Cloud Computing

Weather Analysis and Forecasting Of Australian Suburbs

Check Our Application Here

By:

Clarine Anslum s3687114

Shamini Puthoopppallil Baby s3674381

Summary

A wide range of weather apps are currently available. The increasing proliferation of emerging broadcasting technology has a major impact on how weather forecasts are transmitted and used by people. In the last few years, there has been a huge change in how people get weather information and a large percentage of the population now gets weather forecasts on their mobile phones, tablets and personal computers. Portability, continuous communication and geo-location allow location-specific and time-sensitive weather forecasts. This project focuses on a web interface where user interacts to check the weather on a particular time for a given suburb in Australia and to see the prediction for the same on five consecutive days. In addition to the weather details, “Open weather” integrates the location specific aspects of the suburbs with the help of popular Google map application.

Links:

Public Link to App	https://storage.googleapis.com/weatherapp/index.html
Github	github:OpenWeather
Google Map API	https://developers.google.com
Open Weather Application	Application:OpenWeather

Table of Contents

List of Figures

Introduction

Meteorological forecasting in general involves gathering observations about past or present conditions that define and influence weather, and using those observations in some way to predict what the meteorological conditions will be at some future time.

In this fast-moving world, people are busy with their own schedule and always prefer to search for information in one step. There are weather channels and webpages such as BOM, weatherzone, accuweather etc. But this weather page would help a person to check and forecast the condition of weather while providing the attractions close to that area. They will choose or benefit from accessing/visiting the places as per weather condition. This package is especially designed for travellers who love to visit places often. They can check the current environmental conditions, upcoming weather changes and geographical aspects of the destination for planning a trip.

Related Work

There are tremendous weather prediction applications are available in market for android and iOs Platforms[6].

Some similar soft wares are listing as part of background analysis.

The Weather Channel

The Weather Channel's free, ad-supported weather app is available on Android and iOS. The TWC app provides you with a wealth of meteorological data such as temperature, wind and visibility on an hourly or daily basis, with an extended 10 day forecast available. Interactive maps can display the latest Doppler radar data to show rainfall and weather events and severe weather alerts. The app also includes social sharing features for users to upload images, tweets, videos and photos.

Emergency: Alerts

it's an app you hope you never have to use, but when disaster strikes, you'll be glad you have Emergency: Alerts from the American Red Cross (Android, iOS) on your iPhone or Android device. The app pays attention to when severe weather conditions strike — everything from hurricanes and floods to extreme heat and winter storms. Not only will Emergency monitor where you live, you can also enter cities and people that are important to you. The app features customizable alerts and provides a map with shelter information in the event of an emergency. You can use the app to make sure you're ready for a potential storm by scanning through the disaster preparedness info tucked within Emergency.

NOAA Radar Pro

Weather can change in an instant, so it's handy to be alerted when something severe is headed your way. Apalon's NOAA Radar Pro for iOS promises real-time animated radar images on an interactive app map, along with the kind of detailed weather information you'd expect from a mobile app. But a noteworthy feature is the app's ability to send notifications whenever a weather alert is issued, so that you're never caught unaware by snow storms, flood warnings or other dangerous conditions. The iOS version includes Apple Watch compatibility, so those alerts are delivered to your wrist.

WeatherBug

If there's a developing weather situation you need to track, WeatherBug (iOS) can help you out. (The app is also available on Android, where it's called Weather by WeatherBug.) The free download not only offers Doppler radar for North America, but can provide hurricane forecasts, Spark lightning alerts, air quality info and pollen count data on top of the usual array of weather information. Since it pulls data from weather services, satellites and tracking stations, WeatherBug can provide hyper local real-time conditions with current, hourly and 10-day forecasts.

Software Design/Architecture

High level architecture

A highlevel overview of the architecture for the project is shown below.

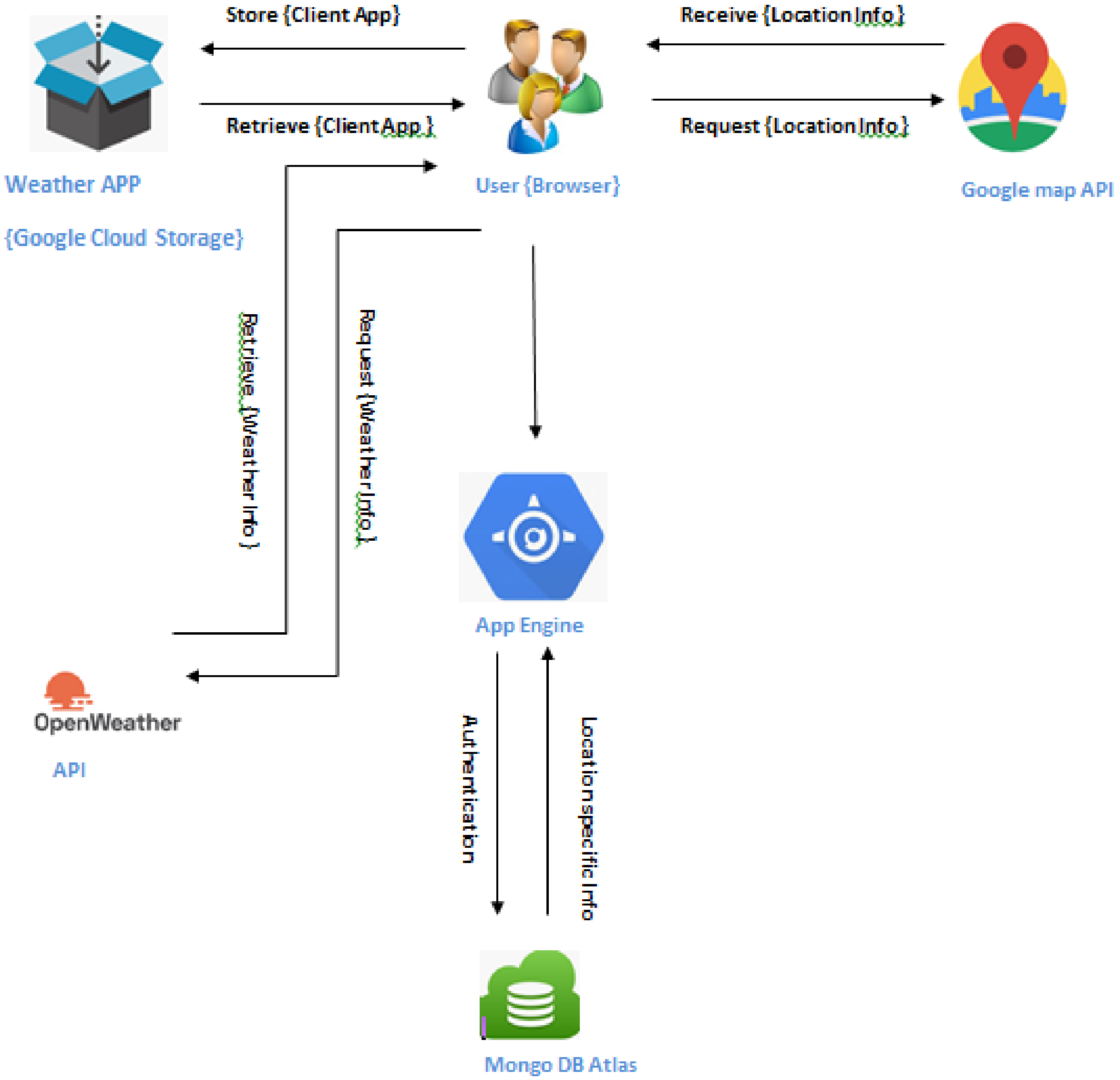


Figure 1: High level design of Open Weather Application

Open weather application work with 6 different components as shown in Figure 1.

1. **User**

   User is the main part of Open weather application. This application starts with the user interaction by clicking the open weather application interface.

2. **Google Cloud Storage**

   Google cloud storage is used to store all the design specific codes related to sever and client. Once the developer starts the application, the developer codes related to project is saved to the Google cloud storage publicly.

3. **App Engine**

   App engine is the central core unit of the application. It controls and integrates all the client and server requests in the distributed environment and provide the results to the user on user demand.

4. **Mongo DB**

   Mongo DB from AWS community is used as a database store for storing the authentication information of the users of Open Weather application. It also stores the information about the country and suburbs which is serving as the request information for Google map API.

5. **Google Map API**

   Google map API is a popular API using in the market which is used in this project to display the suburb information in the Google map

6. **Open Weather API**

   The main part of this project is to provide the current weather information and forecast the weather for next 5 days. This facility is adopted from Open weather API, which is available as an open source platform.

Tools

- UI hosted in Google Storage as public resources

  Technology: Angular 8, Bootstrap, HTML, typescript and CSS

- Backend API hosted in Google App Engine

  Technology: Node.js, Express and JavaScript

- Backend Database

  Technology: Mongo Atlas and NoSQL Cloud

The above tools and technologies are collectively used in development phase of this project.

Front end design is done with the help of html and typescript. To incorporate the styling aspects and formatting, CSS is used. Compatibility between platforms are maintained by bootstrap technology.

Since Java script is a strong and powerful tool which can run on client and server machine, we adopted java script during the app deployment and app engine development phases. In addition to JSS, node js and express used to deploy and maintain the services.

To store the authentication details of the new and existing user, Mongo Atlas is used and data retrieval is achieved by NoSQL Cloud. Location specific information such as country name, suburb name, latitude and longitude details are also retrieving to the client machine in the same manner.

Third Party API

Google map API	: https://developers.google.com
Open weather API	: https://openweathermap.org

The existing open source APIs that are used to fetch the weather information and location information which is integrated with the project with the help of API key and displayed the relevant information to the web browser interface.

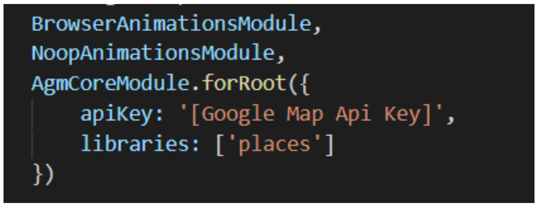
Implementation

Developer Manual

First, the user must clone the project from git hub. There are two folders front end and back end for the source codes. Install nodejs and Angular cli to run the application as these are the pre-requisites of the program.

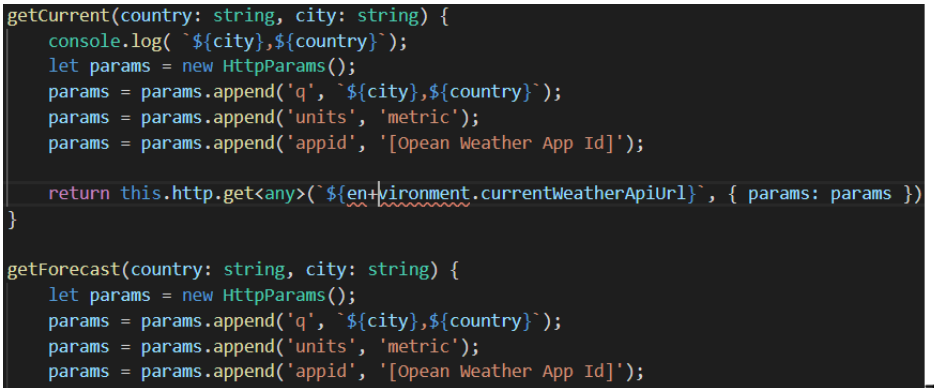
Register with google map and open weather to get the API key and App ID respectively.

Open the **front-end folder** in visual studio code. Apply your google map API key in the app.module.ts file in the src/app folder.



```
BrowserAnimationsModule,
NoopAnimationsModule,
AgmCoreModule.forRoot({
    apiKey: '[Google Map Api Key]',
    libraries: ['places']
})
```

Then, apply the App ID in weather.service.ts file in both functions getcurrent and getforecast as follows. This file is located in src/app/-services



```
getCurrent(country: string, city: string) {
    console.log( `${city},${country}`);
    let params = new HttpParams();
    params = params.append('q', `${city},${country}`);
    params = params.append('units', 'metric');
    params = params.append('appid', '[Opean Weather App Id]');

    return this.http.get<any>(`${environment.currentWeatherApiUrl}`, { params: params })
}

getForecast(country: string, city: string) {
    let params = new HttpParams();
    params = params.append('q', `${city},${country}`);
    params = params.append('units', 'metric');
    params = params.append('appid', '[Opean Weather App Id]');
```

Run the command “npm install” in the frontend folder to download the dependency files.

Run the application through the command prompt in the front-end folder using “ng serve –open”.

Now, the client UI is up and running in the local browser.

For **backend** service, the data base should be setup.

Register free tier in mongoDB cloud (mongoDBAtlas) (link https://cloud.mongodb.com/user#/atlas/login) and create a cluster. Create a user in Database Access to access the database.

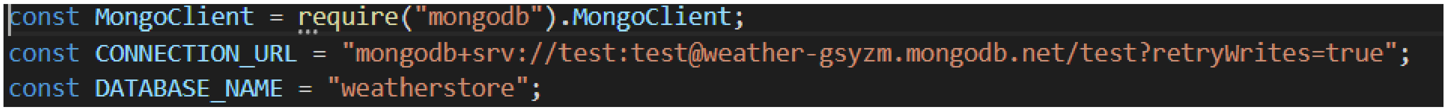
Allow your machine IP in network access section in mongoDB cloud.

Create a new database called weatherstore under cluster section by clicking Collections button. Then, add new collections city, county and user.

Upload documents city.list.json (available in data folder) into city collection (can be downloaded from http://bulk.openweathermap.org/sample/).

Upload county details data_jason.jason (available in data folder) into county collection (can be extracted from https://datahub.io/core/country-list).

Open the backend folder and change the mongoDB connection string in file server.js.



```
const MongoClient = require("mongodb").MongoClient;
const CONNECTION_URL = "mongodb+srv://test:test@weather-gsyzm.mongodb.net/test?retryWrites=true";
const DATABASE_NAME = "weatherstore";
```

How to get connection string

Goto Clusters section in mongoDB and click connect button. Click the second option “Connect your application” and copy the string. Apply the string on the server.js file as above and change the password as well.

To run the dependency, run “npm install” command in the backend folder followed by “npm start” to run the back-end service.

User Manual

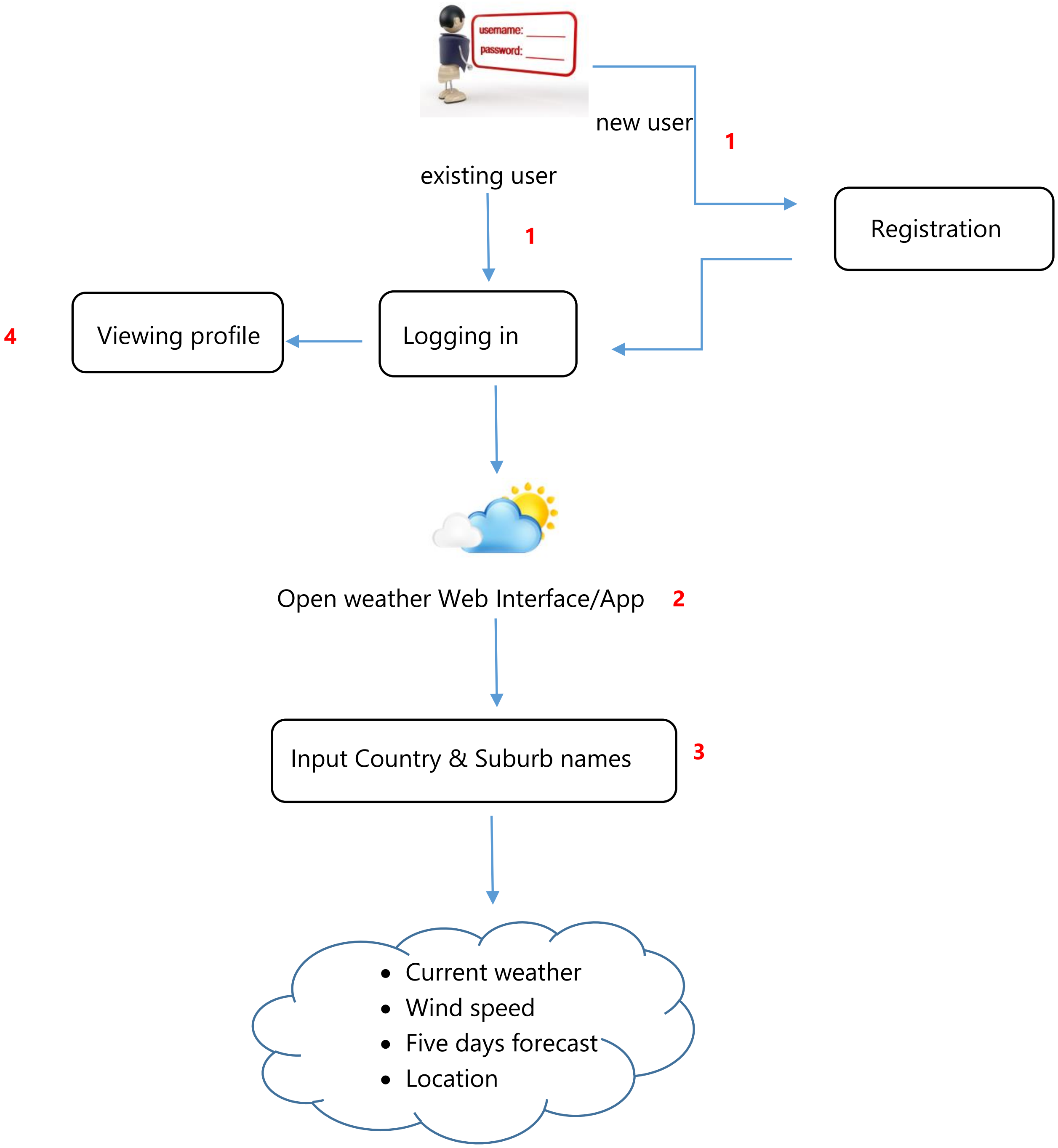


Figure 2:User Interaction of Open Weather Application

User interaction of Open Weather Application is depicted in figure 2

1. Login/Registration

Open weather application starts with a login page as shown below when user opens the interface.

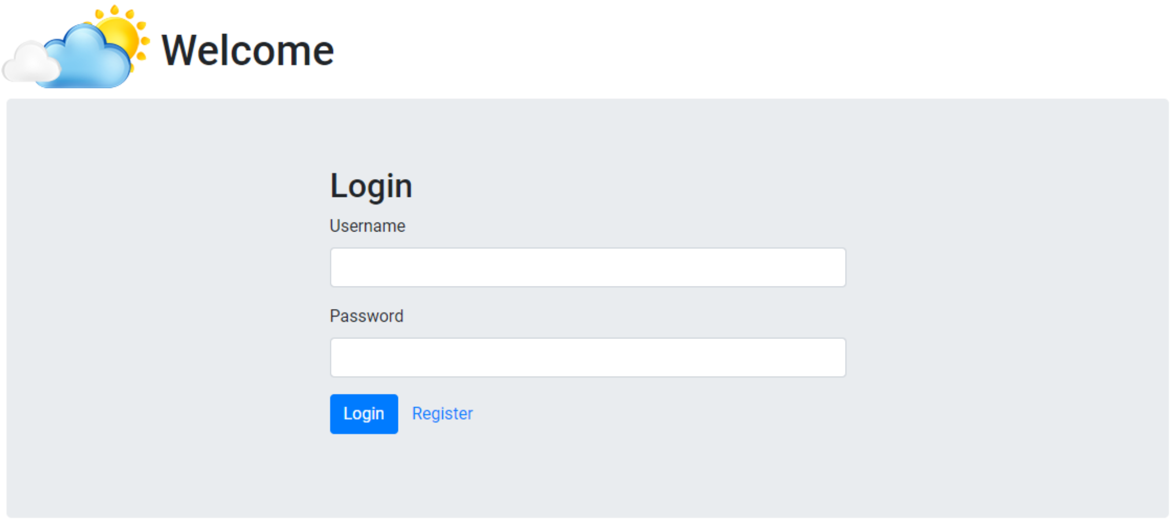
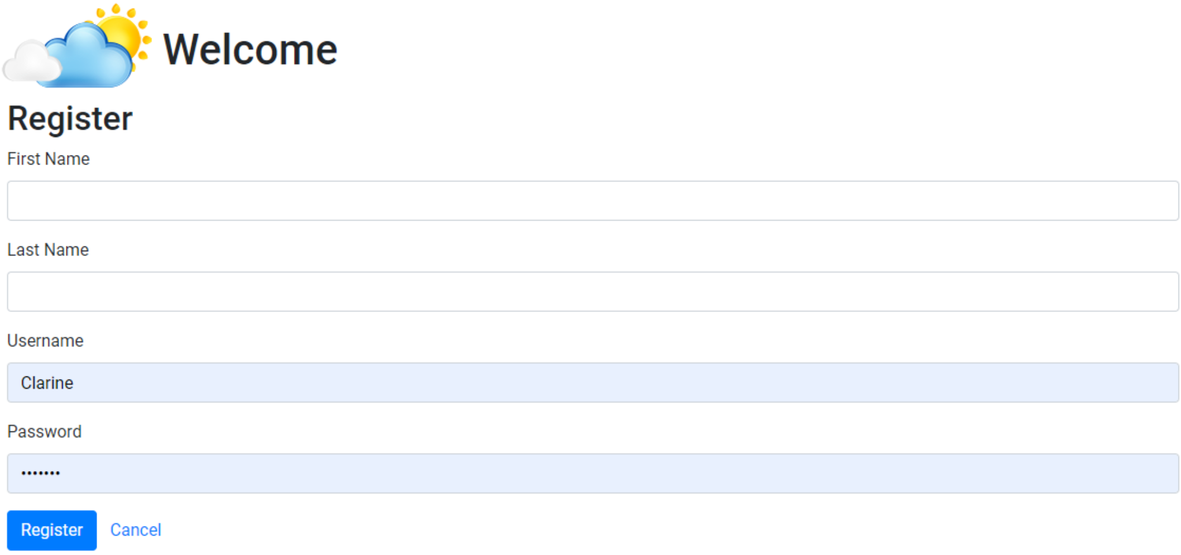


Figure 3:Login Page

.

If the user is already registered with the application, they can login with the existing credentials and that leads to home page of Open Weather Interface.

If a user visiting here for the first time, they have to register with a user name, password and email address as shown below .Once the registration successful, they are redirected to the login page and they can start the application like existing users.



Username

Clarine

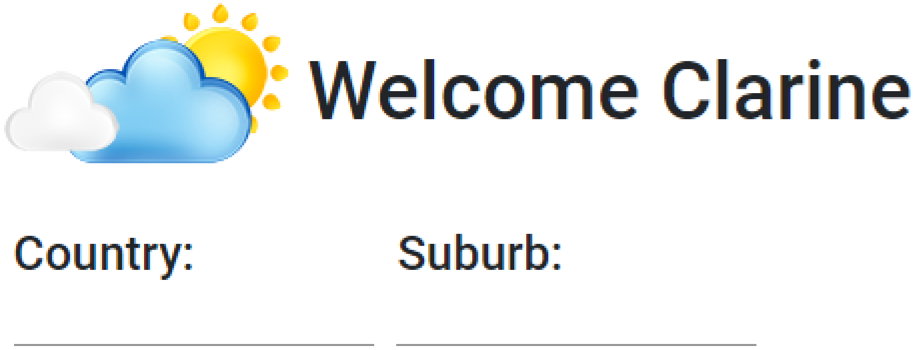
Password

•••••••

Figure 4:Register Page

2. Open Weather Interface/App

User is directed to the home page of Open weather application once they login. From the home page, they can search for a country and suburb.



Welcome Clarine

Country:

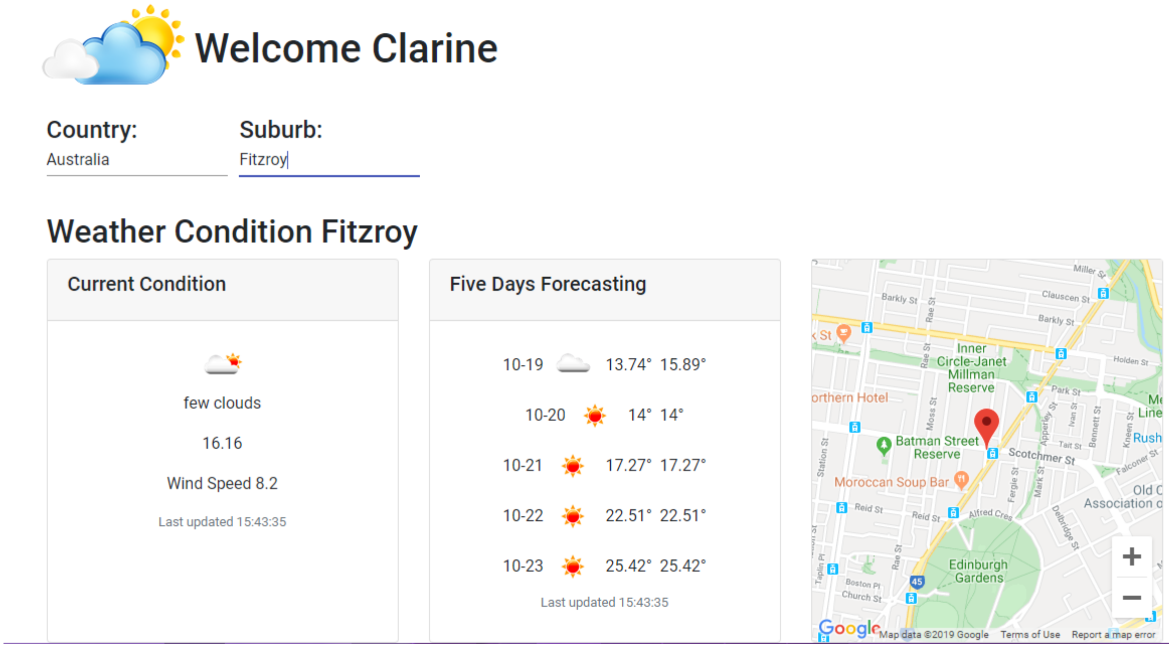
Suburb:

Figure 5:Home Page

3. Open Weather Features

When the user types a country and suburb, Interface provides the following details as shown below.

- Current weather
- Wind speed
- Five days forecast
- Location



Welcome Clarine

Country: Australia

Suburb: Fitzroy

Weather Condition Fitzroy

Current Condition

few clouds

16.16

Wind Speed 8.2

Last updated 15:43:35

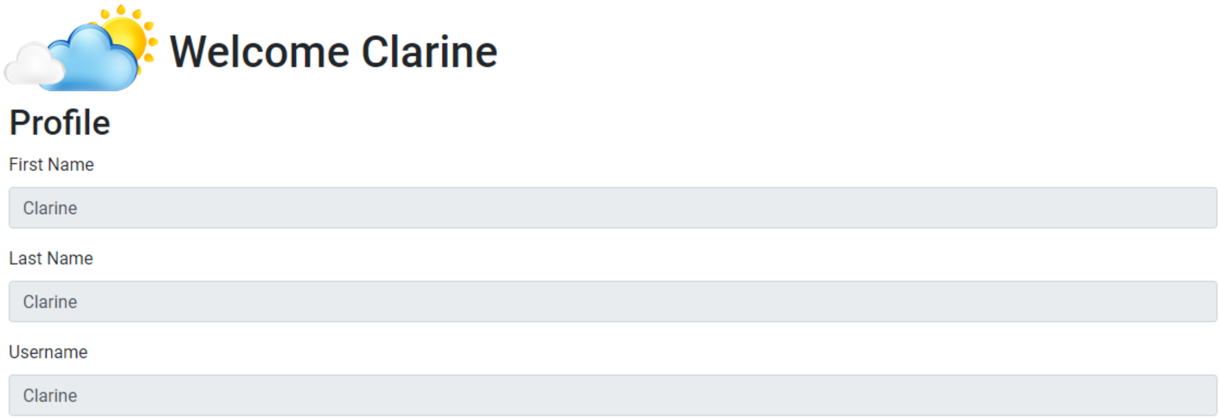
Five Days Forecasting

```
10-19   13.74°  15.89°
10-20   14°  14°
10-21   17.27°  17.27°
10-22   22.51°  22.51°
10-23   25.42°  25.42°
Last updated 15:43:35
```

Figure 6:Open Weather Main Page

4. Profile View

Open weather interface has a tab called “Profile View” which stores the user specific information.



Welcome Clarine

Profile

First Name

Clarine

Last Name

Clarine

Username

Clarine

Figure 7:Profile View

This session can be developed further to add more user specific details for planning a trip in future.

Conclusion and Future Scope

Main focus of this project was to develop an application which is scalable, distributable. We developed an interface which display's the weather information of selected cities in Australia. This project started as a first phase of a scalable project which can be extended further in future. This project was limited to Australian suburbs and the future developer could improve this by adding other suburbs around the world. We would like to develop a small package for tourists who visit famous cities all over the world as future scope. In addition to the location and weather information, personal attractions, short descriptions and a page for trip planning are also in board to consider.

Contribution

Joint Responsibilities:

- Project proposal and Idea generation
- Writing up report
- Integration and collaboration of soft wares

Clarine Anslum (s3687114) - Frontend Developer

Responsibilities

- HTML AND Java Script code to create, store and validate users
- Angular and Typescript code to interface/handle front end
- CSS styling of HTML files
- Hosting all project data on Google App Engine

Shamini Puthooppallil Baby (s3674381) - Backend Developer

Responsibilities:

- Node JS scripts to run the server code and integration of app engine.
- Java script code to fetch and process location information.
- Angular code to store the application file to Google cloud store.
- NoSQL script to fetch the user authentication and location information from Mongo DB.
- HTML files to interface with backend.

References

(1.) Google Cloud. (2019). *@google-cloud/bigquery 1.0.1 » Class: BigQuery | Node.js | Google Cloud*. [online] Available at: https://cloud.google.com/nodejs/docs/reference/bigquery/1.0.x/BigQuery [Accessed 19 Oct. 2019].

(2.) W3schools.com. (2019). *How To Create a Search Bar*. [online] Available at: https://www.w3schools.com/howto/howto_css_searchbar.asp [Accessed 23 May 2019].

(3.) 262588213843476 (2019). *gcp gcloud cheat sheet*. [online] Gist. Available at: https://gist.github.com/pydevops/cffbd3c694d599c6ca18342d3625af97 [Accessed 19 Oct. 2019].

(4.) Babr (2018). *Node refused to apply style because its MIME type ('text/html') is not a supported stylesheet MIME*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/51201869/node-refused-to-apply-style-because-its-mime-type-text-html-is-not-a-support [Accessed 19 Oct. 2019].

(5.) OpenWeatherMap.org (2012). *Current weather and forecast - OpenWeatherMap*. [online] Openweathermap.org. Available at: https://openweathermap.org/ [Accessed 19 Oct. 2019].

(6.) Corpuz, J. (2019). *15 Best Weather Apps*. [online] Tom’s Guide. Available at: https://www.tomsguide.com/us/pictures-story/430-best-weather-apps.html [Accessed 19 Oct. 2019].

(7.) Javascript.com. (2016). *Free JavaScript training, resources and examples for the community*. [online] Available at: https://www.javascript.com/.

(8.) MongoDB. (2019). *The most popular database for modern apps*. [online] Available at: https://www.mongodb.com/.