**Assignment #2**

**Web Engineering**

****

**DEPARTMENT OF SOFTWARE ENGINEERING**

**Submitted To: Ma’am Laiba Imran**

**Submitted By:** (5-E)

1.Rehab Imtiaz (22I-8769)

# Weather Dashboard Project

## Introduction

This project is a responsive Weather Dashboard application designed to provide current weather data, forecasts, and interactive features like filtering, sorting, and searching weather conditions. The app uses the OpenWeather API to fetch weather data and displays it in various visual formats like charts and tables.

## Features

Dashboard: Displays weather information based on city name or coordinates.  
Chart Integration: Includes bar, line, and doughnut charts to visualize weather data.

Search and Filtering: Allows searching through weather data by city name, latitude, and longitude.  
Chatbot Integration: A chatbot that responds to weather-related questions.  
Responsive Design: Optimized for different screen sizes.

## Project Structure

1. HTML: Provides the basic structure and layout of the Weather Dashboard.
2. CSS: Defines the visual appearance and responsiveness of the application.
3. JavaScript: Handles data fetching, dynamic content updates, chart rendering, and interaction logic.
4. External Libraries: Responsive design, and Chart.js for chart rendering.

## Prerequisites

1. A web browser (e.g., Chrome, Firefox).
2. Node.js installed on your machine (for running a local development server).
3. Live Server Extension (for VS Code) or any alternative server like httpserver.

## Setup and Installation

1. Download and Extract the Project:  
    Extract the downloaded ZIP file into your desired location. The extracted folder contains all necessary project files, including HTML, CSS, JavaScript, and any supporting assets.
2. Open the Project Folder: Open the project folder in a code editor like VS Code.
3. Installing Dependencies: Ensure you have an active internet connection for fetching dependencies like Bootstrap and Chart.js.
4. Running the Project: If using VS Code: Right click on the dashboard.html file and select Open with Live Server. This will launch the project in your default web browser on a local server. Alternatively, you can use httpserver:  
    Install httpserver globally by running: npm install g httpserver  
    Navigate to your project folder in the terminal and run:  
    httpserver  
    Open the URL shown in your terminal to access the app.

## Add API Keys

1. Get Your Open Weather API Key: Visit Open Weather and create an account.  
 Generate an API key and update the dashboard.js file as follows JavaScript  
 const apiKey = 'YOUR\_API\_KEY'; // Replace with your OpenWeather API key  
   
2. Get Your Gemini API Key: Obtain an API key and update the dashboard.js file as follows: JavaScript  
 const geminiApiKey = 'YOUR\_API\_KEY'; // Replace with your Gemini API key

## Usage

1. Search Weather by City: Enter a city name to view current weather conditions and a 5day forecast.

2. Search by Latitude and Longitude: Provide coordinates to fetch weather data for a specific location.

3. Interactive Charts: Explore the temperature trends using bar, line, and doughnut charts.

4. Sort and Filter Data: Sort the weather data by temperature or filter out rainy days.

5. Chatbot Interaction: Use the chatbot to ask questions about the weather in different cities.

## Tech Stack

HTML5: Structure and semantic layout.  
 CSS3: Styling and responsive design.  
 JavaScript: Functionality and interaction logic.  
 Chart.js: Data visualization and chart rendering.  
 Open Weather API: Weather data source.