WEATHER FORECAST SYSTEM

BY REAL TIME API

A Paper
Submitted to the Graduate Faculty
of the

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

By

ARGHYA BANERJEE
With the guidance of
Prof. Shilpa Pareek

In Partial Fulfillment of the Requirements
for the Degree of
B.Tech & Computer Science and Engineering

Major Program:
Software Project

JANUARY , 2022

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Title

WEATHER FORECAST SYSTEM

By
Arghya Banerjee

The Supervisory Committee certifies that this ***disquisition*** complies with UNIVERSITY OF ENGINNERING & MANAGEMENT's regulations and meets the accepted standards for the degree of

B.Tech & Computer Science and Engineering

Cartified that this project work was carried out under Our supervision

WEATHER FORECAST SYSTEM BY REAL TIME API

Approved by Department HOD:

Prof. (Dr.) Mrinal Kanti Sarkar
(HOD of Computer Science & Engineering)

Date

Signature

ABSTRACT

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time.

The purpose of this application is to bring knowledge to students about Weather Forecast and how an interactive weather application can be designed from scratch using client-side languages, such as JavaScript and HTML, combined with the server-side Node.js language through Node Server Faces. The server side, mostly Node,js , contains all the implementation related to fetching the API for data, creating session models for joining different user-interface (UI) pages, server pages, routing paths, handle frameworks etc. It is responsible for taking information from the API and making it available to the UI to show the weather values to the respective Areas according to the input. The client side is responsible for showing the entire user interface, containing the CSS, HTML, JavaScript.

ACKNOWLEDGMENTS

We would like to express our deep sense of gratitude and convey thanks to our Mentor who helped us and supported during the completion of this project and our research paper. First, We would like to express a deep sense of gratitude to our Mentor **Prof. Shilpa Pareek** for helping, guiding, and supporting us throughout our Project. We also convey thanks to our all teachers for helping us from time to time and for being on our project. We acknowledge our department for providing the courses and a great atmosphere that helped complete different chapters of this paper. Last but not least, We would like to thank our family members for their constant and unrelenting support towards our education and for their impartial love for us. We would also like to thank our friends, without whom this project would have been impossible.

TABLE OF CONTENTS

Weather Forecast System

CHAPTER 1. INTRODUCTION

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. People have attempted to predict the weather informally for millennia and formally since the 19th century. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere, land, and ocean and using meteorology to project how the atmosphere will change at a given place.

Once calculated manually based mainly upon changes in barometric pressure, current weather conditions, and sky condition or cloud cover, weather forecasting now relies on computer-based models that take many atmospheric factors into account. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, teleconnections, knowledge of model performance, and knowledge of model biases. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the land, and the ocean, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric and related processes. Hence, forecasts become less accurate as the difference between current time and the time for which the forecast is being made (the *range* of the forecast) increases. The use of ensembles and model consensus help narrow the error and provide confidence level in the forecast.

There is a vast variety of end uses to weather forecasts. Weather warnings are important forecasts because they are used to protect life and property. Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets. Temperature forecasts are used by utility companies to estimate demand over coming days. On an everyday basis, many use weather forecasts to determine what to wear on a given day. Since outdoor activities are severely curtailed by heavy rain, snow and wind chill, forecasts can be used to plan activities around these events, and to plan ahead and survive them.

.

1.1. Motivation

The motivation for designing this weather application came because In daily life we need weather application to monitor our daily weather rather than spending lot of time at TV to know weather details. Further, using weather application, there is also the possibility of designing one's own customized weather application from scratch because custom-designed platforms are expensive. Moreover, I value recent learning about the Node.js and JavaScript programming languages as well as seeing how powerful and dynamic they are when it comes to web designing and applications. Apart from helping computer science students understand the concepts of web-application designing, it would be very easy to incorporate the idea of using programming techniques from the available visuals to understand how a piece of code appears on a user interface. The languages used to build this application are JavaScript, HTML, and Node.js (Express.js)because We found them to be extremely useful while working on the technologies at our workplace.

1.2. Aim of the Software

There are several reasons why weather forecasts are important. They would certainly be missed if they were not there. It is a product of science that impacts the lives of many people. The following is a list of various reasons why weather forecasts are important:

1. Helps people prepare for how to dress (i.e. warm weather, cold weather, windy weather, rainy weather)
2. Helps businesses and people plan for power production and how much power to use (i.e. power companies, where to set thermostat)
3. Helps people prepare if they need to take extra gear to prepare for the weather (i.e. umbrella, rain coat, sun screen)
4. Helps people plan outdoor activities (i.e. to see if rain/storms/cold weather will impact outdoor event)

5. Helps curious people to know what sort of weather can be expected (i.e. a snow on the way, severe storms)
6. Helps businesses plan for transportation hazards that can result from the weather (i.e. fog, snow, ice, storms, clouds as it relates to driving and flying for example)
7. Helps people with health related issues to plan the day (i.e. allergies, asthma, heat stress)
8. Helps businesses and people plan for severe weather and other weather hazards (lightning, hail, tornadoes, hurricanes, ice storms)
9. Helps farmers and gardeners plan for crop irrigation and protection (irrigation scheduling, freeze protection)

1.3. Proposed System

User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable.

1.4. Paper Organization

The rest of the document is divided into three parts: Objectives, Implementation, and Testing. The Objectives chapter lists the need for building the system. It provides use cases to help the business and technical users with their understanding. It also gives a detailed explanation for each use case to help with design and implementation, and outlines the constraints regarding the software. The Implementation chapter contains the detailed design of the system, including the ER Diagram, Data flow Diagram. This chapter also includes a detailed explanation for each component as well as the interaction of the class and its components with each other when carrying out certain tasks, besides software's mock screen shots.

CHAPTER 2. OBJECTIVES

All the steps required in the software-analysis process related to this project (product function, user characteristics, functional and nonfunctional requirements, constraints, assumptions, and dependencies for the Weather Forecast application) are described in the following sections.

2.1. Requirements Analysis

The requirements analysis and gathering processes are critical for the success of any software engineering project. Requirements analysis in software engineering is a process that determines the tasks that are required to determine the needs and conditions to design a new product or to make modifications in any existing product/application. This process considers all the stakeholders' conflicting requirements, and analyzes the documentation and validation of the system. The requirements should be actionable, measurable, testable, and related to the defined needs of the system design. From the software-engineering perspective, requirements analysis is a three-step process.

1. Requirements Elicitation: Elicitation of requirements, also known as requirements gathering, includes the task of identifying various requirement types from stakeholders or from project documentation.
2. Requirements Analysis: Analysis of requirements determines if the gathered requirements are clear, complete, and consistent. The analysis also handles any ambiguous requirements that do not clearly state what needs to be implemented, which could create a loss of resources and time if identified later in the development or testing phase.

Requirement analysis requires identifying the stakeholders and taking their needs into

account to help them understand the implications of designing the new system,

along with

what modules are worth implementing and which ones are more cost efficient, and then to create a software-requirement specification document. To clearly elicit the stakeholders' requirements, different processes, such as developing a scenario or user stories, and identifying the use case which is being used for the project, can be utilized.

Stakeholder analysis says that, to clearly gather the requirements of the project, analysts first need to identify the stakeholders. Stakeholders are people or organizations that have a valid interest or use in the system. The steps to identify the stakeholders are as follows:

- Anyone who operates the system.
- Anyone who benefits from the system
- Anyone who is directly or indirectly involved in purchasing the system
- People or organizations opposed to the system
- Organizations responsible for the system design
- Organizations that regulate the financial or safety aspects of the system

Once the stakeholders are successfully identified, interviews are conducted through different processes; the needs and requirements of the system are identified, and a requirements specification document is prepared. The document is then discussed with the major stakeholders to identify any ambiguity with the requirements and understanding of the system.

3. Requirements Documentation: This step involves documenting the requirements in various forms, including summary lists, natural language documents, visual documents, use cases, user stories, or process specifications. A requirement specification document is categorized in different ways according to the stakeholders' need, helping to create a clear contract between development and business. The following sections include the different

categories of requirements specification document that are essential for designing this application: the functional requirements, constraints, system requirements, etc.

2.2. Product Perspective

The Weather Forecast System application is a web-based system. It can be accessed using Internet Explorer 8.0 and above, Mozilla Firefox 2.0, and Google Chrome.

2.3. User Interface

1. **User Interface**: Users are able to view the home page of application.

Can View

- Real Time Weather Update.
- Current Weather.
- Daily Forecast for 7 days.
- Climatic Forecast.
- Get weather for your Location by only One Click.
- Get updates for any location in the Globe.
- Get Location by any City Name, Area Name.

Get :

- Accurate Temperature Update & weather status
- Changes weather in every seconds according to weather changes
- Get Weather Conditions (Humidity, Pressure, Clouds, Wind, Status etc)

2.4. Hardware Interface

The Weather Forecast application shall provide minimum hardware requirements. The following hardware configurations are required for a PC using the Weather Forecast application:

- Pentium processor
- 10 GB Space
- 1.5 GB RAM

2.5. Software Interface

This section lists the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run the application, the driver for running Node.js web applications, the integrated development environment to develop the application, and the third-party tool used for editing purposes are as follows:

1. Operating System: Windows (Vista/Windows 7) or MAC OS
2. Web Brower: Internet Explorer (8.0 and above), Mozilla Firefox (3.0 and above), or Google Chrome
3. Drivers: Node.js
4. Integrated Development Environment: Microsoft Visual Studio Code
5. Third-Party Tool: Microsoft Word

2.6. Product Function

The weather application would have the following basic functions:

- Real Time Weather Update.
- Current Weather.
- Daily Forecast for 7 days.
- Climatic Forecast.
- Get weather for your Location by only One Click.
- Get updates for any location in the Globe.
- Get Location by any City Name, Area Name
- Accurate Temperature Update & weather status
- Changes weather in every seconds according to weather changes

- Get Weather Conditions (Humidity, Pressure, Clouds, Wind, Status etc)

2.7. User Characteristics

1. **Users:** The users of this application are all customers who would View to test the application. These users are anyone who know-how to browse through a weather application. They must have basic understandings about computers and the internet. The users should be able to perform the following functions using this system:
    - View, browse, search on the home page.
    - View, search inputs in the search bar.
    - Know the correct spelling of all cities.
    - Know the values they get are appropriate or not.

2.8. Constraints

1. Hardware Limitations: The minimum hardware requirement for the system is 1.4 GB of Ram and a 10-GB hard-disc drive.
2. Accessibility: Initially, the software should be available as a desktop application for a small set of users to test.
3. Others: The application should be built using Node and JavaScript inscribed in HTML, and it should, initially, be accessible through the VS CODE IDE and later published on a server.

2.9. Assumptions and Dependencies

The assumptions and dependencies are as follows:

1. Users and the administrator are accustomed to the paper-based system and would require training to use the Weather application.
2. The system is dependent on the availability of a Node driver to run.
3. We assume that system users adhere to the system's minimum software and hardware requirements.
4. This system will use third-party software, and it is assumed that system users are familiar with the software.

2.10. Specific Requirements

This section contains details about all the software that is required for designers to create a system to satisfy the users' requirements and for testers to test the given requirements. This section contains the interface description of each GUI for the different system users. These sections also give descriptions about all the system inputs, all the functions performed by the system, and all the system output (responses).

2.11. Functional Requirements

This section contains the requirements for the Weather System application. The functional requirements, as collected from the users, have been categorized as follows to support the types of user interactions that the system shall have.

1. **Educational Purpose:** The main purpose of this Weather Forecast System application is to teach computer science students the basics of the Node js, JavaScript, CSS and HTML programming languages along with the concepts of web-application designing.
    - **FR01:** The students shall be able to view the source code for the entire application.
    - **FR02:** The students shall be able to, individually, view and understand the code for all pieces on the UI.
    - **FR03:** The students shall be able to debug the application's source code using Chrome Development Tools, which is an online tool to inspect, edit, and monitor HTML, CSS, and JavaScript requests directly on the web page.
2. **User: View :** The users shall be able to see the home page of the Weather application when they first run the program.
    - **FR04:** The users shall be able to view the application's home page.
    - **FR05:** The users shall be able to view Weather Page.
    - **FR06:** The users shall be able search the location.

   FR07: The users shall be able to view more information about the weather of the location.

2.12. Design Constraint

This section lists the design requirements for the Weather application.

1. **DC01:** The user interface (UI) must have specific fonts and font sizes. The system shall match the fonts and font sizes used for all the pages of the application.

2.13. Software System Quality Attribute

1. **Integrity**
   - **QA01:** The authorized user shall be allowed to access Weather Forecast application.
   - **QA02:** Based on the user type, the Weather application shall provide a user-specific interface.
2. **Correctness**
   - **QA03:** The assigned task should be received by the specified user.
3. **Availability**
   - **QA04:** The system shall be made available to the user/administrator year round.
4. **Robustness**
   - **QA05:** The system shall be share location and get data from the server.

CHAPTER 3. IMPLEMENTATION

This chapter includes the detailed design used to build the Weather forecast application. The system's design is used to create the functions and operations of the gathered requirements in detail, including screen layouts, Api Fetch for data , diagrams, and other documentation. The output of this chapter describes the new system which is defined as a collection of modules and subsystems. This design stage takes the initial input requirements that were identified in the approved requirements specification document. For each requirement, there is a set of one or more design elements that are produced using the different prototypes. These design elements describe the desired software features, in detail, including functional hierarchy diagrams, screen layouts, diagrams. The intention of these diagrams is to describe the software in detail so that the system can develop the application with less additional design input. The system's mock screen shots are shown later in this chapter.

3.1. Detailed Scope

This project is supposed to be delivered in three phases, with each phase being an add-on to the project that makes it more usable and acceptable.

- Browse information at the home page
- Search locations
- View more information.
- Get results by users location

3.2. Static Decomposition and Dependency Description

This section contains the system use-case diagram for the application and also has a detailed explanation for each use case in the system.

3.2.1. High-Level Use Case Diagram

The system's use case shows the user a detailed view of the system and how the actors would interact with each other and with the system. The explanation for each use case is then provided below the system use case for the user (Figure), helping the user to understand who are the actors areas as well as giving the description for each use case along with its pre- and post-conditions that should be satisfied once the use case is implemented in the software.

Figure 1 demonstrates the use case of for an user where he or she has access to the application.

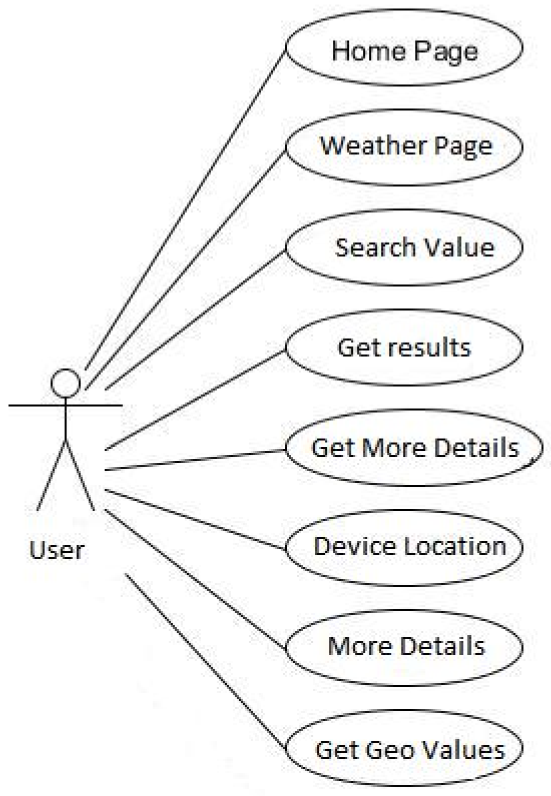


Fig. 1. Weather ForeCast Application System Use-Case Diagram:

Primary Actor: User

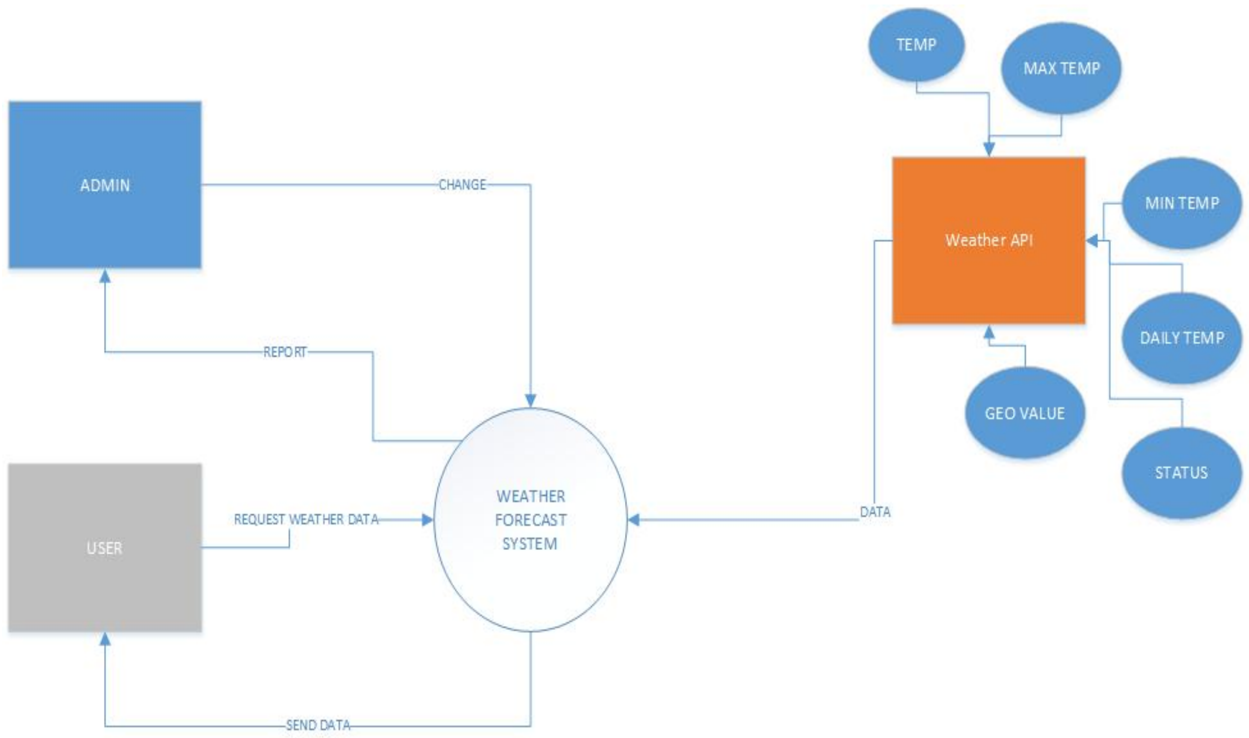
Precondition: Run the application.

Post-condition: The user successfully runs the application and is able to view the app.

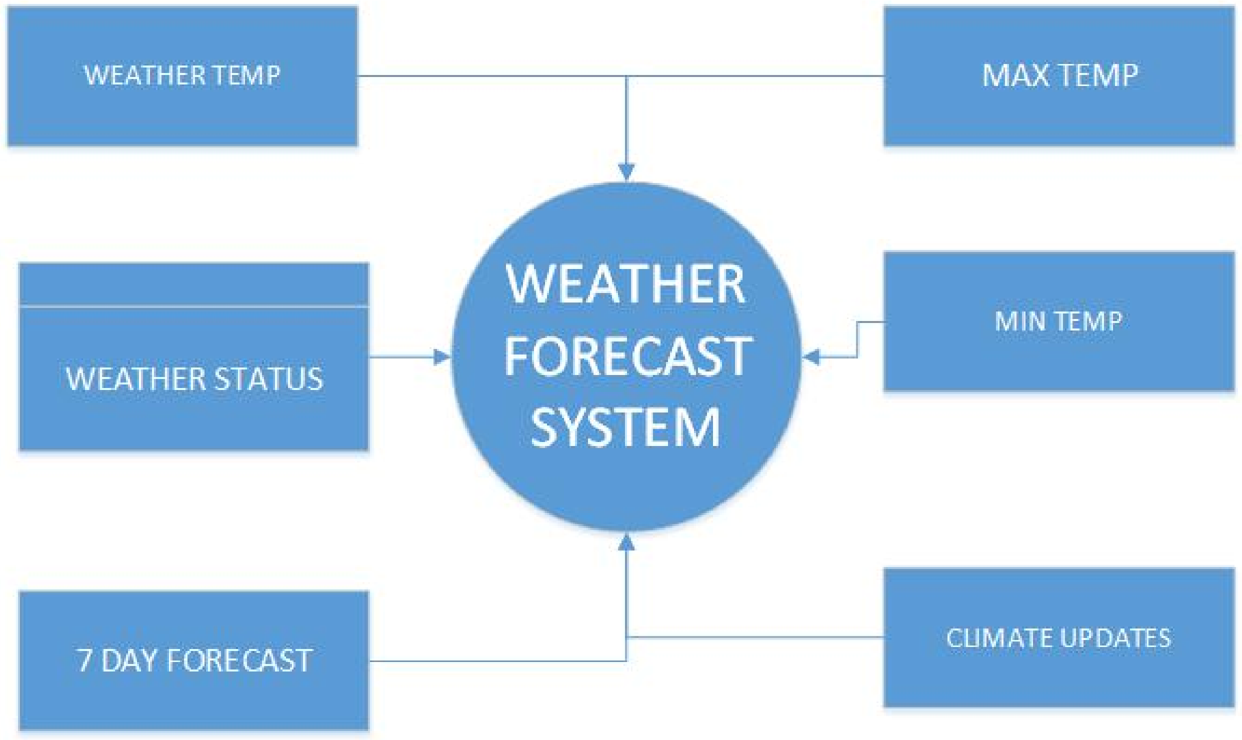
Basic Flow:

- Run the application
- View the home page
- Search the input.

3.2.2. ER Diagram



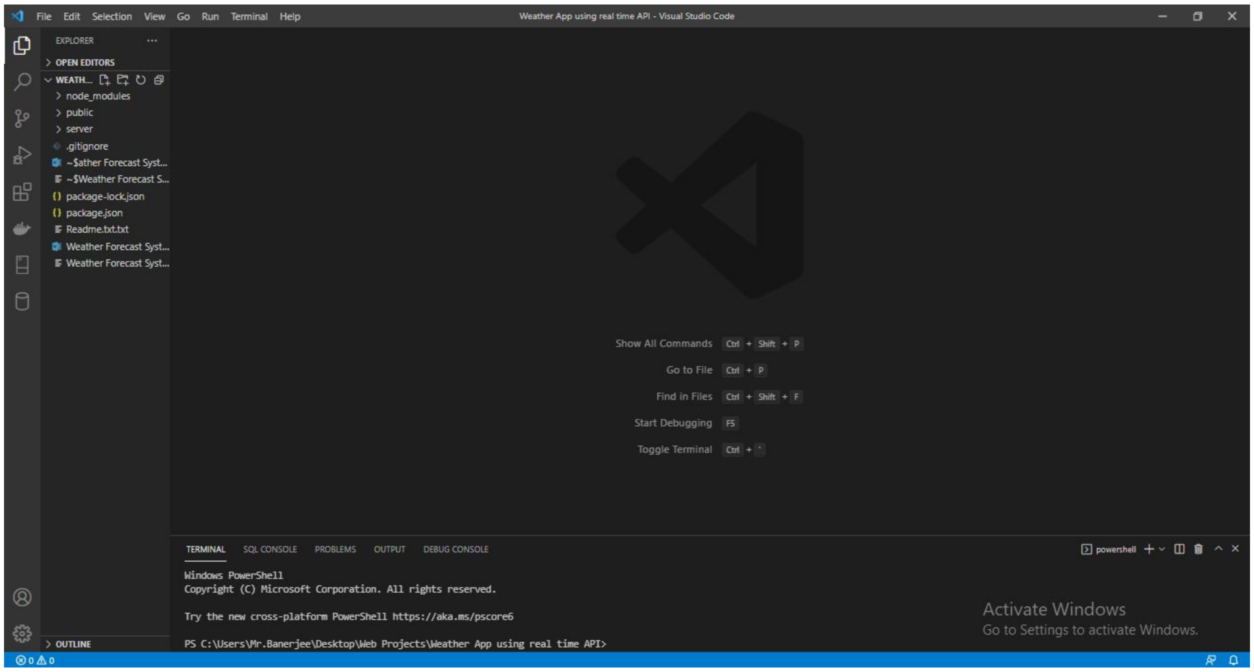
3.2.3. DFD :



3.3. The Fashion Shopping Cart Application Implementation

This section contains the implementation details for different packages and classes of the Weather application. It also contains the coding snippets that help computer science students understand what a particular section of code represents. The following steps are required to run the application successfully:

1. Install the VS Code IDE and Node.js
2. Open powershell, and import the project file into VS Code IDE

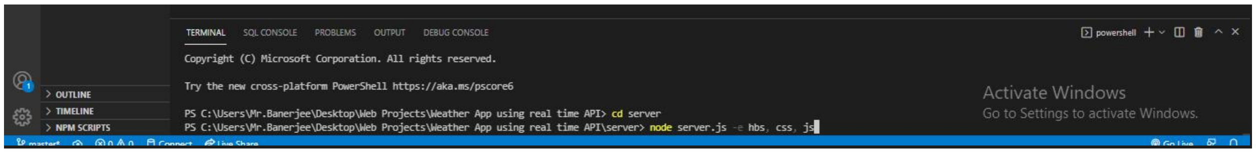


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Mr.Banerjee\Desktop\Web Projects\Weather App using real time API>
```

The Project Manage Center

3. Once the project added use the terminal and rum NPM commands and then install express by using NPM install. It will automatically install all the dependencies automatically .

4) After that in the terminal type cd server to get into server folder then type node server.js -e hbs, css, js to start the server then Visit to the Browser the in the address bar type **Localhost:5500** the hit enter. Then You can finally get the Weather Application.



```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Mr.Banerjee\Desktop\Web Projects\Weather App using real time API> cd server
PS C:\Users\Mr.Banerjee\Desktop\Web Projects\Weather App using real time API\server> node server.js -e hbs, css, js
```

Server :

```
//Express App

//Initialization...
const Express = require("express");
const app = Express();
const path = require("path");
const hbs = require('hbs');
const fetchAPI = require("node-fetch");
const { response } = require("express");
```

```
//Connections / Enviromental variables
const HOST = "Localhost : 127.0.0.1";
const PORT = process.env.PORT || 5500;
const Author = "Arghya Banerjee";
const AppName = "We-ather"
const API_KEY = "86d6b0edaf026c4efc0e9aa5f74d9f75"
```

```
//Joining path for the public folder and partials folder...
const publicFolder = path.join(__dirname, "../public");
const partialFolder = path.join(__dirname, "../server/views/partials")
const serverPath = path.join(__dirname, "../server")
console.log(partialFolder)
```

```
//Built-in middleware to read all file formats...
app.use(Express.static(publicFolder));
app.use(Express.static(serverPath))
```

```
//set handlebars...
app.set('view engine', 'hbs')
```

```
// Set Partials
hbs.registerPartials(partialFolder)
```

```
//API's
```

```
//Server...
app.get("/", (req, res) => {
    res.render('index');
})
app.get('/Weather', async (req, res) => {
    // try{
    //      //Time API :
    //      const timeAPI = await
fetchAPI("https://www.timeapi.io/api/Time/current/coordinate?latitude=18.5196&lon
gitude=73.8553")
    //      const timeAPIJson = await timeAPI.json()
    //      const timeAPIArr = [timeAPIJson]
    //      console.log( "APPI DATA : ", timeAPIArr[0].year)
    //     res.render('weather', {
    //         cityDateDay : timeAPIArr[0].day,
    //         cityDateMonth : timeAPIArr[0].month,
    //         cityDateYear : timeAPIArr[0].year,
    //     });
```

```
    // }catch(error){
    //     res.render('404error')
    // }
```

```
    res.render('weather');
```

```
})
app.get('/about', (req, res) => {
    res.render('about');
})
```

```
//404Error
app.get('*', (req, res) => {
    res.render('404error', {
        pageNotfound: "Opps! Page Not Found",
    });
})
```

```
// app.get('/weather', async (req, res)=>{

// })
```

```
//Server listening...
app.listen(PORT, () => {
    console.log(`Server is running at ${HOST} and Port no. is : ${PORT}
\nCopyright (C) "${AppName}" Corporation by ${Author}. All rights reserved.`);
})
```

Home Page :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>We-ather | Local,National weather forcast</title>
    <link rel="stylesheet" href="./Css/root.css">
    <link rel="stylesheet" href="./Css/style.css">
    <!--Font awesome-->
    <script src="https://kit.fontawesome.com/fd1a296e6c.js"
crossorigin="anonymous"></script>
</head>
```

```
<body>
    <!-- Navigation -->
    {{>nav}} <!--Creating reference of nav from partials-->
```

```
    <!-- Main -->
    <main>
       <section class="container-fluid">
```

```
       <section class="mainSection">
            <div class="welcomGret">
                <p>Welcom to</p>
                <h3 style="font-size: 1.8vw;"><span style="font-size: 3.8vw;
color:var(--primary-color); font-family: 'Satisfy', cursive;">We</span><span
style="color: var(--primary-color) ;font-size: 2.8vw; font-family: 'Satisfy',
cursive">-ather</span> app where you can find National weather forcast</h3>
                <a href="/Weather" class="hmBtn"><span>Check now</span></a>
            </div>
            <div class="welcomGret">
                <img class="illustrations" src="./Assests/Bgs/Snow_Two Color.png"
alt="">
            </div>
        </section>
```

```
        <section class="mainSection">
            <div class="welcomGret">
                <img class="illustrations"
src="./Assests/Bgs/Weather_Isometric.png" alt="">
            </div>
            <div class="welcomGret">
                <p>Developed by Arghya Banerjee</p>
                <h3 style="font-size: 1.8vw;"><span style="font-size: 3vw;
color:var(--primary-color); font-family: 'Satisfy', cursive;">We-ather</span>
offers you to find any city, area's weather information.</h3>
                <a class="hmBtn" href="/about">Know more</a>
            </div>
        </section>
```

```
        <section class="mainSection">
            <div class="welcomGret">
                <p>Reliable with many options</p>
                <h3 style="font-size: 1.8vw;">Get weather report by City/Area
Name, City/Area Postal Code</h3>
                <p>e.g. : [MUMBAI] or [700123]</p>
            </div>
            <div class="welcomGret">
                <img class="illustrations" src="./Assests/Bgs/Weather_Two
Color.png" alt="" srcset="">
            </div>
        </section>
```

```
        <section class="mainSection">
            <div class="welcomGret">
                <img class="illustrations"
src="./Assests/Bgs/undraw_Weather_app_re_kcb1.png" alt="" srcset="">
            </div>
            <div class="welcomGret">
                <p>Available all over the world</p>
                <h3 style="font-size: 1.8vw;">We Provide National and
International both weather updates</h3>
                <p>e.g. : [London] , [Delhi] , [New York] , [Pune]</p>
            </div>
        </section>
```

```
        </section>
    </main>
```

```
    <!-- Footer -->
```

```
</body>
```

```
</html>
```

Weather Page :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Know your local area's Weather forecast</title>
    <link rel="stylesheet" href="./Css/root.css">
    <link rel="stylesheet" href="./Css/weather.css">
    <!--Font awesome-->
    {{!-- {{fontAwesome}} --}}
```

```html
    <script src="https://kit.fontawesome.com/fd1a296e6c.js"
crossorigin="anonymous"></script>
```

```html
</head>
```

```html
<body>
    {{!-- Nav --}}
    {{>nav}}
    {{!-- MAIN --}}
```

```html
    {{!-- Loading Bar --}}
    {{>loadingBar}}
    {{!-- Loading Bar --}}
```

```html
    <main>
        <section class="container-fluid">
            <div id="shortTimeMsg"><span>
                    <p>Please enter the correct City or Area name for accurate
output.</p><i id="cross" class="fas fa-times"></i>
                </span></div>
```

```html
            <div id="cityNameError"><span>
                    <p id="errorContent">Please Enter a valid City or Area name
or Postal Code</p><i id="cross2" class="fas fa-times"></i>
                </span></div>
```

```html
            <div id="geolocation">
                <div>
                <span id="location-txt">Get weather by your location</span><span
id="location-ico" onclick="geoLocation()"><i class="fas fa-map-marker-
alt"></i></span>
                 </div>
            </div>
```

```html
            <div class="searchBar">
                <span id="searchBarBody">
                    <i style="font-size: 1.5rem; text-align:center;" class="fas
fa-cloud-sun"></i>
                    <input name="userData" id="searchBar" type="search"
placeholder="Enter your city or Postcode properly" maxlength="15">
                    <input type="submit" value="Search" id="searchBTN">
                </span>
            </div>
```

```html
            <div class="outputs">
                <div class="outputFeild details">
                    <div class="date_week">
                        <div class="dw"><span id="day">Monda</span></div>
                        <div class="dw"><span id="date-month"><p
id="date"></p>,<p id="month"></p></span></div>
                    </div>
```

```
                    <div class="city_name" id="city_name">
                        <p>City: </p>
                        <span id="cityName">Your city name</span>
                    </div>
```

```
                    <div class="lt-ln">
                        <div id="lt"> Lat: <span id="lat">00.000</span></div>
                        <div id="ln"> Long: <span id="lon">00.000</span></div>
                    </div>
                </div>
```

```
                <div class="outputFeild weatherfeild">
```

```
                    <div id="weather_status"><span
id="wed_status">Rain</span></div>
```

```
                    <div id="weather_info">
                        <div id="temp"><span
id="tempData">0</span><sup>&deg;</sup><span>c</span></div>
                        <div id="weather_status_img">
                            <div><img id="weather-icon"
src="./Assests/weather/64x64/day/113.png" alt=""></div   >
                        </div>
                    </div>
                </div>
                <div class="outputFeild otherdetailsfeeild">
```

```
                    <div id="min-max-temp"><span>Max:&nbsp;<p
id="max"></p>&deg;c</span><span>Min:&nbsp;<P id="min"></p>&deg;c</span></div>
                <div id="othersfeildMain">
                    <div class="tempFeelsLike"><i class="fas fa-temperature-
low"></i><span><p>Feels like: <span id="feelsLike"></span> </p></span></div>
                    <div class="Humidity"><i class="fas fa-
tint"></i><span><p>Humidity: <span id="humidity"></span> </p></span></div>
                    <div class="Pressure"><i class="fas fa-
sort"></i><span><p>Pressure: <span id="press"></span></p></span></div>
                    <div class="Wind"><i class="fas fa-wind"></i><span><p>Wind:
<span id="wind"></span> </p></span></div>
                </div>
                </div>
            </div>
```

```
            <div id="note">
```

```
                <p>Here you can find any city/area/location's weather forecast in
the world for free.</p>
            </div>
            <div id="note2"><p>If you like our services and want to buy me a
coffee the Please <a href="">Click here</a> </p></div>
        </section>
```

```
    </main>
     {{!-- More Details --}}
        <section class="moredetails">
            <div class="MoreDetailscontainrer container-fluid">
```

```
            <div class="moreDetailsbody">
                <div class="moreDetailsHead"><span>More Details</span><span><i
id="dropDown-UpBtn" class="fas fa-chevron-circle-down"></i></span></div>
```

```
            <div id="collaps_details">
                <div class="dateAndTime"><span><p>7 Days Forecust : </p></span>
                <span><p>State Name : </p>&nbsp;<p id="stateName"> </p></span>
                </div>
                <div style="display: flex; justify-content: center ; margin-
bottom: 20px"><hr style="width: 95%;"></div>

                <div id="Ten_day_forecast">
                    <div id="tenDForC-Container">
                        <div><span class="dfd7">Day 1</span><span><span
id="day1">0</span><sup>&deg;</sup><span>c </span> | <span id="day1-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 2</span><span><span
id="day2">0</span><sup>&deg;</sup><span>c </span> | <span id="day2-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 3</span><span><span
id="day3">0</span><sup>&deg;</sup><span>c </span> | <span id="day3-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 4</span><span><span
id="day4">0</span><sup>&deg;</sup><span>c </span> | <span id="day4-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 5</span><span><span
id="day5">0</span><sup>&deg;</sup><span>c </span> | <span id="day5-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 6</span><span><span
id="day6">0</span><sup>&deg;</sup><span>c </span> | <span id="day6-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                        <div><span class="dfd7">Day 7</span><span><span
id="day7">0</span><sup>&deg;</sup><span>c </span> | <span id="day7-L">
30</span><sup>&deg;</sup><span>c</span></span></span></div>
                    </div>
                </div>
                {{!-- <div class="warning-status">
                    <div class="war_stat_container">
                        <h2 class="warning_head">This is warning :</h2>
                        <div id="warning-desc">
                            Lorem ipsum dolor sit amet consectetur adipisicing
elit. Temporibus maiores mollitia, ex esse fugiat illum odit ullam hic
repellendus distinctio dolorem excepturi officia velit laborum quis est in quo
amet sunt iusto impedit. Impedit, quos?
                        </div>
                    </div>
                </div> --}}
            </div>
            </div>
            </div>
        </section>
```

```
        <section id="weather_maps">
            <div class="weatherMaps"></div>
            <div class="weatherMaps"></div>
            <div class="weatherMaps"></div>
        </section>
```

```
<script src="./src/weather.js"></script>
</body>
```

```
</html>
```

Scripting Part :

```
//Selecting elements...
let shortMsgCross = document.getElementById("cross");
let cityNameErrorCross = document.getElementById("cross2")
let shortTimeMsg = document.getElementById("shortTimeMsg")
let searchBar = document.getElementById("searchBar")
let searchBTN = document.getElementById("searchBTN")
let latitude = document.getElementById("lat")
let longitude = document.getElementById("lon")
let weatherStatus = document.getElementById("wed_status")
let temp = document.getElementById("tempData")
let maxTemperature = document.getElementById("max")
let minTemperature = document.getElementById("min")
let tempFeelsLike = document.getElementById("feelsLike")
let Humidity = document.getElementById("humidity")
let pressure = document.getElementById("press")
let wind = document.getElementById("wind")
let weather_icon = document.getElementById("weather-icon")
let day1Temp = document.getElementById("day1")
let day2Temp = document.getElementById("day2")
let day3Temp = document.getElementById("day3")
let day4Temp = document.getElementById("day4")
let day5Temp = document.getElementById("day5")
let day6Temp = document.getElementById("day6")
let day7Temp = document.getElementById("day7")
let day1TempL = document.getElementById("day1-L")
let day2TempL = document.getElementById("day2-L")
let day3TempL = document.getElementById("day3-L")
let day4TempL = document.getElementById("day4-L")
let day5TempL = document.getElementById("day5-L")
let day6TempL = document.getElementById("day6-L")
let day7TempL = document.getElementById("day7-L")
// //Variables...
let date = new Date()
let cityName = ""
const API_KEY = "86d6b0edaf026c4efc0e9aa5f74d9f75"


//Cross BTN function...
shortMsgCross.addEventListener('click', () => {
    shortTimeMsg.style.animationFillMode = "none"
})
cityNameErrorCross.addEventListener("click", () => {
    document.getElementById("cityNameError").style.display = "none"
})
```

```
//Geo Location weather status
let loc = document.getElementById("location-ico");
```

```
function geoLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition, showError);
    } else {
        document.getElementById("location-txt").innerHTML = "Location is not
supported by this device"
    }
}
```

```
async function showPosition(position) {
    document.getElementById("cityNameError").style.display = "none"
    let geoLat = position.coords.latitude;
    let geoLon = position.coords.longitude;
    console.log("Lat and lon : " + geoLat + geoLon)
    try {
        document.getElementById("loadingbar").style.display = "block"
        let geoApiUrl =
`https://api.openweathermap.org/data/2.5/onecall?lat=${geoLat}&lon=${geoLon}&excl
ude=hourly,minutely&units=metric&appid=86d6b0edaf026c4efc0e9aa5f74d9f75`
        let geoAPiFetch = await fetch(geoApiUrl)
        let geoLocationJson = await geoAPiFetch.json();
        let geoLocationOfDevce = [geoLocationJson];
        console.log(geoLocationOfDevce)
```

```
        //City Name :
        let url =
`http://api.openweathermap.org/geo/1.0/reverse?lat=${geoLat}&lon=${geoLon}&limit=
1&appid=${API_KEY}`
        let timeAPI = `http://worldtimeapi.org/api/timezone/Asia/Kolkata`
        let response = await fetch(url)
        let responseJson = await response.json()
        let dataArr = [responseJson]
        // console.log(dataArr)
        cityName = dataArr[0][0].name
        let cityState = dataArr[0][0].state
        document.getElementById('cityName').innerText =
`${cityName},${dataArr[0][0].country}`
        document.getElementById("stateName").innerText = cityState
        let TimeAPiFetcher = await fetch(timeAPI)
        let timeApiJson = await TimeAPiFetcher.json()
        let timeAPIData = [timeApiJson]
        console.log("API", timeAPIData)
```

```
        //Latitude
        latitude.innerText = await geoLat
        latitude.style.color = "white"
        //Longitude
        longitude.innerText = await geoLon
        longitude.style.color = "white"
        //Status
        weatherStatus.innerText = await
geoLocationOfDevce[0].current.weather[0].description
        //Temperature
        let cityTempInFloat = await geoLocationOfDevce[0].current.temp
        let cityTemp = Math.round(cityTempInFloat)
        temp.innerText = cityTemp
        //Status Icon
        //Max And Min temp
        let maxTempFloat = await geoLocationOfDevce[0].daily[0].temp.max
        let minTempFloat = await geoLocationOfDevce[0].daily[0].temp.min
        let maxTemp = Math.round(maxTempFloat)
        let minTemp = Math.round(minTempFloat)
        //console.log("Hiii" + minTempFloat + maxTemp)
        maxTemperature.innerText = maxTemp
        minTemperature.innerText = minTemp
```

```
        //Feels like
        let feelsLikeFloat = await geoLocationOfDevce[0].current.feels_like
        tempFeelsLike.innerText = `${Math.round(feelsLikeFloat)} °c`
```

```
        //Humidity
        let humidityFloat = await geoLocationOfDevce[0].current.humidity
        Humidity.innerText = `${Math.round(humidityFloat)} %`
```

```
        //Pressure
        let pressureData = await geoLocationOfDevce[0].current.pressure
        pressure.innerText = `${pressureData} mb`
```

```
        //Wind
        let windSpeed = await geoLocationOfDevce[0].current.wind_speed
        let windDeg = geoLocationOfDevce[0].current.wind_deg
        wind.innerText = `${windSpeed} km/h (${windDeg}°)`
```

```
        //part 1 : get time status...
        let cityTimeDt = await geoLocationOfDevce[0].current.dt
        let citySunrise = await geoLocationOfDevce[0].current.sunrise
        let citySunset = await geoLocationOfDevce[0].current.sunset
        var currentTimeStatus = ""
        if (cityTimeDt > citySunrise && cityTimeDt < citySunset) {
            //console.log("Day")
            currentTimeStatus = "Day"
        } else if (cityTimeDt > citySunset && cityTimeDt > citySunrise) {
            //console.log("Night")
            currentTimeStatus = "Night"
        } else if (cityTimeDt < citySunrise && cityTimeDt < citySunset) {
            //console.log("Mid Night")
            currentTimeStatus = "Night"
        }
```

```
        //part 2 : Weather status
        var city_weather_status =
geoLocationOfDevce[0].current.weather[0].description
        // console.log(city_weather_status)
        // 1)
Clouds..............................................................................
....................................................
        //Night part
        if (city_weather_status === "overcast clouds" && currentTimeStatus ===
"Night") {
            //console.log("perfect")
            weather_icon.src = "./Assests/weather/64x64/night/122.png"
```

```
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
            document.body.style.background = "url(./Assests/WeatherBgs/overcust-
cloud-1.PNG) fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            //nav
            document.querySelector("#nav").style.backgroundColor = "#00000031"
            //Search
            document.querySelector("#searchBarBody").style.backgroundColor =
"#00000033"
            document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
            //Top
            document.querySelector(".date_week").style.backgroundColor =
"#00000033"
            document.getElementById("weather_status").style.backgroundColor =
"#00000033"
            document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
            //Bottom
            document.querySelector(".details").style.backgroundColor =
"#062c5d00"
            document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
            document.querySelector(".otherdetailsfeeild").style.backgroundColor =
"#062c5d00"
```

```
            //MORE DETAILS
            document.querySelector(".moreDetailsHead").style.backgroundColor =
"#00000033"
```

```
            document.querySelector(".moreDetailsbody").style.backgroundColor =
"#062c5d00"
```

```
        } else if (city_weather_status === "broken clouds" && currentTimeStatus
=== "Night") {
            weather_icon.src = "./Assests/weather/64x64/night/119.png"
            document.body.style.background =
"url(./Assests/WeatherBgs/brokenCloud_night.jpg) fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
            //nav
            document.querySelector("#nav").style.backgroundColor = "#0000001c"
            //Search
            document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
            document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
            //Top
            document.querySelector(".date_week").style.backgroundColor =
"#00000033"
            document.getElementById("weather_status").style.backgroundColor =
"#00000033"
            document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
            //Bottom
            document.querySelector(".details").style.backgroundColor =
"#062c5d00"
            document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
            document.querySelector(".otherdetailsfeeild").style.backgroundColor =
"#062c5d00"
```

```
            //MORE DETAILS
            document.querySelector(".moreDetailsHead").style.backgroundColor =
"#00000033"
            document.querySelector(".moreDetailsbody").style.backgroundColor =
"#062c5d00"
```

```
        } else if (city_weather_status === "scattered clouds" &&
currentTimeStatus === "Night") {
            weather_icon.src = "./Assests/weather/64x64/night/143.png"
            document.body.style.background =
"url(./Assests/WeatherBgs/scatterdCloud_nightt.jpg) fixed no-repeat center
center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
            //nav
            document.querySelector("#nav").style.backgroundColor = "#00000031"
            //Search
            document.querySelector("#searchBarBody").style.backgroundColor =
"#00000033"
            document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
            //Top
            document.querySelector(".date_week").style.backgroundColor =
"#00000033"
            document.getElementById("weather_status").style.backgroundColor =
"#00000033"
            document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
            //Bottom
            document.querySelector(".details").style.backgroundColor =
"#062c5d00"
            document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
            document.querySelector(".otherdetailsfeeild").style.backgroundColor =
"#062c5d00"
```

```
            //MORE DETAILS
            document.querySelector(".moreDetailsHead").style.backgroundColor =
"#00000033"
            document.querySelector(".moreDetailsbody").style.backgroundColor =
"#062c5d00"
```

```
        } else if (city_weather_status === "few clouds" && currentTimeStatus ===
"Night") {
```

```
            weather_icon.src = "./Assests/weather/64x64/night/116.png"
            document.body.style.background =
"url(./Assests/WeatherBgs/Few_clouds.jpg) fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
```

```
            //nav
            document.querySelector("#nav").style.backgroundColor = "#0000001c"
            //Search
            document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
            document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
            //Top
            document.querySelector(".date_week").style.backgroundColor =
"#00000033"
            document.getElementById("weather_status").style.backgroundColor =
"#00000033"
            document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
            //Bottom
            document.querySelector(".details").style.backgroundColor =
"#062c5d00"
            document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
            document.querySelector(".otherdetailsfeeild").style.backgroundColor =
"#062c5d00"
```

```
            //MORE DETAILS
            document.querySelector(".moreDetailsHead").style.backgroundColor =
"#00000033"
            document.querySelector(".moreDetailsbody").style.backgroundColor =
"#062c5d00"
        }
        //Clear sky Mode............
        else if (city_weather_status === "clear sky" && currentTimeStatus ===
"Night") {
            weather_icon.src = "./Assests/weather/64x64/night/113.png"
            document.body.style.background =
"url(./Assests/WeatherBgs/clearSky.jpg) fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
            //nav
            document.querySelector("#nav").style.backgroundColor = "#00000031"
            //Search
            document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
            document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
            //Top
            document.querySelector(".date_week").style.backgroundColor =
"#00000033"
            document.getElementById("weather_status").style.backgroundColor =
"#00000033"
            document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
            //Bottom
            document.querySelector(".details").style.backgroundColor =
"#062c5d00"
            document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
            document.querySelector(".otherdetailsfeeild").style.backgroundColor =
"#062c5d00"
```

```
            //MORE DETAILS
            document.querySelector(".moreDetailsHead").style.backgroundColor =
"#00000033"
            document.querySelector(".moreDetailsbody").style.backgroundColor =
"#062c5d00"
        }
```

```
        //
Haze.................................................................................
.......................................
        else if (city_weather_status === "haze" && currentTimeStatus === "Night")
{
```

```
            weather_icon.src = "./Assests/weather/64x64/night/248.png"
            console.log("Hmmm")
            document.body.style.background = "url(./Assests/WeatherBgs/haze.jpg)
fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
        }
```

```
        //Day Part
        // 1) Clouds
        if (city_weather_status === "overcast clouds" && currentTimeStatus ===
"Day") {
            console.log("perfect")
            weather_icon.src = "./Assests/weather/64x64/day/122.png"
            document.body.style.background = "url(./Assests/WeatherBgs/pexels-
engin-akyurt-6137895.jpg) fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
```

```
        } else if (city_weather_status === "broken clouds" && currentTimeStatus
=== "Day") {
            weather_icon.src = "./Assests/weather/64x64/day/119.png"
            document.body.style.background = "url(./Assests/WeatherBgs/Broken.jpg)
fixed no-repeat center center"
        } else if (city_weather_status === "scattered clouds" &&
currentTimeStatus === "Day") {
            weather_icon.src = "./Assests/weather/64x64/day/143.png"
        } else if (city_weather_status === "few clouds" && currentTimeStatus ===
"Day") {
            weather_icon.src = "./Assests/weather/64x64/day/116.png"
        }
```

```
        // 2)
Rain.............................................................................
..................................................
```

```
        // 3)
Haze.............................................................................
..................................................
        if (city_weather_status === "haze" && currentTimeStatus === "Day") {
            weather_icon.src = "./Assests/weather/64x64/day/248.png"
            document.body.style.background = "url(./Assests/WeatherBgs/haze.jpg)
fixed no-repeat center center"
            document.body.style.backgroundSize = "cover"
            document.querySelector(".details").style.boxShadow = "0px 10px 10px
5px #00000054"
            document.querySelector(".weatherfeild").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".otherdetailsfeeild").style.boxShadow = "0px
10px 10px 5px #00000054"
            document.querySelector("#searchBarBody").style.boxShadow = "0px 10px
10px 5px #00000054"
            document.querySelector(".MoreDetailscontainrer").style.boxShadow =
"0px 10px 10px 5px #00000054"
        }
        // 4)
Fog..............................................................................
..................................................
```

```
        // 2)
Mist.............................................................................
..................................................
```

```
        document.getElementById("loadingbar").style.width = "100%"
        document.getElementById("loadingbar").style.display = "none"
```

```
        // More Details.....
        //     day 1
        day1Temp.innerText = geoLocationOfDevce[0].daily[1].temp.max
        day1TempL.innerText = geoLocationOfDevce[0].daily[1].temp.min
        //     day 2
        day2Temp.innerHTML = geoLocationOfDevce[0].daily[2].temp.max
        day2TempL.innerHTML = geoLocationOfDevce[0].daily[2].temp.min
        //     day 3
        day3Temp.innerText = geoLocationOfDevce[0].daily[3].temp.max
        day3TempL.innerText = geoLocationOfDevce[0].daily[3].temp.min
        //    day 4
        day4Temp.innerText = geoLocationOfDevce[0].daily[4].temp.max
        day4TempL.innerText = geoLocationOfDevce[0].daily[4].temp.min
        //    day 5
        day5Temp.innerText = geoLocationOfDevce[0].daily[5].temp.max
        day5TempL.innerText = geoLocationOfDevce[0].daily[5].temp.min
        //    day 6
        day6Temp.innerText = geoLocationOfDevce[0].daily[6].temp.max
        day6TempL.innerText = geoLocationOfDevce[0].daily[6].temp.min
        //    day 7
        day7Temp.innerText = geoLocationOfDevce[0].daily[7].temp.max
        day7TempL.innerText = geoLocationOfDevce[0].daily[7].temp.min
```

```
    } catch (e) {
        //Validation
        console.log("Error :", e)
        document.getElementById('errorContent').innerText = "Somthing went wront"
        document.getElementById("cityNameError").style.display = "flex"
    }
}
```

```
function showError(error) {
    switch (error.code) {
        case error.PERMISSION_DENIED:
            document.getElementById('errorContent').innerText = "You denied the 
request for Geolocation."
            document.getElementById("cityNameError").style.display = "flex"
            break;
        case error.POSITION_UNAVAILABLE:
            document.getElementById('errorContent').innerText = "Location 
information is unavailable."
            document.getElementById("cityNameError").style.display = "flex"
            break;
        case error.TIMEOUT:
            document.getElementById('errorContent').innerText = "The request to 
get user location timed out."
            document.getElementById("cityNameError").style.display = "flex"
            break;
        case error.UNKNOWN_ERROR:
            document.getElementById('errorContent').innerText = "An unknown error 
occurred."
            document.getElementById("cityNameError").style.display = "flex"
            break;
    }
}
```

```
/*********************************************************************************
*********************************************
 *
*********************************************************************************
******************************************
 ********************************************************************************
**********************************************/
```

```
//Search bar function...
searchBTN.addEventListener("click", async function getWeather() {
    let searchBarValue = searchBar.value
    if (searchBarValue === "") {
        document.getElementById("cityNameError").style.display = "flex"
    } else {
```

```
        try {
            document.getElementById("loadingbar").style.display = "block"
            document.getElementById("cityNameError").style.display = "none"
            //Get weather data...
            let url =
`http://api.openweathermap.org/geo/1.0/direct?q=${searchBarValue}&limit=1&appid=8
6d6b0edaf026c4efc0e9aa5f74d9f75`
            let response = await fetch(url)
            let responseJson = await response.json()
            let dataArr = [responseJson]
            console.log(dataArr)
            let getGeoCorLat = dataArr[0][0].lat
            let getGeoCorLon = dataArr[0][0].lon
            console.log(getGeoCorLat)
            let searchBarGeoApiUrl =
`https://api.openweathermap.org/data/2.5/onecall?lat=${getGeoCorLat}&lon=${getGeo
CorLon}&exclude=hourly,minutely&units=metric&appid=86d6b0edaf026c4efc0e9aa5f74d9f
75`
            let srchBarApiFetcher = await fetch(searchBarGeoApiUrl)
            let srchBarApiJson = await srchBarApiFetcher.json();
            let srchBarApi = [srchBarApiJson]
            //City Name
            cityName = dataArr[0][0].name
            document.getElementById('cityName').innerText =
`${cityName},${dataArr[0][0].country}`
```

```
            //Latitude
            let latitudeCoord = srchBarApi[0].lat
            latitude.innerText = latitudeCoord
            latitude.style.color = "white"
            //Longitude
            let longitudeCoord = srchBarApi[0].lon
            longitude.innerText = longitudeCoord
            longitude.style.color = "white"
            //Status
            weatherStatus.innerText =
srchBarApi[0].current.weather[0].description
            //Temperature
            let cityTempInFloat = srchBarApi[0].current.temp
            let cityTemp = Math.round(cityTempInFloat)
            temp.innerText = cityTemp
            //Status Icon
```

```
            //part 1 : get time status... After Exam I have to Update this...
            let cit_lat = dataArr[0][0].lat
            let city_long = dataArr[0][0].lon
            let ExtendedURL =
`https://api.openweathermap.org/data/2.5/onecall?lat=${cit_lat}&lon=${city_long}&
exclude=hourly,daily&appid=86d6b0edaf026c4efc0e9aa5f74d9f75`
            let responseExtenden = await fetch(ExtendedURL)
            let responseJsonExt = await responseExtenden.json()
            let dataArrExt = [responseJsonExt]
            let cityTimeDt = dataArrExt[0].current.dt
            let citySunrise = dataArrExt[0].current.sunrise
            let citySunset = dataArrExt[0].current.sunset
            let currentTimeStatus = ""
            if (cityTimeDt > citySunrise && cityTimeDt < citySunset) {
                //console.log("Day")
                currentTimeStatus = "Day"
            } else if (cityTimeDt > citySunset && cityTimeDt > citySunrise) {
                //console.log("Night")
                currentTimeStatus = "Night"
            } else if (cityTimeDt < citySunrise && cityTimeDt < citySunset) {
                //console.log("Mid Night")
                currentTimeStatus = "Night"
            }
            console.log("Status ", currentTimeStatus)
            console.log("Time iss ", dataArrExt)
```

```
            //part 2 : Weather status
            let city_weather_status =
srchBarApi[0].current.weather[0].description
            console.log("Main :", city_weather_status)
```

```
            //Night part
            if (city_weather_status === "overcast clouds" && currentTimeStatus
=== "Night") {
                console.log("perfect")
                weather_icon.src = "./Assests/weather/64x64/night/122.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/overcust-cloud-1.PNG) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#00000031"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#00000033"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#00000033"
                document.getElementById("weather_status").style.backgroundColor =
"#00000033"
                document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#00000033"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
```

```
                //Broken clouds mode............
            } else if (city_weather_status === "broken clouds" &&
currentTimeStatus === "Night") {
                weather_icon.src = "./Assests/weather/64x64/night/119.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/brokenCloud_night.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#0000001c"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#00000033"
                document.getElementById("weather_status").style.backgroundColor =
"#00000033"
                document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#00000033"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
```

```
                //Scattered clouds mode.........
            } else if (city_weather_status === "scattered clouds" &&
currentTimeStatus === "Night") {
                weather_icon.src = "./Assests/weather/64x64/night/143.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/scatterdCloud_nightt.jpg) fixed no-repeat center
center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#00000031"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#00000033"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#00000033"
                document.getElementById("weather_status").style.backgroundColor =
"#00000033"
                document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#00000033"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
```

```
                //Few Clouds Mode..................
            } else if (city_weather_status === "few clouds" && currentTimeStatus
=== "Night") {
```

```
                weather_icon.src = "./Assests/weather/64x64/night/116.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/Few_clouds.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#0000001c"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#00000033"
                document.getElementById("weather_status").style.backgroundColor =
"#00000033"
                document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#00000033"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
```

```
            }
```

```
            //Clear sky Mode...........
            else if (city_weather_status === "clear sky" && currentTimeStatus ===
"Night") {
                weather_icon.src = "./Assests/weather/64x64/night/113.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/clearSky.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#00000031"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#0b295838"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#00000033"
                document.getElementById("weather_status").style.backgroundColor =
"#00000033"
                document.getElementById("min-max-temp").style.backgroundColor =
"#00000033"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#00000033"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
            }
            //
Haze...............................................................................
.......................................
            else if (city_weather_status === "haze" && currentTimeStatus ===
"Night" || city_weather_status === "mist" && currentTimeStatus === "Night" ||
city_weather_status === "smoke" && currentTimeStatus === "Night" ||
city_weather_status === "sand" && currentTimeStatus === "Night" ||
city_weather_status === "dust" && currentTimeStatus === "Night") {
                weather_icon.src = "./Assests/weather/64x64/night/248.png"
                console.log("Hmmm")
                document.body.style.background =
"url(./Assests/WeatherBgs/haze.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#35334894"
                document.getElementById("weather_status").style.backgroundColor =
"#35334894"
                document.getElementById("min-max-temp").style.backgroundColor = "
#35334894"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#35334894"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"

            }

            //Fog..............................................................
......................................................
```

```
            else if (city_weather_status === "fog" && currentTimeStatus ===
"Night") {
                weather_icon.src = "./Assests/weather/64x64/night/fog.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/nfog.PNG) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor =
"#00000031"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor =
"#0000007d"
                document.querySelector("#searchBar").style.backgroundColor =
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#0000004f"
                document.getElementById("weather_status").style.backgroundColor =
"#0000004f"
                document.getElementById("min-max-temp").style.backgroundColor =
"#0000004f"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#00000052"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#00000052"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#00000052"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#0000004f"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#00000052"
```

```
            }
            //Day
Part...........................................................................
.........
            // 1) Clouds
            if (city_weather_status === "overcast clouds" && currentTimeStatus
=== "Day") {
                console.log("perfect")
                weather_icon.src = "./Assests/weather/64x64/day/122.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/pexels-engin-akyurt-6137895.jpg) fixed no-repeat center
center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
```

```
            } else if (city_weather_status === "broken clouds" &&
currentTimeStatus === "Day") {
                weather_icon.src = "./Assests/weather/64x64/day/119.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/Broken.jpg) fixed no-repeat center center"
            } else if (city_weather_status === "scattered clouds" &&
currentTimeStatus === "Day") {
                weather_icon.src = "./Assests/weather/64x64/day/143.png"
            } else if (city_weather_status === "few clouds" && currentTimeStatus
=== "Day") {
                weather_icon.src = "./Assests/weather/64x64/day/116.png"
            }
```

```
            // 2) Rain
```

```
            //
Haze...........................................................................
.......................................
            else if (city_weather_status === "haze" && currentTimeStatus ===
"Day" || city_weather_status === "mist" && currentTimeStatus === "Day" ||
city_weather_status === "smoke" && currentTimeStatus === "Day" ||
city_weather_status === "dust" && currentTimeStatus === "Day") {
                weather_icon.src = "./Assests/weather/64x64/night/248.png"
                console.log("Hmmm")
                document.body.style.background =
"url(./Assests/WeatherBgs/haze.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".otherdetailsfeeild").style.boxShadow =
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow
= "0px 10px 10px 5px #00000054"
                //Top
                document.querySelector(".date_week").style.backgroundColor =
"#35334894"
                document.getElementById("weather_status").style.backgroundColor =
"#35334894"
                document.getElementById("min-max-temp").style.backgroundColor = "
#35334894"
                //Bottom
                document.querySelector(".details").style.backgroundColor =
"#062c5d00"
                document.querySelector(".weatherfeild").style.backgroundColor =
"#062c5d00"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#062c5d00"
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor
= "#35334894"
                document.querySelector(".moreDetailsbody").style.backgroundColor
= "#062c5d00"
```

```
            }
            //Fog.................................................................
......................................................
            else if (city_weather_status === "fog" && currentTimeStatus === "Day")
{
                weather_icon.src = "./Assests/weather/64x64/night/fog.png"
                document.body.style.background =
"url(./Assests/WeatherBgs/fog.jpg) fixed no-repeat center center"
                document.body.style.backgroundSize = "cover"
                document.querySelector(".details").style.boxShadow = "0px 10px
10px 5px #00000054"
                document.querySelector(".weatherfeild").style.boxShadow = "0px
10px 10px 5px #00000054"
```

```
                document.querySelector(".otherdetailsfeeild").style.boxShadow = 
"0px 10px 10px 5px #00000054"
                document.querySelector("#searchBarBody").style.boxShadow = "0px 
10px 10px 5px #00000054"
                document.querySelector(".MoreDetailscontainrer").style.boxShadow 
= "0px 10px 10px 5px #00000054"
                //nav
                document.querySelector("#nav").style.backgroundColor = 
"#00000031"
                //Search
                document.querySelector("#searchBarBody").style.backgroundColor = 
"#0000007d"
                document.querySelector("#searchBar").style.backgroundColor = 
"Transparent"
```

```
                //Top
                document.querySelector(".date_week").style.backgroundColor = 
"#0000004f"
                document.getElementById("weather_status").style.backgroundColor =
"#0000004f"
                document.getElementById("min-max-temp").style.backgroundColor = 
"#0000004f"
                //Bottom
                document.querySelector(".details").style.backgroundColor = 
"#00000052"
                document.querySelector(".weatherfeild").style.backgroundColor = 
"#00000052"
                document.querySelector(".otherdetailsfeeild").style.backgroundCol
or = "#00000052"
```

```
                //MORE DETAILS
                document.querySelector(".moreDetailsHead").style.backgroundColor 
= "#0000004f"
                document.querySelector(".moreDetailsbody").style.backgroundColor 
= "#00000052"
```

```
            }
```

```
            document.getElementById("loadingbar").style.width = "100%"
            document.getElementById("loadingbar").style.display = "none"
```

```
            //Max And Min temp
            let maxTempFloat = srchBarApi[0].daily[0].temp.max
            let minTempFloat = srchBarApi[0].daily[0].temp.min
            //console.log(dataArr)
            let maxTemp = Math.round(maxTempFloat)
            let minTemp = Math.round(minTempFloat)
            maxTemperature.innerText = maxTemp
            minTemperature.innerText = minTemp
```

```
            //Feels like
            let feelsLikeFloat = srchBarApi[0].current.feels_like
            tempFeelsLike.innerText = `${Math.round(feelsLikeFloat)} °c`
```

```
            //Humidity
            let humidityFloat = srchBarApi[0].current.humidity
            Humidity.innerText = `${Math.round(humidityFloat)} %`
```

```
            //Pressure
            let pressureData = srchBarApi[0].current.pressure
            pressure.innerText = `${pressureData} mb`
```

```
            //Wind
            let windSpeed = srchBarApi[0].current.wind_speed
            let windDeg = srchBarApi[0].current.wind_deg
            wind.innerText = `${windSpeed} km/h (${windDeg}°)`
```

```
            // More Details.....
            //    day 1
            day1Temp.innerText = srchBarApi[0].daily[1].temp.max
            day1TempL.innerText = srchBarApi[0].daily[1].temp.min
            //    day 2
            day2Temp.innerHTML = srchBarApi[0].daily[2].temp.max
            day2TempL.innerHTML = srchBarApi[0].daily[2].temp.min
            //    day 3
            day3Temp.innerText = srchBarApi[0].daily[3].temp.max
            day3TempL.innerText = srchBarApi[0].daily[3].temp.min
            //    day 4
            day4Temp.innerText = srchBarApi[0].daily[4].temp.max
            day4TempL.innerText = srchBarApi[0].daily[4].temp.min
            //    day 5
            day5Temp.innerText = srchBarApi[0].daily[5].temp.max
            day5TempL.innerText = srchBarApi[0].daily[5].temp.min
            //    day 6
            day6Temp.innerText = srchBarApi[0].daily[6].temp.max
            day6TempL.innerText = srchBarApi[0].daily[6].temp.min
            //    day 7
            day7Temp.innerText = srchBarApi[0].daily[7].temp.max
            day7TempL.innerText = srchBarApi[0].daily[7].temp.min
```

```
            let cityState = dataArr[0][0].state
            document.getElementById("stateName").innerText = cityState
```

```
        } catch (e) {
            //Validation
            console.log("Error :", e)
            document.getElementById('errorContent').innerText = "Please Enter a 
valid City Name or Postalcode"
            document.getElementById("cityNameError").style.display = "flex"
```

```
        }
```

```
    }
```

```
})
```

```
//More Details function....
let h = document.getElementById("")
```

```
document.getElementById("dropDown-UpBtn").addEventListener('click', function 
moreDetails() {
```

```
    if (document.getElementById("dropDown-UpBtn").className == "fas fa-chevron-
circle-up") {
        document.getElementById("dropDown-UpBtn").className = "fas fa-chevron-
circle-down"
        document.getElementById("collaps_details").style.height = "0"
    } else {
        document.getElementById("dropDown-UpBtn").className = "fas fa-chevron-
circle-up"
        document.getElementById("collaps_details").style.height = "100%"
    }
})
```

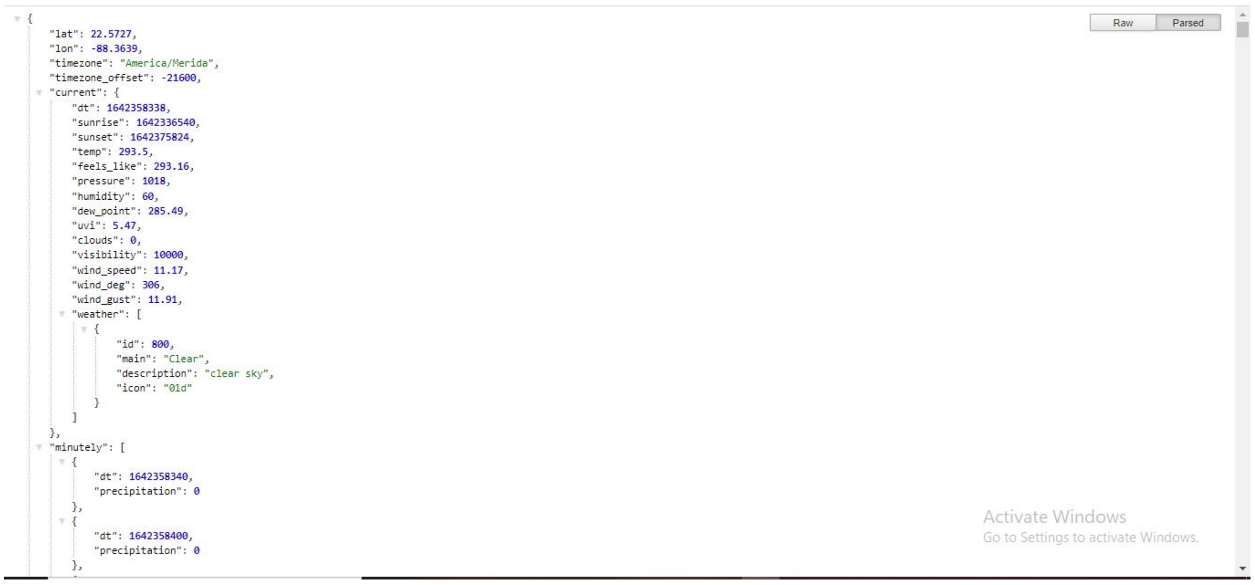
```
//Date and weak...
let gettingMonth = date.getMonth() + 1;
let getDate = date.getDate()
let getWeek = date.getDay()
console.log(getWeek)
let month = ""
let day = ""
//month
switch (gettingMonth) {
    case 1:
        month = "Jan"
        break
    case 2:
        month = "Feb"
        break
    case 3:
        month = "Mar"
        break
    case 4:
        month = "Apr"
        break
    case 5:
        month = "Mey"
        break
    case 6:
        month = "Jun"
        break
    case 7:
        month = "Jul";
        break
    case 8:
        month = "Aug"
        break
    case 9:
        month = "Sep"
        break
    case 10:
        month = "Oct"
        break
    case 11:
        month = "Nov"
        break
    case 12:
        month = "Dec"
        break
    default:
        month = "Sorry there is a technical problem"
}
```

```
//week
```

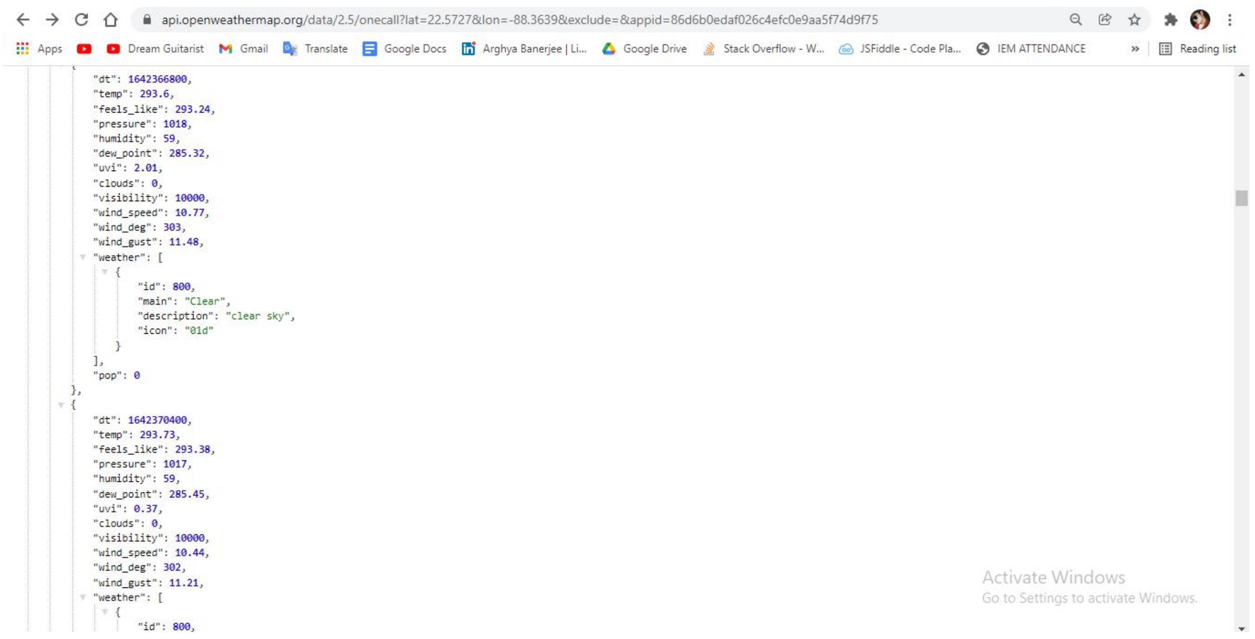
```
switch (getWeek) {
    case 1:
        day = "Monday"
        break
    case 2:
        day = "Tuesday"
        break
    case 3:
        day = "Wednesday"
        break
    case 4:
        day = "Thursday"
        break
    case 5:
        day = "Friday"
        break
    case 6:
        day = "Saturday"
        break
    case 0:
        day = "Sunday"
        break
    default:
        day = "sorry technical issue"
}
```

```
document.getElementById("month").innerText = month;
document.getElementById("date").innerText = getDate
document.getElementById("day").innerText = day
//onsole.log(getWeek)
```

DataBase (API) :



```
{
    "lat": 22.5727,
    "lon": -88.3639,
    "timezone": "America/Merida",
    "timezone_offset": -21600,
    "current": {
        "dt": 1642358338,
        "sunrise": 1642336540,
        "sunset": 1642375824,
        "temp": 293.5,
        "feels_like": 293.16,
        "pressure": 1018,
        "humidity": 60,
        "dew_point": 285.49,
        "uvi": 5.47,
        "clouds": 0,
        "visibility": 10000,
        "wind_speed": 11.17,
        "wind_deg": 306,
        "wind_gust": 11.91,
        "weather": [
            {
                "id": 800,
                "main": "Clear",
                "description": "clear sky",
                "icon": "01d"
            }
        ]
    },
    "minutely": [
        {
            "dt": 1642358340,
            "precipitation": 0
        },
        {
            "dt": 1642358400,
            "precipitation": 0
        },
```



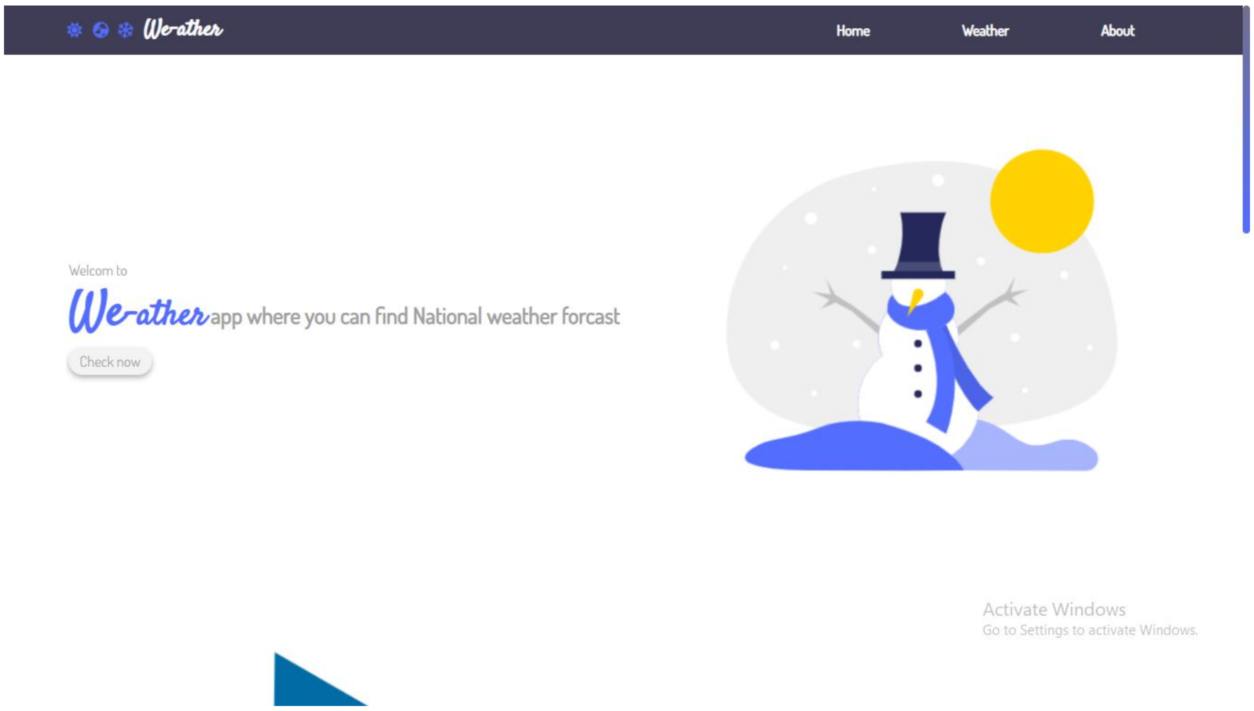
```
        "dt": 1642366800,
        "temp": 293.6,
        "feels_like": 293.24,
        "pressure": 1018,
        "humidity": 59,
        "dew_point": 285.32,
        "uvi": 2.01,
        "clouds": 0,
        "visibility": 10000,
        "wind_speed": 10.77,
        "wind_deg": 303,
        "wind_gust": 11.48,
        "weather": [
            {
                "id": 800,
                "main": "Clear",
                "description": "clear sky",
                "icon": "01d"
            }
        ],
        "pop": 0
    },
    {
        "dt": 1642370400,
        "temp": 293.73,
        "feels_like": 293.38,
        "pressure": 1017,
        "humidity": 59,
        "dew_point": 285.45,
        "uvi": 0.37,
        "clouds": 0,
        "visibility": 10000,
        "wind_speed": 10.44,
        "wind_deg": 302,
        "wind_gust": 11.21,
        "weather": [
            {
                "id": 800,
```

3.4. The Fashion Shopping Cart Application Interface

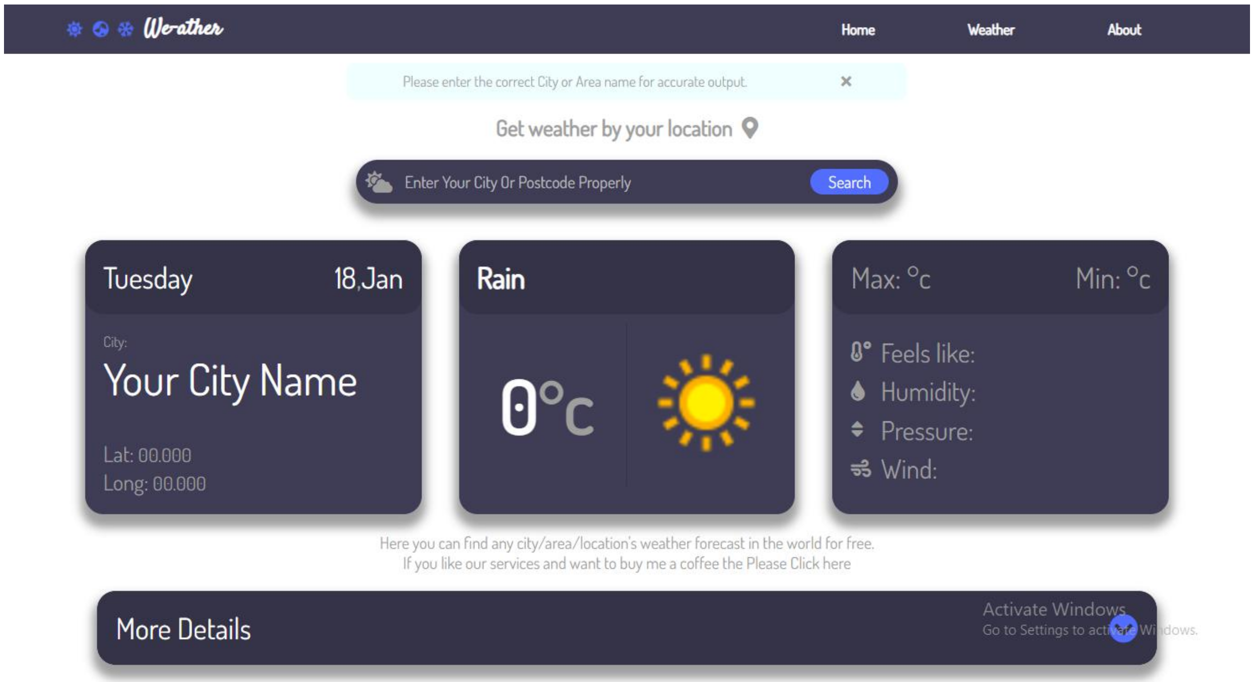
This section describes the different interfaces for the Weather Forecast System application. It contains a detailed description about each interface along with a screen shot of the interface.

1. **Home Page:** The home page of the application is common to all the system users. This interface is available through the web application.

Each Section links to an individual page containing the items related to the category to which it is assigned.


We-ather
Home
Weather
About
Welcom to
We-ather app where you can find National weather forcast
Check now
Activate Windows
Go to Settings to activate Windows.

Screenshot of the Home Page



Please enter the correct City or Area name for accurate output.

Get weather by your location

Enter Your City Or Postcode Properly

Tuesday 18,Jan

City:
Your City Name

Lat: 00.000
Long: 00.000

Rain

0°c

Max: °c Min: °c

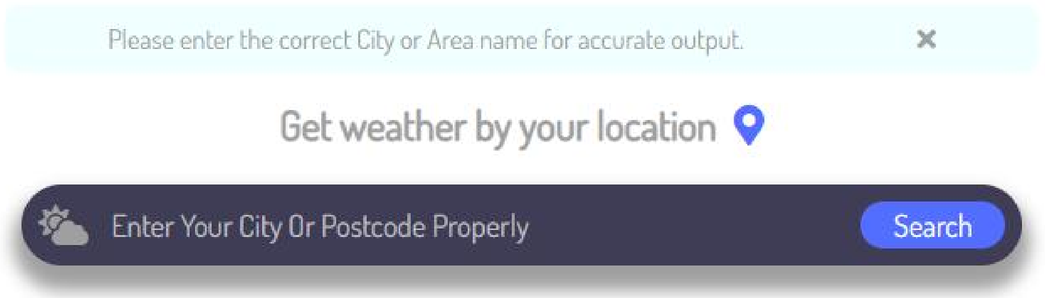
Feels like:
Humidity:
Pressure:
Wind:

Here you can find any city/area/location's weather forecast in the world for free.
If you like our services and want to buy me a coffee the Please Click here

More Details

Activate Windows
Go to Settings to activate Windows.

Screenshot of the Weather page

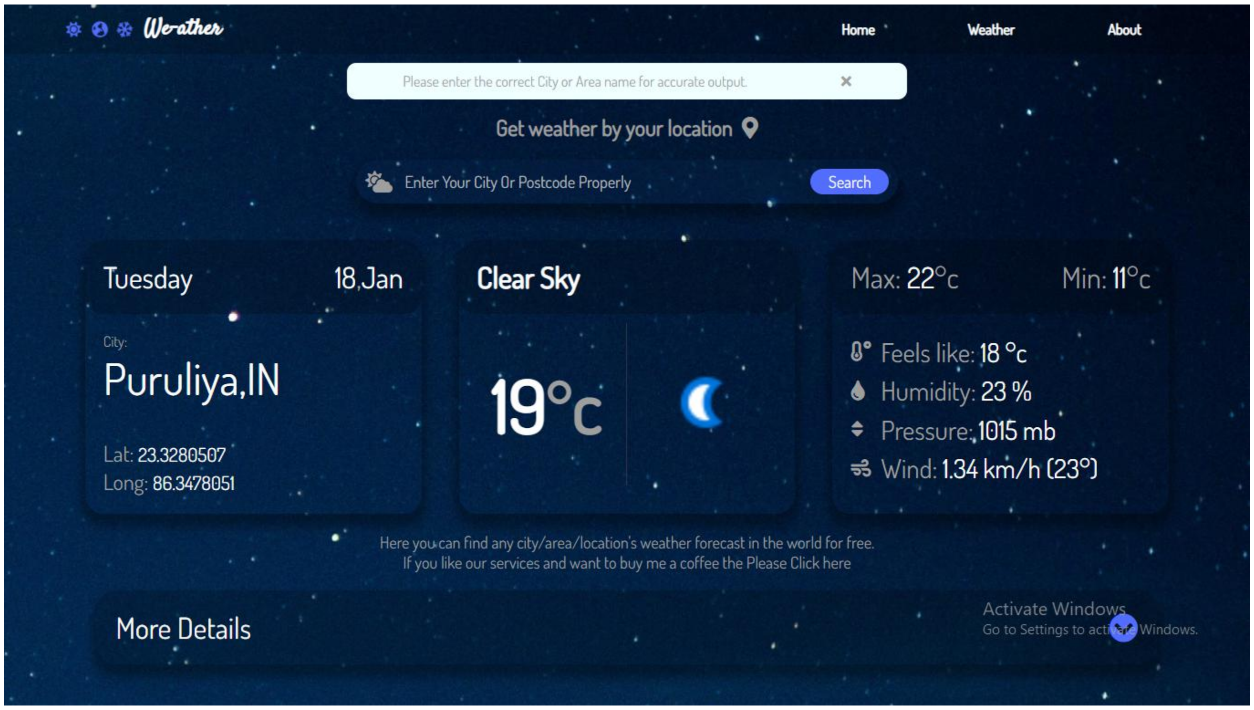


Please enter the correct City or Area name for accurate output.

Get weather by your location

Enter Your City Or Postcode Properly

Screenshot of the Input Takers


Weather
Home
Weather
About
Please enter the correct City or Area name for accurate output.
Get weather by your location
Enter Your City Or Postcode Properly
Search
More Details
Activate Windows
Go to Settings to activate Windows.

Tuesday 18,Jan

City:
Puruliya,IN

Lat: 23.3280507
Long: 86.3478051

Clear Sky

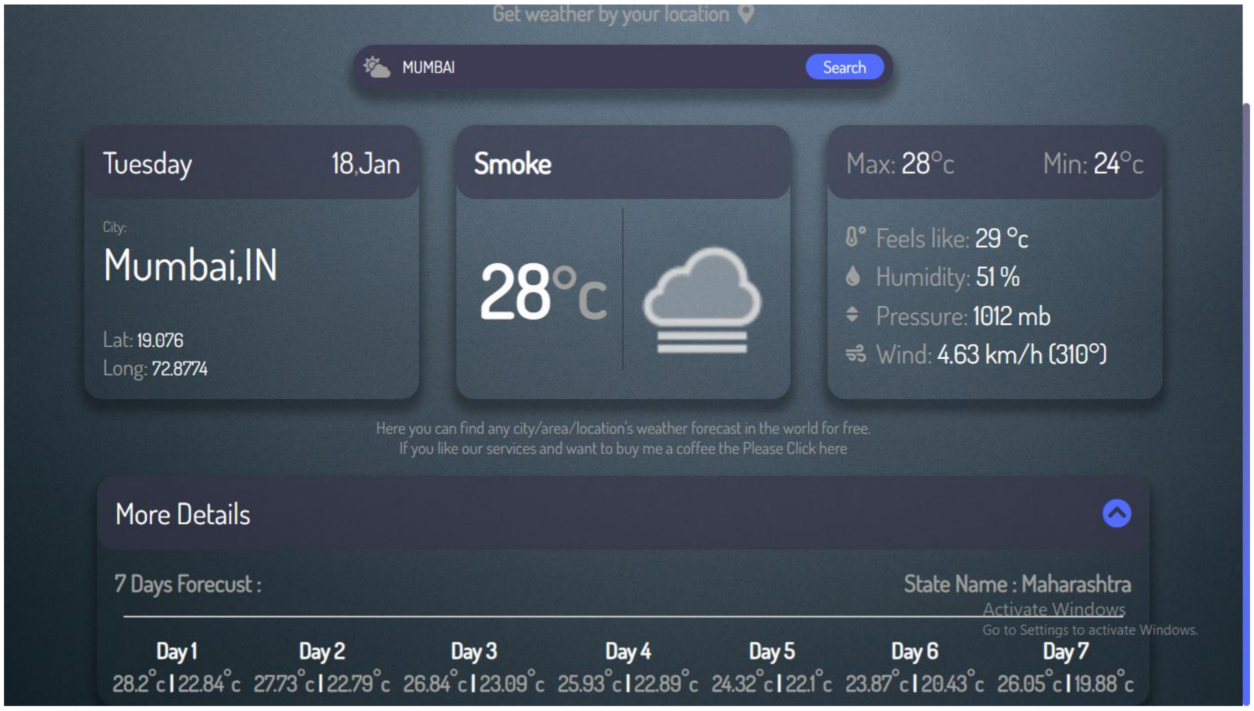
19°c

Max: 22°c Min: 11°c

Feels like: 18 °c
Humidity: 23 %
Pressure: 1015 mb
Wind: 1.34 km/h (23°)

Here you can find any city/area/location's weather forecast in the world for free.
If you like our services and want to buy me a coffee the Please Click here

Screenshot of the Search Result


Get weather by your location
MUMBAI
Search
More Details
Activate Windows
Go to Settings to activate Windows.

Tuesday 18,Jan

City:
Mumbai,IN

Lat: 19.076
Long: 72.8774

Smoke

28°c

Max: 28°c Min: 24°c

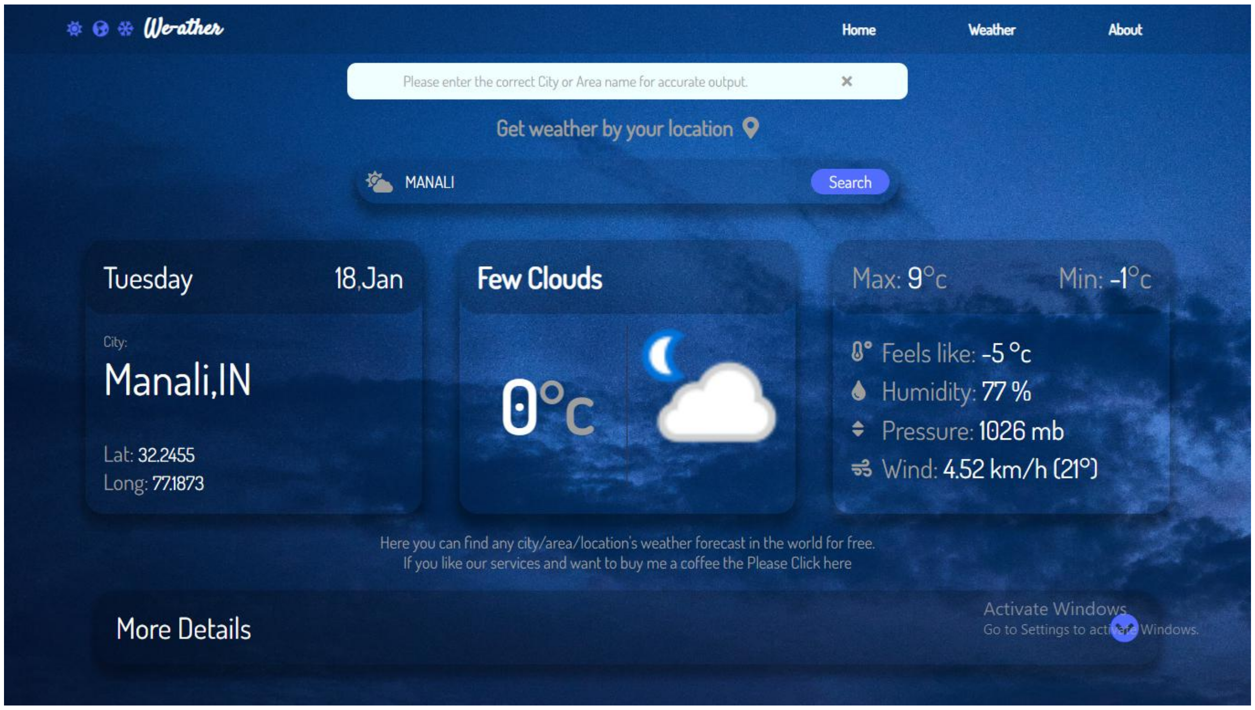
Feels like: 29 °c
Humidity: 51 %
Pressure: 1012 mb
Wind: 4.63 km/h (310°)

Here you can find any city/area/location's weather forecast in the world for free.
If you like our services and want to buy me a coffee the Please Click here

7 Days Forecust : State Name : Maharashtra

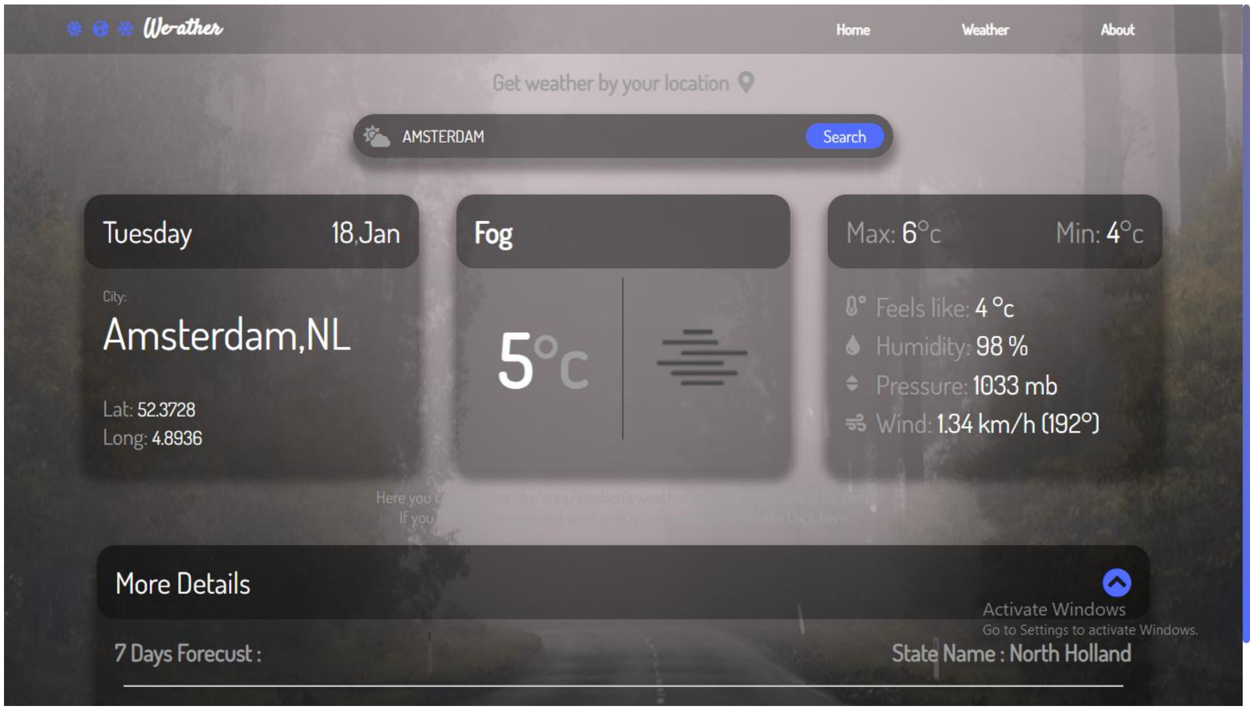
Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
28.2°c | 22.84°c	27.73°c | 22.79°c	26.84°c | 23.09°c	25.93°c | 22.89°c	24.32°c | 22.1°c	23.87°c | 20.43°c	26.05°c | 19.88°c

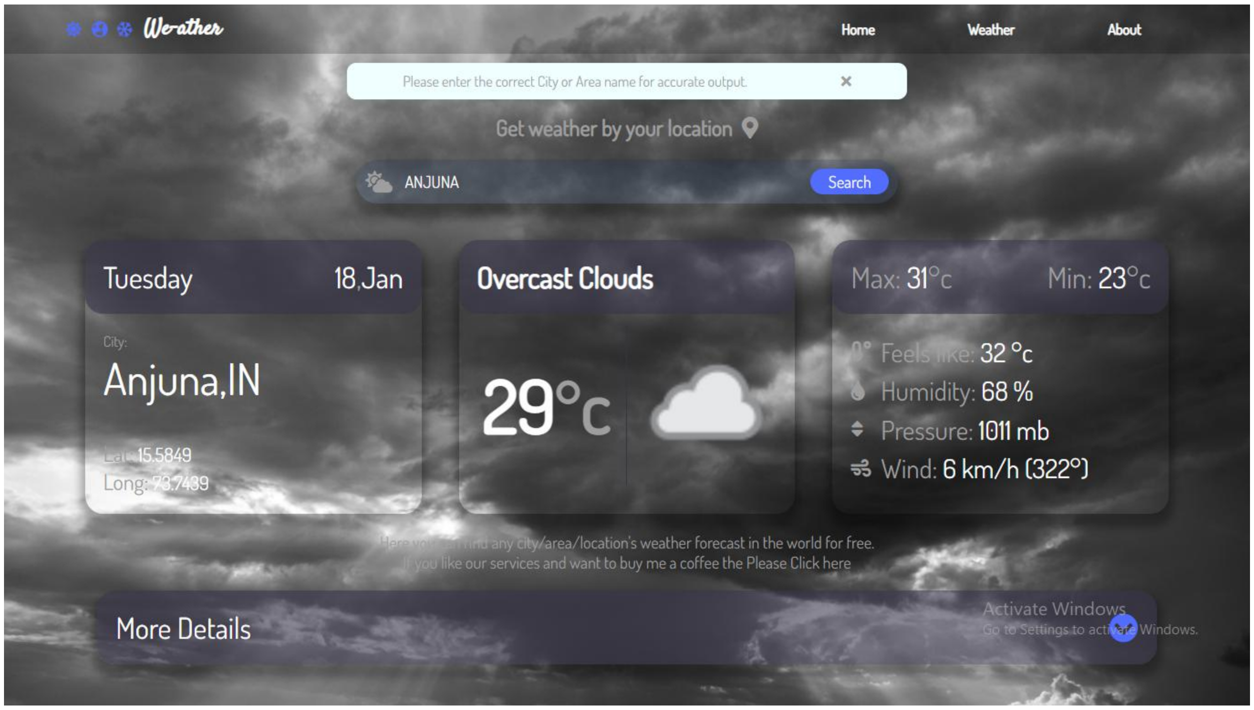
Screenshot of the Diffrent Sreach Results.



Weather
Home
Weather
About
Please enter the correct City or Area name for accurate output.
Get weather by your location
MANALI
Search
Tuesday
18,Jan
City:
Manali,IN
Lat: 32.2455
Long: 77.1873
Few Clouds
0°c
Max: 9°c
Min: -1°c
Feels like: -5 °c
Humidity: 77 %
Pressure: 1026 mb
Wind: 4.52 km/h (21°)
Here you can find any city/area/location's weather forecast in the world for free.
If you like our services and want to buy me a coffee the Please Click here
More Details
Activate Windows
Go to Settings to activate Windows.

Screenshot of the **Few Clouds Weather Value.**





Please enter the correct City or Area name for accurate output.

Get weather by your location

ANJUNA

Tuesday 18,Jan

City:
Anjuna,IN

Lat: 15.5849
Long: 73.7439

Overcast Clouds

29°c

Max: 31°c Min: 23°c

Feels like: 32 °c
Humidity: 68 %
Pressure: 1011 mb
Wind: 6 km/h (322°)

Here you can find any city/area/location's weather forecast in the world for free.
If you like our services and want to buy me a coffee the Please Click here

Activate Windows
Go to Settings to activate Windows.

CHAPTER 4. TESTING

This chapter includes the methods that were used for testing, validating, and evaluating the system. The Conclusion and the Future Work for the software are also given.

4.1. Methodology

With this testing approach, all the specs were ready for a prototype, and a plan was already built to be shown; the tester started writing his or her code and saw if he or she could obtain the same results that the specs mentioned. This way, the specs were tested on each prototype, and continuous testing was applied. This also helped to minimize the testing that would have to be implemented at the end of the software lifecycle. In the process, all aspects of the software were tested. Steps to follow while implementing the methodology are as follows:

1. Start with a base function that you want to implement.
2. Create a document with the detailed requirement definition, an activity diagram with a description of the flow, database tables to be used, a component diagram, and a description of each component with the precondition and tables that would be affected by the component.
3. Give the document to the tester, and work with the tester while he or she writes the code to check if the steps in the document can be implemented and if the result of each use case can be achieved.
4. If the tester finds a step difficult to implement or thinks he or she is missing additional information to implement the functionality, then go to step 2; otherwise, go to step 3.
5. Ask the tester to log on all the errors and difficulties he or she encountered while working on the prototype implementation.
6. Once the prototype is done and the results between the developer's prototype and tester's prototype match, work on the other requirement, and expand the prototype to final software.

7. When the testing approach was implemented, the following pros and cons regarding the testing approach were realized.

   Pros of using the methodology

   - Helps give a better understanding about the requirements.
   - Better design at the end of the cycle.
   - Reduced testing to be performed at the end of the cycle
   - Documents produced would be of higher quality.

   Cons of using the methodology

   - The person working on the document should be experienced.
   - There are increased time and money involved with testing.
   - Different viewpoints for the same problem can lead to varying results.

4.2. Interface Testing

This section lists the functional requirements used for creating the test-case table, the test cases that were used to verify the interface table, and the results for the test-cases table.

Table 1 lists the functional requirements for the interface built for the online shopping-cart application, along with a short description of each requirement

CHAPTER 5. CONCLUSION & FUTURE WORK

This chapter includes the Conclusion reached after creating the current version of the software to meet the system objectives. The comparison is done between the system that was built and original requirements that were designed at the beginning of the project. It also describes the Future Work that is intended to be accomplished with later versions of the software.

5.1. Conclusion

The main objective of the application is to help computer science students understand the basics of Node.js , JavaScript, API, CSS and HTML. By browsing through the application and looking at the code for each graphical interpretation, students should be able to easily understand the implementation. The following results have been achieved after the completing the system and relate back to the system's objective.

1. **Should allow computer science students to browse through the code and application:** This is achieved when users, i.e., computer science students, are able to run and install the application. When they run the application, they can browse through the implementation of different objects.
2. **Should allow users to search all values and get the desire output:** This is achieved when the user first runs the application and is directed to a home page and the search city names. The user can browse and search on any city and they will get their desire output with various details.

5.2. Future Work

The following section discusses the work that will be implemented with future releases of the software.

- Adding Rode Risk prediction
- Hourly & Minutly forecast
- Agriculture analytics (Crop map, Recognised field boundaries , Vegetation indicies statistics by each recognised field, Climate data for regions and particular field, Soil and weather data)
- Travel Plan scheduler by weather of the place.
- Statistical Weather report.
- Precipitation in maps & Map weather visualization
- Air Pollution

CHAPTER 6. BIBLIOGRAPHY

1. Node js & API Web Development

2. HTML , CSS , Javascrip In Web Development

3. Node js the complete Reference

4. Learning Node js with API

5. Programming in JavaScript

6. Learn Application Program Interface

-------------- O -------------