CA377 Programming Fundamentals (Project)

Name	Aaron Gillespie
Student Number	14314471
Programme	EC3
Module Code	CA377
Assignment Title	Programming Fundamentals (Project)
Submission date	12/12/16
Module coordinator	Suzanne Little & Marija Bezbradica

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name: Aaron Gillespie Date: 12/12/16

Introduction

Welcome to my report outlining my project. During this project I built an online web service which is called Park at DCU and can be found at the following URL: http://parkatdcu.azurewebsites.net/ . The objective of this website was to provide a user with information about the different parking faculties within Dublin City University. There are different sections to the website which are targeted to different users. The sections are laid out as follows.

Web Service

This section of the website allows you to access real time information about the capacity of each of the car parks across DCU's three campuses. Simply input a carpark name into the text box and select which type of space you want and you will be given live information.

Closest Car Park

This section of the website allows you to input the building you would like to park closest to. Once you click enter you will be given the closest carpark to that building.

Campus Space Information

This section of the website allows you to get information on the carpark spaces across DCU's three campuses. Simply input a campus and you will be given the total number of regular spaces and disabled spaces for that campus.

Historical Information

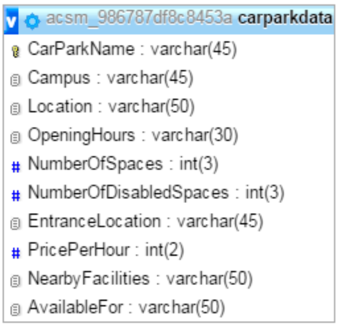
This section allows you to input a Carpark and Week (1-52) and you will be returned a list of the occupancy times for that week at each hour of the day for the previous year.

The website was designed using HTML and CSS with the back end script being written in PHP. The website is linked to a database which is hosted on azure along with the website itself. In this report I am going to describe how I completed this project going from two excel spreadsheets with data to a fully functional website in 10 weeks. The report is going to be broken down into the sections bellow.

- System Architecture: Breaking down the back end and database structure.
- UI description: Breaking down why the website looks the way it does.
- Using Source Code Repository: How I used gitlab.
- Additional Functionality: What I added to the website outside of the specification.
- Conclusion: What I learned by completing this project.

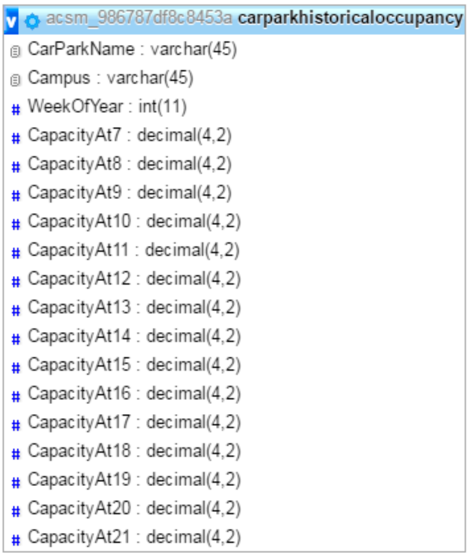
System Architecture

The project originally started out with two csv files. One csv file gave us information about each car park e.g. opening times and what facilities they were located beside. The second file gave us historical information about each car park for the previous year including average occupancy rates for each hour of the day on each week of the year. There are six carparks in total located across the three DCU campuses. Four are located on the Glasnevin campus, one on the Drumcondra campus and one on the innovation campus DCU Alpha. My first step was the build a database to host these csv files. I decided to use azure to host both my database and website because of my experience in using it as part of other modules. To get the website on azure I created a student account and deployed a web application with MySQL. To manage my database I downloaded the server extension phpMyAdmin. I then imported the csv files to this database using the import function on phpMyAdmin. For my SQL statements to work I had to convert the percentage figures on the historical file to decimal figures. They would be still displayed as percentages on the website. Below is the schema of my database as you can see I kept things quite simple while making sure it would be fully functional.



acsm_986787df8c8453a carparkdata

CarParkName : varchar(45)
Campus : varchar(45)
Location : varchar(50)
OpeningHours : varchar(30)
NumberOfSpaces : int(3)
NumberOfDisabledSpaces : int(3)
EntranceLocation : varchar(45)
PricePerHour : int(2)
NearbyFacilities : varchar(50)
AvailableFor : varchar(50)



acsm_986787df8c8453a carparkhistoricaloccupancy

CarParkName : varchar(45)
Campus : varchar(45)
WeekOfYear : int(11)
CapacityAt7 : decimal(4,2)
CapacityAt8 : decimal(4,2)
CapacityAt9 : decimal(4,2)
CapacityAt10 : decimal(4,2)
CapacityAt11 : decimal(4,2)
CapacityAt12 : decimal(4,2)
CapacityAt13 : decimal(4,2)
CapacityAt14 : decimal(4,2)
CapacityAt15 : decimal(4,2)
CapacityAt16 : decimal(4,2)
CapacityAt17 : decimal(4,2)
CapacityAt18 : decimal(4,2)
CapacityAt19 : decimal(4,2)
CapacityAt20 : decimal(4,2)
CapacityAt21 : decimal(4,2)

Once the database was created it was time to create some SQL quires that would perform the functions described in the introduction. The quires would allow me to implement the closest car park, campus space information and historical information functions. I originally used set quires without variables to test that the SQL worked. For example I created a query that would allow me to view the nearest car parks to the helix. This looked like the following.

“SELECT CarParkName, NearbyFacilities FROM CarParkData WHERE NearbyFacilities LIKE "%helix%"

I had versions of these statements for my two other quires as well. To get these working with a user input I created a PHP file with the SQL scripts inside that were linked to different forms on my website. I used a switch statement to differ between the quires and the forms. E.g. form one was linked to case 1in the switch statement.
When I put my SQL scripts into the PHP file I changed them to allow the use of variables. This allowed users to make their own selections. While the above SQL statement would only allow the user to see the results of carparks near the helix, the bellow SQL statement will allow them to see the carparks which are beside the facility they select from the following list. The list is contained within a html form.(Helix ,Business ,Main reception ,Crèche ,1838 , Invent , Gardens , Sports grounds , Met Eireann)

The new SQL

```
$SQL = "SELECT CarParkName FROM CarParkData WHERE NearbyFacilities LIKE
"."\"%$facility%\"";
```

The next part of the website to implement was the real time web service. This function brings real time data to the website. A user can enter the name of a car park into the text box provided and they will be provided with live information about the number of carparks available in the carpark at that moment. To implement this into my website I had to make use of the pre-existing web service located on http://suzannelittle.pythonanywhere.com. This web service works by allowing the user to put name of the carpark they want information on after the URL e.g. http://suzannelittle.pythonanywhere.com/carpark/invent . This displays the following information for the invent car park “Current time, space type Regular/disabled/all, Car park name and the spaces available”. To implement this into my website I used the PHP json function to parse the results into variables which I would then call on my html page. I got the users required carpark from their html form input which is then inputted to the following URL http://suzannelittle.pythonanywhere.com/carpark/$entry. The form allows the user to enter any car park they like into a text box and select whether they would like to find out information on disabled, regular or all spaces. Because of this I have also implemented some error checking. This means the user can enter any combination of capitalised letters when writing St Pats e.g. St Pats or St pats or St PAtS and they will still be given results for the St Pats car park. I have also checked for invalid car park entries along with blank entries. This error checking was completed with if statements within the PHP file.

UI Description

When designing the user interface I had to take into account what type of people would be using the website. After some analysis I concluded that the main demographics would be

- Current students.
- Current / Guest Lecturers.
- Conference Attendees.
- University Guests.
- Theatre/Concert Attendees (Helix).

Due to this wide user base the technological literacy rate could vary from basic to advanced. Because of this I decided it would be best for all users if I made the website simple and easy to use. When the users get to the homepage they are greeted with a blue and white colour scheme with black and orange text. I chose these colours because they blend well together and it makes the user immediately focus on what is inside the white area. Inside the text box on the homepage is an introduction to the website along with instructions on how to use each of the sections and what they do. This allows the user to immediately get an explanation on how to use the website so they don't have to figure things out by themselves through trial and error. On the top left corner of the webpage we have the Park at DCU logo which when clicked on brings you back to the homepage. This function was added through css and html and is just for ease of use as this is something users will already be familiar doing to get back to the homepage e.g. Facebook and YouTube.

On the top right corner of every page on the website is the navigation bar. This is how the user can switch between the different services. There are four query pages and the homepage on the navigation bar. I created the bar simply by styling links using CSS and creating a div called nav which was at the top of every page. I also added to the CSS so that when the user hovers over one of the selections on the navigation bar it is highlighted in orange. This is to make the process simpler for the user so they know what selection they are about to make before they click their mouse. Each of the functional sections follows this design principal with the information being stored in a white box in the middle with a blue background above and below it. On the real time information section there is a text box which allows the user to enter one of the carparks located across DCU's four campuses. If they do not know the name of the campus they or looking for or they enter in an invalid carpark an error is returned and returns a list of the carparks that are valid for entry. The text box is combined with a drop down menu which allows the user to select ordinary, disabled or all spaces. The layouts for the three other quires are all the same as each other. All there is on the page is a dropdown box and submit button on each page. Along with a line above the box explaining once again what action the form completes. I wanted all of the pages to look similar to keep things simple and so they had a unified design.

Using the source code repository

Throughout the course of the project I used the school of computing hosted repository manager gitlab. Gitlab is a web application that allows you to host, manage and collaborate on programming projects. I used it to host and manage my code. I also used it to get that fortnight's tasks from the project manager's repo which all of the class had access to. This repo included the source code we were to work from and it is also where I got my original csv files from. I forked this project to create my own version. Every two weeks we were given a new task to complete. While I was working on this task I used gitlab to commit any changes I made to my files at the end of the day. This meant there was a constant backup of the changes made to my code and I could go back and review my code at any time and also have access to it where ever I was be that at home or in the labs. This made keeping track of my changes easier and also allowed me to revert back to previous versions if needed. At the end of each fortnight I would make a final commit which would then be received by the project managers and graded.

The structure of my folders is as follows. First I have a client_notes folder this folder contains the original project briefing I received at the start of the module. The next folder is the data folder. This folder contains the original raw data that I received at the start of the module including both of the csv files. It also includes my early test SQL statements. The next folder is the doc's folder. This folder includes documentation on the project including this report and a user manual. The final folder is the src folder. This folder is broken down into two sub folders one for the main version of the website which is in the folder labelled main the other folder includes the files for the mobile website labelled mobile. The root directory includes a readme file describing the project.

In conclusion this was my first time using a source code repository manager while completing a project and I will defiantly make use of it in the future be that gitlab or an alternative. It made the management of my code so much easier since I had it all in the one place and I could review my changes at any time. I also hope make use of its collaborative functions on future group assignments and my final year project.

Extra Credit

To gain my extra credit for this project I decided to create a mobile version of the website. To achieve this I used the open source user interface system jQuery mobile [1]. JQuery mobile comes with premade JavaScript and CSS templates that allow you to make a simple mobile website. I combined my original code with their CSS and JavaScript to create a mobile interface. Once I configured my code in this new layout I ran into an issue. The results from the historical and campus information tabs were not scrollable. Therefore you could only view a section of the results. After some research online I came up with a fix for the issue. I included the following CSS in the heading of the webpage and a div for the table-container right before the PHP code.

```
<style>
        .table-container
        {
         overflow: hidden;
        overflow-x: scroll;
        -webkit-overflow-scrolling: touch;
        }
</style>
```

This fixed the issue but lead to another issue. The results were now scrollable but only after you refreshed the page. So the user has to load the results then refresh the page. After more research online I could not figure out a fix for this issue. Once the mobile version was created I wanted to make my desktop site automatically re-direct to the mobile version if the user was using a mobile device. To do this I included some JavaScript in the header of the desktop index page.

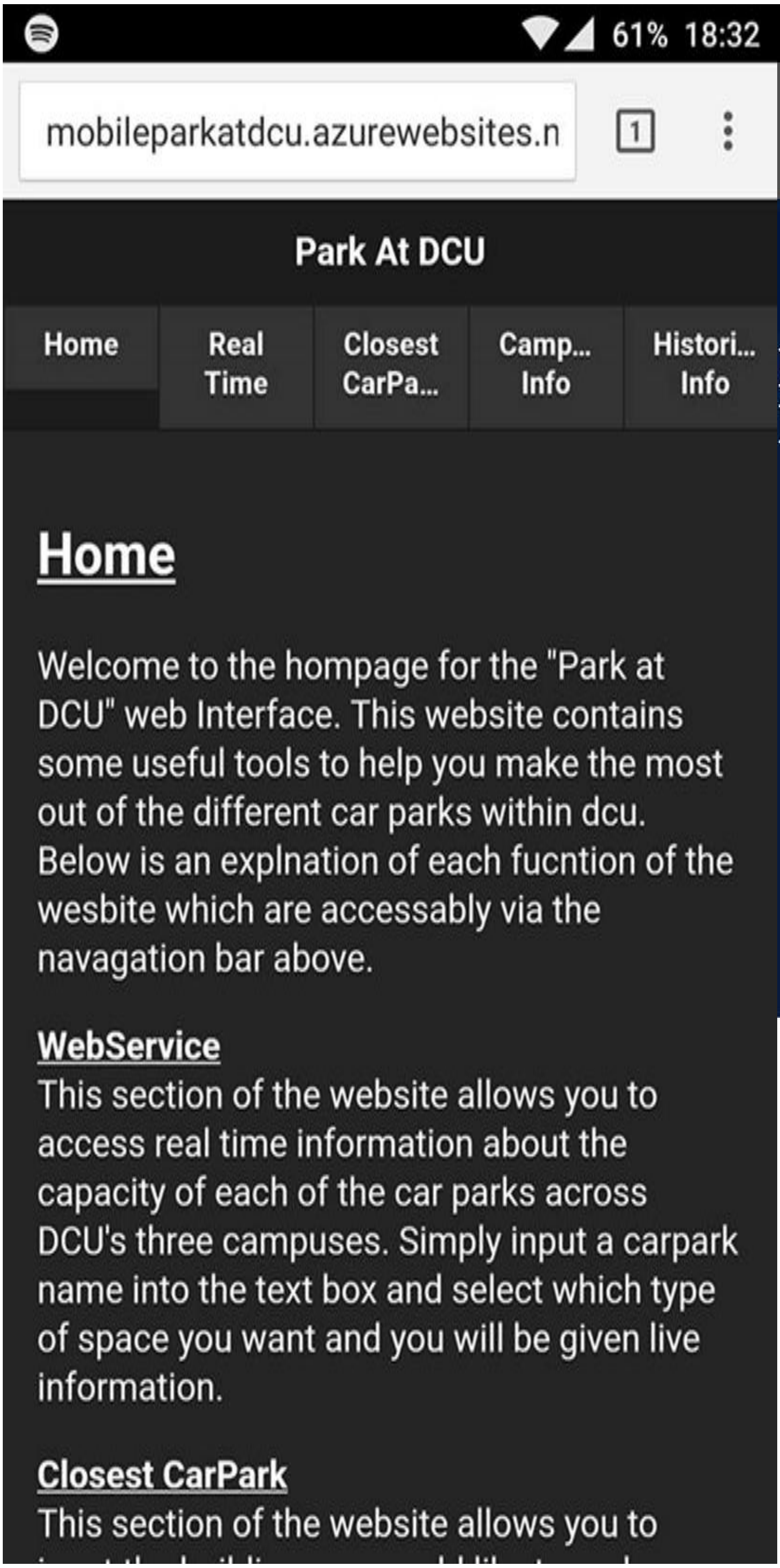
```
<script type="text/javascript">

  if (screen.width <= 800) {
    window.location = "http://mobileparkatdcu.azurewebsites.net";
  }
```

This JavaScript checks if the device screen width on the users device is less than 800 pixels wide if it is it redirects them to the mobile version of my website hosted at mobileparkatdcu.azurewebsites.net. If not they stay on the regular website. I feel like the mobile portion of my website has been implemented well but I would like to fix the scrolling issue.

The layout of the website has been kept simple with the content being the exact same as the desktop version. There is a simple navigation bar up top which allows the user to navigate the each section. The other sections are laid out the same as the desktop version with dropdown menus / text boxes along with a large submit button. Below is a comparison of how the website looks once optimised and how it looks before it was optimised for mobile.

<u>Comparison Of Mobile Optimised vs Non Optimised</u>

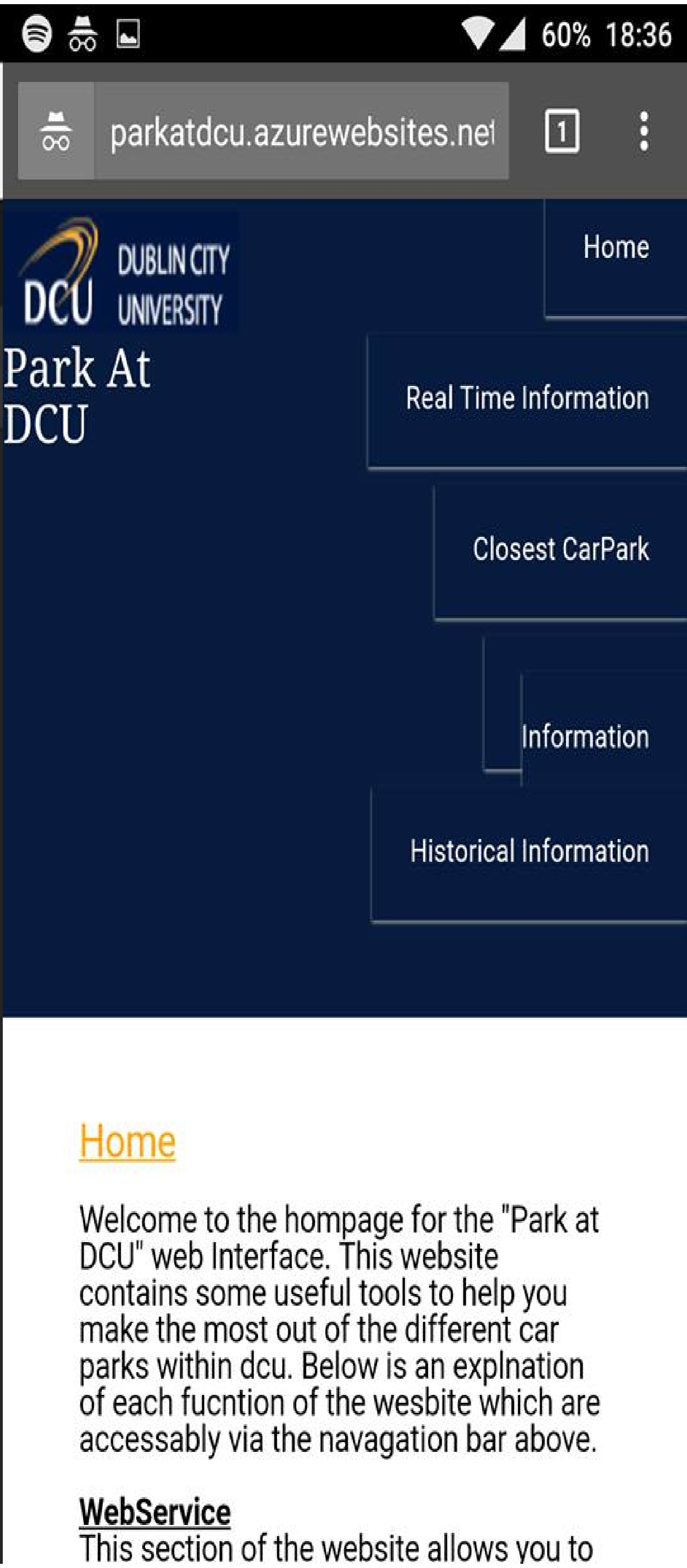


Home

Welcome to the hompage for the "Park at DCU" web Interface. This website contains some useful tools to help you make the most out of the different car parks within dcu. Below is an explnation of each fucntion of the wesbite which are accessably via the navagation bar above.

WebService
This section of the website allows you to access real time information about the capacity of each of the car parks across DCU's three campuses. Simply input a carpark name into the text box and select which type of space you want and you will be given live information.

Closest CarPark
This section of the website allows you to



Home

Welcome to the hompage for the "Park at DCU" web Interface. This website contains some useful tools to help you make the most out of the different car parks within dcu. Below is an explnation of each fucntion of the wesbite which are accessably via the navagation bar above.

WebService
This section of the website allows you to

Conclusion

In just 10 weeks I went from two csv files to a fully functional website which I developed both the front end and back end for myself. I feel by completing this project I have revised and built on the skills I learned in my first and second year of this degree. Completing this project has allowed me to gain skills in all the stages of web development. I developed the back end application side by using PHP and SQL which I had experience with in second year separately but didn’t have much experience in linking both of the services together. I also learned about web design using HTML and CSS which I was not great at as I had not done any since semester one of first year and I felt this helped me brush up on those skills and improve on them. I also learned how to deploy my work to my own server hosted on azure. Before this I only ever hosted my work on the student.computing.dcu.ie server but this time I learned how to do it myself. I feel like this is a skill which will be useful not only in future modules but also in my future career.

I also learned a small amount about mobile web development using jQuery mobile. I researched this by myself and combined it with my existing code to make a mobile website. If it were not for this module I wouldn’t have researched this. I am glad I did as I had no prior experience with mobile web development and now I feel more comfortable with it. I would have liked to get the scrollbar fully working and do some more customisation but due to time constraints this was not possible.

I also learned about source code management in the form of gitlab which led me to do my own research on GitHub. I have now set up a few small personal projects on GitHub and hope to use it or gitlab next year when working on my final year project to collaborate on and manage our code.

Sources

[1] http://jquerymobile.com/