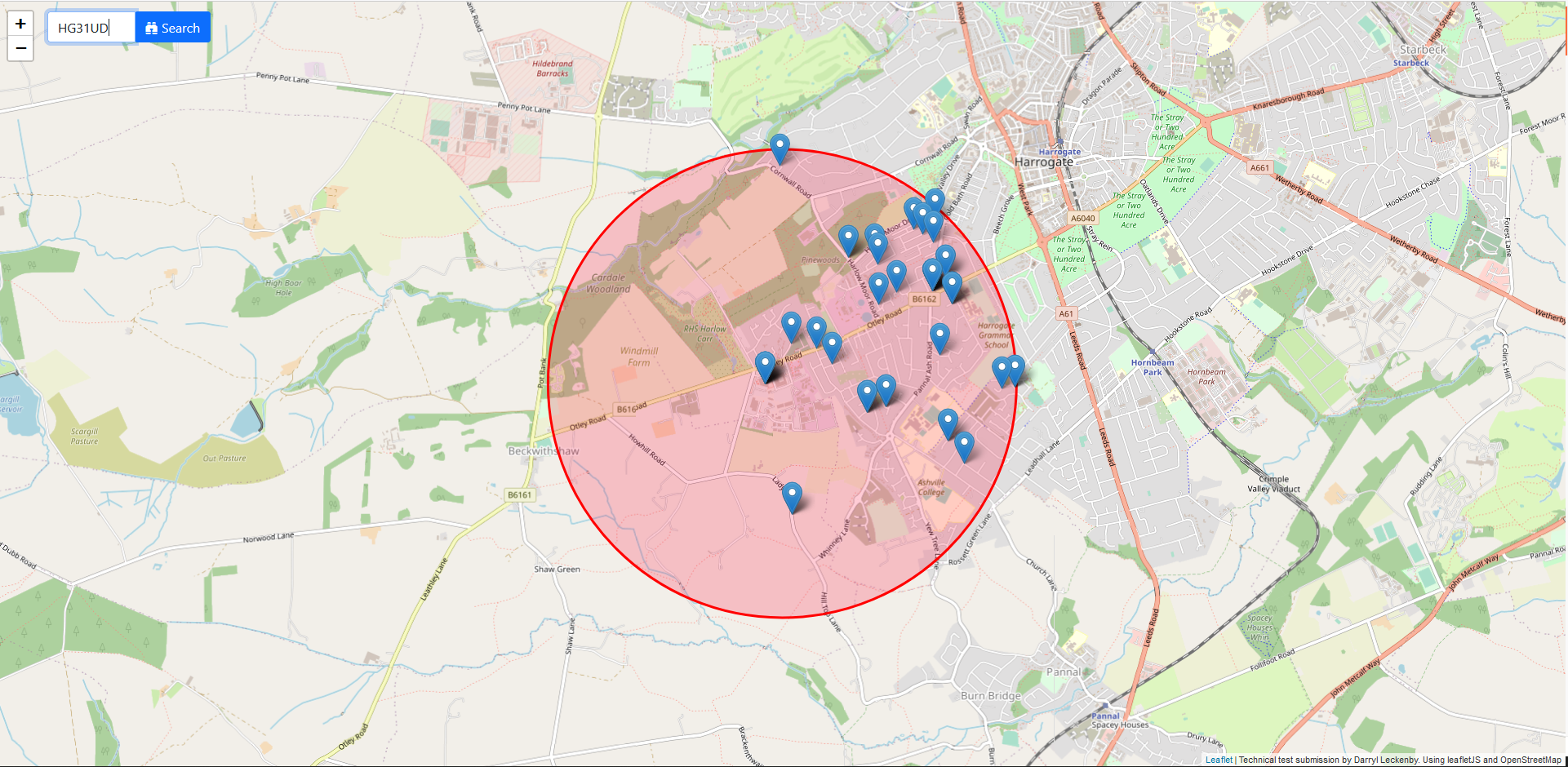
**Summary:**  
I have created a .Net Core WebAPI as well as a basic Javascript client for calling and presenting the webAPI data.

**Getting started:**

1. Run the CrimeStatsAPI project from within visual studio to launch the .NET Core WebAPI
2. I have created a basic HTML client (3 files) which I have put into the wwwroot folder, this will automatically open when you run the WebAPI project.
3. If you have any issues with the client running, please open CrimeStatsAPI/wwwroot/crimestats.js and change the port number of line 4 to match whatever port number your visual studio environment is building and running on.



**Web API Overview:**

A basic webapi has been created with three end-points. The postcodelookup and crimelookup endpoints were initially created just to test responses with both data providers API’s and I have left these in.

A folder has been provided with the exports of the postman tests I’ve used for both the data sources as well as the WebAPI.

|  |  |
| --- | --- |
| localhost:port/detailedpostcodelookup/POSTCODE | This is the main end-point used by the client, it will return a JSON response containing the crimereports for 1 mile radius of the postcode entered along with the location co-ordinates. |
| localhost:port/postcodelookup/POSTCODE | This is not required, it provides an endpoint for returning the latitude and longitude of the postcode in JSON format. |
| localhost:port/crimelookup/LON/LAT | This is not required, it provides an endpoint for returning the crimereports for the longitude/latitude provided. |

**Additional Notes**

This technical test response has been written using some brutal project management to cut as many corners as required to turn-around a minimally viable response due to short notice/little time around existing family/work/interview commitments etc (<3 hrs). For the sake of awareness of these I have listed them below:

* Security:
  + I have allowed full cross-site-scripting using a horrible ‘allow all’ CORS policy in the WebAPI, usually I’d drill down the CORS policy as much as possible.
  + I have also disabled SSL to reduce the potential issues running in another environment, all endpoints and tests with the API are not over secure sockets.
* Detail of Crime:
  + I would liked to have added more definitions to the data models returned from the police API to enrich the UI with more information.
* Defensive Coding:
  + Usually I would wrap 100% of all code with try-catches for the purposes of forwarding all exceptions and edge-case occurrences to central logging.
* Unit Tests:
  + I would have considered creating a series of unit tests with assertions to check expected behavior when (at least):
    - When time-outs to third party API’s occur
    - A postcode cannot be resolved to lat/lon co-ordinates
    - A correct postcode has no crime reports (such as mine YO411QG)
* Logging:
  + I have not implemented any logging; a personal favourite solution is [www.logdna.com](http://www.logdna.com) to send all events to a third party logging ingress solution for storage and aggregation. Usually I would have watchers setup on this kind of service to ping 1st line support, dev team or to create a support ticket when seeing never-events without relying on user feedback to be aware of them.
* Documentation:
  + In a real-world scenario I would introduce a swagger definition to index the end-points and features of the API.
* Client Design:
  + If time allowed I may have considered a react.js UI. I have undergone training into react with a previous employer and supported react apps but I wouldn’t class myself as a react expert. I prefer to deal with known quantities when time is not available like now; for this reason I have provided a simple JS client using a leafletJS instance.
* Better Map
  + If I wanted a wow -factor I would have used <https://www.wrld3d.com> which displays the map with all buildings as 3D and it even includes live cars driving around the map for that ‘live’ feeling. For a demo see <https://www.wrld3d.com/wrld.js/latest/docs/examples/>
* Data Aggregation
  + With an application like this I’d expect to summarise the data somewhat, maybe a pie chart of crime types etc. This is beyond the scope of the time I have for this.