VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Weather App Using API

Project Report

Open Source Development for Google Applications
(EXC1081)

Name-Akshath Ninjoor
Reg no. – 17BCE0884

Submitted To:-
DSC VIT

ACKNOWLEDGEMENTS

A deepest gratitude and sincere thanks to DSC VIT in helping us complete our Project with several learning outcomes. We feel deeply obliged to thank the SCOPE (School of Computer Science and Engineering) Department and the VIT University for their services rendered and for giving us an opportunity to make such projects along with our studies at the University.

(Akshath Ninjoor)

Reg. No. 17BCE0884

Abstract

The purpose of the project entitled as “Weather Forecast App Using API” is to display the weather of a place using the API. This project will display the current weather and the 5 days of forecast in different activity. It consists of temperature, pressure, humidity, pressure, city, date and time. This will help the user to easily know about the weather of any place just by entering the city name and country code according to ISO 3166.

Introduction

The project “Weather Forecast App Using API” is Android or IOS app which displays the weather information for any place. It has been made using ionic 4 framework. Ionic Framework is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies (HTML, CSS, and JavaScript).

Ionic Framework is focused on the frontend user experience, or UI interaction of an app (controls, interactions, gestures, animations). It’s easy to learn, and integrates nicely with other libraries or frameworks, such as Angular, or can be used standalone without a frontend framework using a simple script include.

Currently, Ionic Framework has official integration with Angular, but support for **Vue** and **React** are in development.

Using API from the weather forecast sites we can get the data related to weather such as Temperature, Humidity, Maximum Temperature, Minimum Temperature for a place that the user has input. This application has different activities to show the various result according to user choice. The User can view the current temperature in one activity and the upcoming days forecast in other activity. The different activity can be managed by changing tabs.

Methodology

This app is made using ionic 4 framework. Ionic framework makes it easy to use components such as tabs and make a beautiful and attractive layout. Ionic Framework is built to be a blank slate that can easily be customized and modified to fit a brand, while still following the standards of the different platforms. Theming Ionic apps is now easier than ever. Because the framework is built with CSS, it comes with pre-baked default styles which are extremely easy to change and modify.

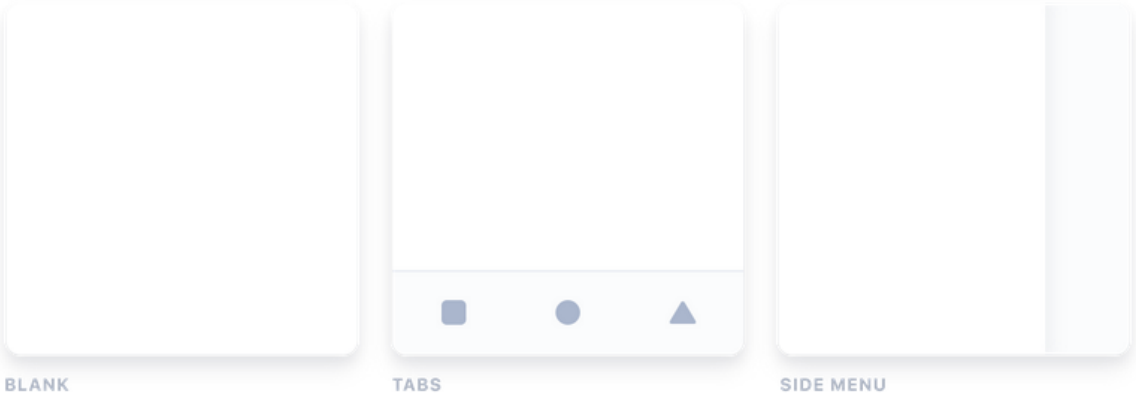
Install ionic CLI globally.

$npm install -g ionic

Then move to the directory in which you want to make the app in the terminal and start the app using the command

$ionic start appname tabs

“tabs” specify the type of app you want to make. Other types are blank and sidemenu. In tabs you can create tabs to navigate.



Go to the app folder and run command ionic serve to watch the development as you code.

We create three tabs one for current weather, one for forecast and other for taking inputs for location.

We also create two services one weather services for using the api key and rrunning the api url. The other services we use for storing the location information as a global data. Thus tab1 and tab2 both can use the location input obtained from tab3.

1.Weather services

```
import { Injectable } from '@angular/core';
import { environment } from '../environments/environment';
import { HttpClient } from '@angular/common/http';
const API_URL1 = environment.apiUrl1;
const API_URL2 = environment.apiUrl2;
const API_KEY = environment.apiKey;
@Injectable({
  providedIn: 'root'
})
export class WeatherService {

  constructor(private http : HttpClient) { }

  getdata(url){
    if(url===undefined||null)
    {
      url="Mumbai";
    }
    return this.http.get(`${API_URL1}${url}&APPID=${API_KEY}`);
  }
  getfordata(cit1,country){
    if(cit1===null||undefined)
    {
      cit1="Mumbai";
    }
    if(country===null||undefined)
    {
      country="IN";
    }
    return this.http.get(`${API_URL2}${cit1},${country}&APPID=${API_KEY}`);
  }
```

```
}
```

We create service by giving command - ionic g services weather. We import HttpClient for using the http.get function and the environment import contains the information about the api key and api url.

```
export const environment = {
  production: false,
  apiUrl1:' https://cors-
anywhere.herokuapp.com/api.openweathermap.org/data/2.5/weather?q=',
  apiUrl2:' https://cors-
anywhere.herokuapp.com/api.openweathermap.org/data/2.5/forecast?q=',

  apiKey: '5cbf4cc923a99b217280a27fc475a97f'
};
```

2.Global Service

```
export class GlobalService {

  constructor() { }
  city:any;
  cid:any;
}
```

Creating a function with global variables for storing location information.

3.Tab 3 (Taking input)

```
<ion-content >
  <ion-grid >
    <ion-row>
      <ion-col width-100>
        <ion-list >
          <form>
            <ion-item>
              <ion-label position="floating">
                Enter City name:
              </ion-label>
              <ion-input placeholder="First letter capital rest all small"
name="city" type="text" [(ngModel)]="inputValue1"></ion-input>
            </ion-item>
            <ion-item>
            <ion-label position="floating">
              Enter Country code:
            </ion-label>
            <ion-input placeholder="IN/US/UK or country code according to  ISO
3166" name="cid" type="text" [(ngModel)]="inputValue2"></ion-input>
          </ion-item>
            <ion-button (click)="onclick(inputValue1,inputValue2)" block
color="secondary" size="small">Search location</ion-button>
          </form>
        </ion-list>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

The above code is html code used for taking the user input the ngModel is used Creates a FormControl instance from a domain model and binds it to a form control element.

```
export class Tab3Page{
name : string;
 constructor(public global : GlobalService) {}

 onclick(city : string , cid : string){
  this.global.city = city;
  this.global.cid = cid;
```

```
  console.log(city);
  console.log(cid);
}
```

This the function used for storing the user input in global variable.

4.Tab1

```
export class Tab1Page implements OnInit {
  data : any;
  constructor(private weatherService: WeatherService, public
global:GlobalService){}
  ngOnInit(){}
  ionViewWillEnter()
  {
    this.weatherService.getdata(this.global.city).subscribe(data=>{
      console.log(data);
      this.data=data;
    })

  }
```

In tab1 we use the following function to get the data form the api url. The function has one argument in which we pass the value of the global variable storing the value of city. We store the data in a variable called data. ionViewWillEnter() used for refreshing the page each time we click the tab icon.

```
<ion-content class="background">

    <ion-list lines="none">
      <ion-list-header color="secondary">

        <ion-label>Current Conditions for {{data.name}}</ion-label>
      </ion-list-header>
```

```
      <ion-item >
          <ion-avatar>
              <img src="/assets/img/temperature.png">
            </ion-avatar>
        <ion-label>  Current Temperature: {{data.main.temp}} K</ion-label>
      </ion-item>
      <ion-item >
          <ion-avatar>
              <img src="/assets/img/wind.png">
            </ion-avatar>
        <ion-label>  Wind: {{data.wind.speed}} m/s {{data.wind.deg}}
degrees</ion-label>
      </ion-item>
      <ion-item >
          <ion-avatar>
              <img src="/assets/img/pressure.png">
            </ion-avatar>
        <ion-label>  Pressure: {{data.main.pressure}} Pa</ion-label>
      </ion-item>
      <ion-item >
          <ion-avatar>
              <img src="/assets/img/humidity.png">
            </ion-avatar>
        <ion-label>  Humidity: {{data.main.humidity}} %</ion-label>
      </ion-item>

    </ion-list>
</ion-content>
```

The above html code is used to display the current weather we use ion list with various ion items displaying each component of weather like temperature, humidity, pressure, wind speed. And ion avatar is used for displaying an avatar for each component of list.

5.Tab2

```
export class Tab2Page {
  data: any;
  page = 1;
  constructor(private weatherService: WeatherService, public global:
GlobalService) {}

  ngOnInit() {}
```

```
  ionViewWillEnter(){

this.weatherService.getfordata(this.global.city,this.global.cid).subscribe(dat
a => {
        console.log(data);
        this.data = data;
      });
  }

}
```

This function is used for getting the forecast data from the url it takes the global variables of city and country code as arguments and stores the data in a variable called data.

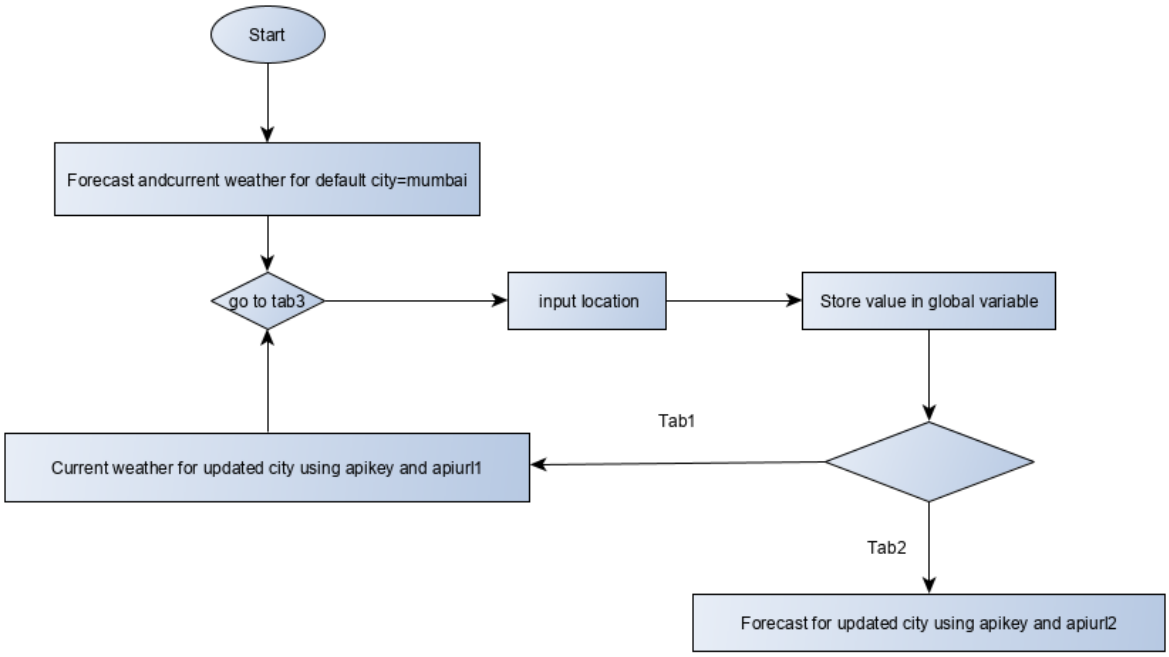
```
<ion-content class="background">
  <ion-card class="welcome-card" *ngFor="let lis of data?.list">
    <ion-item >
        <ion-avatar>
            <img src="/assets/img/date.png">
          </ion-avatar>
      <ion-label>Forecast for: {{lis.dt_txt}}</ion-label>
    </ion-item>

    <ion-item >
        <ion-avatar>
            <img src="/assets/img/temperature.png">
          </ion-avatar>
      <ion-label>  Temperature : {{lis.main.temp}} K</ion-label>
    </ion-item>

    <ion-item >
        <ion-avatar>
            <img src="/assets/img/humidity.png">
          </ion-avatar>
      <ion-label>Humidity : {{lis.main.humidity}} %</ion-label>
    </ion-item>

    <ion-item>
        <ion-avatar>
            <img src="/assets/img/pressure.png">
          </ion-avatar>
      <ion-label>Pressure : {{lis.main.pressure}} Pa</ion-label>
    </ion-item>
  </ion-card>
</ion-content>
```

The above html code uses the card component of the ionic to display the weather components.



Before apps can be deployed to Android simulators and devices, the native project must be configured.

1. **Generate the native project, if it does not already exist.**

For Cordova, run the following:

ionic cordova prepare android

2. **Set the** Package ID**.**

For Cordova, open the config.xml file and modify the id attribute of the root element, <widget>.

The Ionic CLI can build, copy, and deploy Ionic apps to Android simulators and devices with a single command. It can also spin up a development server, like the one used in ionic serve, to provide live-reload functionality.

Run the following to start a long-running CLI process that boots up a live-reload server:

ionic cordova run android -l

Now, when changes are made to the app's source files, web assets are rebuilt and the changes are reflected on the simulator or device without having to deploy again.

```
apiUrl1:'
https://corsanywhere.herokuapp.com/api.openweathermap.org/data/2.5/weather
?q=${city}&APPID=${API_KEY}',',
  apiUrl2:'
https://corsanywhere.herokuapp.com/api.openweathermap.org/data/2.5/forecast
?q=${city},${countrycode}&APPID=${API_KEY}',
 apiKey: '5cbf4cc923a99b217280a27fc475a97f'
```

RESULTS:

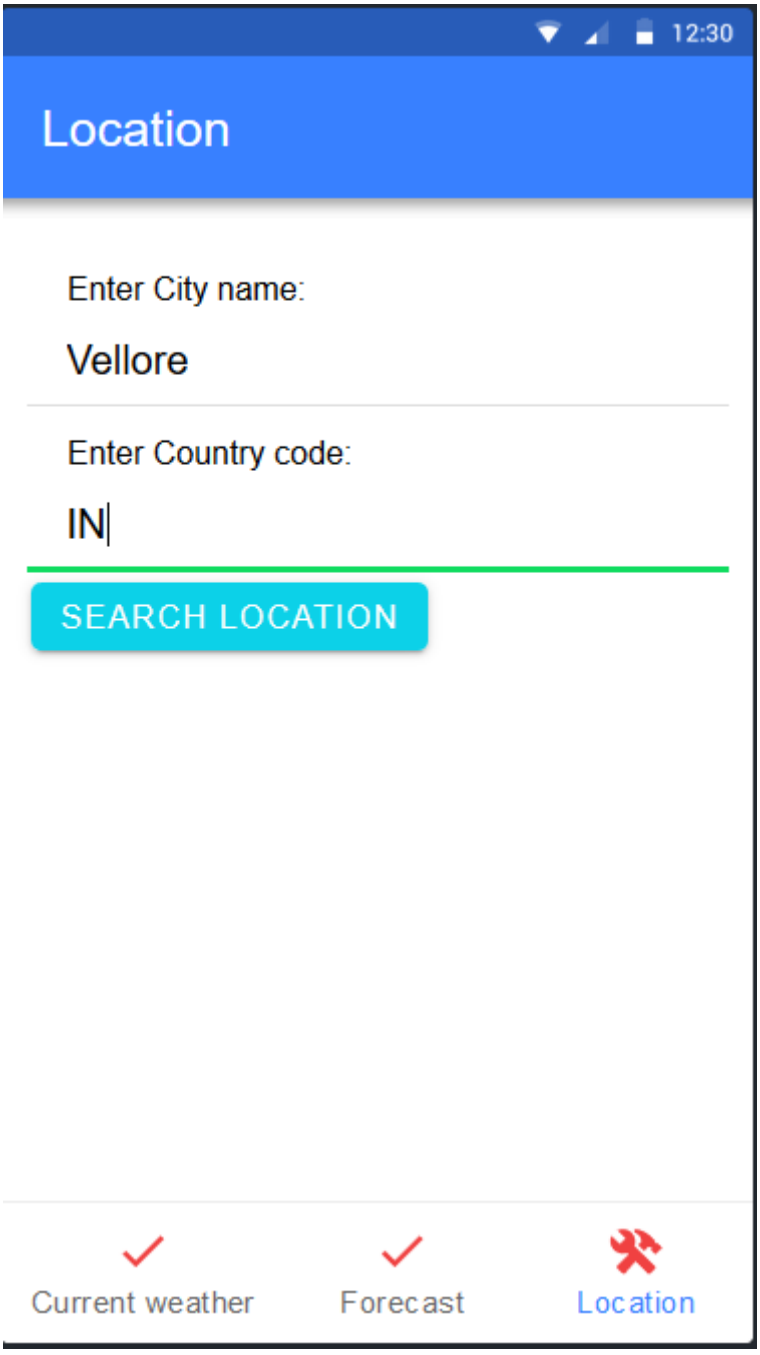

12:30
Location
Enter City name:
Vellore
Enter Country code:
IN
SEARCH LOCATION
Current weather
Forecast
Location

FIG:TAB3

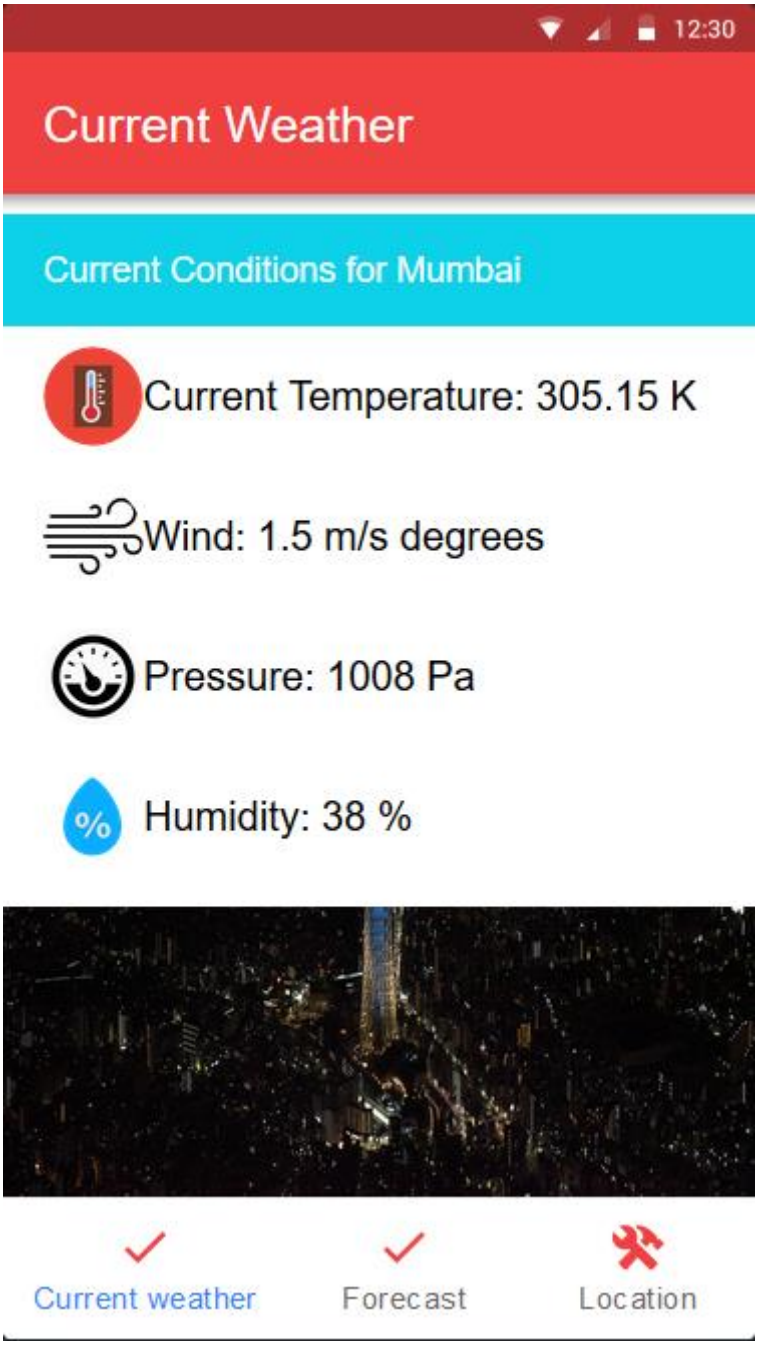

12:30
Current Weather
Current Conditions for Mumbai
Current Temperature: 305.15 K
Wind: 1.5 m/s degrees
Pressure: 1008 Pa
Humidity: 38 %
Current weather
Forecast
Location

FIG: TAB1

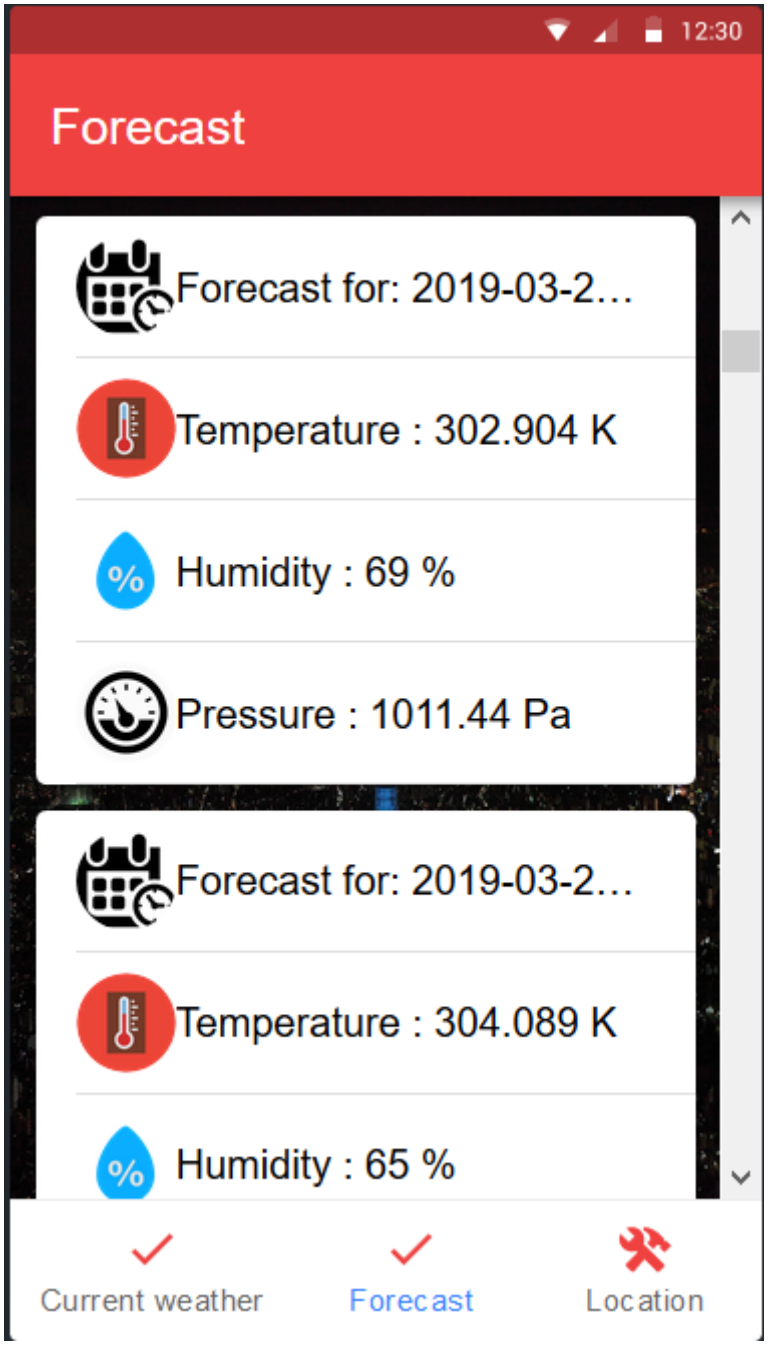

12:30
Forecast
Forecast for: 2019-03-2…
Temperature : 302.904 K
Humidity : 69 %
Pressure : 1011.44 Pa
Forecast for: 2019-03-2…
Temperature : 304.089 K
Humidity : 65 %
Current weather
Forecast
Location

FIG: TAB2

Storing the input in global variable was a bit difficult when using query selector of ts it was giving null value for the input variable thus had to solve it with ngModel. The extracting of the information from url was also giving some errors but solved using appropriate measures. The page was not updating but using ionViewWillEnter() instead of ngOnit() solved the problem.

REFERENCES:

https://openweathermap.org/api

https://ionicframework.com/docs/

https://forum.ionicframework.com/