ClimaRoute

ClimaRoute is a web application which displays the weather conditions along proposed route when a user enters a destination and source for a journey. For this purpose, ClimaRoute uses googlemaps api to get the route along the path, and openweathermap api to display the weather conditions.



Figure 1ClimaRoute interface

# Motivation

Many a times a person makes plans to take a trip from one city to another only for the plan to be ruined by the uninviting weather conditions. ClimaRoute helps a user to make educated decision about trip plans with providing the said user with last minute weather conditions.

# Getting Started

In this section, instructions have been provided so that the source code for this application can be copied to any local machine and get the project up and running.

## Frameworks used

This application is built with the help of Angular and Node.js. So, run this code, you will need to install node and angular on your machine.

**Installing angular**

First step would be to install angular command line interface or Angular CLI. The instructions can be found on [https://cli.angular.io](https://cli.angular.io/)

To install the angular Cli, run the following command in your terminal:

install -g @angular/cli

**Note**: If you are a MacBook user, you might get a “permission denied” error. This error can be removed by typing the keyword “sudo” before your command

sudo install -g @angular/cli

Once you have installed angular, change your directory to the code directory and run ng serve command:

cd /sourcedirectory/src/app

ng serve

Once ng serve has run, a browser tab with URL of [https://localhost:4200](https://localhost:4200/) will open up which will display the webpage of the application. ng serve runs continuously, so whenever you make a change to the code and save it, the browser tab will refresh and display the corresponding changes.

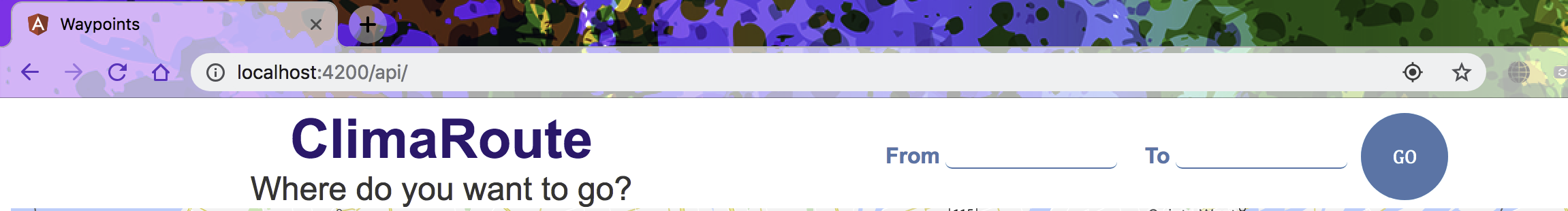


Figure 2 browser tab opens on ng serve

**Installing Node**

Download the node installer from https://nodejs.org/en/download/ for your system and go through the setup process. Once the setup is complete, run this command to verify the installation:

node --version

There are other options to install node through package managers. You can explore these options on this link: <https://nodejs.org/en/download/package-manager/>

Then run the following command to run the backend server on node:

cd sourcedirectory

npm init -y

node app.js

## Packages required

This application requires certain Angular and node packages that must be installed on your local machine for the code to run properly.

**Angular packages**

An information about the installed packages for the code can be found in “package.json” file in src directory. If an error regarding any package occurs, then it is a good idea to check the package.json file to see whether the dependency for that package is installed in your code. If you are unable to find the package, then you may need to reinstall the package to remove and warnings and errors.

@types/googlemaps: The typescript googlemaps package is used in this application to display the map and weather conditions along the path. You can install the package using this command on your machine in your source directory:

npm install @types/googlemaps --save

**Node packages**

The node packages required for this project are request, async, cors, express. These dependencies can be viewed in the package.json file of the node directory.

MySQL: MySQL package helps with implementing database for climaroute

npm install mysql

Request: The request package helps to simplify the http calls and supports the https format. It can be installed by running the following command.

npm install request

async: This utility helps in handling the asynchronous calls. Install it using:

npm install async --save

cors: This package is used to give access to angular for node server so that the frontend can make a request to the node server.

npm install cors --save

Express: Express is the middleware that helps in providing simple interface for creating request endpoints.

npm install express

# Directory Structure

The directory for phase 1 contains two folders called backend and frontend. The frontend directory contains the code for the webpage based on angular. The src folder contains the source code and then in src, the app folder contains the important files for implementation. These files are divided into three types: typescript, html and CSS. Typescript files mainly contain the implementation and class structures whereas html file the webpage code. CSS files contain the styling for the components of html files. There is a file “index.html” which contains the script flag which sources the googlemaps api to display the map on the web page.

The backend directory, on the other hand, contains the code for server based on node and contains two folders maps and weathers. Maps folder file implements the query to googlemaps api and returns the waypoints along the way. Weather file takes a pair of longitude and latitude and returns the weather parameters for that location. The app.js is the entry location and calls the above-mentioned files for the execution of code. Config.js contains the configuration for connecting with database.

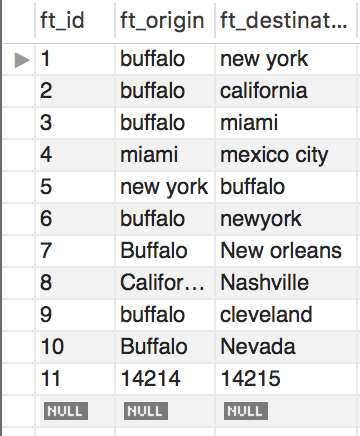


Figure database Table 1

Table 1 is a database which contains the origins and destinations whereas table 2 of the database contained all of the longitudes, latitudes and weather parameters.

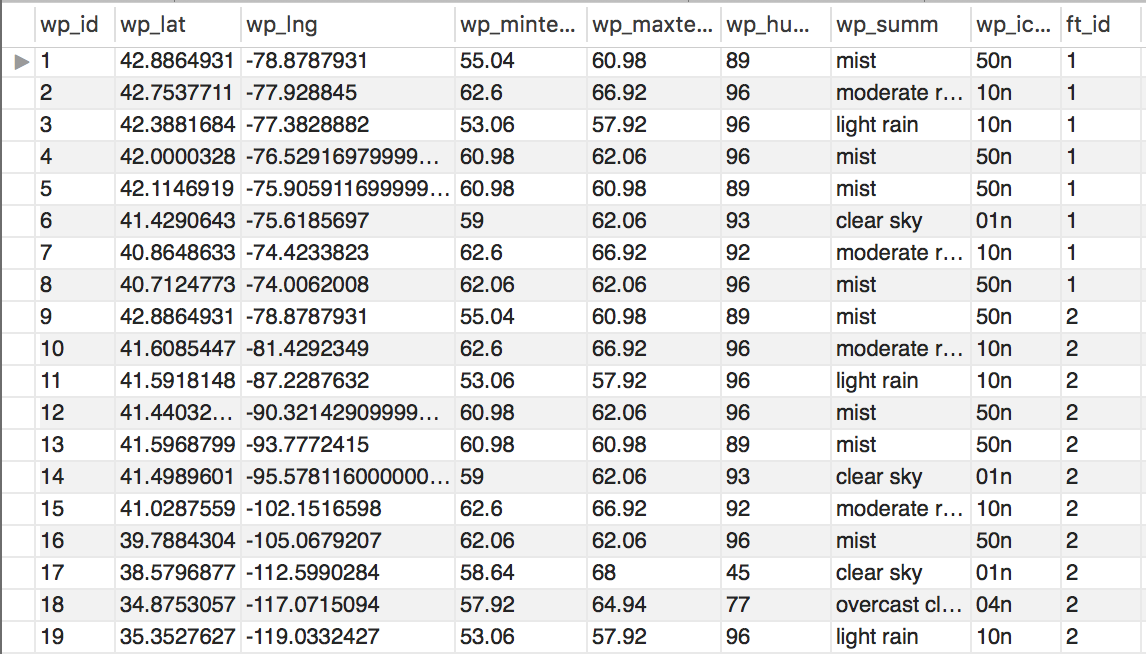


Figure database table 2

API keys

You have to get api keys for googlemaps and openweathermaps and insert that API keys in the code in index.html as well as maps.js and weather.js

Index.html

<script src="https://maps.googleapis.com/maps/api/js?libraries=places&key=API\_KEY"></script>

Googlemaps API in maps.js

const mapsKey= '';

Openweathermap API in weather.js

const weatherKey= '';

Figure 5Directory structure