Apps Map.

**Purpose.**

During the initial planning of this project, it was decided that a map was to be an intractable component of this app. There are three purposes for the existence of this map.

Firstly, the map allows users to be able to see the locations of their selected places of interest.

It also allows them to be able to be physically directed to these places, due to the map also displaying the user’s location. Finally, it is used for allowing users to add their own chosen locations to the application, by allowing the user to add their own custom markers onto the map.

**Appearance.**

Upon launch of the app, the map shows user’s surrounding location, along with their exact location displayed using a distinctive marker. When clicked, this marker displays an information window, further identifying itself as displaying the user’s location. When the use changes location, the user’s location on the map will be updated in response.

Also shown on the map are places of interest near to the user, also displayed as markers, and identified as such through their difference to the user’s location marker. These specific markers have been determined by the user’s selection of which types of places of interest they’d like to view. When these markers are clicked, the names of these places of interest are displayed to the user in information windows.

There is also a button on the top center of the map stating ‘Add new location’. This allows the user to user to add their own custom markers to the map, customising the database of the app. When this is selected two more buttons are dynamically created, allowing the user to confirm their choice of marker location or cancel their attempt at making a new marker. When either of these buttons has been selected, both buttons are then removed, as they are then unnecessary.

**Development process.**

**Initial set up.**

The first step I took in creating the map component of this project was to decide what to use to develop it in. I chose Google Maps JavaScript API. I chose this due to the simplicity in integrating it into the rest of the angular based front end of the application, along with its robust feature support and clear to use documentation.

Having chosen to use Google Maps, the first thing I needed to do according to the documentation was to register the application project on the Google API console and create an API key to add to the app. This would then allow me to make use of the Google Maps API functionality. I did this by following the steps required in the documentation, and then went about the process of learning how to create what was needed.

**List of requirements**

I then made a list of the functionality required, and with the help of the official documentation and a tutorial on YouTube, went about learning how to develop this functionality. The things I needed this map to do where as follows:

* **Present a map of the user’s location.** I wanted the map to be centred on the user’s location. This would require the user of the geolocation functionality within the Google Maps API. I then wanted the user’s exact location to be identified by a marker placed upon the map, using the extracted geolocation coordinates. To distinguish from other markers upon the map, I wanted the marker icon to have its own unique design, and an information window to be displayed when clicked, identifying it as the user’s own location. I also wanted the position of the user’s location on the map to be routinely checked, in order to track the user’s location as they moved. Allowing the user to make use of the map, in order to be directed to chosen locations.
* **Show the user’s selected sites of interest.** When the user selects the type of locations they wish to view the details of, I wanted the map to display the locations of these to the user. This would be done through the use of importing the necessary data into the map, and displaying these locations using markers. These markers would also display information windows when clicked, identifying the locations by their imported names.
* **Allow user to add custom locations to the map.** I wanted to allow the user to be able to add custom markers to the map. To do this, a button would be provided to the user allowing them to perform this option when clicked. When done so, I wanted the map to be able to add a new marker to any location then clicked upon it. With the options then provided to the user to then confirm or delete the chosen location for the new marker. This would be done through the use of an event listener which would be triggered once the initial button was clicked.

**Development.**

I tackled this by creating a number of individual maps. Each one containing the described individual elements needed in the finished map. This helped my learning process by allowing me to focus on only learning one piece of functionality at a time. These were made from following the official Google Maps API documentation online. Initially I also began following a tutorial on YouTube, but found the official documentation to be sufficient for learning.

I used Github to keep track of my progress, by creating my own branch in order to commit my work to. I used Github issues and milestones to keep account of certain steps I deemed necessary to help me with managing the work. The code was written using Brackets text editor, and tested in Google Chrome, Microsoft Edge and Mozilla Fireworks.

A version of the final map before implementation into the app has been submitted with this project called ‘Map.html’. This can be found within the folder ‘Map’. It contains all of the functionality listed below. Found on the lines referenced.

The first map I made from the documentation consisted simply of adding a basic map element to a html page, centered on the coordinates of Galway, and containing a marker at a fixed position. This was to test that the Google Maps API was loading correctly through the use of my personalised API key. This worked as expected. (Map.html. Lines 56 – 68)

The next page dealt with importing external data into the map. To do this, I made an external Json file holding some test data for reading and displaying on the map. This held the coordinates of some locations, and populated the map accordingly. (Map.html. Lines 71 – 107)

Geolocation was then worked on. In the initial test for this I added a simple marker with information to a map, displaying my location through the use of the geolocation feature of the Google Maps API.

The implementation of click events to markers was then added. These will be for showing the names of the locations to the user when clicked. (Map.html. Lines 95– 103)

Updating of the user’s location was then developed. This was done by having the application call a function every three seconds which checks the user’s location, and adjusts the position of the user’s marker on the map accordingly if necessary. (Map.html. Lines 123– 175)

The final piece of development was the adding of custom markers to the map. This was done by adding a button for the user to initially click when they’d like to add a new marker to the map. This sits within a floating panel in the center top of the browser. When clicked, this creates an event listener, which will add a new marker to the map when the map is clicked. It also calls a function which dynamically creates two more buttons. Upon creation these buttons are assigned onclick functions. The save button deals with saving the newly created marker, whilst the cancel button clears it from the map. Both buttons also remove these newly created buttons, as they are not necessary when a new marker isn’t being created, and therefore viewing space for the map isn’t wasted. (Map.html. Lines 36 – 38. 188 – 235. 239 – 246. 264 - 302)