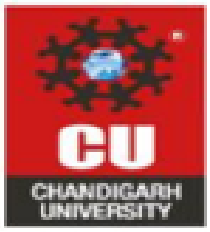



# CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

## UNIVERSITY INSTITUTE OF COMPUTING

## BRANCH: BACHELOR OF COMPUTER APLLICATION

**Weather Forecast Web App – Mini Project**

**SUBMITTED BY-**

**NAME: JASLEEN KAUR**

**UID: 24BCA20022**

**SEMESTER: 2nd**

**SECTION: 24 BCV-1**

**SUBJECT: WEB DESIGNING**

**CODE: 24CAH-153**

**SUBMITTED TO-**

MR. JAGVINDER SINGH

# 📘 1. INTRODUCTION

This project is a simple web application that shows real-time weather information for a given city. It uses HTML, CSS, JavaScript, and the Open Weather Map API to fetch live data such as temperature, humidity, and weather conditions. In today's fast-paced world, weather forecasting plays a vital role in planning daily activities, travel, agriculture, and even large-scale events. People often rely on web applications or mobile apps to get real-time weather updates and forecasts. This mini project aims to build a simple yet functional weather forecast web app using basic frontend technologies such as HTML, CSS, and Java Script. The application allows users to enter the name of a city and get the current weather conditions for that location. It fetches data from the Open Weather Map API, processes it, and displays the results in a user-friendly format.

This project also emphasizes the importance of responsive user interfaces and seamless API communication. It serves as an excellent starting point for beginners who are learning how to interact with APIs and structure modern web applications using HTML, styling with CSS, and adding interactivity using JavaScript.

# 2. ABSTRACT

The aim of this project is to build a lightweight and user-friendly web application that allows users to check weather updates from anywhere. The app connects to a weather API and provides current weather data in real-time. It's a great example of how frontend technologies can be integrated with external APIs to create practical applications. The Weather Forecast Web Application is a lightweight, interactive tool designed to provide users with real-time weather information for any city in the world. Built using HTML, CSS, and JavaScript, the project integrates with the Open Weather Map API to fetch live weather data and present it in a clean, user-friendly interface. This mini project highlights how simple web technologies can be used to develop practical, real-world applications.

The key objective of this project is to demonstrate how client-side scripting and API integration can work together to retrieve and display data dynamically. This real-time functionality provides a seamless user experience and helps users make informed decisions based on current weather conditions.

# 💻 3. SYSTEM CONFIGURATION

To successfully develop and run the Weather Forecast Web App, certain system and software requirements must be met. These ensure the application performs as expected and provides a smooth user experience.

| |
|---|
| ▯ frontend: html, css, javascript |
| ▯ A pi used: open weather map api |
| ▯ browser compatibility: modern browser, |
| ▯ editor used: any code editor (vs code recommended) |
| ▯ internet connection: required for fetching api data |

This configuration ensures that the application runs efficiently in a typical development environment and can be easily deployed or shared for demonstration purposes.

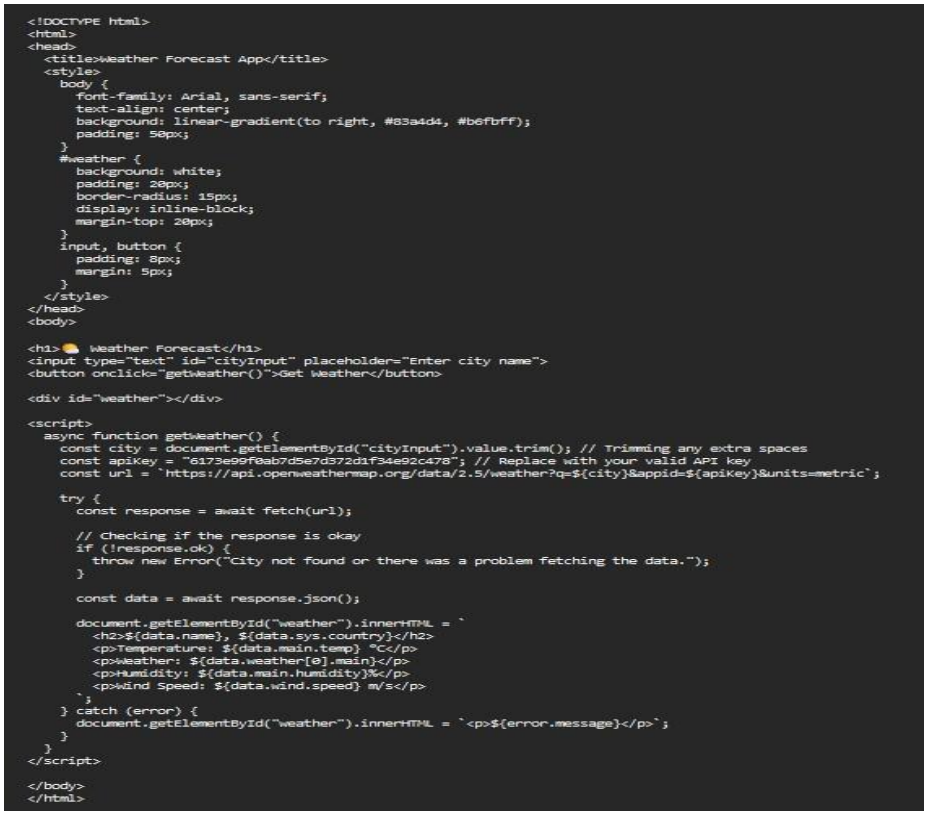
# 💡 4: Code

The following code demonstrates the complete implementation of the Weather Forecast Web App. It includes the HTML structure for the layout, CSS styling for a clean interface, and JavaScript to handle user input and fetch real-time weather data from the Open Weather Map API.

```
<!DOCTYPE html>
<html>
<head>
  <title>Weather Forecast App</title>
  <style>     body {       font-family: Arial, sans-serif;       text-
align: center;       background: linear-gradient(to right,
#83a4d4, #b6fbff);       padding: 50px;
    }
    #weather {
background: white;
padding: 20px;
border-radius: 15px;
display: inline-block;
margin-top: 20px;
    }
    input, button {
padding: 8px;       margin:
5px;
    }
  </style>
```

```
</head>
<body>

<h1>  Weather Forecast</h1>
<input type="text" id="cityInput" placeholder="Enter city name">
<button onclick="getWeather()">Get Weather</button>

<div id="weather"></div>

<script>  async function
getWeather() {
  const city = document.getElementById("cityInput").value.trim(); // Trimming any extra spaces  const apiKey = "6173e99f0ab7d5e7d372d1f34e92c478"; // Replace with your valid API key
  const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;  try {
    const response = await fetch(url);

    // Checking if the response is okay  if (!response.ok) {  throw new Error("City not found or there was a problem fetching the data.");
    }

    const data = await response.json();

    document.getElementById("weather").innerHTML = `
      <h2>${data.name}, ${data.sys.country}</h2>
      <p>Temperature: ${data.main.temp} °C</p>
      <p>Weather: ${data.weather[0].main}</p>
      <p>Humidity: ${data.main.humidity}%</p>
```

```
      <p>Wind Speed: ${data.wind.speed} m/s</p>
    `;
  } catch (error) {
    document.getElementById("weather").innerHTML = `<p>${error.message}</p>`;
  }
 }
</script>

</body>
</html>
```

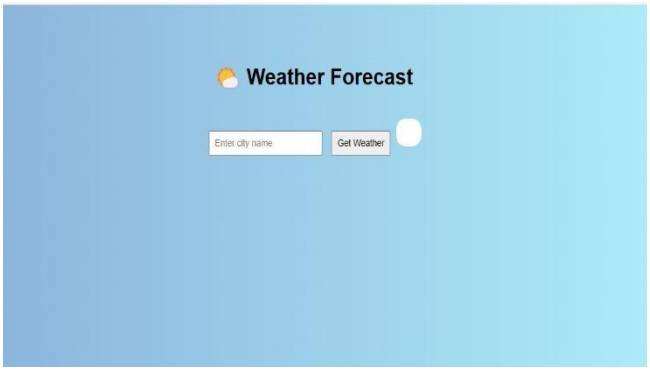



## 5: Output

When the user enters a city and clicks "Get Weather", the app displays:

- City and Country
- Temperature

- Weather Condition
- Humidity
- Wind Speed





# 6: REFERENCES

The development of this Weather Forecast Web App was supported by various online resources that provided useful guidance, tutorials, and documentation. The key references include:


- OpenWeatherMap API Documentation – For fetching real-time weather data.

- W3Schools – JavaScript Fetch API – For understanding how fetch() works in JavaScript.
- MDN Web Docs – JavaScript – Comprehensive JavaScript documentation.
- MDN Web Docs – HTML & CSS – For HTML structure and CSS styling.
- Google Fonts – To enhance typography in web design (optional).
- Stack Overflow – Helpful community for solving coding issues during development.
- CodePen – For testing and sharing front-end code snippets.

These resources played an important role in completing the project effectively and improving coding and design skills.

# ✅ 7: CONCLUSION

This project demonstrates how to create a real-time weather application using basic web technologies and an external API. It's a simple but powerful example of combining frontend skills with data fetching to build interactive and useful web tools. In conclusion, this Weather Forecast Web App demonstrates how real-time data from external APIs can be effectively used in modern web development. By integrating HTML, CSS, and JavaScript with the OpenWeatherMap API, the project successfully delivers dynamic weather updates for any city input by the user.

This mini project helped in understanding essential web development concepts such as asynchronous data fetching, DOM manipulation, and

responsive UI design. It also strengthened practical skills in working with third-party APIs and handling real-world data formats like JSON.

Overall, this project not only serves as a functional weather checking tool but also as a foundational learning experience for future web-based applications. With further enhancements, such as adding weather icons, forecast charts, or local storage, this app can be extended to provide even richer user experiences.