Abstract

Weather forecasting is the prediction of the state of the atmosphere for a given location using the application of science and technology. This includes temperature, rain, cloudiness, wind speed, and humidity. Weather warnings are a special kind of short-range forecast carried out

for the protection of human life. Weather warnings are issued by the governments throughout the world for all kinds of threatening weather events including tropical storms and tropical cyclones depending upon the location. The forecast may be short-range or Long-range. It is a very interesting and challenging task. This report provides a basic understanding of the purpose and scope of weather forecasts, the basic principles and the general models developed for CH

CONTENTS

CHAPTER -3

CHAPTER -4

CHAPTER -5

CHAPTER-1

Introduction

1.1 Overview

In this weather app project, users can enter any city name to get the 5-day weather forecast or simply click on the “Use Current Location” button to get their current location’s weather details, including temperature, wind speed, humidity, and more. This project is also mobile-friendly, which means it looks great on all devices.

1.2 Objective of the System

This project will serve the following objectives:-

- Provides the user with an easy and friendly interface.
- Provides the user with the temperature of a particular region.
- It will also show humidity, wind speed and cloud.

1.3 JUSTIFICATION AND NEED FOR THE SYSTEM

Weather is something everybody deals with, and accurate data about it like what is coming can help users to make informed decisions. With weather apps for iOS and Android, people can exactly know when to expect a change in the weather conditions. Weather apps can give urgent alerts too.

Undoubtedly, weather forecasting has come a long way, helping people to know about weather conditions. So, if you are in an area where weather frequently changes from sunny to torrential rain in a matter of minutes, then what is the easiest way to make sure to be prepared? A suitable answer is a weather application.

1.4 Advantages of the system

- Real-Time Data

One of the biggest advantages of weather monitoring systems and also the reason why people have been going in for weather stations is because of the ability to get their information in Realtime.

- ➢ Accurate Local Forecast

In reality, the meteorological department may be located far from your home and weather forecasts are made for regions, not a specific area. That's a reason why in these instances, the weather predictions that they give are not always the most accurate.

- ➢ Ease Of Use

Ease to use is definitely a big advantage of the weather monitoring system. Weather stations like all other weather devices are designed to be efficient and straightforward, therefore, everyone can use them It is so convenient and comfortable for users to get the most accurate information in the simplest way possible.

1.5 Previous work or related systems; how they are used.

Before we begin a new system it is important to study the system that will be improved or replaced (if there is one). We need to analyze how this system uses hardware, software, network and people resources to convert data resources, such as transaction data, into information products.

Following are the problems associated with the previous project which led to the creation of the proposed project:-

1. Not user-friendly: The existing system is not user-friendly because the information like humidity cloud and wind etc are not in one place.

2. Not a good UI: The user interface of the previous systems are not that good.

CHAPTER -2

REQUIREMENT ANALYSIS

2.1 Functional Requirements

Functional requirements are the requirements that describe the functionalities of the system elements. It may involve functional user requirements or functional system requirements.

For example:

- The operator shall be able to input the region to the system to view the desired weather parameters.
- The system shall provide the following weather parameters: temperature, pressure, wind speed, date / time and humidity.

2.2 Analysis study

- Lower Installation Charges:

We neither require any high-configuration systems for the smooth running of the server program nor do we require any high-configuration systems for the smooth running of a client program. This application is designed with ease to support any ordinary system having an internet connection

- Secured and Reliable:

The reliability of the system is to make sure the website does not go offline

- Availability

The availability of the system is that the website will be active on the Internet and people will be able to browse it.

2.3 Feasibility Study

All projects are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. The three considerations are involved in the feasibility analysis.

2.4 Technical Feasibility

Technical feasibility centres on the existing mobile system (hardware, software...etc) and to what extent it can support the proposed addition if the budget is a serious constraint, then the project is judged not feasible. The technical feasibilities are an important role in our project because here we're using HTML, CSS and JavaScript. It requires Visual Studio Code(software) to develop this application. A easily available software and easy to use.

2.5 Economical Feasibility

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. If the benefits outweigh the cost then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed systems have to be made if it is having a chance of being approved. This is an ongoing effort that improves any feasibility costs spent on this project because here we're using open-source environments.

2.6 Operational Feasibility

People are inherently resistant to change and mobiles have been known to facilitate change. There is no need of technical background is required to work on the application. All the information needed can be seen with just one click.

2.7 Feasibility Study

All projects are feasible if they have unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. The three considerations are involved in the feasibility analysis.

2.8 Technical Feasibility

Technical feasibility centres on the existing mobile system (hardware, software...etc) and to what extent it can support the proposed addition if the budget is a serious constraint, then the project is judged not feasible. The technical feasibilities are an important role in our project because here we're using HTML,CSS and JavaScript. It requires Visual Studio Code(software) to develop this application. A easily available software and easy to use.

2.9 Economical Feasibility

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. If the benefits outweigh the cost then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed systems have to be made if it is having a chance of being approved. This is an ongoing effort that improves any feasibility costs spent on this project because here we're using open-source environments.

2.10 Operational Feasibility

People are inherently resistant to change and mobiles have been known to facilitate change. There is no need of technical background is required to work on the application. All the information needed can be seen with just one click

3. USER REQUIREMENTS

The system specifications that a user may want are as follows:

➢ It should be easy to understand.

➢ Must be interactive.

➢ Should provide a good user interface.

➢ Security should be maintained.

4. Final Requirements

User Oriented: A system should be more user friendly not from the technical point of view

Better GUI: All the elements used in the system should be interactive in nature so that its look and feel are not so boring that the user could get bored while using it.

Reliability: The system should be reliable and fast in processing.

Data security: Access to the organizational data is not to be granted to any unknown person who is not a part of the transaction.

Confidentiality: Whatever the user is providing to the organization, the user has the full rights to modify it and it could be not be accessed/modified without the user's permission.

Better Management of information: All the information should be managed so that is the flow of the information is to be in the right track.

Presentation: The content that is to be presented to the user is to be presented in such a way that is self- explanatory to the user and he/she is satisfied with the data.

CHAPTER-3

DESIGN OF THE SYSTEM

3.1 Software requirements

Platform	**The Operating System**
Framework	CSS
Front-End Tool	Firefox
API	Open Whether Map

3.2 Hardware Requirements

Processor	Intel Pentium IV 2.9 GHz Other
RAM	Minimum 4 GB
Graphics	Integrated graphics card
Hard Disk	Minimum 500 GB

3.3 System Requirements

To know the detailed system requirements an SRS has to be prepared. Software requirement specification abbreviated as SRS is a means of translating the idea of files into a formal document. The main features of SRS include:

- Establishing the basis for an agreement between the client and the developer.
- Producing a reference for validation of the final product. SRS assist clients in determining if the software meets their requirements.

Mainly there are six requirements which an SRS must satisfy.

- It should specify the external behaviour.
- It should specify the constraints.
- It should be easy to change.
- It should be a reference tool.
- It should record throughout the lifecycle.
- It should have the capacity to expect an undesired event.

3.4 Functional Requirements

Functional requirements are the requirements that describe the functionalities of the system elements. It may involve functional user requirements or functional system requirements.

For example:

- The operator shall be able to input the region to the system to view the desired weather parameters.
- The system shall provide the following weather parameters: temperature, pressure, wind speed & direction, rainfall, and humidity.

3.5 Design Requirements

3.5.1 The main objectives of input design are: (a)

Controlling the amount of input.

- Keeping the process simple.
- The best thing in the input design is to achieve all the objectives mentioned in the simplest manner possible.

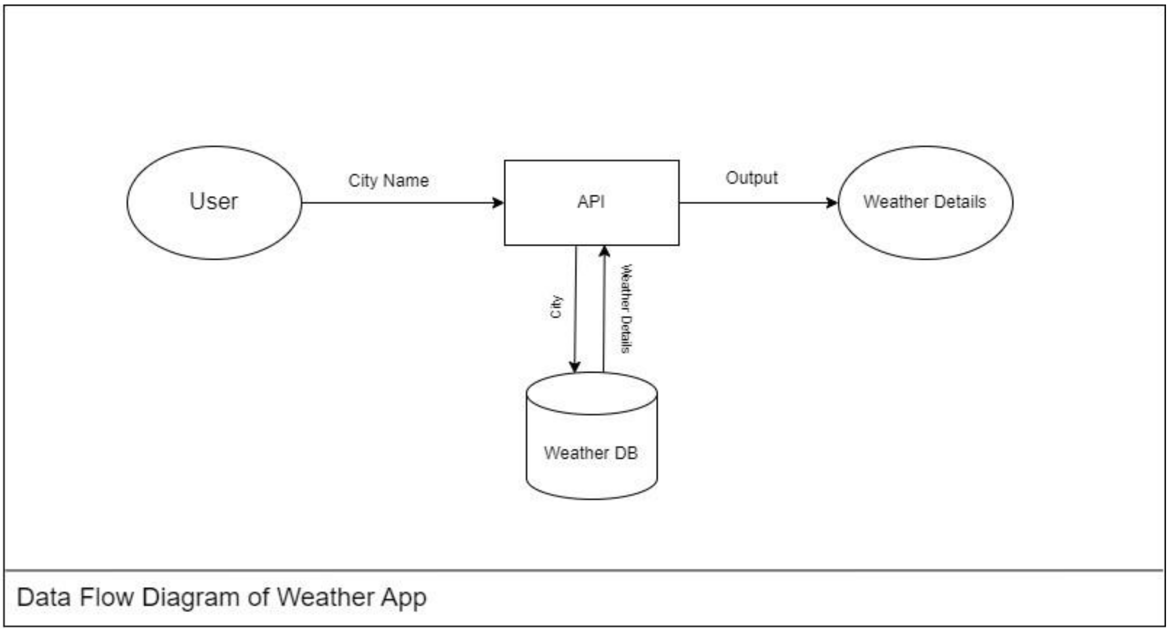
3.5.2 The main objectives of output design are:

(a) Identifying the specific outputs.

The primary goal of the system analysis is to improve the efficiency of the existing system. For that the study of specification of the requirements is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted.
Investigation done whether the upgradation of the system into an application program could solve the problems and eradicate the inefficiency of the existing system

3.6 DATA FLOW DIAGRAM (DFD)

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

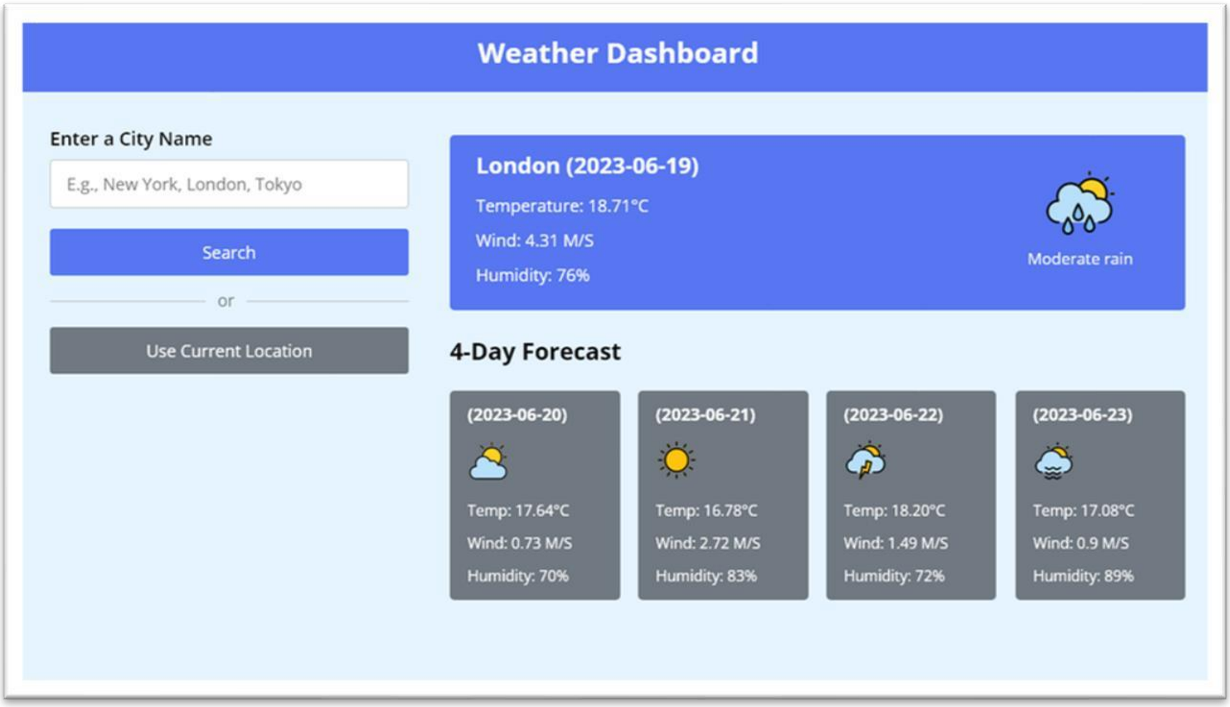


Data Flow Diagram of Weather App

A DFD shows what kinds of information will be input into and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

CHAPTER -4

IMPLEMENTATION AND CODING



Weather Dashboard

Enter a City Name

London (2023-06-19)

Temperature: 18.71°C

Wind: 4.31 M/S

Humidity: 76%

Moderate rain

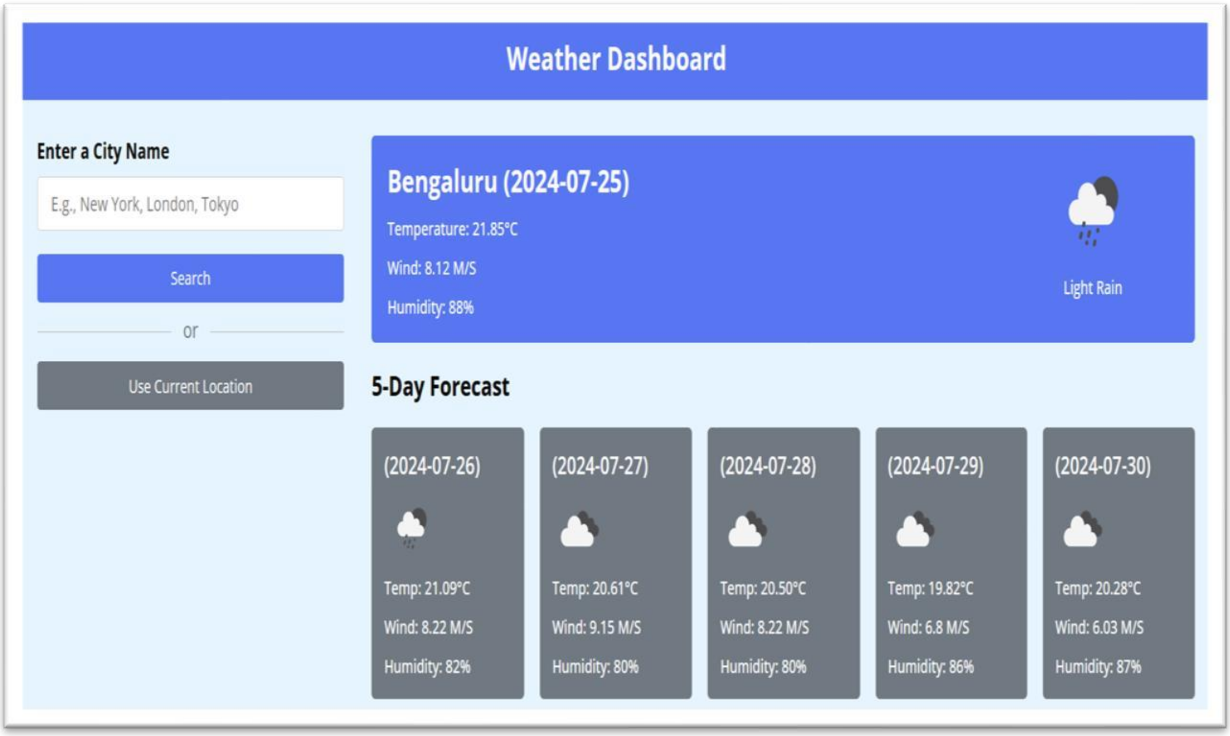
4-Day Forecast

(2023-06-20)
Temp: 17.64°C
Wind: 0.73 M/S
Humidity: 70%

(2023-06-21)
Temp: 16.78°C
Wind: 2.72 M/S
Humidity: 83%

(2023-06-22)
Temp: 18.20°C
Wind: 1.49 M/S
Humidity: 72%

(2023-06-23)
Temp: 17.08°C
Wind: 0.9 M/S
Humidity: 89%



Weather Dashboard

Enter a City Name

Bengaluru (2024-07-25)

Temperature: 21.85°C

Wind: 8.12 M/S

Humidity: 88%

Light Rain

5-Day Forecast

(2024-07-26)
Temp: 21.09°C
Wind: 8.22 M/S
Humidity: 82%

(2024-07-27)
Temp: 20.61°C
Wind: 9.15 M/S
Humidity: 80%

(2024-07-28)
Temp: 20.50°C
Wind: 8.22 M/S
Humidity: 80%

(2024-07-29)
Temp: 19.82°C
Wind: 6.8 M/S
Humidity: 86%

(2024-07-30)
Temp: 20.28°C
Wind: 6.03 M/S
Humidity: 87%

4.1 OPERATING SYSTEM

Platform Independent: Since the project is done completely in HTML, CSS and JavaScript, it also executes the main properties of the language. The application is platform-independent.

4.2 Languages used

HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS

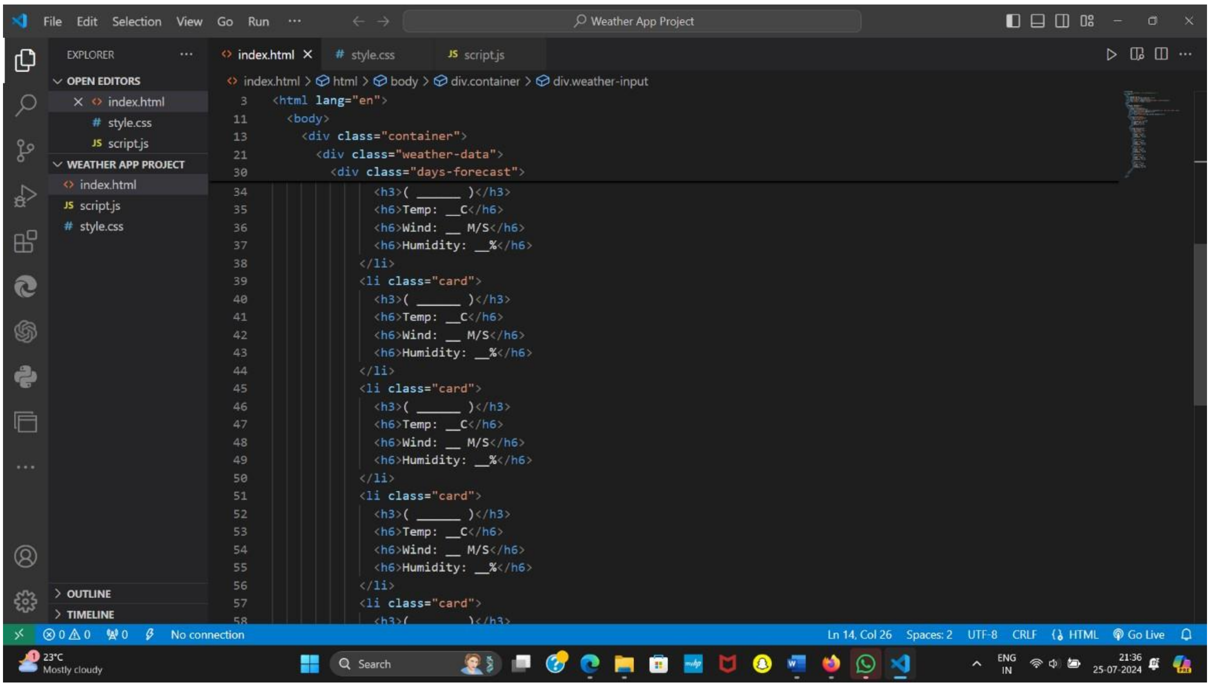
CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

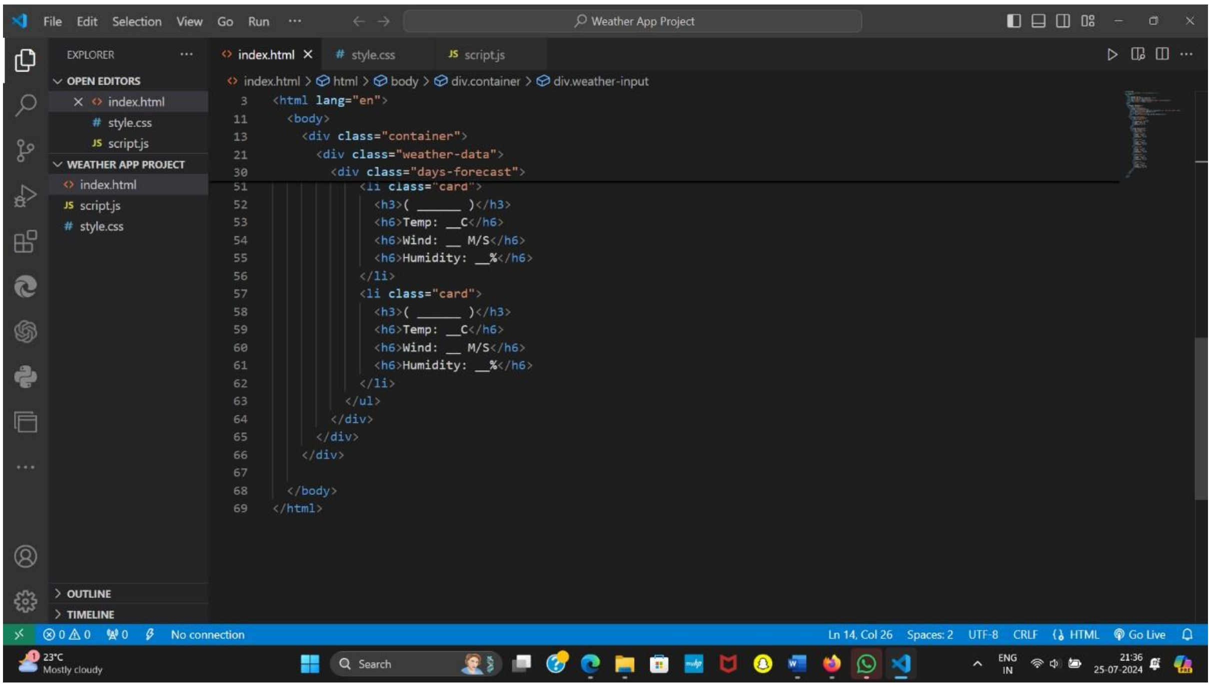
JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

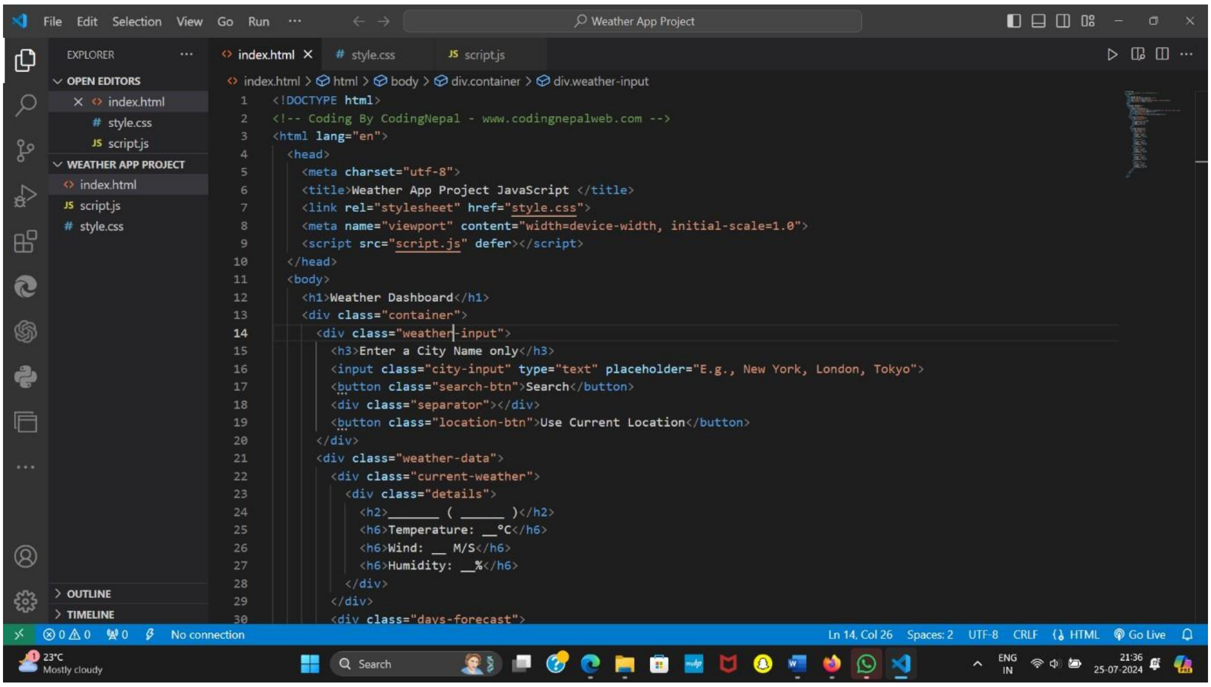
4.3 Snapshots



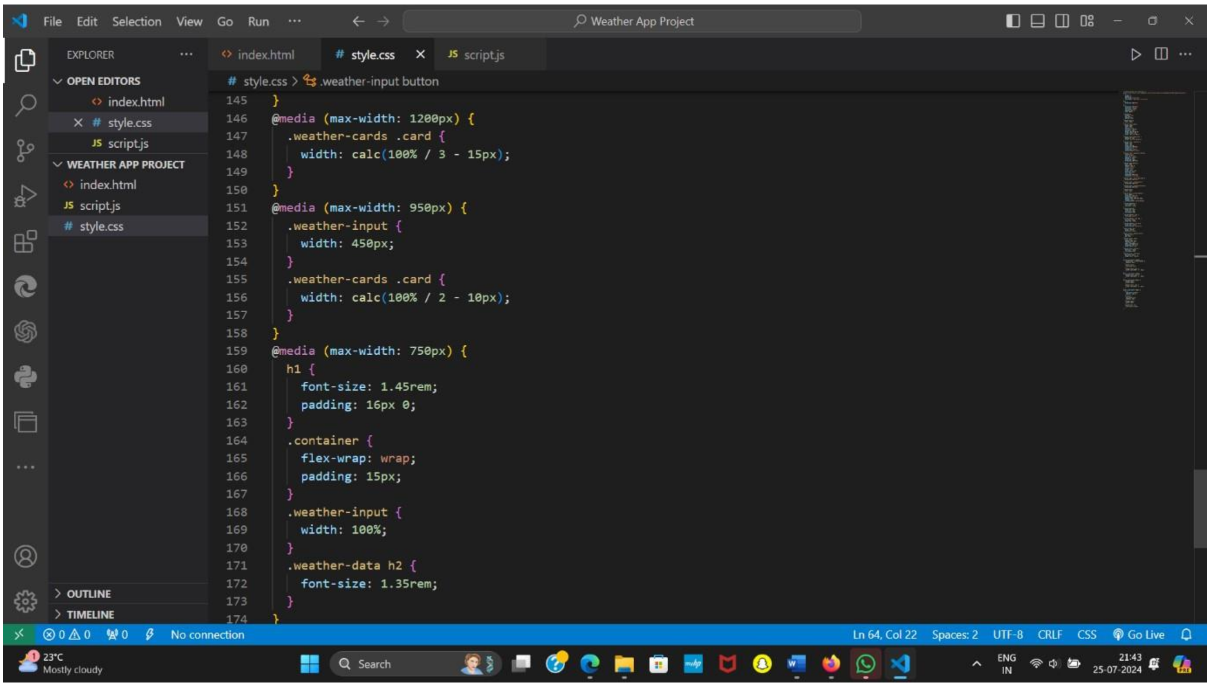
```
<html lang="en">
  <body>
    <div class="container">
      <div class="weather-data">
        <div class="days-forecast">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
            <li class="card">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
            <li class="card">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
            <li class="card">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
            <li class="card">
              <h3>( ______ )</h3>
```



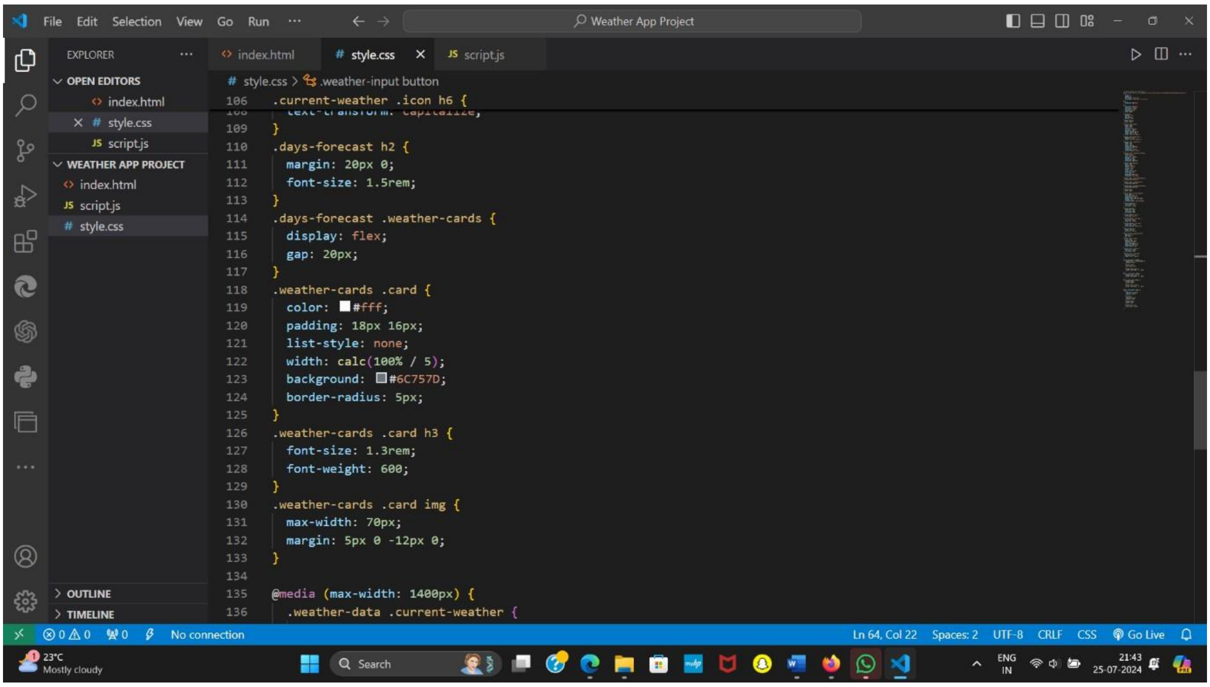
```
<html lang="en">
  <body>
    <div class="container">
      <div class="weather-data">
        <div class="days-forecast">
            <li class="card">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
            <li class="card">
              <h3>( ______ )</h3>
              <h6>Temp: __C</h6>
              <h6>Wind: __ M/S</h6>
              <h6>Humidity: __%</h6>
            </li>
          </ul>
        </div>
      </div>
    </div>

  </body>
</html>
```



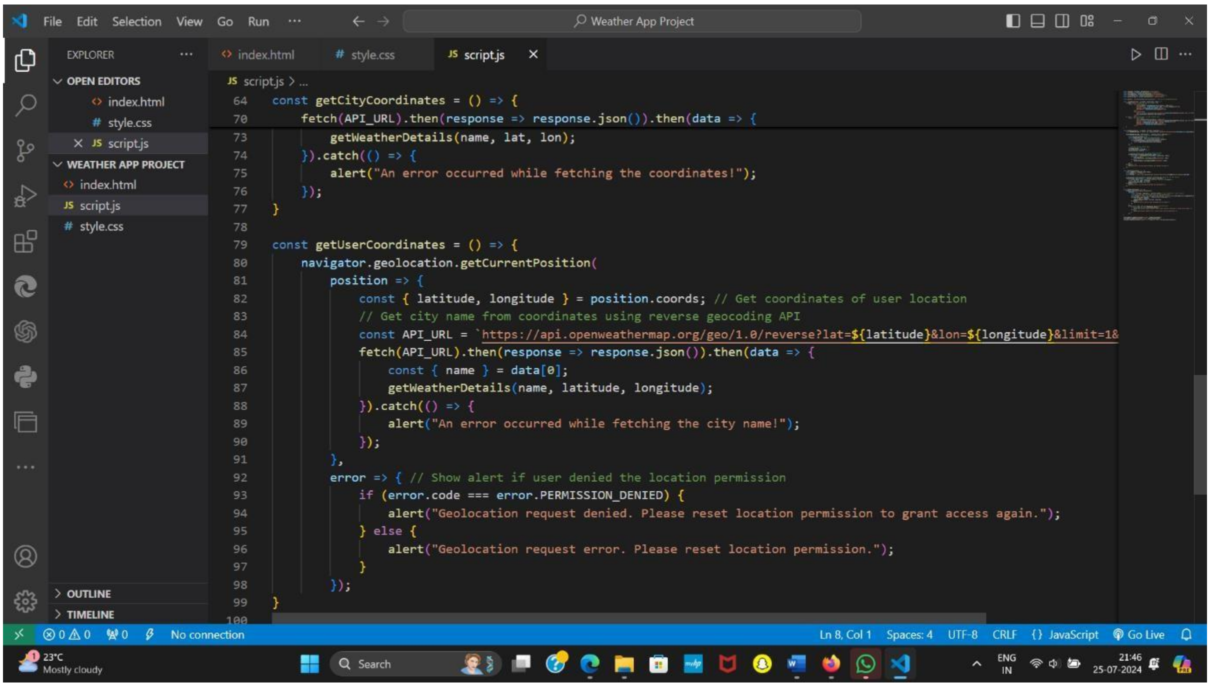
```html
<!DOCTYPE html>
<!-- Coding By CodingNepal - www.codingnepalweb.com -->
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Weather App Project JavaScript </title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="script.js" defer></script>
  </head>
  <body>
    <h1>Weather Dashboard</h1>
    <div class="container">
      <div class="weather-input">
        <h3>Enter a City Name only</h3>
        <input class="city-input" type="text" placeholder="E.g., New York, London, Tokyo">
        <button class="search-btn">Search</button>
        <div class="separator"></div>
        <button class="location-btn">Use Current Location</button>
      </div>
      <div class="weather-data">
        <div class="current-weather">
          <div class="details">
            <h2>_______ ( ______ )</h2>
            <h6>Temperature: __°C</h6>
            <h6>Wind: __ M/S</h6>
            <h6>Humidity: __%</h6>
          </div>
        </div>
        <div class="days-forecast">
```



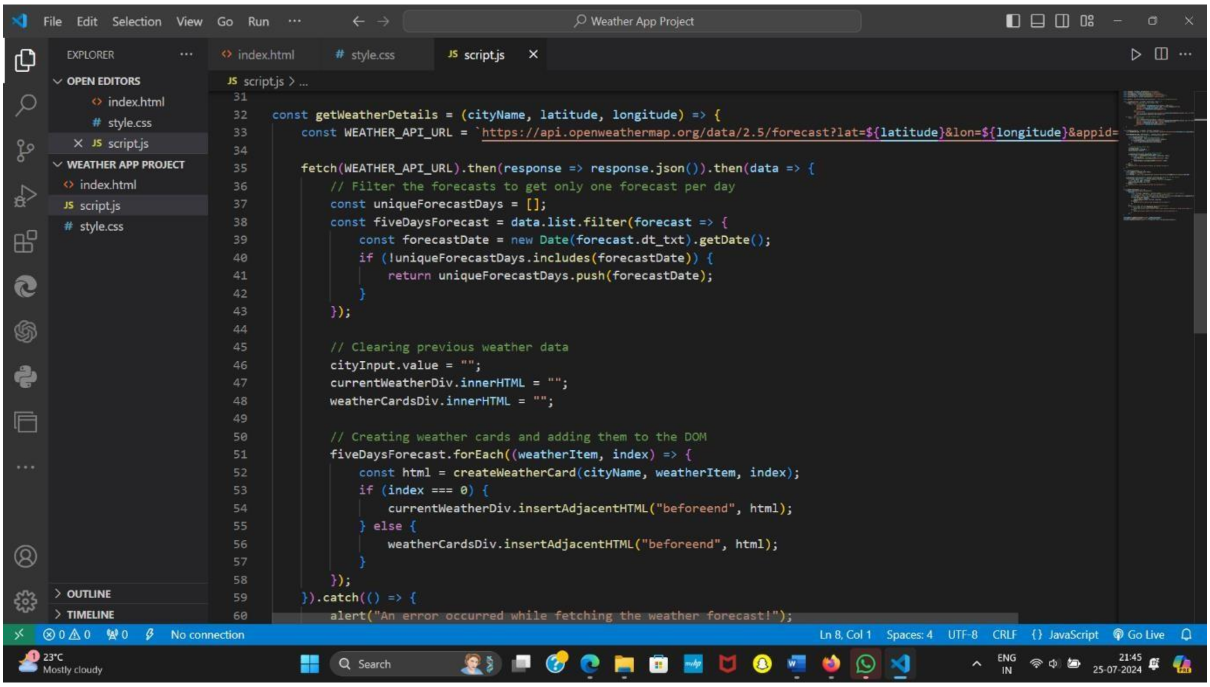
```css
}
@media (max-width: 1200px) {
  .weather-cards .card {
    width: calc(100% / 3 - 15px);
  }
}
@media (max-width: 950px) {
  .weather-input {
    width: 450px;
  }
  .weather-cards .card {
    width: calc(100% / 2 - 10px);
  }
}
@media (max-width: 750px) {
  h1 {
    font-size: 1.45rem;
    padding: 16px 0;
  }
  .container {
    flex-wrap: wrap;
    padding: 15px;
  }
  .weather-input {
    width: 100%;
  }
  .weather-data h2 {
    font-size: 1.35rem;
  }
}
```



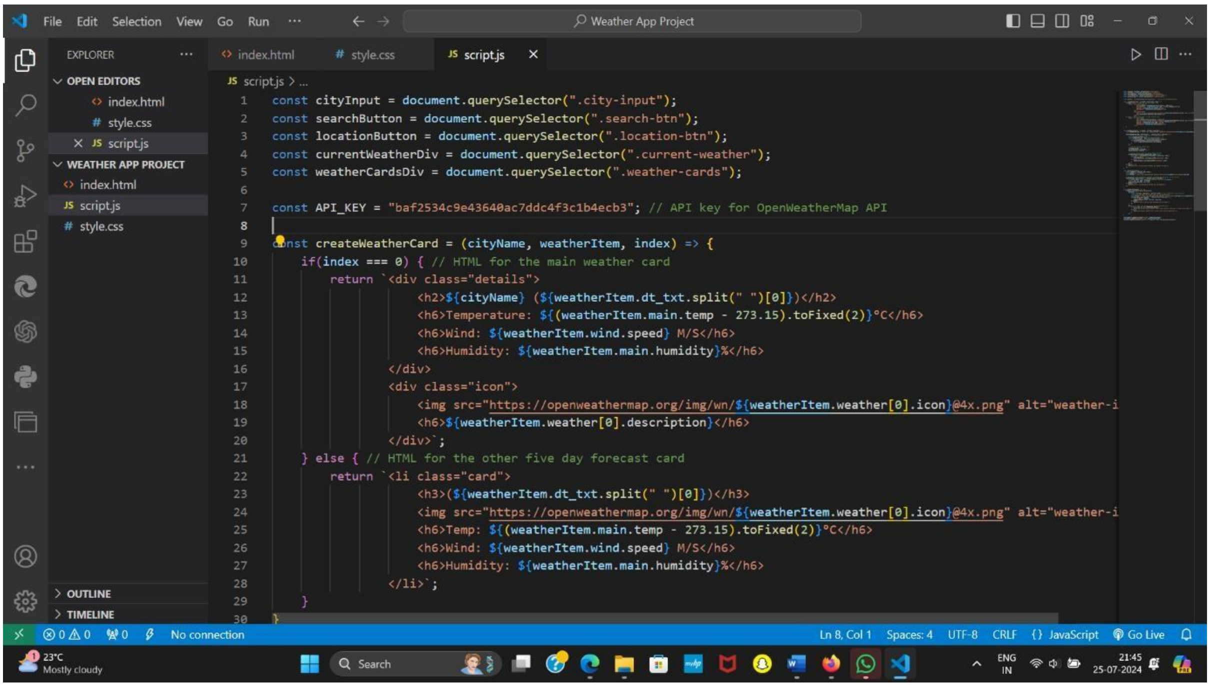
```css
.current-weather .icon h6 {
  text-transform: capitalize;
}
.days-forecast h2 {
  margin: 20px 0;
  font-size: 1.5rem;
}
.days-forecast .weather-cards {
  display: flex;
  gap: 20px;
}
.weather-cards .card {
  color: #fff;
  padding: 18px 16px;
  list-style: none;
  width: calc(100% / 5);
  background: #6C757D;
  border-radius: 5px;
}
.weather-cards .card h3 {
  font-size: 1.3rem;
  font-weight: 600;
}
.weather-cards .card img {
  max-width: 70px;
  margin: 5px 0 -12px 0;
}

@media (max-width: 1400px) {
  .weather-data .current-weather {
```



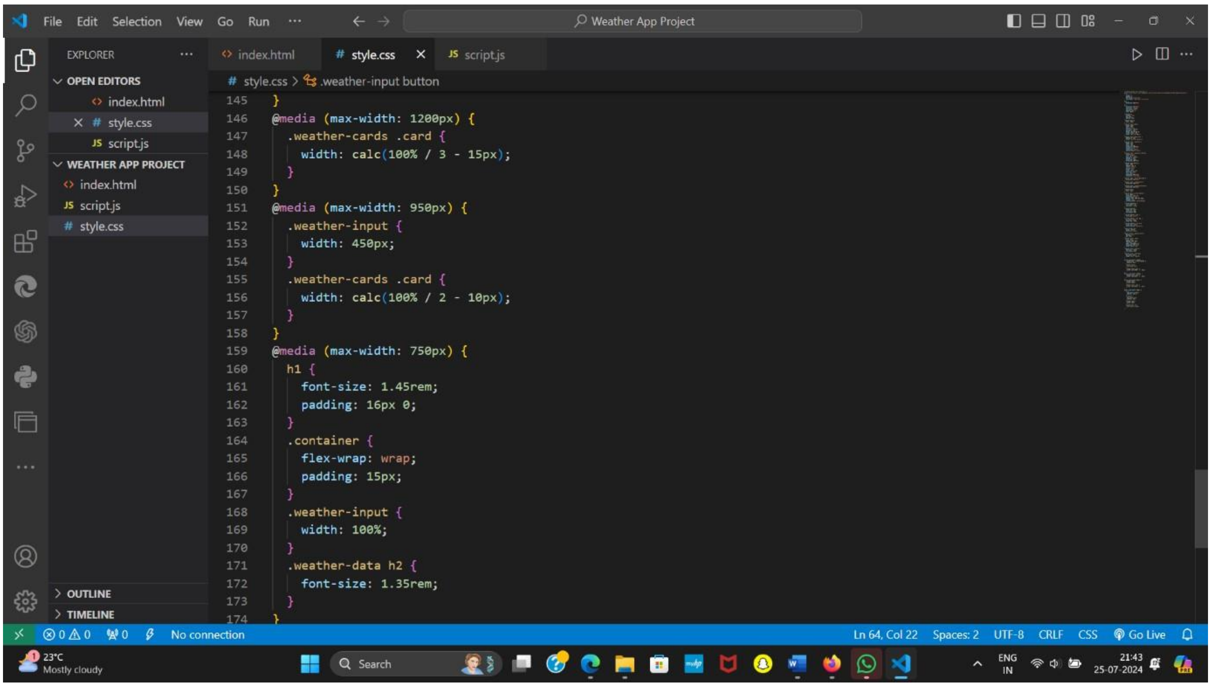
```javascript
const getCityCoordinates = () => {
    fetch(API_URL).then(response => response.json()).then(data => {
        getWeatherDetails(name, lat, lon);
    }).catch(() => {
        alert("An error occurred while fetching the coordinates!");
    });
}

const getUserCoordinates = () => {
    navigator.geolocation.getCurrentPosition(
        position => {
            const { latitude, longitude } = position.coords; // Get coordinates of user location
            // Get city name from coordinates using reverse geocoding API
            const API_URL = `https://api.openweathermap.org/geo/1.0/reverse?lat=${latitude}&lon=${longitude}&limit=1&
            fetch(API_URL).then(response => response.json()).then(data => {
                const { name } = data[0];
                getWeatherDetails(name, latitude, longitude);
            }).catch(() => {
                alert("An error occurred while fetching the city name!");
            });
        },
        error => { // Show alert if user denied the location permission
            if (error.code === error.PERMISSION_DENIED) {
                alert("Geolocation request denied. Please reset location permission to grant access again.");
            } else {
                alert("Geolocation request error. Please reset location permission.");
            }
        });
}
```



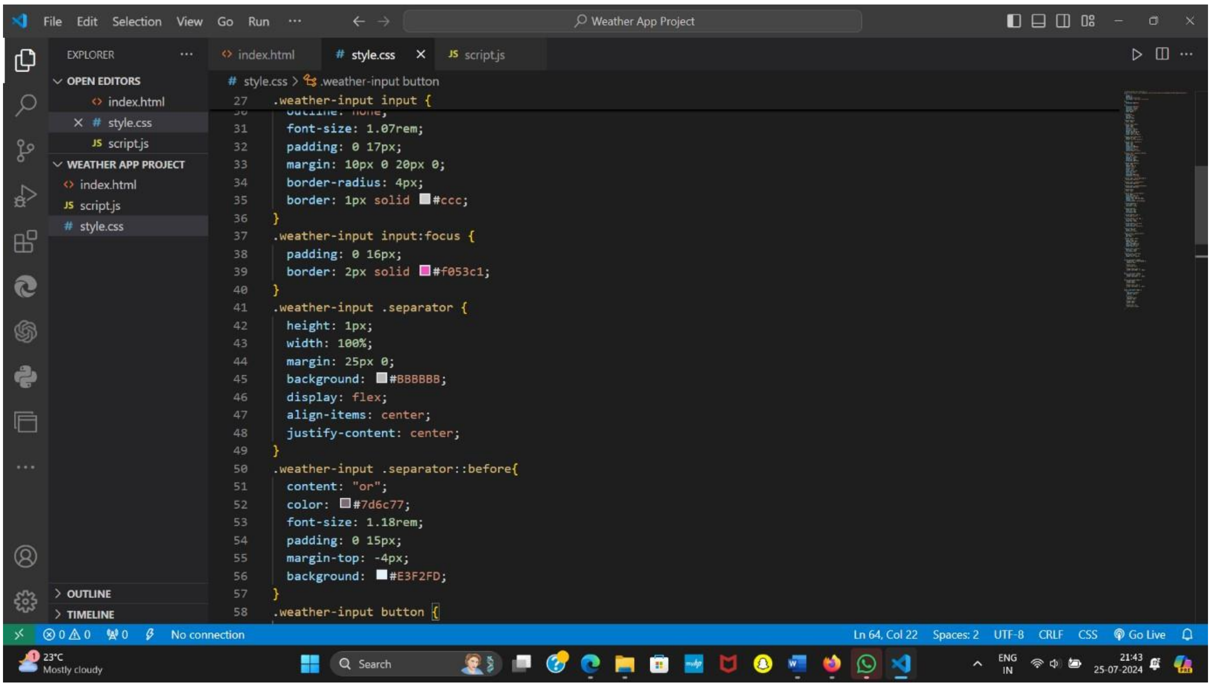
```
const getWeatherDetails = (cityName, latitude, longitude) => {
    const WEATHER_API_URL = `https://api.openweathermap.org/data/2.5/forecast?lat=${latitude}&lon=${longitude}&appid=

    fetch(WEATHER_API_URL).then(response => response.json()).then(data => {
        // Filter the forecasts to get only one forecast per day
        const uniqueForecastDays = [];
        const fiveDaysForecast = data.list.filter(forecast => {
            const forecastDate = new Date(forecast.dt_txt).getDate();
            if (!uniqueForecastDays.includes(forecastDate)) {
                return uniqueForecastDays.push(forecastDate);
            }
        });

        // Clearing previous weather data
        cityInput.value = "";
        currentWeatherDiv.innerHTML = "";
        weatherCardsDiv.innerHTML = "";

        // Creating weather cards and adding them to the DOM
        fiveDaysForecast.forEach((weatherItem, index) => {
            const html = createWeatherCard(cityName, weatherItem, index);
            if (index === 0) {
                currentWeatherDiv.insertAdjacentHTML("beforeend", html);
            } else {
                weatherCardsDiv.insertAdjacentHTML("beforeend", html);
            }
        });
    }).catch(() => {
        alert("An error occurred while fetching the weather forecast!");
```



```
const cityInput = document.querySelector(".city-input");
const searchButton = document.querySelector(".search-btn");
const locationButton = document.querySelector(".location-btn");
const currentWeatherDiv = document.querySelector(".current-weather");
const weatherCardsDiv = document.querySelector(".weather-cards");

const API_KEY = "baf2534c9e43640ac7ddc4f3c1b4ecb3"; // API key for OpenWeatherMap API

const createWeatherCard = (cityName, weatherItem, index) => {
    if(index === 0) { // HTML for the main weather card
        return `<div class="details">
                    <h2>${cityName} (${weatherItem.dt_txt.split(" ")[0]})</h2>
                    <h6>Temperature: ${(weatherItem.main.temp - 273.15).toFixed(2)}°C</h6>
                    <h6>Wind: ${weatherItem.wind.speed} M/S</h6>
                    <h6>Humidity: ${weatherItem.main.humidity}%</h6>
                </div>
                <div class="icon">
                    <img src="https://openweathermap.org/img/wn/${weatherItem.weather[0].icon}@4x.png" alt="weather-i
                    <h6>${weatherItem.weather[0].description}</h6>
                </div>`;
    } else { // HTML for the other five day forecast card
        return `<li class="card">
                    <h3>(${weatherItem.dt_txt.split(" ")[0]})</h3>
                    <img src="https://openweathermap.org/img/wn/${weatherItem.weather[0].icon}@4x.png" alt="weather-i
                    <h6>Temp: ${(weatherItem.main.temp - 273.15).toFixed(2)}°C</h6>
                    <h6>Wind: ${weatherItem.wind.speed} M/S</h6>
                    <h6>Humidity: ${weatherItem.main.humidity}%</h6>
                </li>`;
    }
}
```



```css
}
@media (max-width: 1200px) {
  .weather-cards .card {
    width: calc(100% / 3 - 15px);
  }
}
@media (max-width: 950px) {
  .weather-input {
    width: 450px;
  }
  .weather-cards .card {
    width: calc(100% / 2 - 10px);
  }
}
@media (max-width: 750px) {
  h1 {
    font-size: 1.45rem;
    padding: 16px 0;
  }
  .container {
    flex-wrap: wrap;
    padding: 15px;
  }
  .weather-input {
    width: 100%;
  }
  .weather-data h2 {
    font-size: 1.35rem;
  }
}
```



```css
.weather-input input {
  outline: none;
  font-size: 1.07rem;
  padding: 0 17px;
  margin: 10px 0 20px 0;
  border-radius: 4px;
  border: 1px solid #ccc;
}
.weather-input input:focus {
  padding: 0 16px;
  border: 2px solid #f053c1;
}
.weather-input .separator {
  height: 1px;
  width: 100%;
  margin: 25px 0;
  background: #BBBBBB;
  display: flex;
  align-items: center;
  justify-content: center;
}
.weather-input .separator::before{
  content: "or";
  color: #7d6c77;
  font-size: 1.18rem;
  padding: 0 15px;
  margin-top: -4px;
  background: #E3F2FD;
}
.weather-input button {
```

CHAPTER -5

TESTING & TEST RESULTS

5.1 SOFTWARE TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is an exposure of a system to trial input to see whether the software meets the correct output. Testing cannot be determined whether the software meets the user's needs, only whether it appears to conform to requirements. Testing can show that a system is free of errors, only that it contains errors. Testing finds errors, it does not correct errors. Software success is a quality product, on time and within cost. Testing can reveal critical mistakes. Testing should, therefore,

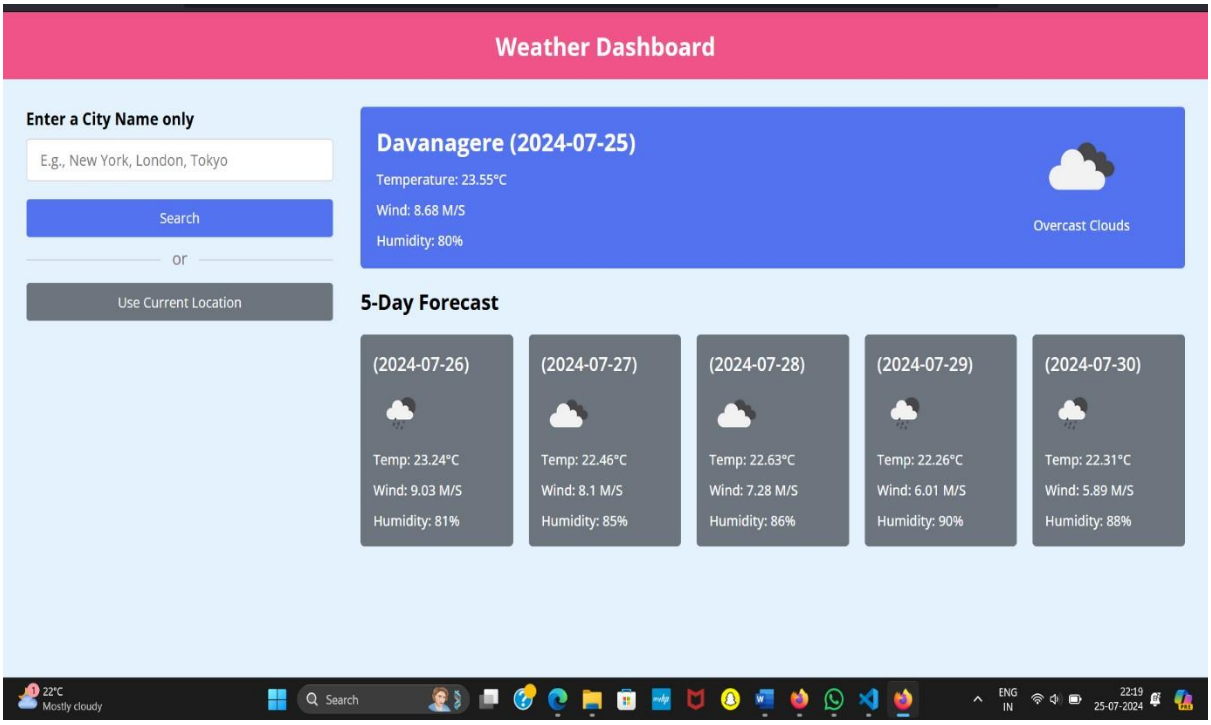
Validate Performance

Detects Errors

1 Uploads

Identify Inconsistencies.

5.2 POSSIBLE OUTCOMES



Davanagere (2024-07-25)

Temperature: 23.55°C

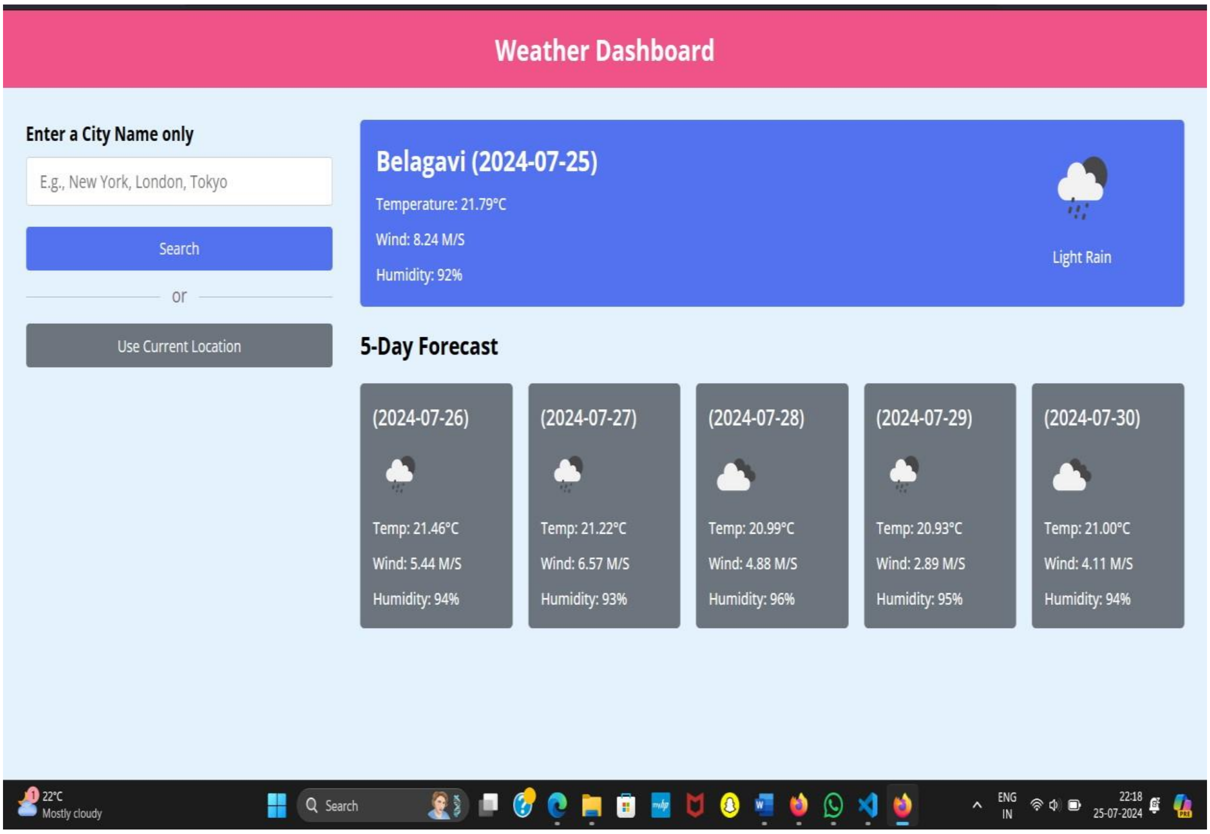
Wind: 8.68 M/S

Humidity: 80%

Overcast Clouds

5-Day Forecast

(2024-07-26)	(2024-07-27)	(2024-07-28)	(2024-07-29)	(2024-07-30)
Temp: 23.24°C	Temp: 22.46°C	Temp: 22.63°C	Temp: 22.26°C	Temp: 22.31°C
Wind: 9.03 M/S	Wind: 8.1 M/S	Wind: 7.28 M/S	Wind: 6.01 M/S	Wind: 5.89 M/S
Humidity: 81%	Humidity: 85%	Humidity: 86%	Humidity: 90%	Humidity: 88%



Weather Dashboard

Enter a City Name only

E.g., New York, London, Tokyo

Search

or

Use Current Location

Belagavi (2024-07-25)

Temperature: 21.79°C

Wind: 8.24 M/S

Humidity: 92%

Light Rain

5-Day Forecast

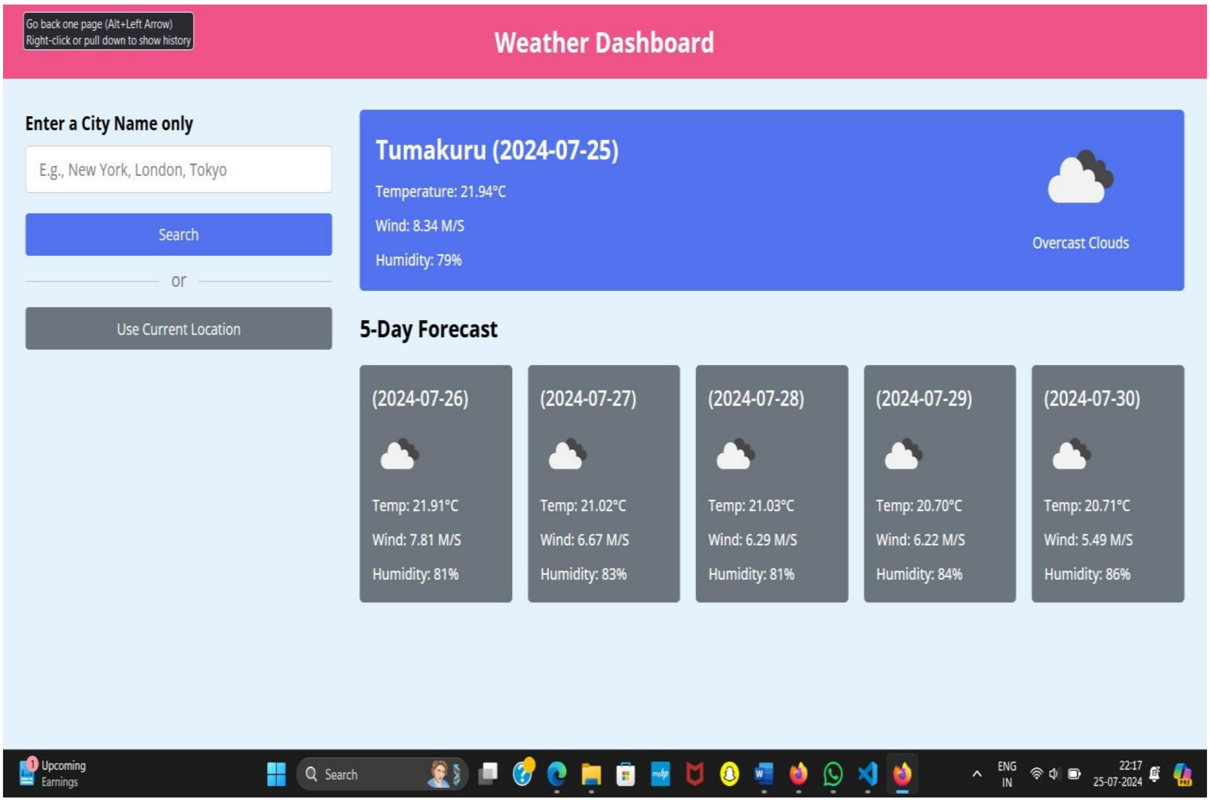
(2024-07-26)
Temp: 21.46°C
Wind: 5.44 M/S
Humidity: 94%

(2024-07-27)
Temp: 21.22°C
Wind: 6.57 M/S
Humidity: 93%

(2024-07-28)
Temp: 20.99°C
Wind: 4.88 M/S
Humidity: 96%

(2024-07-29)
Temp: 20.93°C
Wind: 2.89 M/S
Humidity: 95%

(2024-07-30)
Temp: 21.00°C
Wind: 4.11 M/S
Humidity: 94%



Go back one page (Alt+Left Arrow)
Right-click or pull down to show history

Weather Dashboard

Enter a City Name only

E.g., New York, London, Tokyo

Search

or

Use Current Location

Tumakuru (2024-07-25)

Temperature: 21.94°C

Wind: 8.34 M/S

Humidity: 79%

Overcast Clouds

5-Day Forecast

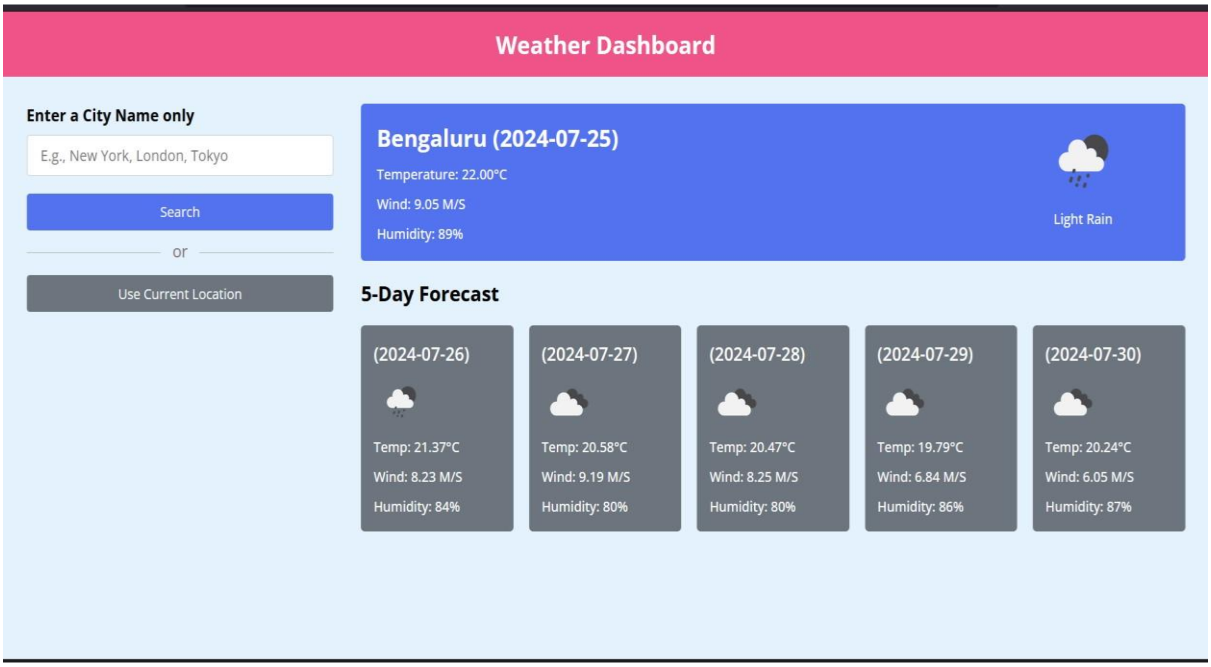
(2024-07-26)
Temp: 21.91°C
Wind: 7.81 M/S
Humidity: 81%

(2024-07-27)
Temp: 21.02°C
Wind: 6.67 M/S
Humidity: 83%

(2024-07-28)
Temp: 21.03°C
Wind: 6.29 M/S
Humidity: 81%

(2024-07-29)
Temp: 20.70°C
Wind: 6.22 M/S
Humidity: 84%

(2024-07-30)
Temp: 20.71°C
Wind: 5.49 M/S
Humidity: 86%



Weather Dashboard

Enter a City Name only

Bengaluru (2024-07-25)

Temperature: 22.00°C

Wind: 9.05 M/S

Humidity: 89%

Light Rain

5-Day Forecast

(2024-07-26)
Temp: 21.37°C
Wind: 8.23 M/S
Humidity: 84%

(2024-07-27)
Temp: 20.58°C
Wind: 9.19 M/S
Humidity: 80%

(2024-07-28)
Temp: 20.47°C
Wind: 8.25 M/S
Humidity: 80%

(2024-07-29)
Temp: 19.79°C
Wind: 6.84 M/S
Humidity: 86%

(2024-07-30)
Temp: 20.24°C
Wind: 6.05 M/S
Humidity: 87%

CONCLUSION AND FUTURE SCOPE

Now a day's there is a big demand of different types of applications, which is because IT has become the main part of our New World. There is a big need of different applications. People want application for every specific task from work to entertainment. We have developed the application "Weather WebApp" which works easy on any given web browser.

The application has been tested and found to be working as per the given criteria. It can be safely concluded that the application possesses a highly efficient UI system and is working properly and meeting to all the requirements of the user. The application gives the user maximum flexibility in the types of touch and other device movements

Every project whether large or small has some limitations no matter how diligently developed. In some cases, limitations are small while in other cases they may be broad also. The new system has got some limitations. Major areas where modifications can be done are as follows:

- Our system does not have weather information for foreign countries or cities.
- There is no provision for complaint handling so further it can be added.

REFERENCE

- http://www.w3schools.com
- http:// www .stackoverflow.com
- http://wikipedia.com
- https://www.geeksforgeeks.org/forecast-weather-project-checktoday-weather-for-any-location
- https://www.codingnepalweb.com/weather-app-project-htmljavascript