A.

API – ASP.NET, ENTITY FRAMEWORK, SQL

* Used entity framework because it simplifies everything by generating the necessary methods in controller and DB connection.

FRONTEND – ANGULAR 2, FOUNDATION CSS

- Used angular 2 because I handled a recent project using it. Wanted to try what I learned.

B.

API – 4 hours

FRONTEND – 8 hours

C.

Limitations:

* **No unit testing was applied.** Tried but always get the cannot initialize DB error when Test project access DB. Alternative was to use mockups. Not sure if this approach is useful so gave up.
* **No Login and Registration.** Didn’t have the time. When you enter the same name if the user exists the notes of the records will be updated instead of adding a new account.
* **No Error handling in API –** It is difficult to modify response of entity framework to JSON.
* **Linq wasn’t used –** Didn’t find a reliable js.

Issues:

* Latitude and Longitude only support up to 2 decimal places. Again, complicated to modify in entity framework.
* Current location of user is shown by marker with C label. Notes saved by current user is shown in a different marker. Which means there are 2 markers in the same location.
* Users that saved notes on the same location cannot display together because they have different markers.

Note: NodeJs project is not really used. I only created the project in order to be able to import the Angular2 project to Visual Studio.

Angular2 project is this folder: GitHub\Landmark\_Remark\Landmark-Remark-UI\Landmark-Remark-UI.

Please install NodeJS. Then in cmd run the command**: npm install -g @angular/cli**

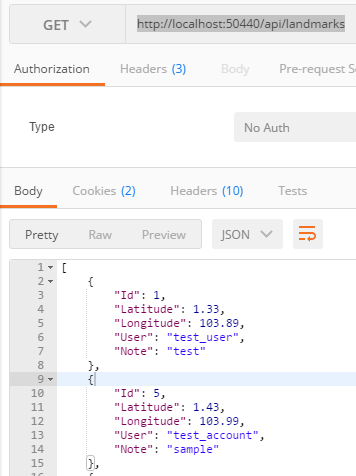
Afterwhich, cd /{path}/Landmark-Remark-UI.

Then run the **ng serve** command.

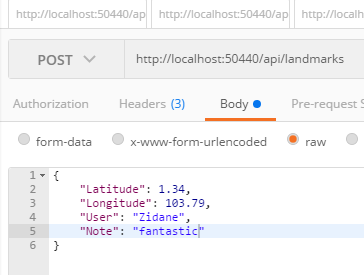
An error may occur and you may need to do an **npm install** again in your current path.

If you want to add other locations not based on current location use the API as shown below:

GET: <http://localhost:50440/api/landmarks>



POST: http://localhost:50440/api/landmarks



DELETE: [http://localhost:50440/api/landmarks/{id}](http://localhost:50440/api/landmarks/%7bid%7d)

id – corresponds to userId

