**MEAN STACK APPLICATION**

**FOR REAL-TIME WEATHER**

**FORECAST**

**TEAM-4**

**TEAM MEMBERS:**

**PRANAY KRISHNA RACHAMADUGU(CLASS ID: 31)**

**SOWMYA REDDY KATUKURI(CLASS ID: 20)**

**ANMISHA BRAHMADEVARA(CLASS ID: 7)**

**SAI PRASAD RAJU BOLLEPALLI(CLASS ID: 6)**

**Motivation:**

Our application is meant to share the weather details like temperature, air pressure, chance of rain and hourly forecast.

**Significance:**

The weather affects almost everything we do in our daily lives. It affects what we wear and what types of activities that we do. In many professions, it affects either where we work or when we work, and often it affects if we work at all that day. So our app is useful in day to day life.

**Objectives:**

* To implement a MEAN Stack application which retrieves the current weather conditions of a location and predictions for the next five hours.
* To implement a login and registration page which takes the details from the UI and stores them in the local MongoDB.
* To develop a home page where we can search the weather details using The Dark Sky API and Mapbox API for the geolocation.

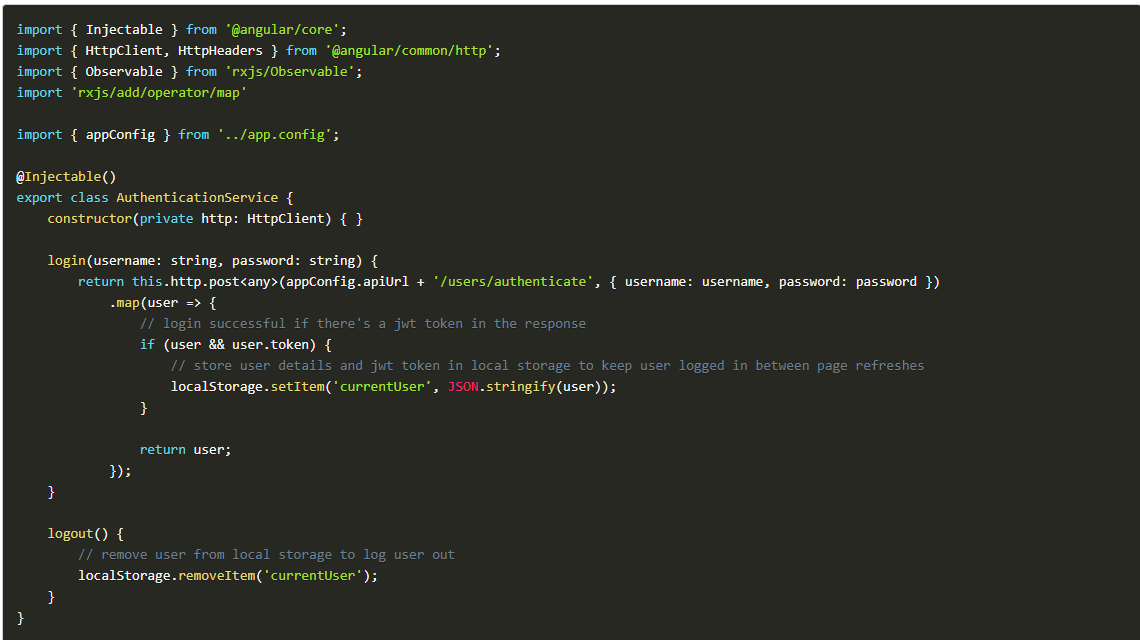
**Features:**

* MongoDB for user details.
* Angular for the front-end part.
* Node.js for the server part.
* API Implementation for geolocation and weather details.

**Workflow:**

First the login and registration page is created.

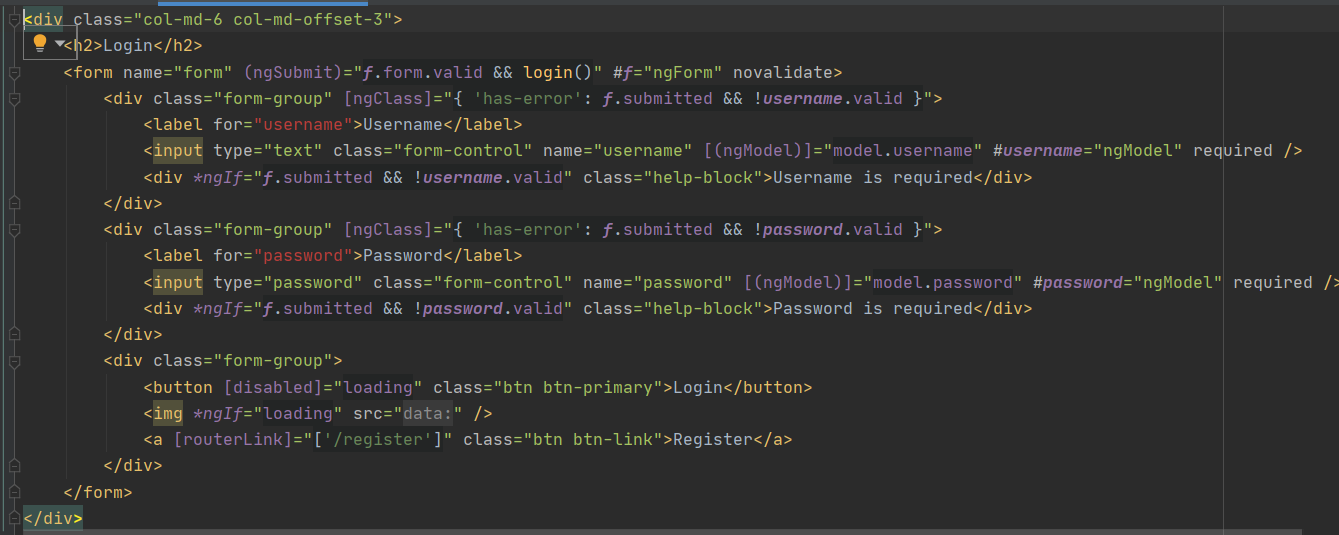
The authentication service is used to login and logout of the application, to login it posts the users credentials to the API and checks the response for a JWT token, if there is one it means authentication is successful so the user details including the token are added to local storage.



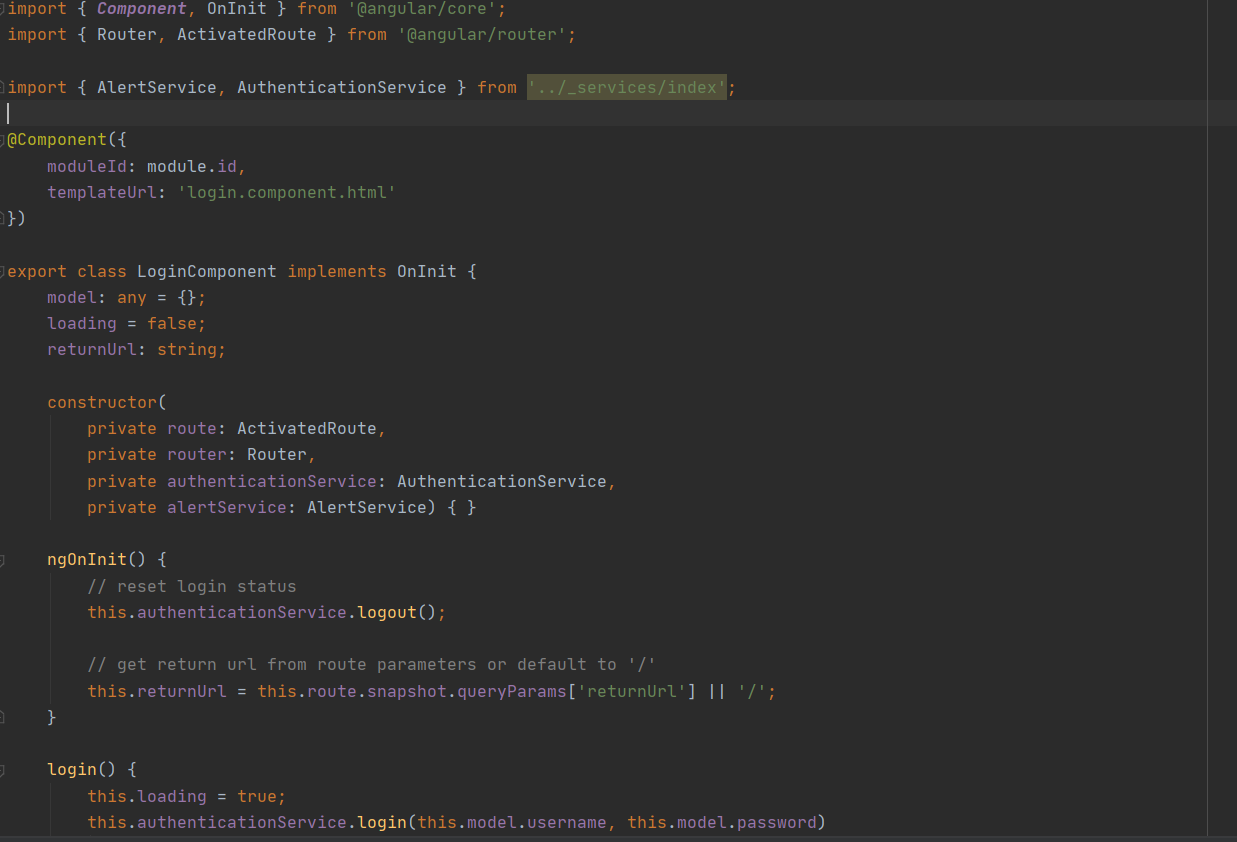
The user service contains a standard set of CRUD methods for managing the users via the API.



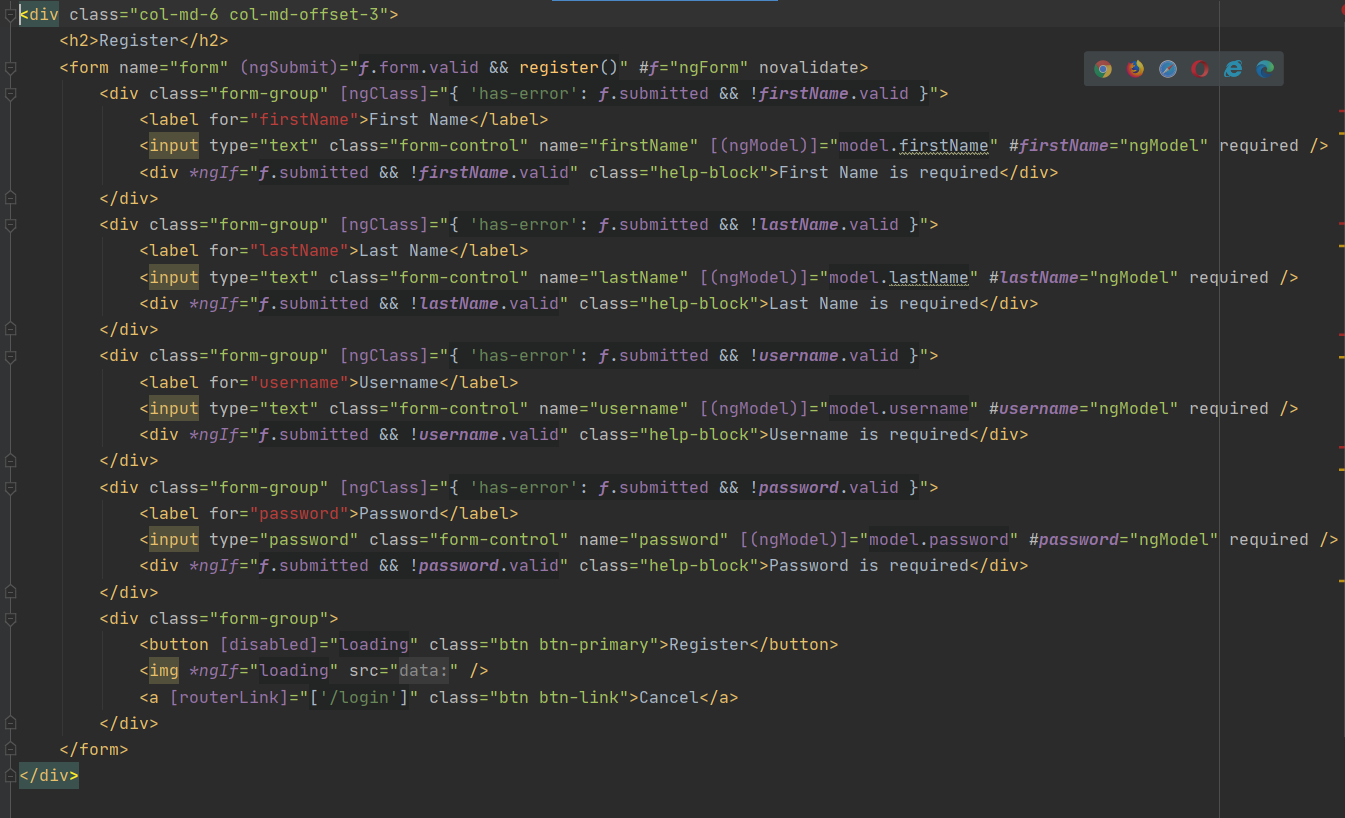
The login component template contains a login form with username and password fields. It displays validation messages for invalid fields when the submit button is clicked. On submit the login() method is called as long as the form is valid.



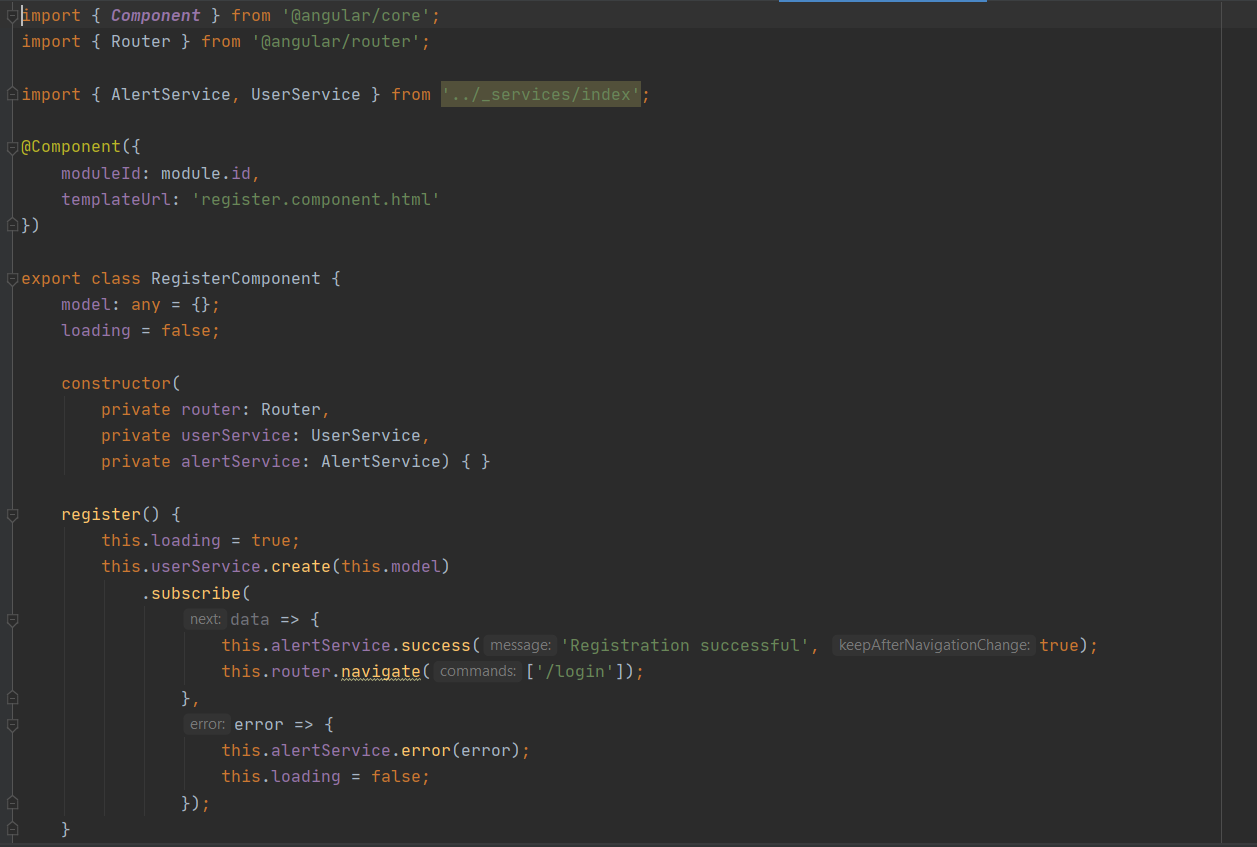
The ts part of the login component:



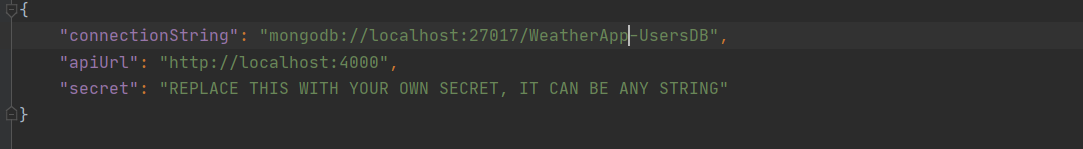
Register component Template:



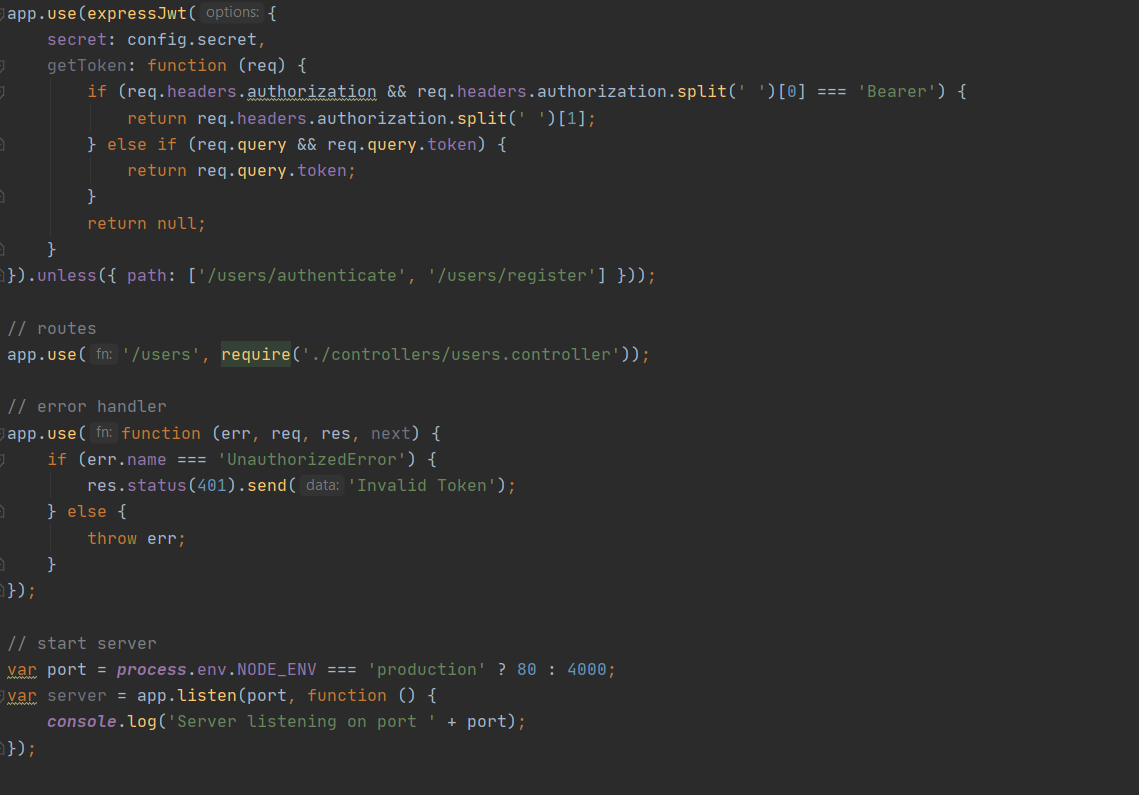
Register Component:



The config.json file where we declare the connections to the MongoDB:



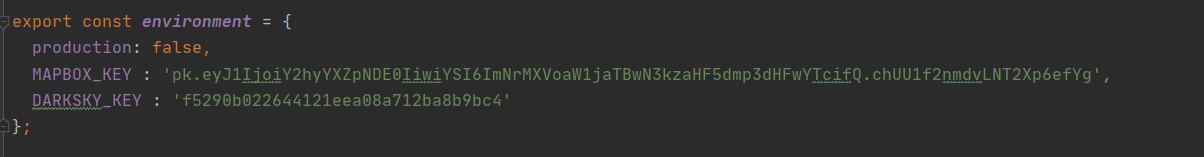
Server.js part:



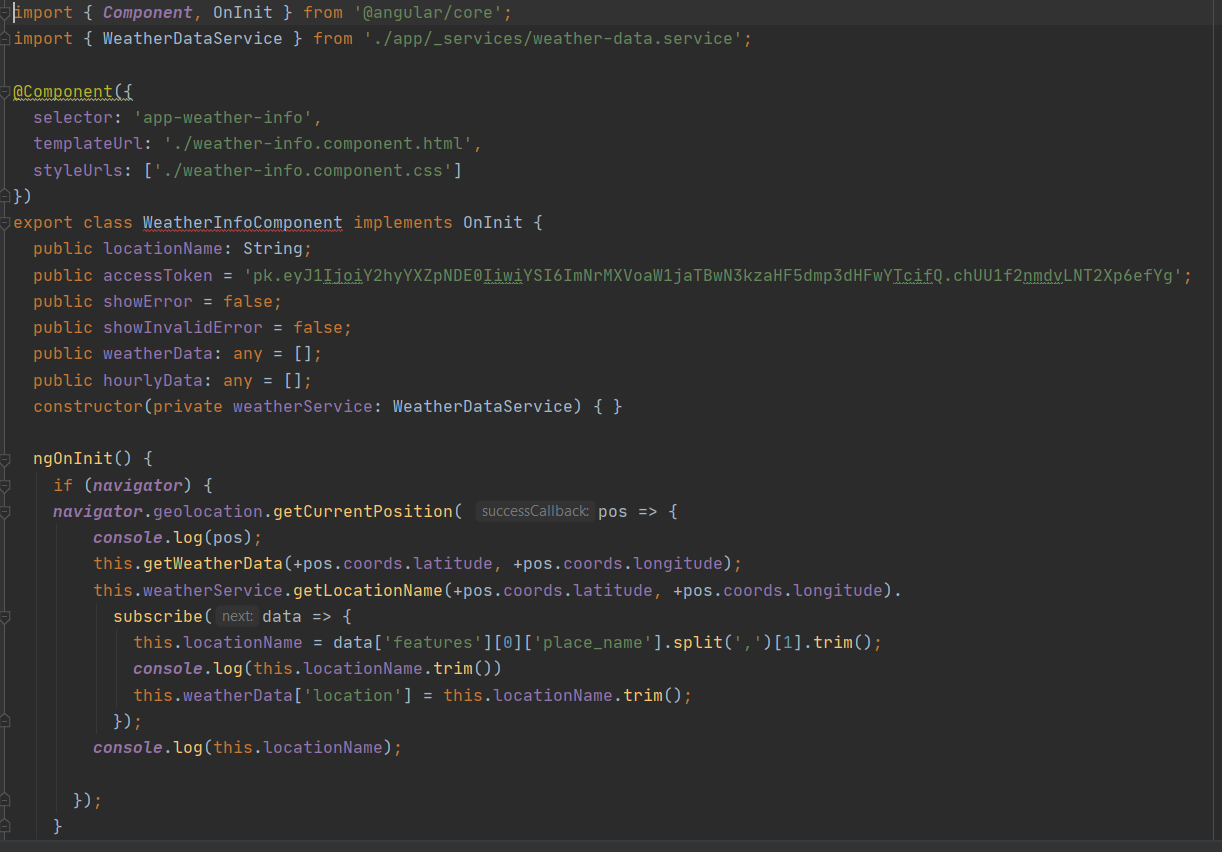
API Services for the application:



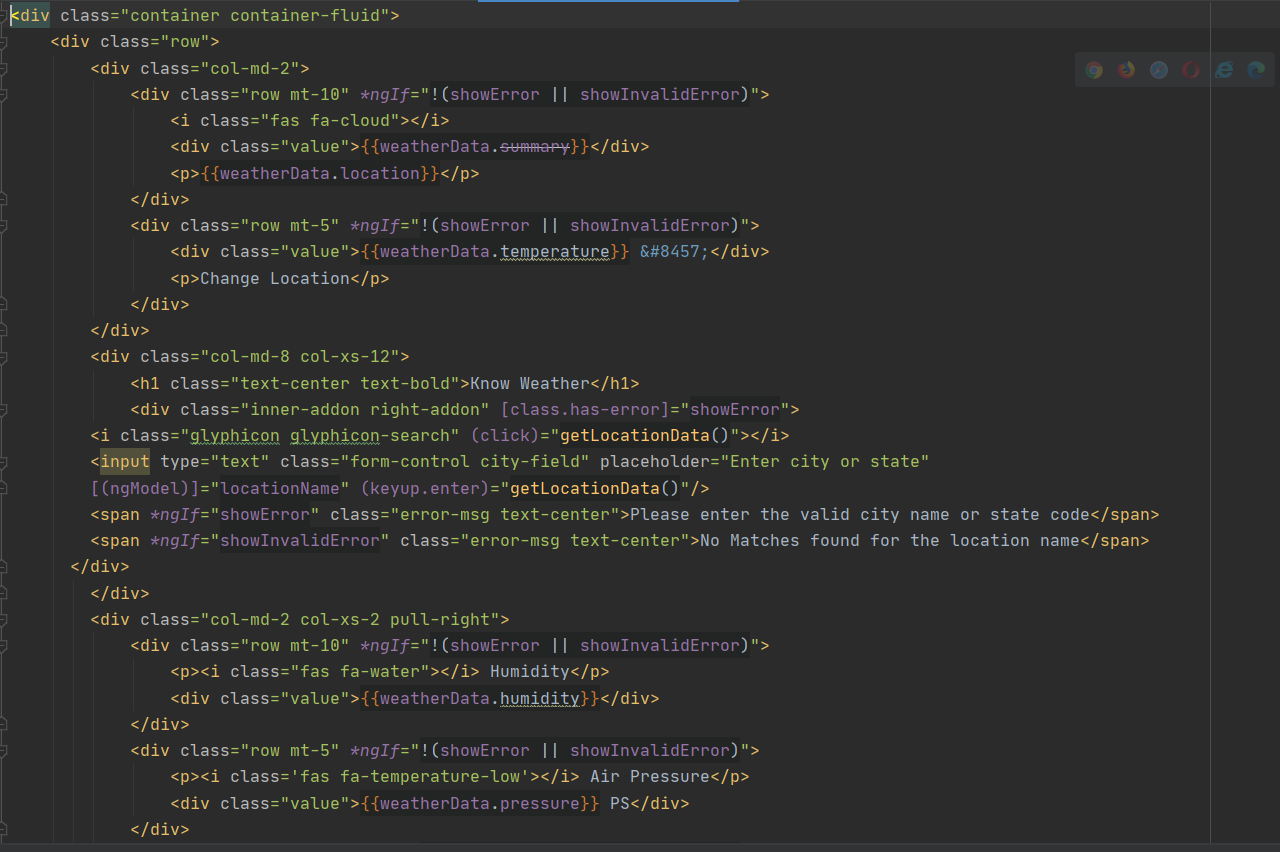
Environment:



Weather-info component where the functions to fetch the geolocation and weather details are implemented:

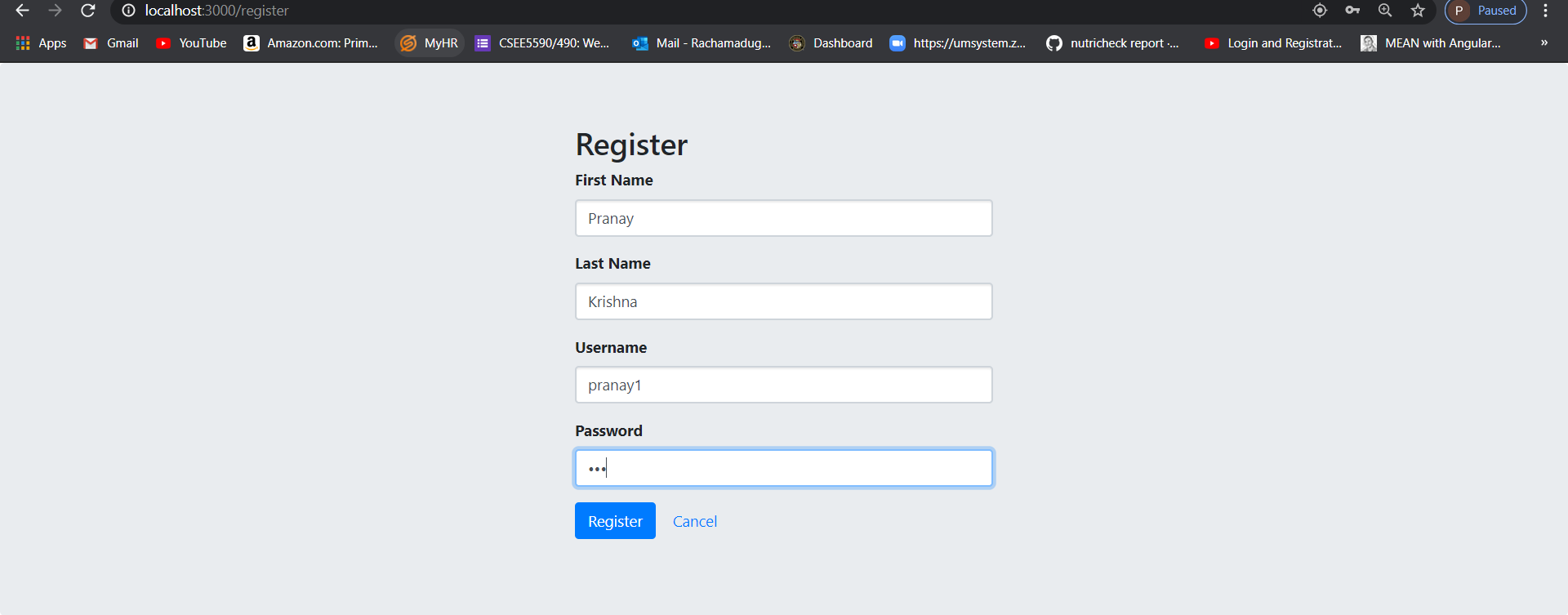


Weather-info Template:

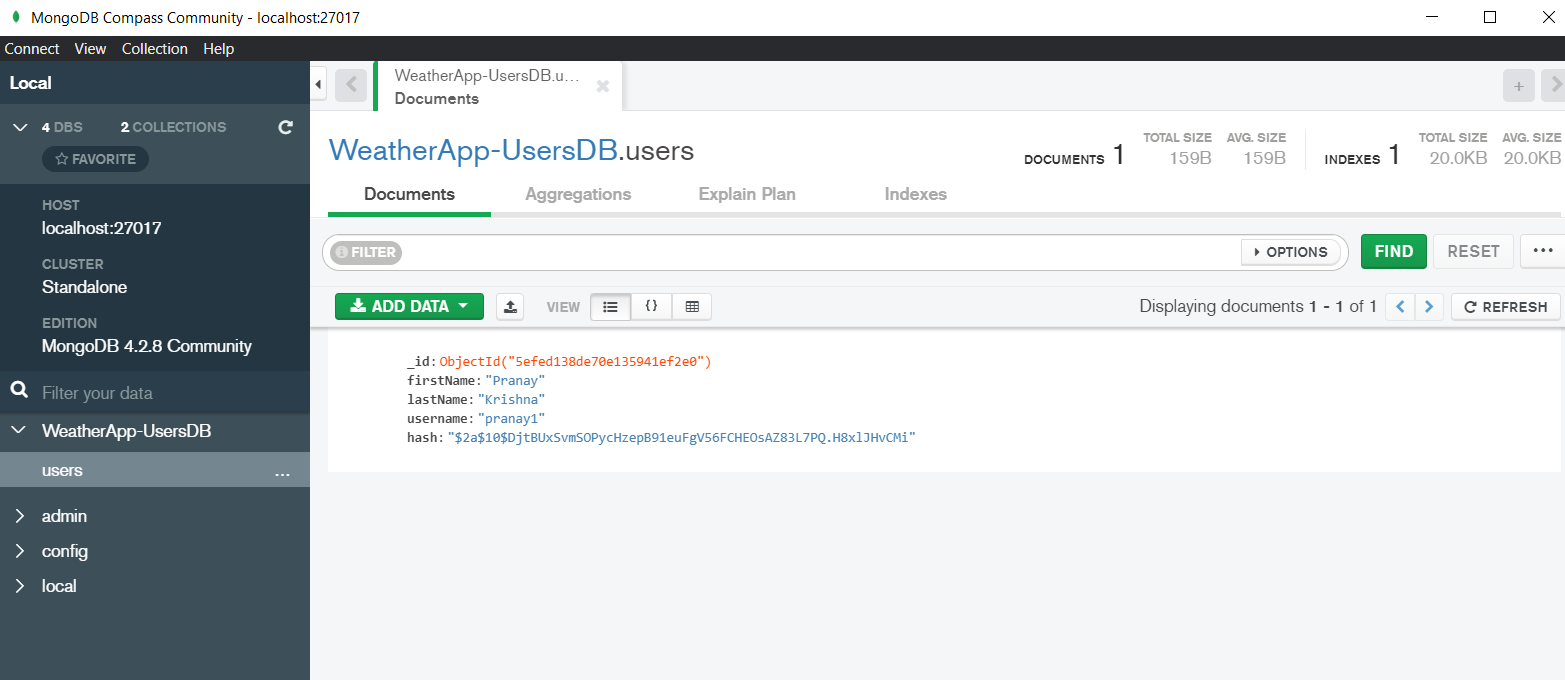


**Output:**

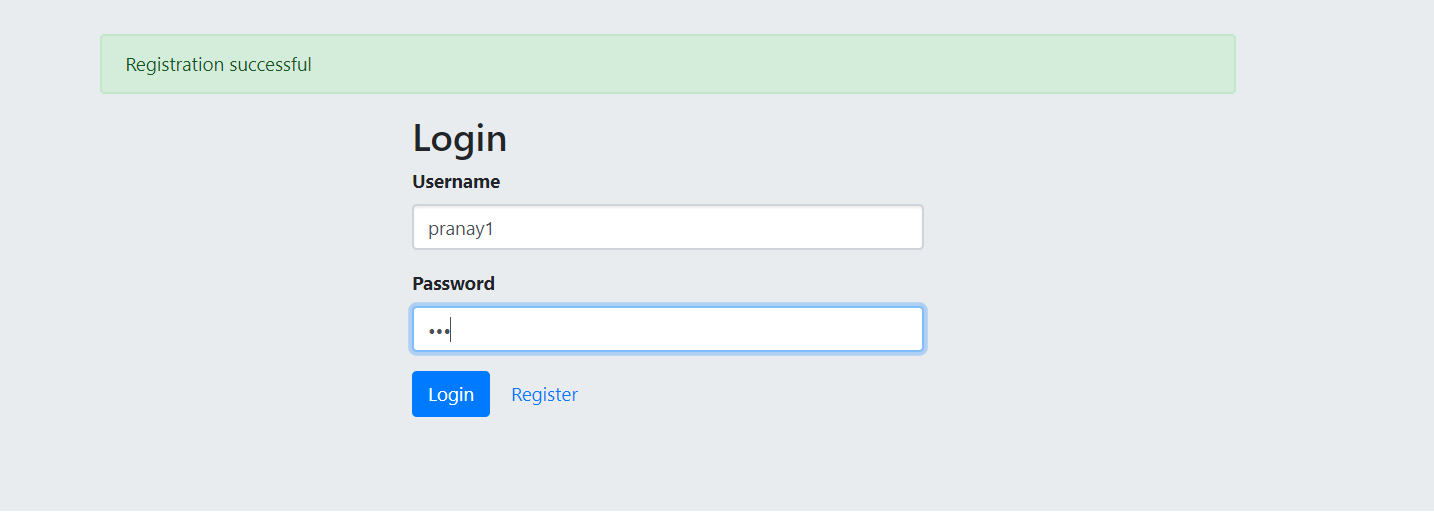
Registration Part:



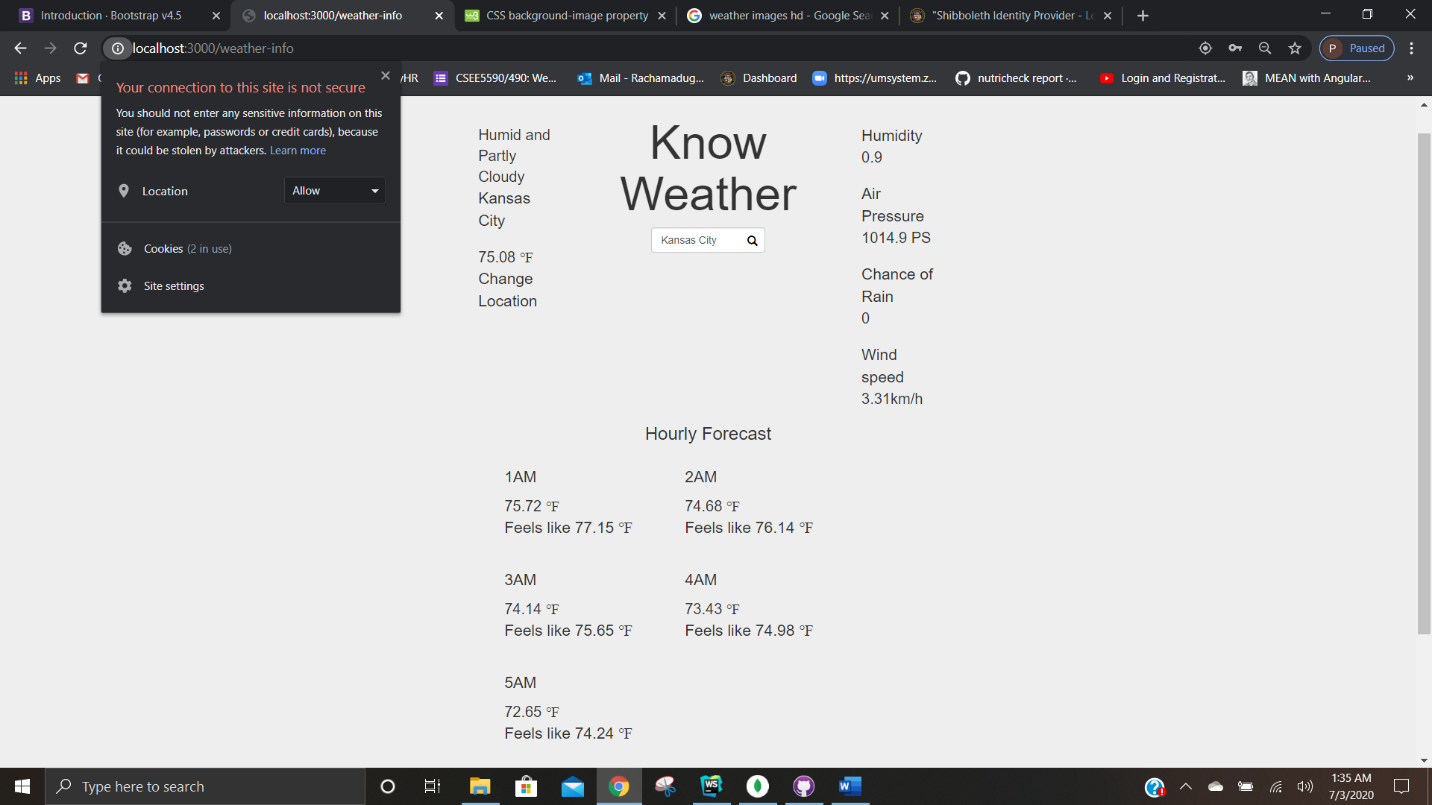
MongoDB:



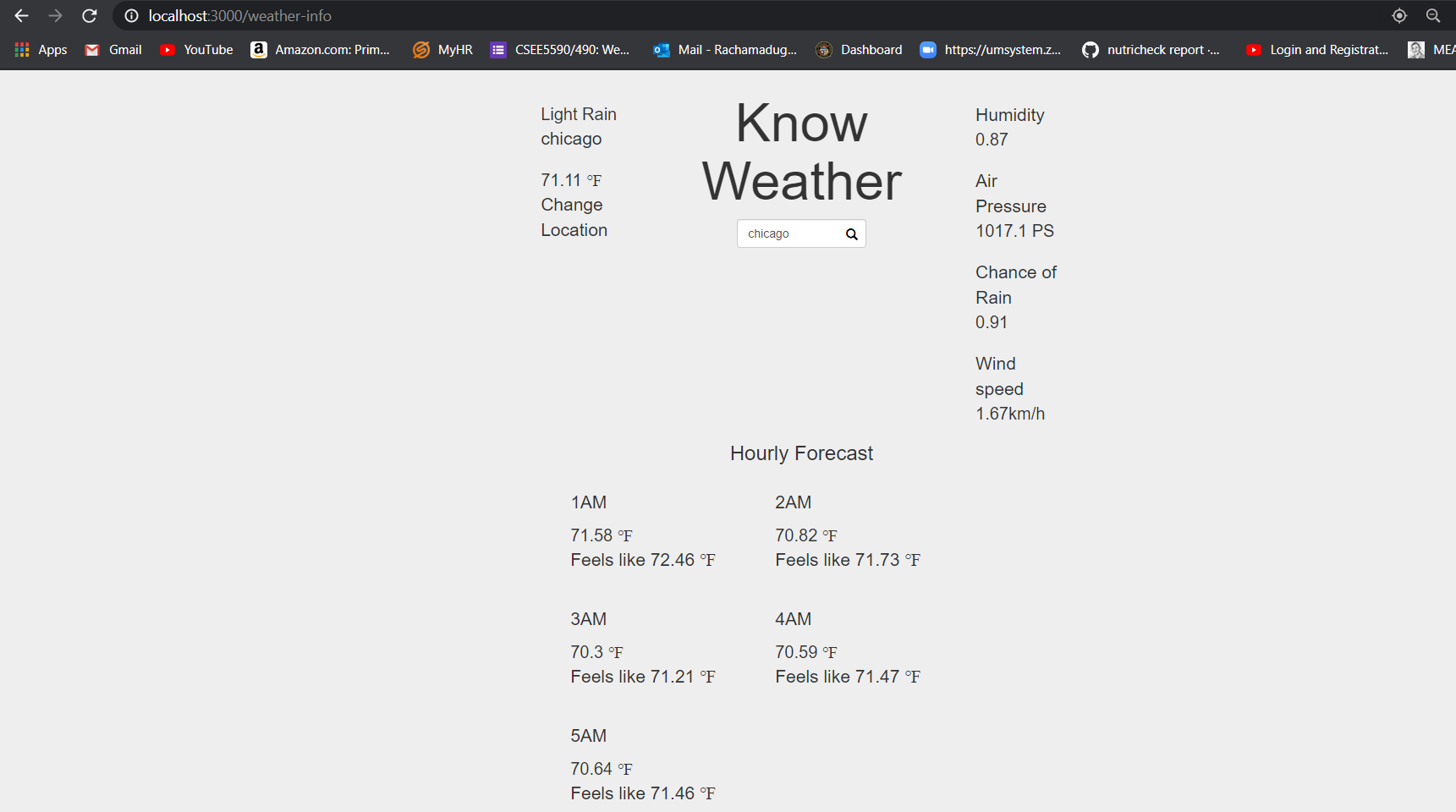
Login:



Weather details using the Geo-location:



Weather details using the search element:



**Issues:**

* Issues in understanding the routing for the application.
* Difficulty in creating the authentication part of the users.

**Github Link:**

<https://github.com/pranaykrishna1/WeatherApplication>

**References:**

<https://jasonwatmore.com/post/2017/02/22/mean-with-angular-2-user-registration-and-login-example-tutorial>

<https://darksky.net/dev/docs>

<https://docs.mapbox.com/api/>

**Conclusion:**

A weather application in MEAN Stack has been created by using the Darksky and Mapbox API which gives the real-time info about all the necessary weather details.