

Assignment 2

by Tomos Joseph Leggett

FILE	17844_TOMOS_JOSEPH_LEGGETT_ASSIGNMENT_2_308223_446715972. PDF (274.62K)		
TIME SUBMITTED	21-JAN-2021 01:54PM (UTC+0000)	WORD COUNT	1267
SUBMISSION ID	142941891	CHARACTER COUNT	6817

Web and Mobile Technologies

Assignment 2

Author: Tomos Leggett · **Student Code:** 1901852 · **Module Code:**ACCA5031

Web and Mobile Technologies Assignment 2

- Introduction

- Home Page

 - Screenshots & Methodology

 - The Hamburger Menu when Open

 - The Random Fact Expandable when Expanded

 - Weather details

 - “Last Updated” tag

 - Annotated Code Listings

 - Hamburger Menu

 - Inserting the data onto the weather cards

 - Inserting the data onto the fact card

- Fact Page

 - Screenshots and Methodology

 - Header

 - Fact Card

 - Refresh Button

 - Annotated Code Listings

- About Page

 - Screenshots and Methodology

 - Header

 - Page Content

 - Annotated Code Listings

- Conclusion

Introduction

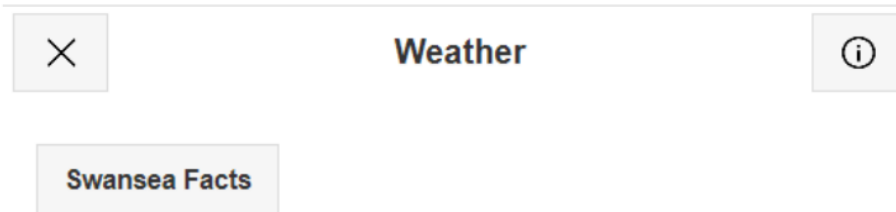
In this documentation, the author demonstrates a mobile website that makes a call to a weather API and displays the information neatly on the web page. The website itself uses JQuery Mobile components and features, such as single page transitions, mobile navigation, touch components such as dropdowns and more. The website itself also uses a personal API developed by the author through MySQL, exported as a public local JSON file embedded in the websites' directory. This is a simple JSON API that relays several facts about Swansea the author found on the internet. The

website itself is minimalistic, it uses a cover image from unsplash with the keyword “Swansea” that changes every time you refresh.

Home Page

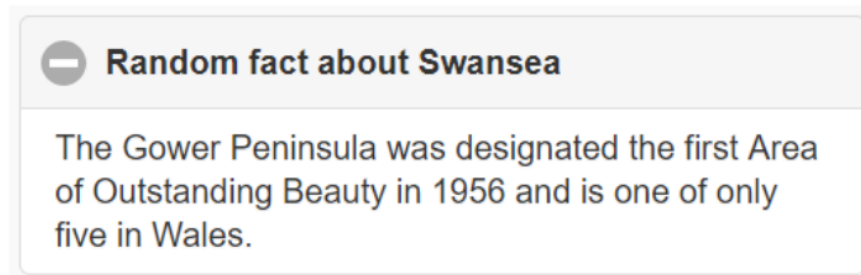
Screenshots & Methodology

The Hamburger Menu when Open



When the user opens the hamburger menu, the menu itself expands. If the user were to press the close button, the hamburger menu would collapse. Inside the hamburger menu is a single button that redirects you to the Swansea Facts page. Originally, the about page and a search page were supposed to be here, but the button for the about page was moved to the top right and the search function was deprecated.

The Random Fact Expandable when Expanded



A random fact is displayed each time you visit the page from an AJAX call to a private JSON file created by the author. It can be collapsed if you're more interested in the weather.

Weather details



Partly cloudy
7°C

Is it daytime here?

yes

Wind Speed:

28

Wind Direction:

↺

Local Time/Date:

2021-01-21 12:12

Weather details are split into cards, each card representing a different attribute sourced from the public weather API the author used. The icon on the top left belongs and is provided by the weather API, originally the author was to produce their own JSON file collection of weather icons, but due to complications (there are 217 weather codes that all have separate icons), the author chose to use the icons provided by the weather API instead.

“Last Updated” tag

Last updated: 12:12 PM

A tag at the bottom of the page informing the user when the API was last updated for convenience can also be seen.

Annotated Code Listings

Hamburger Menu

```

1 function openhamburgermenu() {
2     // an If statement to check if the hamburger menu is already open.
3     if ($('#menubar').css('display') == 'none') { // If the hamburger
        menu is hidden:
4         $('#menubar').css("display", "block") // Show it,
5         $('#menuicon').attr("src", "Icons/Delete.png") // Change the
        icon into an X to indicate the ability to close the menu,
6     } else { // If the hamburger menu is already open, hide it and
        show a menu icon instead of a close icon.
7         $('#menubar').css("display", "none") // Hide the hamburger
        menu,
8         $('#menuicon').attr("src", "Icons/Menu.svg") // Change the
        icon back into a hamburger.
9     }
10 }

```

Inserting the data onto the weather cards

```

1 // API Call for Weather Cards
2 var xmlhttp = new XMLHttpRequest();
3 xmlhttp.onreadystatechange = function () {
4     if (this.readyState == 4 && this.status == 200) { // If the
        browser has connected to the API succesfully:
5         var myObj = JSON.parse(this.responseText);
6
7         // Making every query a variable so it's easier on the
        eyes:
8         var tempquery = myObj["current"]["temperature"]
9         var descquery = myObj["current"]["weather_descriptions"]
10        [0]
11        var icoquery = myObj["current"]["weather_icons"][0]
12        var IsItDaytime = myObj["current"]["is_day"]
13        var windspeed = myObj["current"]["wind_speed"]
14        var winddirection = myObj["current"]["wind_degree"]
15        var lastupdated = myObj["current"]["observation_time"]
16        var localtimedate = myObj["location"]["localtime"]
17        var placetype = myObj["request"]["type"]
18        var country = myObj["location"]["country"]
19
20        // Injecting those variables into my HTML elements and
        breathing life into them:
21        $('#CurrentTemperature').html(tempquery) // Set the
        current temperature to the appropriate label
22        $('#CurrentWeather').html(descquery) // Set the current
        weather conditions to the appropriate label
23        $('#WeatherIcon').attr("src", icoquery) // Set the current
        weather icon to the appropriate image
24
25        $('#favico').attr("href", icoquery) // Change the favicon

```

```

24         to represent the weather condition cause it looks nice, okay?
25         $('#WindSpeed').html(windspeed) // Change the windspeed
26         accordingly.
27         $('#WindDegree').css({
28             "transform": "rotate(" + winddirection + "deg" + ")"
29         // Set the rotation of the arrow based on the weather direction angle.
30         });
31         $('#IsItDaytime').html(IsItDaytime) // Set the "day time"
32         card.
33         $('#lastupdated').html(lastupdated) // Sets the last time
34         mark accordingly.
35         $('#localtimedate').html(localtimedate) // Set the local
36         time card.
37     }
38 };

```

Inserting the data onto the fact card

```

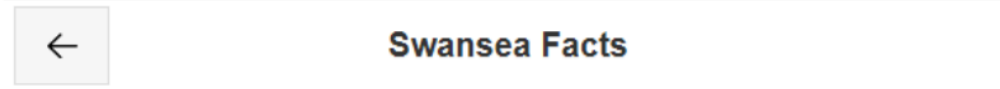
1 // API call for facts
2 var xmlhttp2 = new XMLHttpRequest();
3 xmlhttp2.onreadystatechange = function () {
4     if (this.readyState == 4 && this.status == 200) {
5
6         var myObj2 = JSON.parse(this.responseText);
7         var random = Math.floor(Math.random() * 8); // Pick a
8         random number between 1 and 7
9         var fact = myObj2[2]["data"][random]["Fact"] // Inject
10        that number into the query (there are 7 facts)
11        var factlabel = random + 1; // Since we start at 0, set
12        the fact label to random but add 1.
13
14        $("#randomfact").html(fact); // Set the fact card as an
15        actual fact.
16        $(".fact-card-body").html(fact); // And the fact card on
17        the homepage too.
18        $(".fact-number").html(factlabel); // Insert the fact
19        number above the fact.
20    }
21 };
22 xmlhttp2.open("GET", "swanseafacts.json", true);
23 xmlhttp2.send();

```

Fact Page

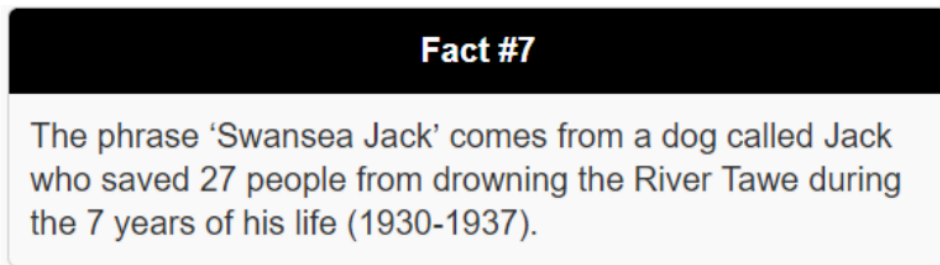
Screenshots and Methodology

Header



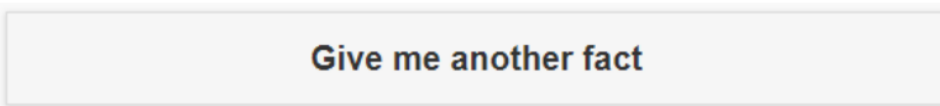
The header contains the page title and a back button. JQuery Mobile handles the functionality of the back button, which takes you back to the homepage.

Fact Card



The fact card contains a fact number and the body of the card. The code simply picks a random fact and injects it into the body, and the fact number is the fact ID.

Refresh Button



A button labelled “Give me another fact” is underneath the card, when pressed, it refreshes the page and displays a new card.

Annotated Code Listings

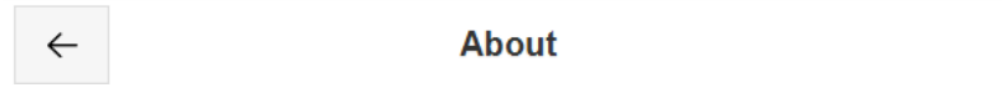
```
1 // API call for facts
2 var xmlhttp2 = new XMLHttpRequest();
3 xmlhttp2.onreadystatechange = function () {
4     if (this.readyState == 4 && this.status == 200) {
5
6         var myObj2 = JSON.parse(this.responseText);
7         var random = Math.floor(Math.random() * 8); // Pick a
random number between 1 and 7
8         var fact = myObj2[2]["data"][random]["Fact"] // Inject
that number into the query (there are 7 facts)
9         var factlabel = random + 1; // Since we start at 0, set
the fact label to random but add 1.
10
11         $("#randomfact").html(fact); // Set the fact card as an
```

```
12         actual fact.
13         $(".fact-card-body").html(fact); // And the fact card on
the homepage too.
14         $(".fact-number").html(factlabel); // Insert the fact
number above the fact.
15     }
16 };
17 xmlhttp2.open("GET", "swanseafacts.json", true);
18 xmlhttp2.send();
```

About Page

Screenshots and Methodology

Header



The header contains the page title and a back button. JQuery Mobile handles the functionality of the back button, which takes you back to the homepage.

Page Content



Weather

A simple Weather Web App developed to demonstrate AJAX/JSON calls using JQuery Mobile by Tomos Leggett

Icons

All icons by icons8.com

Images

Images by Unsplash

Weather API

API by weatherstack.com

Weather Icons

Icons included in API by weatherstack.com

Swansea Facts API

Swansea Facts by yours truly

The about page content showcases information about the page itself and credits the API and more.

Annotated Code Listings

This page doesn't have any code

Conclusion

In conclusion, the author has developed their skills and understandings significantly over the course of this assignment's productions, especially in the field of JQuery Mobile, JSON files and AJAX calls. Although disappointed in the lack of a search function, the author is satisfied with the overall layout and functionality of the website.

Assignment 2

GRADEMARK REPORT

FINAL GRADE

76/100

GENERAL COMMENTS

Instructor **WEBSITE**

Range of code demonstrated - The Swansea facts API is good, and I like the inclusion of the random number to choose different facts from the API. The weather is also good and I like how you have integrated this into your site using JQuery - it looks quite seamless. Some good use of JQuery mobile features, although the website a bit sparse - maybe you could have included some more information about Swansea into your accordion to flesh is out a bit for example. 46/60

Overall integration of website - The styling of the website is a bit minimalist - the shades of grey are very subtle and difficult to distinguish from each other. Website looks very stretched in desktop format, but better in mobile mode 6.5/10

REPORT

Analysis of technical development - Quite well explained, although more about what you did rather than how efficient you think it was. Try to do more of a technical analysis. Avoid the temptation to refer to yourself as "the author" and keep in the passive tense. eg "The Author chose to user the icons provided by xxx" -> "Icons provided by xxx were used" 10/15

Annotated code listing - Easy to follow and understand. 13/15

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8
