



**UM - Internship || WEB DEVELOPMENT || (Full Stack
JavaScript, MEAN, MERN) 1 Month**

INTERNSHIP ID: UMIP23139

WEATHER APP PROJECT REPORT
BASIC PROJECT

Project Title: WEATHER FORECASTING APPLICATION

Developed By: Deeksha Chauhan

TABLE OF CONTENTS

1. Introduction
2. Objectives
3. Technology Stack
4. Implementation details
 - 4.1 HTML Structure
 - 4.2. CSS Styling
 - 4.3. JavaScript Functionality
 - 4.4 Api integration
5. Features
6. Challenges Faced
7. Conclusion
8. Future Enhancements
9. Appendices

1. Introduction

This project is a web-based application that provides users with a 5-day weather forecast for a specified city. Users can input the city name, and the application fetches weather data from the OpenWeatherMap API. The application is designed to be user-friendly, visually appealing, and responsive.

2. Objectives

- To develop a web application that displays the weather forecast for a given city.
- To enable users to search for weather data by entering city names.
- To present the forecast in a clear and organized manner, including temperature, weather conditions, and relevant icons.
- To ensure the application is responsive and accessible on various devices.

3. Technology Stack

- **HTML:** Structure of the web application.
- **CSS:** Styling and layout of the application.
- **JavaScript:** Functionality and interaction, including API calls.
- **OpenWeatherMap API:** Source of weather data.

4. Implementation Details

4.1 HTML Structure

The HTML file contains the following key components:

- A header section for the title and search input.
- An input field for the user to enter a city name.
- A button to trigger the weather data fetch.
- A container to display the weather forecast.

4.2 CSS Styling

The CSS file defines styles for:

- Body and header elements, ensuring a visually appealing design.
- Weather cards, which are styled to be interactive and provide a clear representation of the forecast.
- Responsive design using grid layout for different screen sizes.

4.3 JavaScript Functionality

The JavaScript code handles user interactions and API calls:

- It listens for the button click event to fetch the weather data.
- Validates user input and handles API responses.
- Displays the weather data dynamically in the designated container.

4.4 API Integration

- The application uses the OpenWeatherMap API to fetch 5-day weather data based on user input.
- The API returns data in JSON format, which is then parsed and displayed in the application.

5. Features

- **City Search:** Users can enter any city name to get the weather forecast.
- **Error Handling:** The application displays meaningful error messages when a city is not found or when no input is provided.
- **Responsive Design:** The layout adjusts based on the screen size, ensuring usability on both desktop and mobile devices.
- **Interactive Cards:** Weather data is presented in visually appealing cards that respond to user interactions.

6. Challenges Faced

- Ensuring API calls handled errors gracefully and provided user feedback.
- Designing a responsive layout that works well across different devices and screen sizes.

7. Conclusion

The 5-Day Weather Forecast application successfully meets the project objectives by providing an intuitive interface for users to obtain weather information. With the integration of the OpenWeatherMap API, users can dynamically access weather data, making it a practical tool for daily use. Future enhancements may include adding more detailed forecasts, integrating additional weather data, or allowing users to save their favorite locations.

8. Future Work

- Implementing user location detection for default weather data.
- Adding more detailed weather statistics such as humidity, wind speed, and pressure.

- Enhancing the user interface with animations and transitions.

9. Appendices

Appendix A: Home Page And API integration Structure

Appendix B: Screenshots of the Application UI

Source Code Of The Home Page Saved As The Name Of Weather.Html File

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>10-Day Weather Forecast</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f8ff;
      color: #333;
      padding: 20px;
    }

    header {
      background-color: #0066cc;
      color: white;
      text-align: center;
      padding: 1rem 0;
    }

    header input {
```

```
width: 300px;
height: 40px;
border: 2px solid blue;
font-size: larger;
margin-right: 10px;
}
```

```
button {
  height: 45px;
  width: 100px;
  border: 2px solid blue;
  background-color: #4CAF50;
  color: white;
  font-size: large;
}
```

```
.weather-container {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
  margin-top: 20px;
}
```

```
.weather-card {
  background-color: #e7f3fe;
  border-radius: 10px;
  padding: 15px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
.weather-card h3 {  
  margin: 0 0 10px;  
}
```

```
.weather-card img {  
  vertical-align: middle;  
}
```

```
.big {  
  width: 99%;  
  height: 95vh;  
  float: left;  
  border: 2px solid grey;  
  overflow-y: scroll;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="big">
```

```
<header>
```

```
<h1>5-Day Weather Forecast</h1>
```

```
<input type="text" id="city-input" placeholder="Enter city">
```

```
<button id="search-btn">Search</button>
```

```
</header>
```

```
<div id="weather-container" class="weather-container"></div>
```

```
</div>
```

```
<script>
```

```
const apiKey = '7528a958807c8245cdce61f51a1b98d4';
```

```
document.getElementById('search-btn').addEventListener('click', getWeatherForecast);
```

```
async function getWeatherForecast() {
```

```
    const city = document.getElementById('city-input').value;
```

```
    const apiUrl =
```

```
    `https://api.openweathermap.org/data/2.5/forecast?q=${city}&units=metric&appid=${apiKey}`;
```

```
    try {
```

```
        const response = await fetch(apiUrl);
```

```
        if (!response.ok) throw new Error(`City not found`);
```

```
        const data = await response.json();
```

```
        displayWeatherForecast(data);
```

```
    } catch (error) {
```

```
        document.getElementById('weather-container').innerHTML = `<p>Error: ${error.message}</p>`;
```

```
    }
```

```
}
```

```
function displayWeatherForecast(data) {
```

```
    const weatherContainer = document.getElementById('weather-container');
```

```
    weatherContainer.innerHTML = '';
```

```
    const dailyData = data.list.filter((reading, index) => index % 8 === 0).slice(0, 10);
```

```
    dailyData.forEach(day => {
```

```
        const date = new Date(day.dt_txt);
```



```
const dayName = date.toLocaleDateString('en-US', { weekday: 'long' });

const weatherHTML = `

  <div class="weather-card">

    <h3>${dayName}</h3>

    <p>Temperature: ${day.main.temp}°C</p>

    <p>Condition: ${day.weather[0].description}</p>

  </div>

`;

weatherContainer.innerHTML += weatherHTML;

});

}

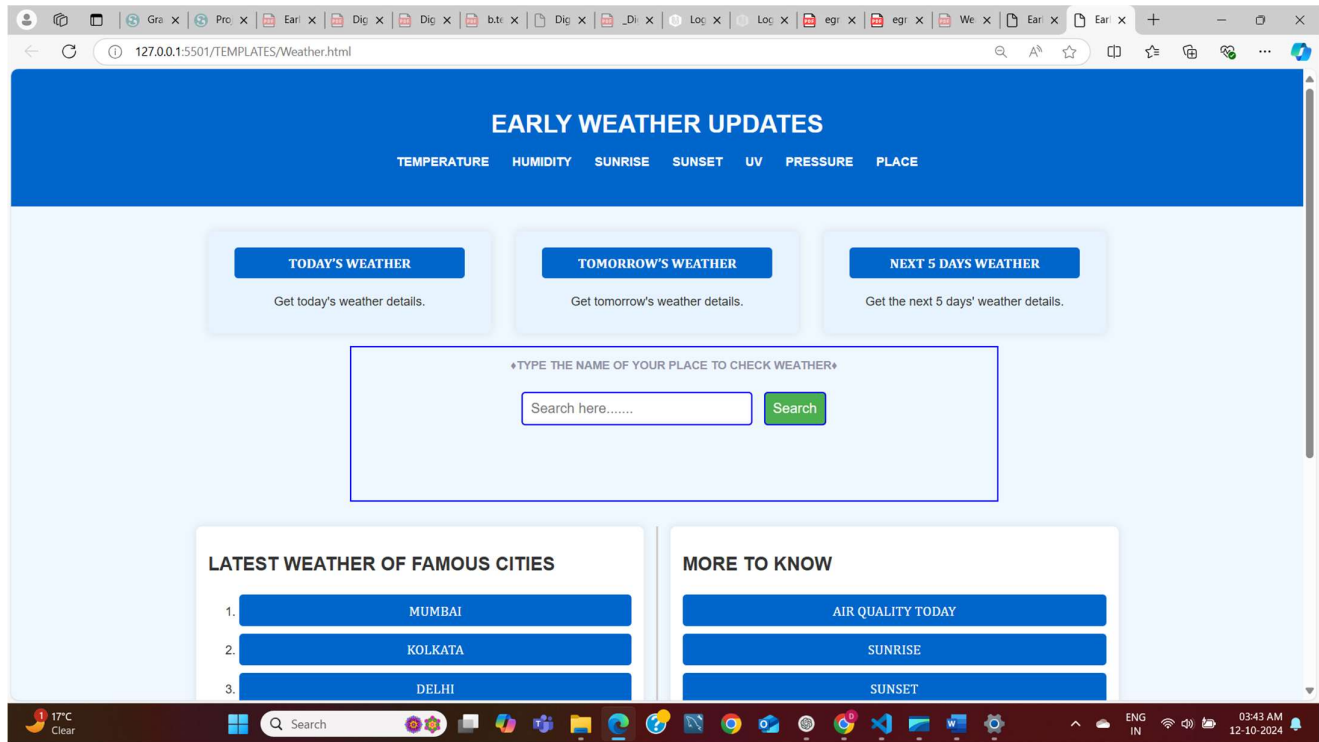
</script>

</body>

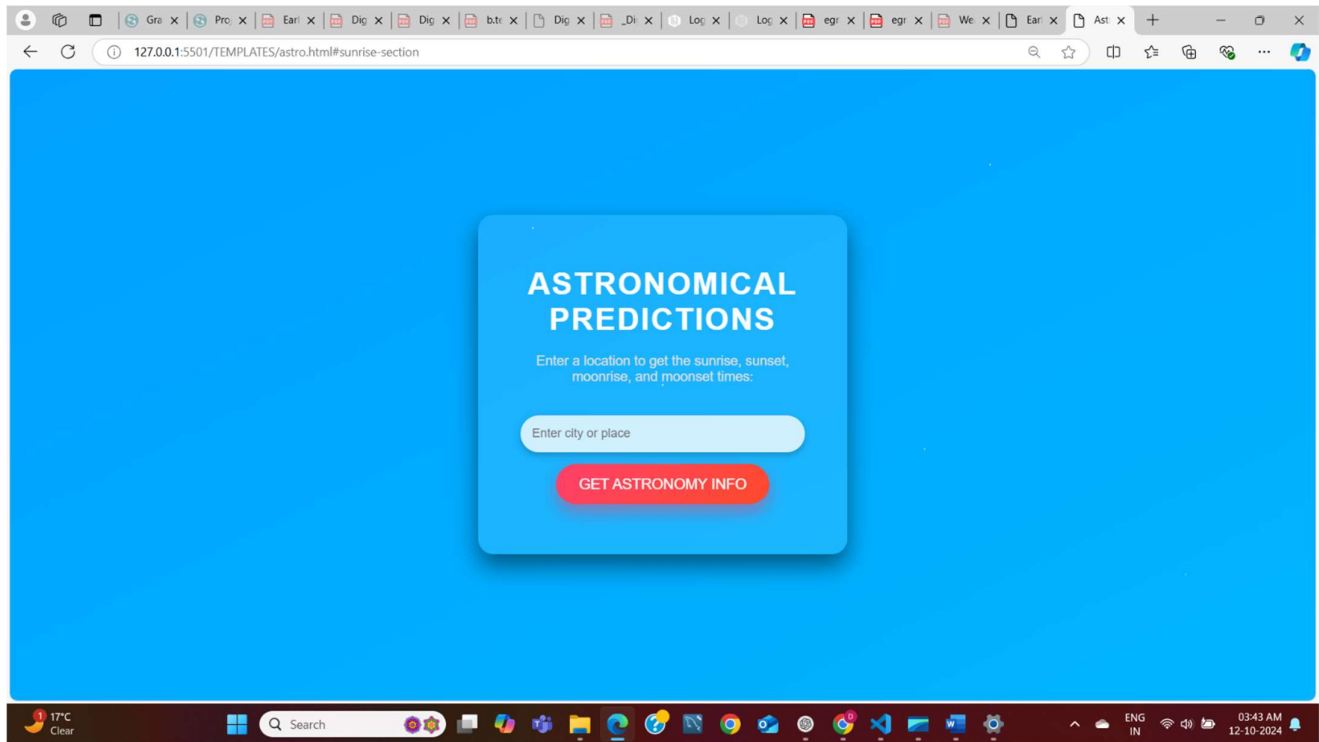
</html>
```

SCREENSHOTS OF THE APPLICATION UI

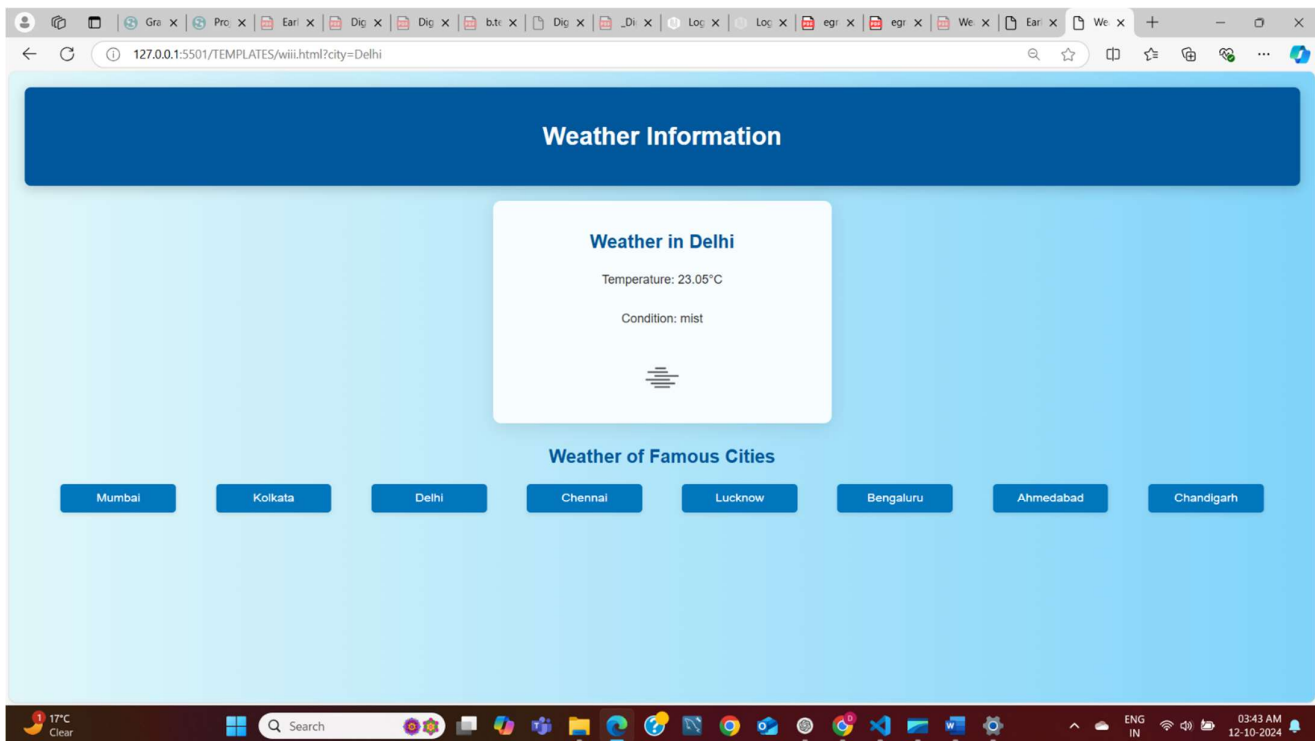
HOME PAGE VIEW



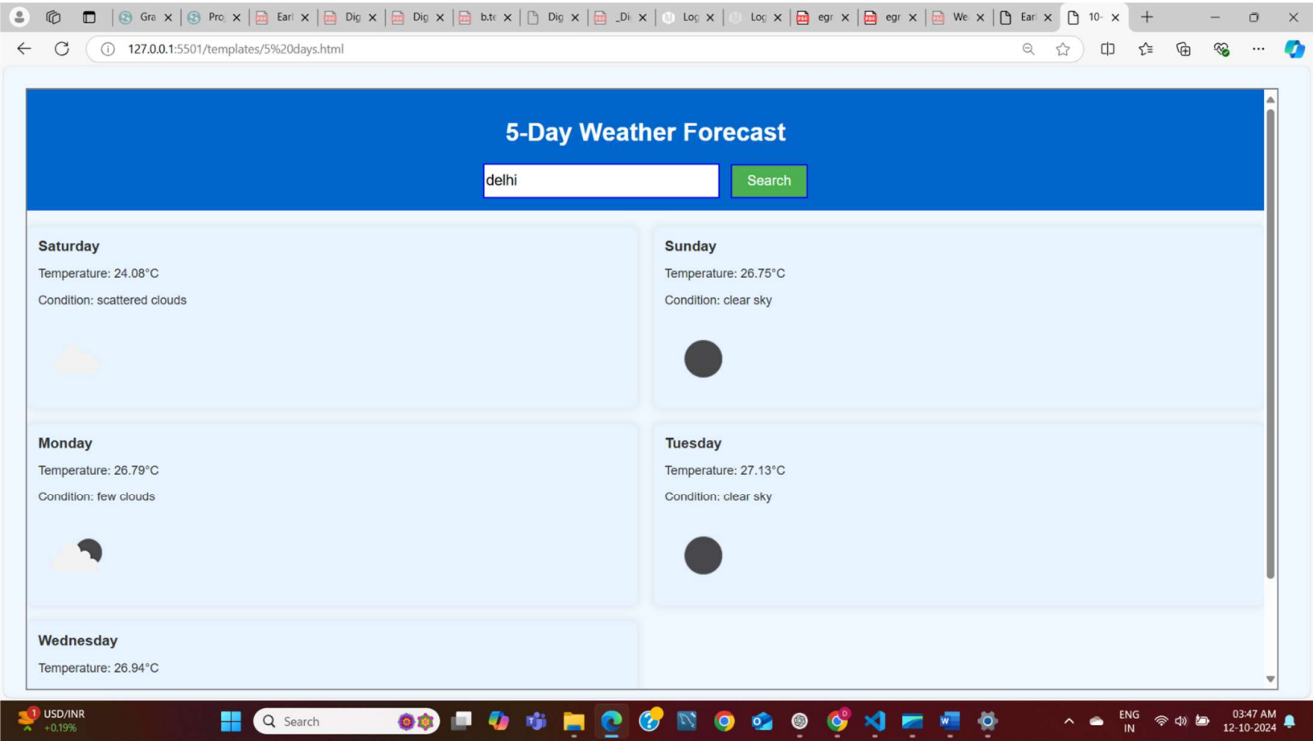
ASTRONOMICAL PREDICTIONS LIKE SUNRISE MOONRISE SUNSET



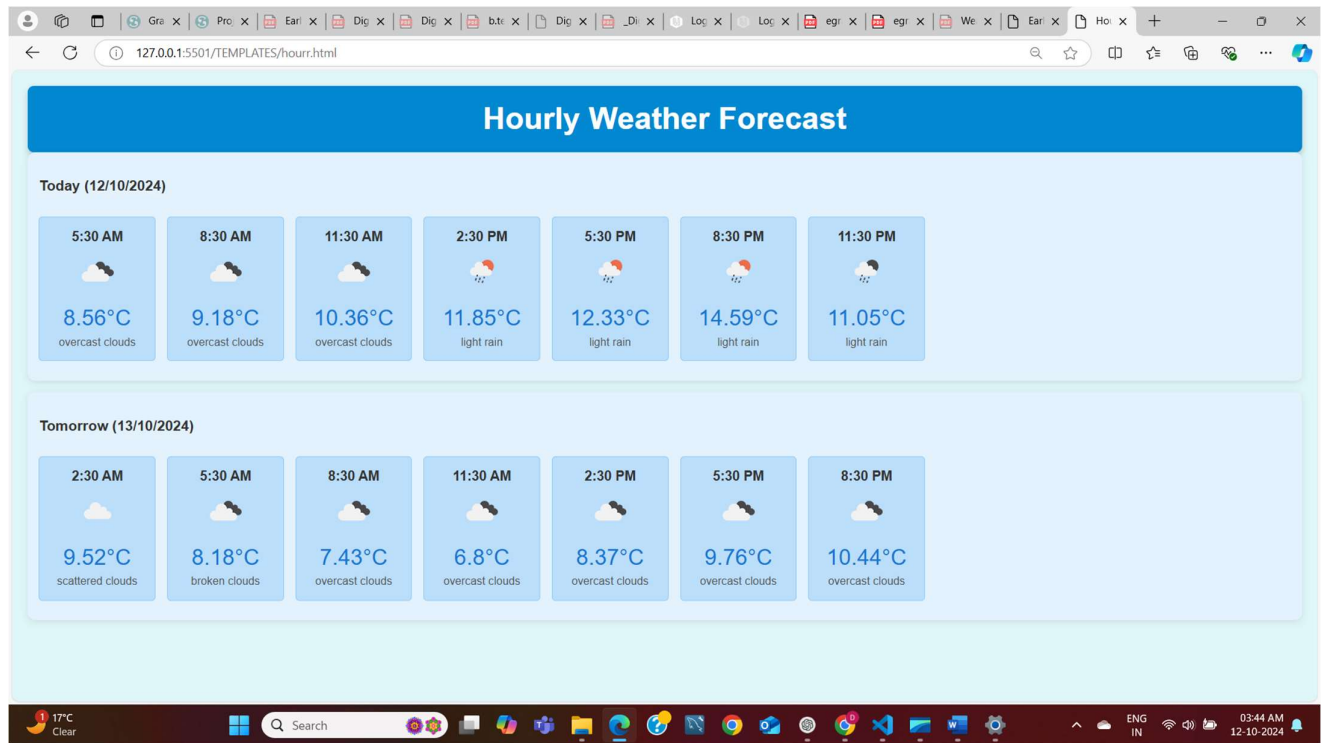
KNOW THE WEATHER OF FAMOUS CITIES



NEXT 5 DAYS WEATHER FORECAST



CURRENT DAY HOURLY FORECAST



TOMORROW'S WEATHER

