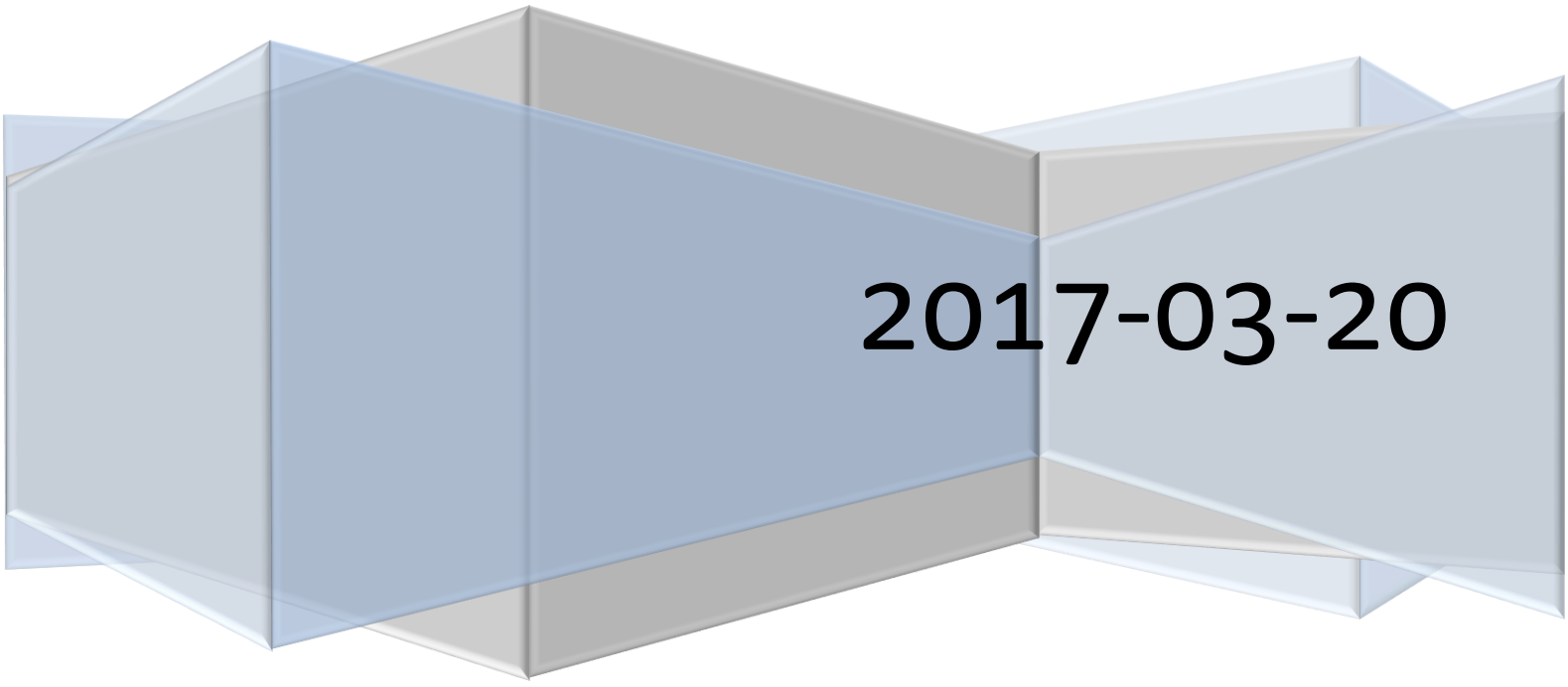
Simple Route

MSc. Communications Software – Enterprise Web Development

Colm Carew - 20053766



2017-03-20

Contents

Plagiarism Declaration

I certify that this assignment is all my own work and contains no Plagiarism. By submitting this assignment, I agree to the following terms:

Any text, diagrams or other material copied from other sources (including, but not limited to, books, journals and the internet) have been clearly acknowledged and referenced as such in the text by the use of ‘quotation marks’ (or indented italics for longer quotations) followed by the author’s name and date [eg (Byrne, 2008)] either in the text or in a footnote/endnote. These details are then confirmed by a fuller reference in the bibliography.

I have read the sections on referencing and plagiarism in the handbook or in the WIT Plagiarism policy and I understand that only assignments which are free of plagiarism will be awarded marks. I further understand that WIT has a plagiarism policy which can lead to the suspension or permanent expulsion of students in serious cases. (WIT, 2008).

Signed: Colm Carew

Date: 20/03/2017

Introduction

Simple Route is a Web Application written using the React JavaScript Library. The core idea of the application is to; using a Map view, create a Single Page Applications (SPA) which gives the user the easiest directions to their desired locations. In creating this solution the goal is also to exhibit some of the React concepts learned throughout the React lectures of the Enterprise Web Development module.

Project Layout

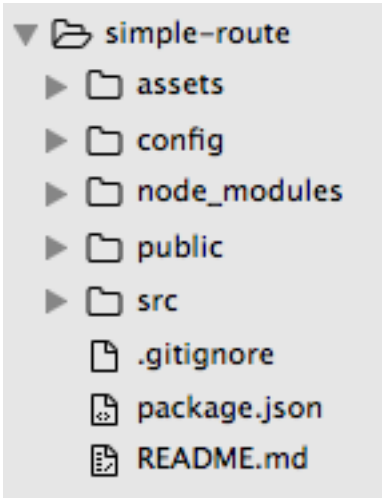


Figure 1: Project Layout

As can be seen the folder structure is :

- assets (assets required for the webpage such as images)
- config – configuration files go here such as a config file for the Google Maps API Key
- node_modules – this is the folder containing all external components used in the project which are pulled using npm
- public – this is where the website favicon and index.html file are located
- src – this is where all the React Components that make up the website are located
- .gitignore – the file to tell git what files/folders should be ignored when committing
- package.json – the project configuration file which includes all relevant dependencies
- README.md – the read me of the project

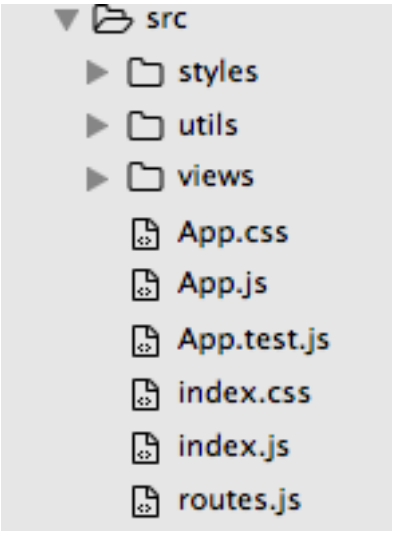


Figure 2: src project layout

Src is broken up into :

- Styles – where some of the css files reside
- Utils – where the utilities such as database writing or Google Maps services reside
- Views – where the actual react components are which comprise the website
- Route.js – used for the routing of the application
- Index – the entry point of the application
- App – where all of the components are gathered before being rendered in index

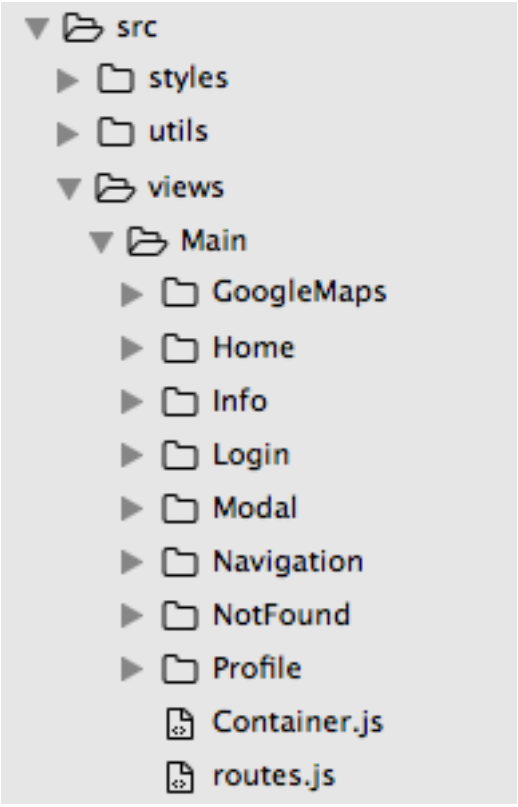


Figure 3: Views Directory

The views directory is inside the Main directory which is inside the src directory, it's content are :

- GoogleMaps – all the components for GoogleMaps to be viewable and interactive
- Home – the main component of the application is located in this folder
- Info – the About page of the application is located here
- Login – the components related to the Login screen are here
- Modal – used to contain a custom modal which is used for displaying forms
- Navigation – folder containing the custom navigation bar
- NotFound – used for containing error pages, in this case just the NotFound page
- Profile – contains the components for the user profile
- Container – used as the main container of the home page
- Routes.js – used for the application routing – the other routes.js in the src directory uses this file

NPM Modules Used

- auth0-lock – used for user authentication
- classnames – used for CSS so className is used in the JSX rather than class for defining css styling
- firebase – for this project firebase is being used as the database store – more on why later as auth0 could also have been used
- jwt-decode – used with auth0 authentication
- react – the react library
- react-bootstrap – a bootstrap library so bootstrap components can be used in React
- react-google-maps – a React module for Google Maps
- react-router – the react router used for routing the application
- react-router-bootstrap – a module that allows the linking of the site within bootstrap components
- react-select – a module used for dropdowns
- superagent – a HTTP request library – used in obtaining location/address data from Google Maps API

3rd Party Libraries and Reasoning

Auth0 (https://auth0.com/) - Auth0 was used for sign in as at the time of project creation it was a popular choice for SSO and has an extensive list of providers that can be used for sign in. In this project it was used for Facebook, Twitter and Google sign in as well as a default registration if the user does not want to link an existing account.

Firebase (https://firebase.google.com/) – Firebase and Auth0 can have very similar uses. Auth0 primary goal is authentication and Firebase's goal is overall application infrastructure. In this project it is being used for the database side of things storing the user's preferences and directions they have searched. A major contributing factor in picking Firebase was that, in parallel with this project an Android App is being developed as part of another module that uses the similar idea of simple navigation. The Android App uses Firebase and thus would be nice if both applications could share

the same user data so a user could use the website or the Android App with both containing the same data.

Google Maps for React (https://github.com/tomchentw/react-google-maps) - this is a library which essentially wraps the Google Maps JavaScript API giving React Components.

How to Setup the Project

The project can be found on github here : https://github.com/1carew1/simple-route .

- Clone the repo and cd into the directory created.
- Ensure you have Node installed –Node version 6.10.0 and npm version 3.10.10 were used which were the latest LTS at the time of creation.
- Obtain API keys from Google for Google Maps, Auth0 for the lock login and Firebase for the database.
- For Auth0, allowed URLs need to be specified so the user can login at that URL/s.
- Create a folder called config in the root of the project and cd into this folder.
- Create 3 files in this folder
    - auth0Config.json
    - firebaseConfig.json
    - googleMapsAPIKey.json
- auth0Config.json should contain the following replacing the placeholders with the appropriate API Keys
    - { "apiKey": "XXXXX", "userUrl": "XXXXX"}
- firebaseConfig.json should contain the following replacing the placeholders with the appropriate API Keys
    - {
      "apiKey": "XXXXXXX",
      "authDomain": "XXXXXXX",
      "databaseURL": "XXXXXXX",
      "storageBucket": "XXXXXXX",
      "messagingSenderId": "XXXXXXX"
      }
- googleMapsAPIKey.json should contain the following replacing the placeholders with the appropriate API Keys
    - { "apiKey" : "XXXXX" }
- Due to having to include the Google Maps JS library in the index.html file which is located in the root of the project in the public folder, the API Key here also needs to be updated to be the Google API Key obtained earlier. There may be one present in this file already but this is locked down to specific IP addresses so it will not work.
- Once all configs are in place cd into the root of the project and run “npm install” – this will install the required dependencies.

- Once all dependencies are installed run “npm start” which will open the running project in the environment’s default web browser at “http://localhost:3000”.

Application Layout and Features

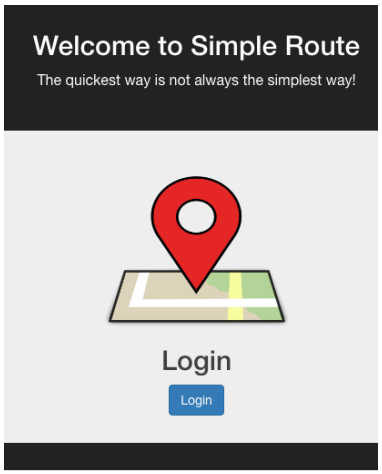


Figure 4: Login Screen

Figure 4 is the starting point of the application. Not much to say here other than it has the application logo and the Login button. Clicking on the Login button reveals the following popup :

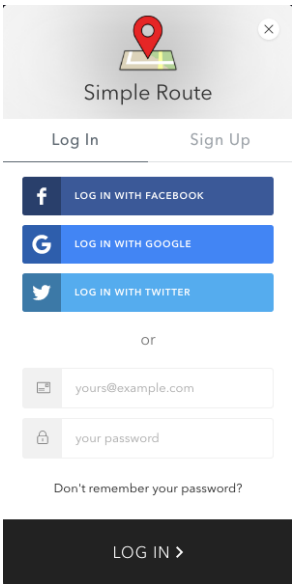


Figure 5: Login Menu

Figure 5 shows the Login Menu. This is the default Auth0 Lock menu which has been customised to change the LOG IN colour and the Logo to be the Simple Route logo. This allows for the user to Sign in with Facebook, Twitter, Google, their Auth0 account or they can register for an Auth0 account.

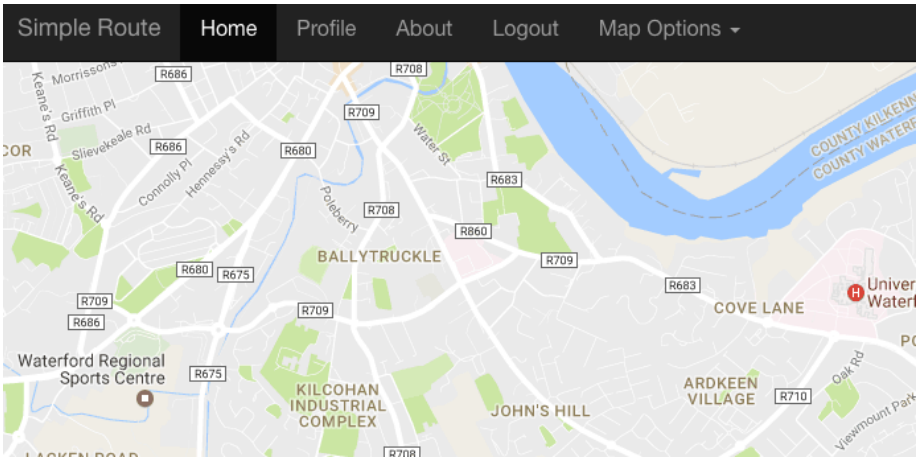


Figure 6: Home Screen

Once logged in the user is brought to the home screen, which should be a navigation menu on top and the rest of the screen should be a map. Initial design has the map as a smaller container however it was decided to make the feel more like Google Maps website which is full screen.

Map Options are currently enabled in the home view only. Clicking on Map Options Reveals a dropdown :

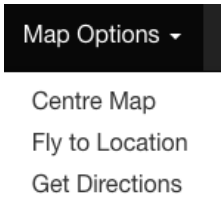


Figure 7: Map Options Menu

Centre Map centres the map and puts a marker at the user's location, which is pulled from the browser, provided the user allows this.

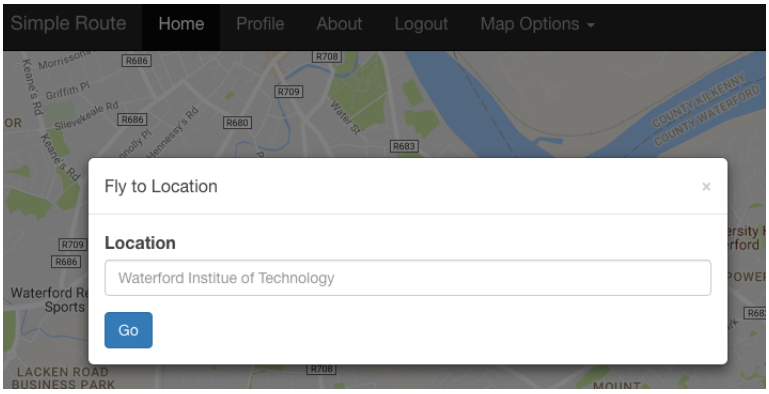


Figure 8: Fly to Location

Within the navigation bar, there is a CustomModal that is used for popups. The Modal is constructed in such a way that a form should be passed in to it and the modal displays that form. This eliminates the need for multiple modals for each form. In this case it is used for the Fly to Location functionality. If the user enters an Invalid Location the browser will give an alert and the Modal Will not close. If the user gives a valid address, the map will fly to that location, place a marker and close the modal. The user can also close the modal by clicking anywhere else on the site that is not the modal or by clicking 'x' in the top right hand corner of the Modal.

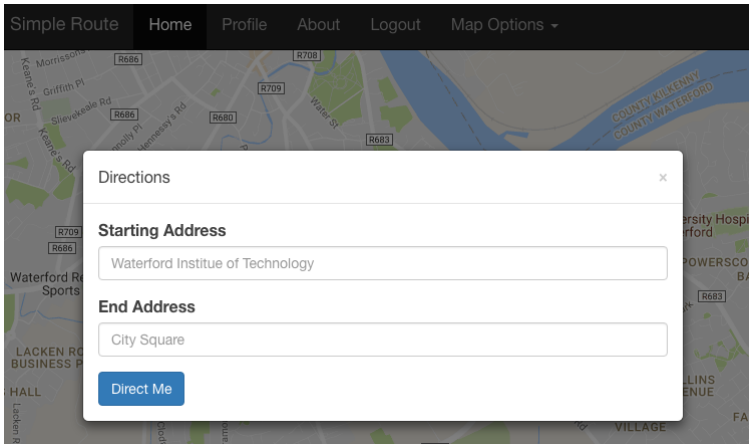


Figure 9: Directions

Directions also use the Modal as clicking the 'Get Directions' option passes in the Directions Form to the modal and displays the modal. If incorrect locations are searched or there is no route between the two locations an alert box pops up alerting the user. When valid addresses are entered however the simplest route between the two points will be plotted on the map and the directions will appear on the right hand side of the page :

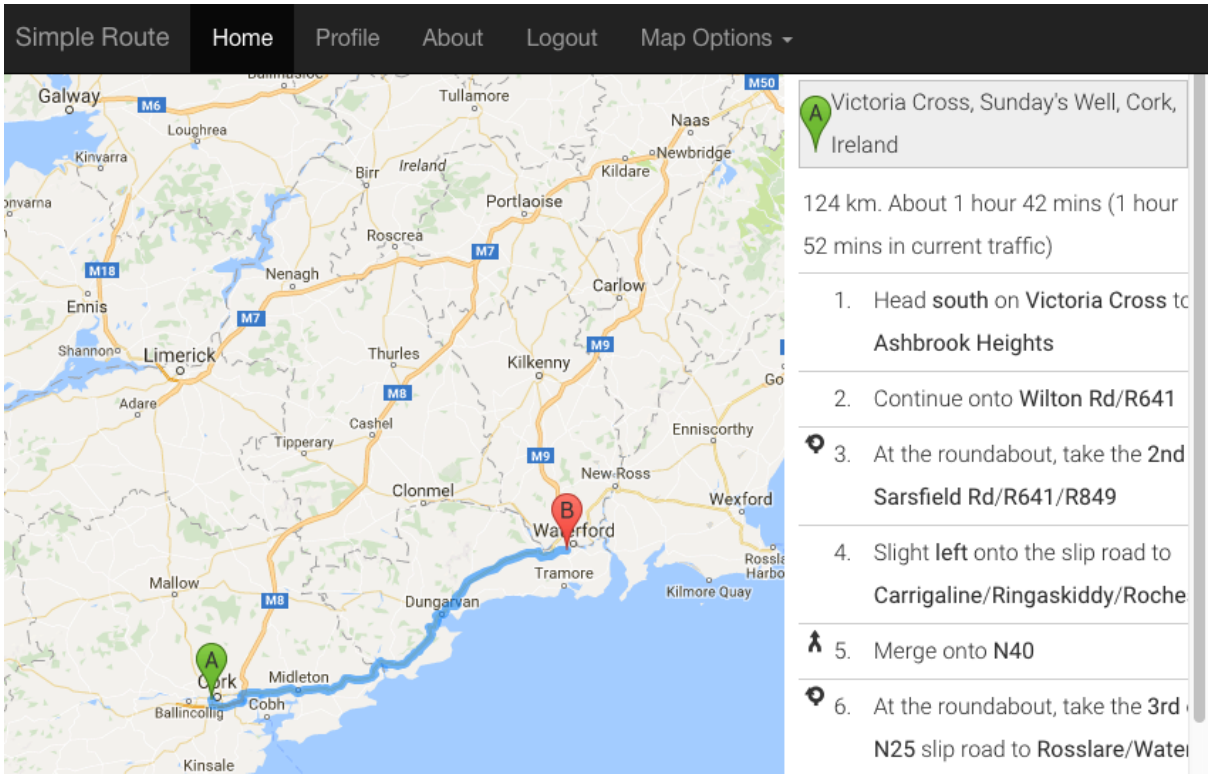


Figure 10: Directions Example

As well as displaying the directions a parameterised URL is generated such that the link can be sent to someone else. The URL for this is : /home/directions/:fromAddress/:toaddress . This was the only URL parameterised as the Fly to Location functionality and the Centre Map functionality did not seem as useful to share. Centring the map, flying to a location or clicking home again will get rid of these directions on the right hand side. A point to note is that a User may not want the directions to be for a driver and would prefer walking directions or bus directions. This is covered in the Profile.

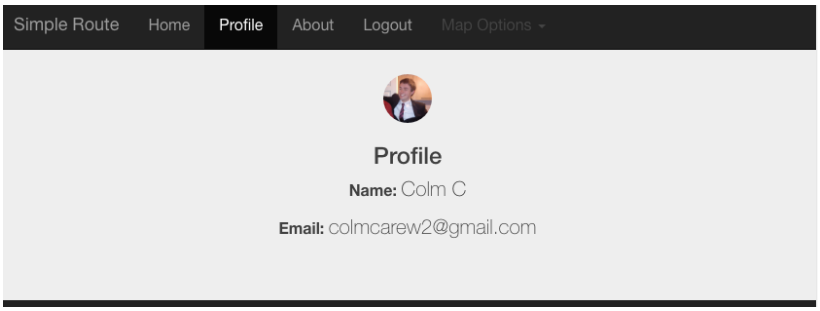


Figure 11 : Profile Basic Info

The Profile is broken into 3 sections. The basic information, username, email and profile photo, the user's preferences and finally the user's last 10 searched directions.

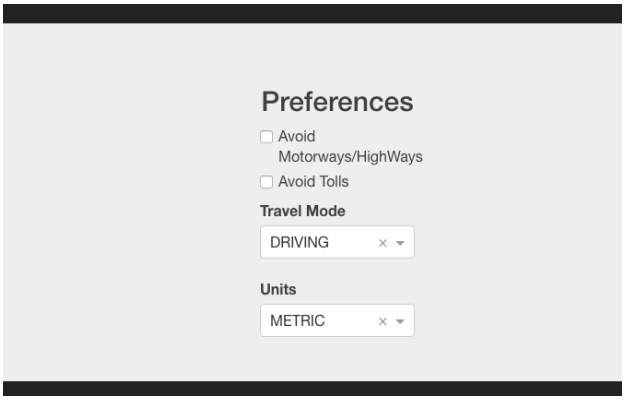


Figure 12: Profile Preferences

Each of the options available are used in calculating the user's directions. The values are stored in a Firebase database table titled 'user'. Travel Mode and Units are each dropdowns. All 4 available options are set to default as above and if the user clicks on one it is automatically updated in the Firebase Database. Note when the Profile page loads the default values will be present but the user's actual preferences should load by the time they click on one.

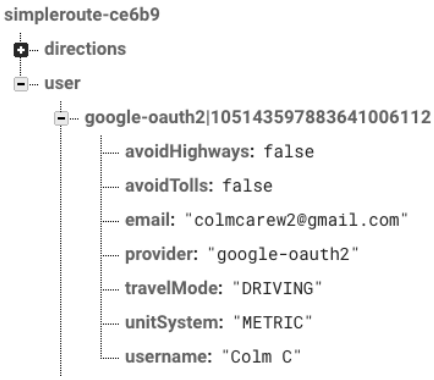
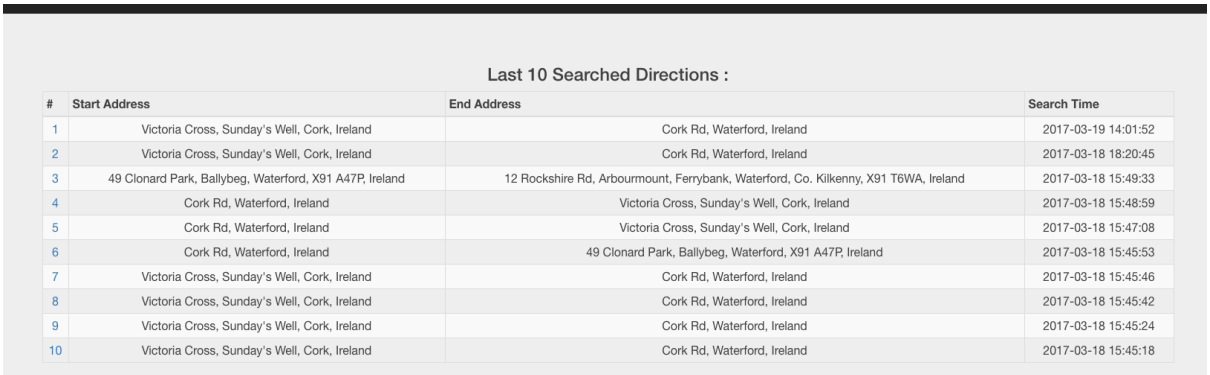


Figure 13: User Preferences

Example of a user's preferences in the Firebase Database.



Last 10 Searched Directions :

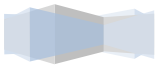
#	Start Address	End Address	Search Time
1	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-19 14:01:52
2	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-18 18:20:45
3	49 Clonard Park, Ballybeg, Waterford, X91 A47P, Ireland	12 Rockshire Rd, Arbourmount, Ferrybank, Waterford, Co. Kilkenny, X91 T6WA, Ireland	2017-03-18 15:49:33
4	Cork Rd, Waterford, Ireland	Victoria Cross, Sunday's Well, Cork, Ireland	2017-03-18 15:48:59
5	Cork Rd, Waterford, Ireland	Victoria Cross, Sunday's Well, Cork, Ireland	2017-03-18 15:47:08
6	Cork Rd, Waterford, Ireland	49 Clonard Park, Ballybeg, Waterford, X91 A47P, Ireland	2017-03-18 15:45:53
7	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-18 15:45:46
8	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-18 15:45:42
9	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-18 15:45:24
10	Victoria Cross, Sunday's Well, Cork, Ireland	Cork Rd, Waterford, Ireland	2017-03-18 15:45:18

Figure 14: User Searched Directions

This is a table generated from the user's most recent 10 searched directions with the number on the left hand side of the table having a link which will bring up the directions via the parameterised URL when clicked. These are stored in the 'directions' table in the Firebase database. This table may take a second to load when the user goes to the profile page. This is just a React component re-render and not an entire page re-render.

Figure 15: Example Directions

As can be seen the table is used to store the date searched, to and from location and the id of the user who searched it. Note that this table has a rule in firebase so that the user_id field of this directions table is indexed to make querying the table faster.



That is the main functionality of the application. The next link to look at is the About Page.

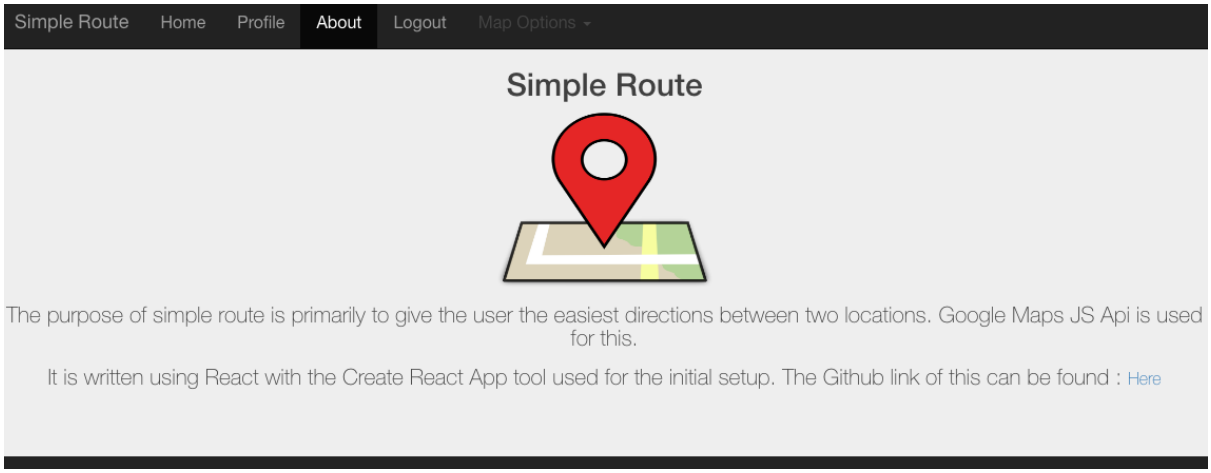


Figure 16: About Page

The about page is a very simple page which gives the purpose of the project and the link to the Github project. Notice that the Map Options is greyed out and when the dropdown is clicked that the menu does not drop down.

The logout link will log the user out and bring them back to the login page. A final point to note, as this is a React App it is scalable for various devices and does scale well such that the menu collapses when the width goes below a certain size and the application is still useable.

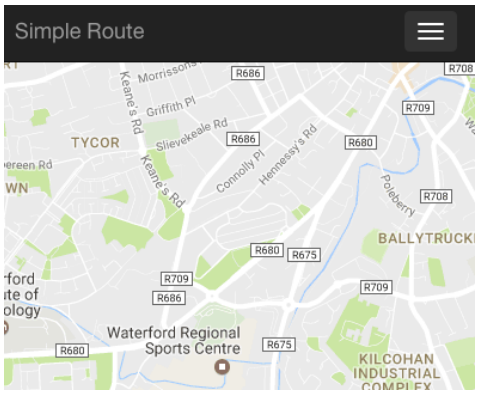


Figure 17: Smaller Screen Main Page

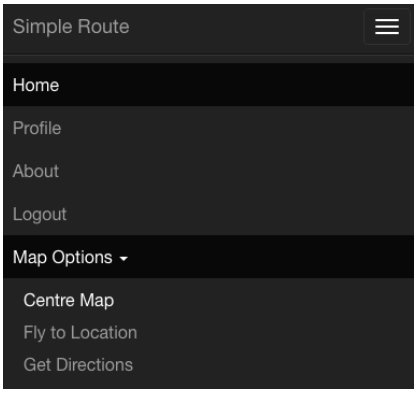


Figure 18: Smaller Screen Menu