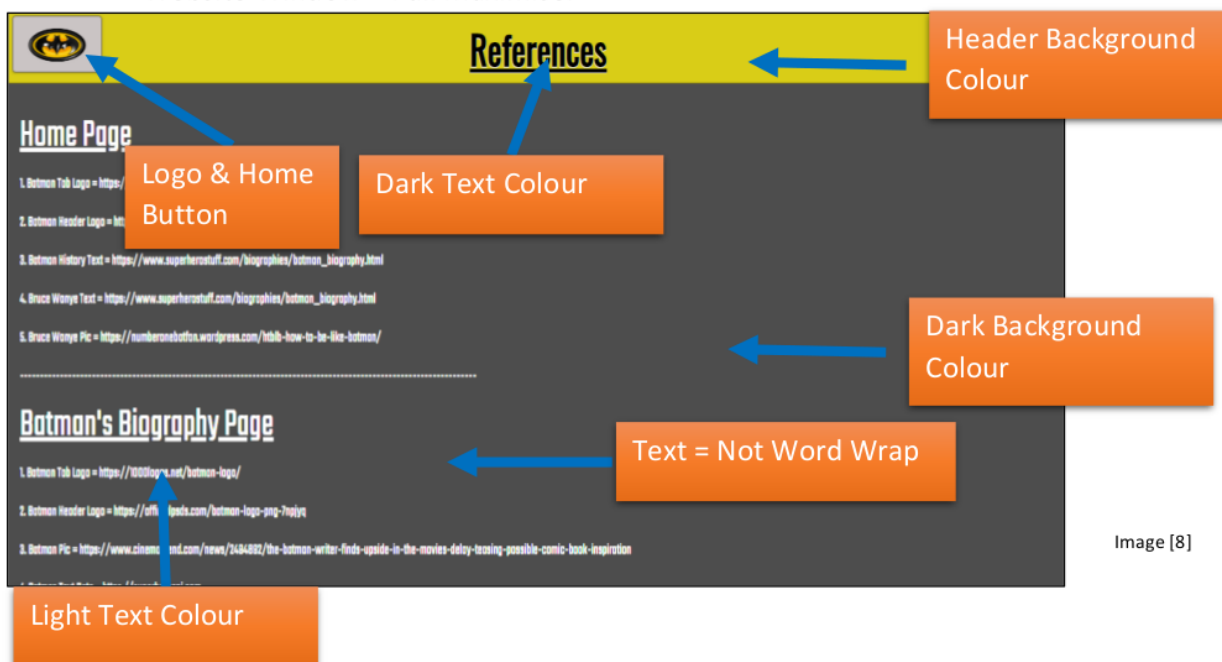
With this all of this code listing above, this feature has successfully been created and worked on the created website.

2.10. References Page

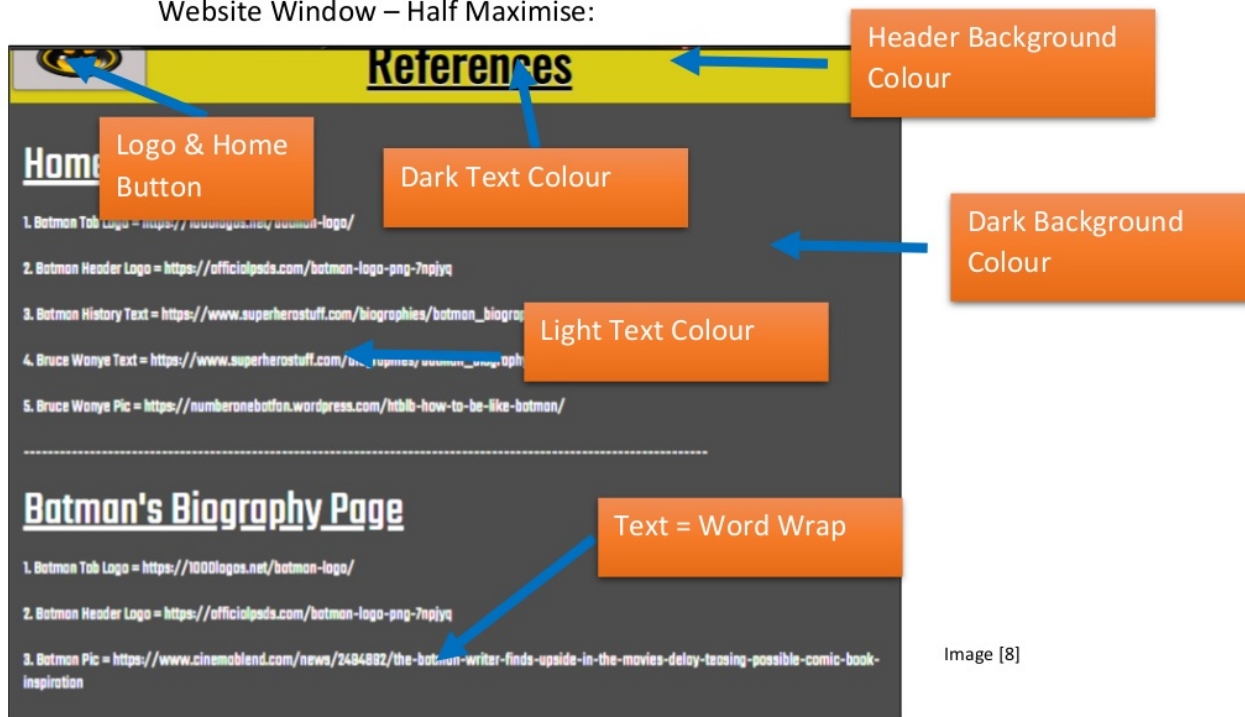
The last web page on the website is called References Page. This webpage will allow users to view all of the URL links that the data and images have been used throughout the whole Batman website. The feature that was used for this page was the Single Page Application (which has been mentioned earlier). No new feature will be used and shown on this webpage.

In the image below, it shows the website interface when the user goes on the References page.

Website Window – Full Maximise:



Website Window – Half Maximise:



As there no new feature on this web page, an annotated code listing will not be shown in this documentation.

3. Conclusion

In conclusion, this assignment has showed the creation of basic and interactive websites by using the languages of HTML, CSS, JavaScript, jQuery, AJAX, JSON and PHP as required in the assignment brief. This has also showed that that the ability to create multiple AJAX, JSON and PHP features has been gained. which has improved skills of creativity and adaptability towards a website development environment. The most important element of this website would be the data receiving features as this is the major element that remains relevant from the beginning of the design to the final product.

4. References

References

- [1] S. Morris, "AJAX—WHAT IT IS, HOW IT WORKS, AND WHAT IT'S USED FOR," N/A N/A N/A. [Online]. Available: <https://skillcrush.com/blog/what-is-ajax/>. [Accessed 6 December 2020].
- [2] A. Cepalia, "What Is JSON?," N/A N/A N/A. [Online]. Available: <https://realpython.com/lessons/what-is-json/>. [Accessed 6 December 2020].
- [3] KnowledgeHut, "Web Development Using PHP And MySQL," 12 January 2017. [Online]. Available: <https://www.knowledgehut.com/blog/programming/web-development-using-php-mysql>. [Accessed 6 December 2020].
- [4] A. Sherif, "Creating A More Efficient Business Venture Using Web And Mobile Technologies," N/A N/A N/A. [Online]. Available: <https://reboundmedia.com.ng/blog/creating-a-more-efficient-business-venture-using-web-and-mobile-technologies>. [Accessed 6 December 2020].
- [5] KnowledgeHut, "What Is the Purpose of AJAX in JavaScript.," 10 November 2020. [Online]. [Accessed 6 December 2020].
- [6] KnowledgeHut, "Working with Json in Javascript," 3 November 2020. [Online]. Available: <https://www.knowledgehut.com/blog/web-development/json-in-javascript>. [Accessed 6 December 2020].
- [7] jQuery, "ThemeRoller for jQuery Mobile," N/A N/A N/A. [Online]. Available: <https://themeroller.jquerymobile.com/>. [Accessed 6 December 2020].
- [8] Zaeem, "index.html," Swansea, 2020.
- [9] Zaeem, "bio.html," Swansea, 2020.
- [10] Zaeem, "connections.html," Swansea, 2020.
- [11] Zaeem, "showinfo_heroes.php," Swansea, 2020.
- [12] Zaeem, "showinfo_villains.php," Swansea, 2020.

5. Bibliography

Bibliography

- [1] S. Morris, "AJAX—WHAT IT IS, HOW IT WORKS, AND WHAT IT'S USED FOR," N/A N/A N/A. [Online]. Available: <https://skillcrush.com/blog/what-is-ajax/>. [Accessed 6 December 2020].
- [2] A. Cepalia, "What Is JSON?," N/A N/A N/A. [Online]. Available: <https://realpython.com/lessons/what-is-json/>. [Accessed 6 December 2020].
- [3] KnowledgeHut, "Web Development Using PHP And MySQL," 12 January 2017. [Online]. Available: <https://www.knowledgehut.com/blog/programming/web-development-using-php-mysql>. [Accessed 6 December 2020].
- [4] A. Sherif, "Creating A More Efficient Business Venture Using Web And Mobile Technologies," N/A N/A N/A. [Online]. Available: <https://reboundmedia.com.ng/blog/creating-a-more-efficient-business-venture-using-web-and-mobile-technologies>. [Accessed 6 December 2020].
- [5] KnowledgeHut, "What Is the Purpose of AJAX in JavaScript.," 10 November 2020. [Online]. [Accessed 6 December 2020].
- [6] KnowledgeHut, "Working with Json in Javascript," 3 November 2020. [Online]. Available: <https://www.knowledgehut.com/blog/web-development/json-in-javascript>. [Accessed 6 December 2020].
- [7] jQuery, "ThemeRoller for jQuery Mobile," N/A N/A N/A. [Online]. Available: <https://themeroller.jquerymobile.com/>. [Accessed 6 December 2020].
- [8] Zaeem, "index.html," Swansea, 2020.
- [9] Zaeem, "bio.html," Swansea, 2020.
- [10] Zaeem, "connections.html," Swansea, 2020.
- [11] Zaeem, "showinfo_heroes.php," Swansea, 2020.
- [12] Zaeem, "showinfo_villains.php," Swansea, 2020.

(FINAL) Zaeem Assignment 2 - Gaynor

GRADEMARK REPORT

FINAL GRADE

74/100

GENERAL COMMENTS

Instructor

WEBSITE

Range of code demonstrated - I'm a bit confused why you have included all the code for all the pages in every page of code (bio.html, connections.html etc) when they are linked together anyway. This is not necessary - would have been better to use a single page application throughout rather than having linked pages instead. Some good use of JQuery mobile features. API work is good - especially the connections one which is more complex (drill down into the Batman's relatives section) The AJAX call to your own data is good and it is good that you have used a parameter to vary the data retrieved. You could have implemented features such a filtering a JSON file or using your own JSON file produced from your database for a higher mark. 44/60

Overall integration of website - Good overall - colours are very suitable to the subject of the website. It's a bit random having a side panel for villains but I appreciate that you were using it as a vehicle to demonstrate jQuery mobile features. More could have been done with the single page application but overall very good. Good to see that you have used ThemeRoller too. 8/10

REPORT

Analysis of technical development - You've done this well with some commentary on why you took the approach you did particularly in the JQuery Mobile sections. You could have done a bit more of a technical analysis on the API work. 11 /15

Annotated code listing - The JSON parts are a bit under-commented but OK on the whole 11/15

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46
