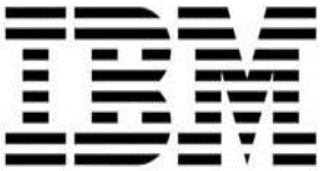


COLLEGE CODE :9222

COLLEGE NAME:THENI KAMMAVAR SANGAM COLLEGE OF TECHNOLOGY

DEPARTMENT:B.TECH IT

STUDENT NM-ID: 64A750175CBE6C9AC8844420357E64BB

ROLL NO:922223205053

DATE:17.10.2025

Completed the project named as Phase5

TECHNOLOGY PROJECT NAME: WEATHER DASHBOARD

SUBMITTED BY,

NAME:WASHIM KHAN M

MOBILE NO:7010092516

Project Demonstration & Documentation

Title: Weather Dashboard

Final Demo Walkthrough – Weather Dashboard

1. Introduction:

- "Hi everyone, today I'll walk you through my Weather Dashboard."
- "This app provides real-time weather updates for any city around the world."

2. Search Functionality: "Let's start by searching for a city. I'll type in *'New York'*."

- "As you can see, the dashboard fetches and displays the current weather conditions."

Highlight:

- City name
- Current temperature
- Weather condition (e.g., sunny, cloudy)
- Icon representing the weather

3. Additional Weather Details:

- "Below the main temperature, we show other details like:"
  - Humidity
  - Wind speed
  - 'Feels like' temperature
  - Date and time

Bonus if added: Sunrise/Sunset, Pressure

4. 5-Day Forecast (if included):

- "Here's the 5-day forecast, giving a quick glance at upcoming weather."
- "Each day shows the expected high/low temperatures and conditions."

5. UI Overview:

- "The interface is clean and responsive."
- "Icons and colors adjust based on the weather condition—for example, blue for rain, orange for sunny."

Responsive Design? Show it briefly on a smaller screen or mobile layout.

6. Tech Stack (Optional Slide or Mention):

- "This app was built using:" ○ HTML/CSS/JavaScript (or React, etc.) ○ OpenWeatherMap API (or whichever API used) ○ Optional: Tailwind, Bootstrap, Chart.js, etc.

7. Final Thoughts:

- "This Weather Dashboard is useful for anyone needing quick weather updates, and it can be expanded further with features like location auto-detection or severe weather alerts."

Project Report :

Introduction:

The Weather Dashboard is a simple Python-based command-line application that provides real-time weather information for any city. It fetches current weather data and a 3-day forecast using the free wttr.in API.

Objectives:

- Display current temperature, weather condition, wind speed, humidity, and feels-like temperature.
- Show a concise 3-day weather forecast.
- Create an easy-to-use terminal interface for quick weather checks.

Tools & Technologies:

- Python
- Requests library for API calls
- wttr.in API for weather data

Program Overview:

The program prompts the user to enter a city name, fetches weather data from the API, and displays current conditions and a short forecast. It handles errors gracefully and allows multiple queries until the user exits.

Conclusion:

The Weather Dashboard is a functional and lightweight tool to get instant weather updates via the terminal. It can be enhanced with additional features or a graphical interface in the future.

Program:

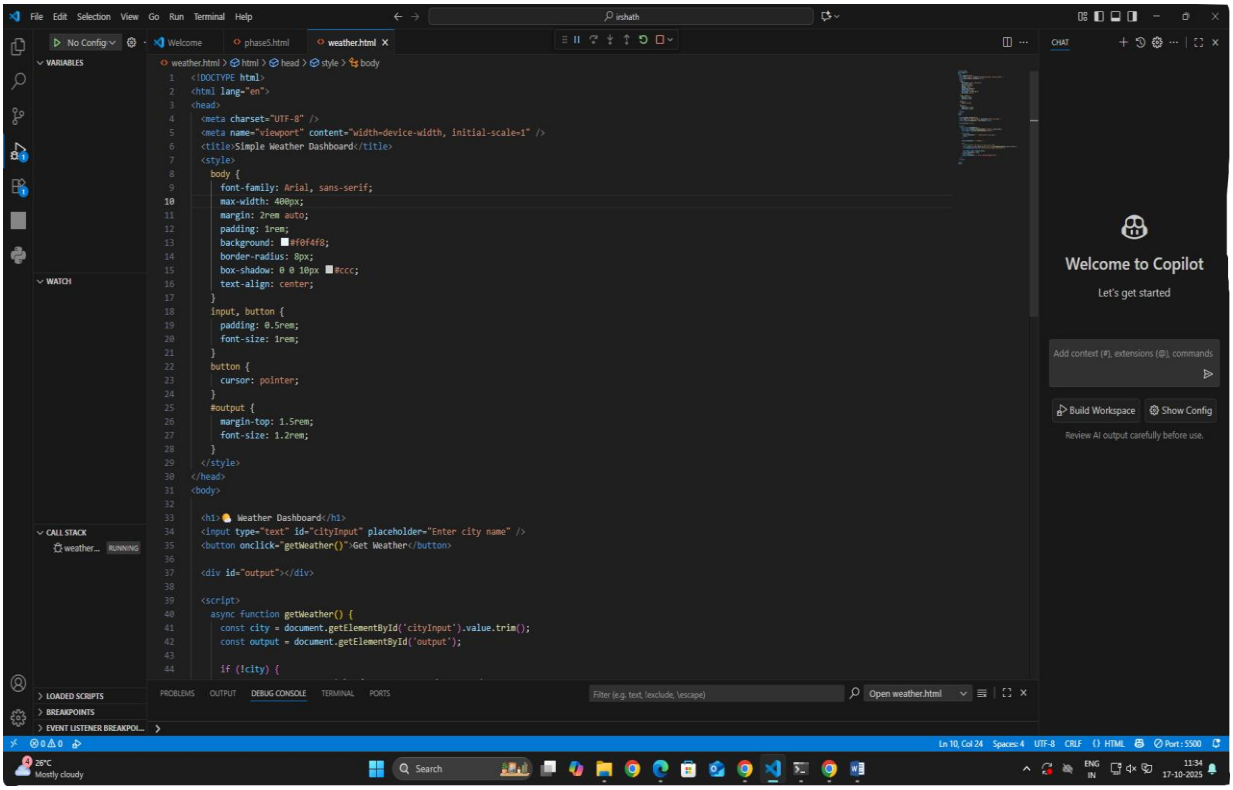
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Simple Weather Dashboard</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      max-width: 400px;
      margin: 2rem auto;
      padding: 1rem;
      background: #f0f4f8;
      border-radius: 8px;
      box-shadow: 0 0 10px #ccc;
      text-align: center;
    }
```

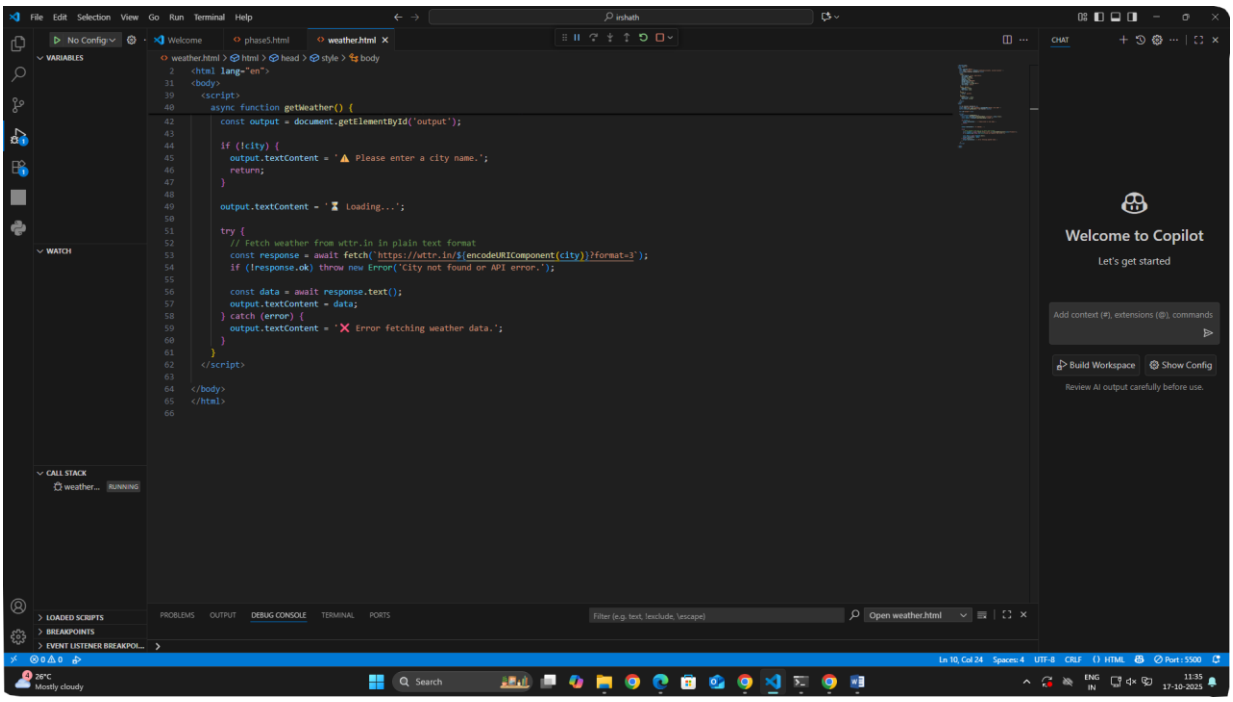
```
    input, button {
      padding: 0.5rem;
      font-size: 1rem;
    }
    button {
      cursor: pointer;
    }
    #output {
      margin-top: 1.5rem;
      font-size: 1.2rem;
    }
  </style>
</head>
<body>

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

  <div id="output"></div>

  <script>
```

```
async function getWeather() {
  const city = document.getElementById('cityInput').value.trim();
  const output = document.getElementById('output');

  if (!city) {
    output.textContent = '⚠ Please enter a city name.';
    return;
  }

  output.textContent = '⏳ Loading...';

  try {
    // Fetch weather from wttr.in in plain text format
    const response = await fetch(`https://wttr.in/${encodeURIComponent(city)}?format=3`);
    if (!response.ok) throw new Error('City not found or API error.');

    const data = await response.text();
    output.textContent = data;
  } catch (error) {
```

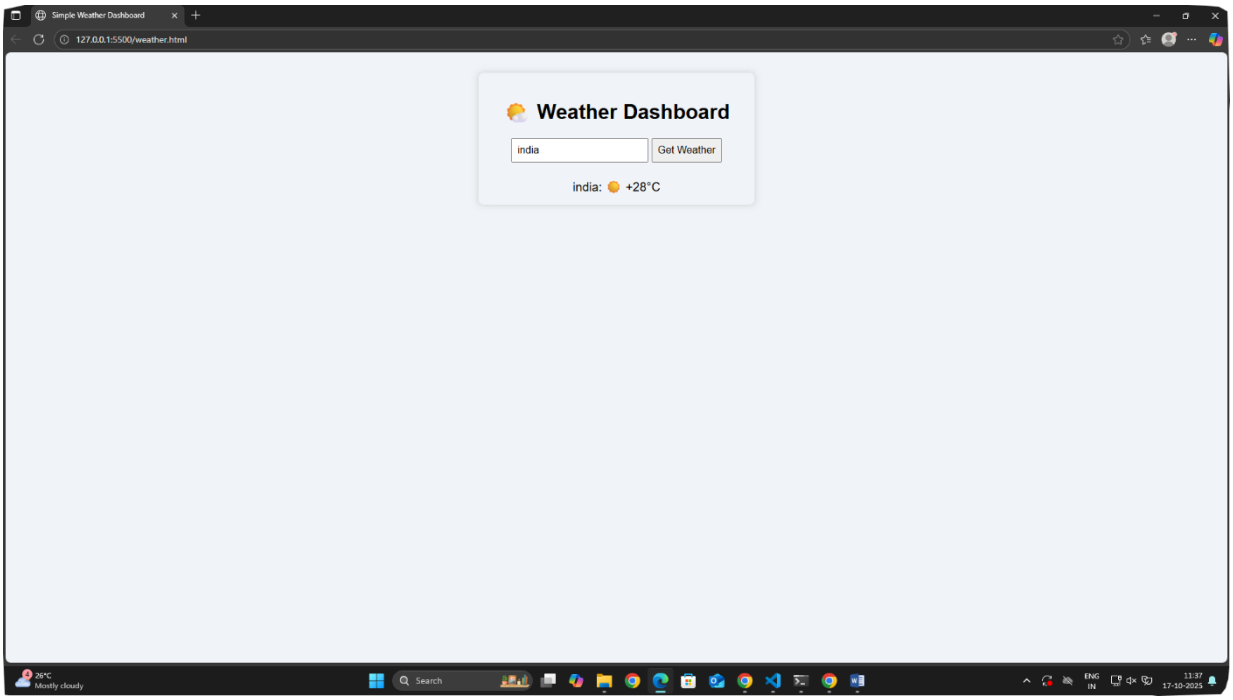
```
      output.textContent = '❌ Error fetching weather data.';
    }
  }
</script>

</body>
</html>
```

Screenshot/API Documentation:




File Edit Selection View Go Run Terminal Help
weather.html
Welcome to Copilot
Let's get started
Build Workspace
Show Config

OUTPUT:


Simple Weather Dashboard
127.0.0.1:5500/weather.html
Weather Dashboard
india
Get Weather
india: +28°C

Overview

wttr.in is a free weather service that provides weather information via HTTP requests in various formats, including JSON. It is designed for easy integration into command-line apps and scripts.

Endpoint Used:

http://wttr.in/{city}?format=j1

- **{city}**: Name of the city or location for which weather data is requested.
- **format=j1**: Requests the data in JSON format.

Response Structure (Key Parts):

- **current_condition**: List containing current weather details like:
  - temp_C: Current temperature in Celsius.
  - weatherDesc[0].value: Text description of weather condition (e.g., "Partly cloudy").
  - windspeedKmph: Wind speed in kilometers per hour.
  - humidity: Current humidity percentage.
  - FeelsLikeC: 'Feels like' temperature in Celsius.

weather:

List containing weather forecasts for upcoming days, each with:

- date: Date of the forecast. ○ maxtempC: Maximum temperature of the day. ○ mintempC: Minimum temperature of the day.
- hourly[4].weatherDesc[0].value: Midday weather condition.

Usage Notes:

- The API is public and does not require authentication or API keys.
- It returns detailed weather data, which may vary slightly depending on the location queried.
- The data is updated frequently but not guaranteed to be realtime to the second.

Challenges & Solutions:

Challenges:

1. **Handling Invalid City Inputs**

 Users might enter misspelled or nonexistent city names, causing API errors or no data returned.

2. **Network and API Errors**

 Network issues or API downtime could lead to failed requests or delayed responses.

3. **Parsing Complex JSON Data**

 The weather data from wttr.in is nested and requires careful extraction of relevant information.

4. **Maintaining a Clean User Interface in Terminal**

- Ensuring the terminal output is clear and user-friendly, especially after multiple queries.

Solutions:

1. **Input Validation and Error Messages**

- The program checks API responses and informs users when data can’t be retrieved, prompting them to try again.

2. **Exception Handling**

- Implemented try-except blocks around API calls to catch network-related errors gracefully.

3. **Structured Data Extraction**

- Carefully navigated JSON structure to reliably extract current weather and forecast details.

4. **Screen Clearing and Formatting**

- Used terminal clear commands to refresh the display for each query, keeping output organized and readable.

Github README LINK: