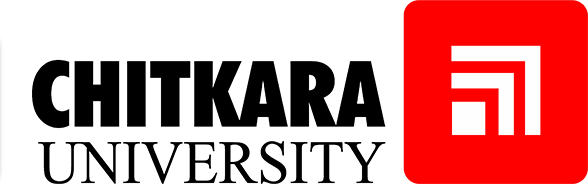
**Front End Engineering-II**

Project Report

Semester-IV (Batch-2022)

**Weather App**



**Supervised By: Submitted By:**

Mr. Raveesh Samkaria Sanpreet Singh

2210990790

G-12

**Department of Computer Science and Engineering Chitkara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

# Abstract

"Pong Game" is an exciting web application designed to bring the classic game of ping pong to users in an interactive way. Utilizing HTML, CSS, and JavaScript for the frontend, Pong game provides a seamless platform for players to enjoy thrilling matches. It incorporates libraries like Phaser.js for smooth animations and responsive design, creating an immersive gaming experience.

Ping Pong Game aims to deliver both entertainment and competition, allowing players to challenge themselves or compete with friends. With its engaging interface and intuitive controls, Pong game caters to players of all skill levels. Whether you're a seasoned pro or a casual gamer, Ping Pong offers endless fun and excitement.

****

# INDEX

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page Number(s)** |
| 1 | Introduction | 4 |
| 2 | Problem Statement | 5 |
| 3 | Software Requirements | 5 |
| 4 | Proposed Design | 6-12 |
| 5 | Results | 12-15 |
| 6 | References | 15 |

# Introduction

"Welcome to WeatherWise, your gateway to accurate and intuitive weather forecasting. In an era where staying informed is essential, WeatherWise provides a seamless and comprehensive experience for users seeking up-to-date weather information. This introduction offers a glimpse into the world of WeatherWise, emphasizing its precision, user-friendly features, and the convenience it brings to those navigating daily activities impacted by weather condition.

**1.1 Background:**

Amidst the digital landscape of constant connectivity and entertainment, the necessity for engaging platforms that seamlessly integrate education with enjoyment becomes evident. Introducing WeatherWise, designed to meet this growing demand by amalgamating the realms of fun and learning. In a world saturated with online activities, WeatherWise stands out by offering a diverse array of features, from detailed forecasts to interactive maps, catering to users with varying interests and needs. Moreover, as social interactions increasingly unfold in virtual realms, WeatherWise serves as a hub for sharing weather-related insights, fostering a sense of community and collaboration among users. In an age where weather impacts every facet of daily life, WeatherWise emerges as a vital tool, empowering users to make informed decisions and stay connected to the world around them.

**1.2 Objectives:**

At WeatherWise, our primary objective is to provide users with a comprehensive and engaging platform for accessing accurate weather forecasts and information. We strive to entertain, inform, and connect users through our intuitive interface and diverse range of features. Our aim is to foster a community of weather enthusiasts by offering interactive tools, insightful content, and opportunities for collaboration. We prioritize accessibility and user-friendly design to ensure that individuals of all backgrounds and expertise levels can navigate our app with ease and derive value from its functionalities.

**1.3 Significance:**

WeatherWise goes beyond mere forecast delivery—it's a dynamic platform that informs, empowers, and connects users. By providing accurate weather data and insightful analysis, WeatherWise equips individuals with the knowledge they need to plan and navigate their lives effectively. Its significance lies in promoting safety, facilitating informed decision-making, and fostering a sense of community among users worldwide. Whether for daily planning or long-term preparedness, WeatherWise enriches lives and strengthens connections through its commitment to reliable weather information and user engagement.

# Problem Statement

In the realm of weather forecasting apps, a common issue arises—a lack of diversity in features and personalized experiences, which can limit user engagement and utility. Many weather apps provide basic forecasts without offering interactive elements or tailored insights, resulting in mundane and unfulfilling interactions. There's a growing need for a versatile, engaging, and informative weather app that provides users with a wide range of customizable features and personalized content. Moreover, the absence of adaptability and customization options in existing weather apps can hinder user satisfaction and retention, as they struggle to find an app that aligns with their individual preferences and lifestyle needs.

# Software Requirements

1. **Integrated Development Environment (IDE):** 
   * Visual Studio Code (VS Code) for code editing and project management.

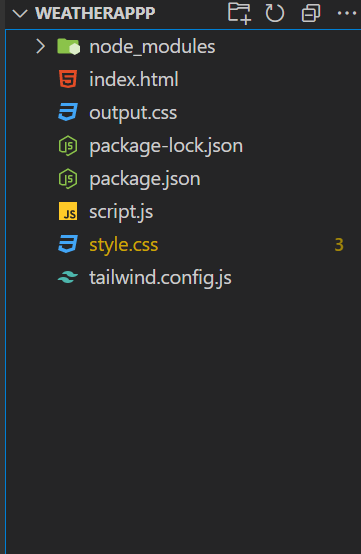
1. **Frontend Technologies:** 
   * HTML: Markup language for structuring the web application.
   * CSS: Styling language for enhancing the presentation and layout.
   * JavaScript (JS): Programming language for implementing interactive features and quiz logic.

1. **User Interface (UI) Framework:** 
   * Tailwind CSS : Utility-first CSS framework for building responsive and customizable user interfaces.

1. **Version Control:** 
   * Git: Distributed version control system for tracking changes in the project codebase and collaborating with team members effectively.

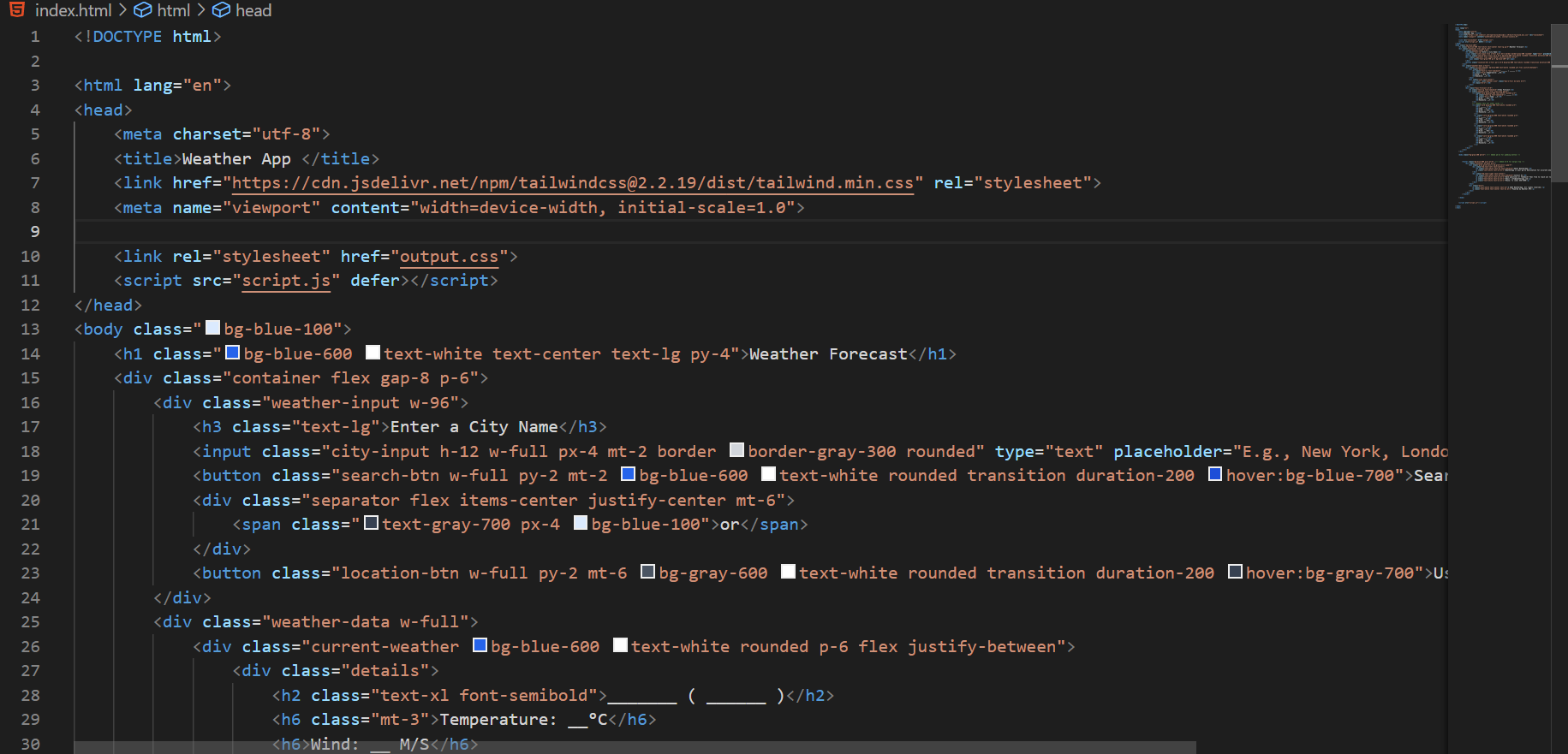
# Proposed Design

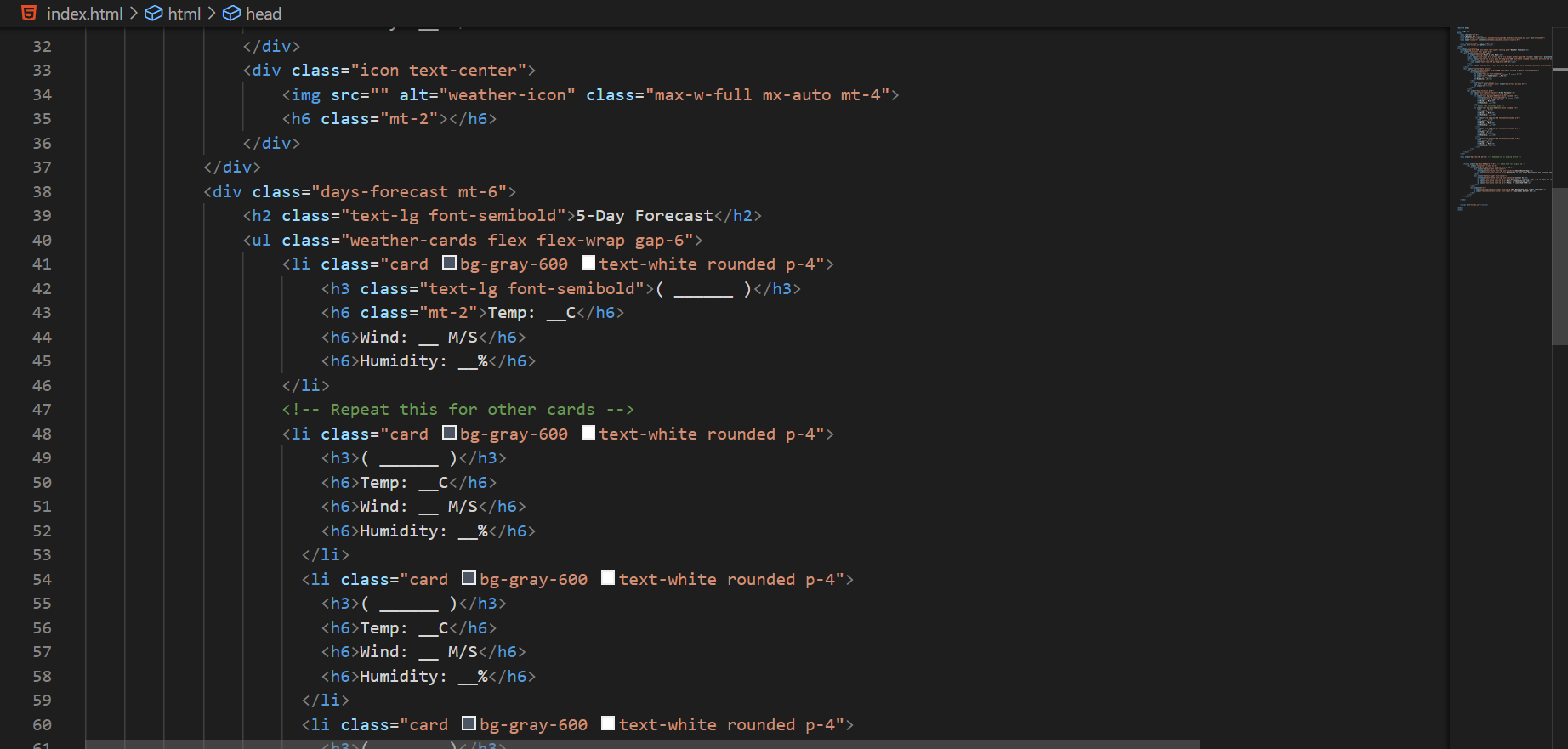
* **User Interface Design:** Employ Tailwind CSS for a modern and responsive layout, focusing on simplicity and clarity.
* **Frontend Development:** Develop the weather application using HTML, CSS, and JavaScript. Leverage HTML5 semantics for structural elements, CSS for styling, and JavaScript for dynamic functionalities.
* **User Experience Optimization:** Prioritize user interaction by providing real-time feedback on answer selection. Implement smooth transitions and animations using Tailwind CSS utilities to enhance the overall user experience.
* **Testing and Quality Assurance:** Conduct rigorous testing, including both manual and automated tests, to ensure the reliability and functionality of the weather app application across different browsers and devices.
* **Documentation and Deployment:** Provide detailed documentation. Deploy on web server with domain. Maintain and update documentation regularly.
  1. **File Structure**

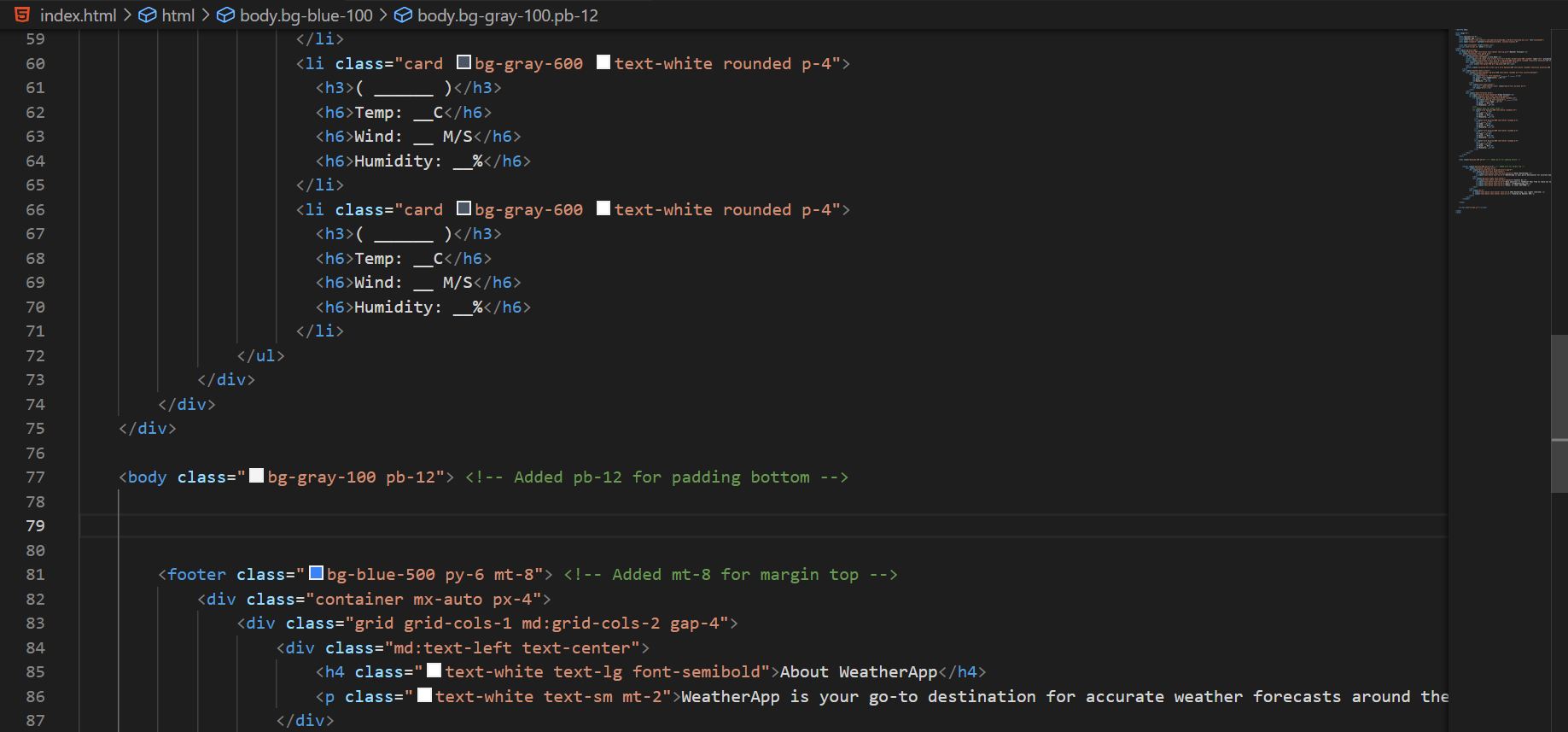


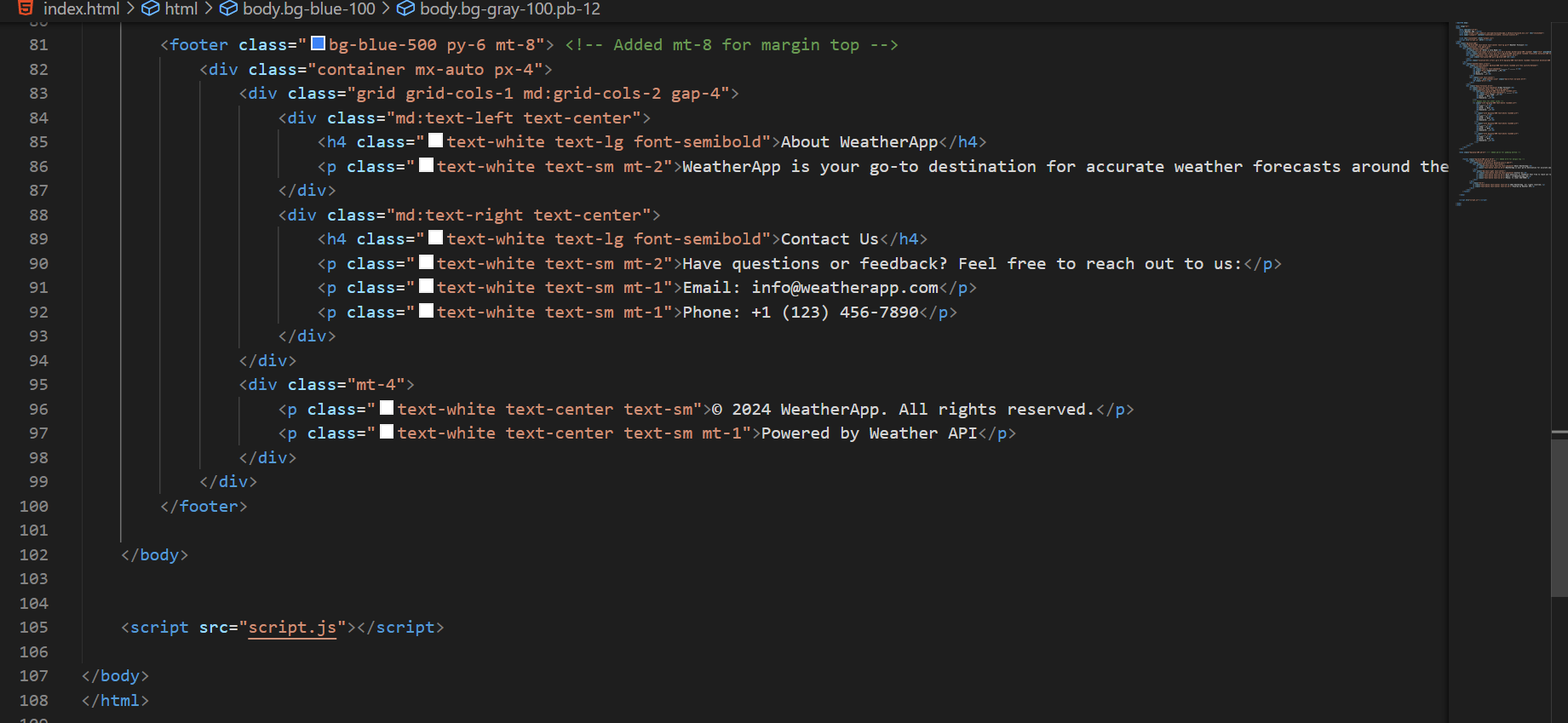
* 1. **HTML Code**

These screenshots present the HTML code for our Weather App project, revealing the layout and content of our web pages in a code format.





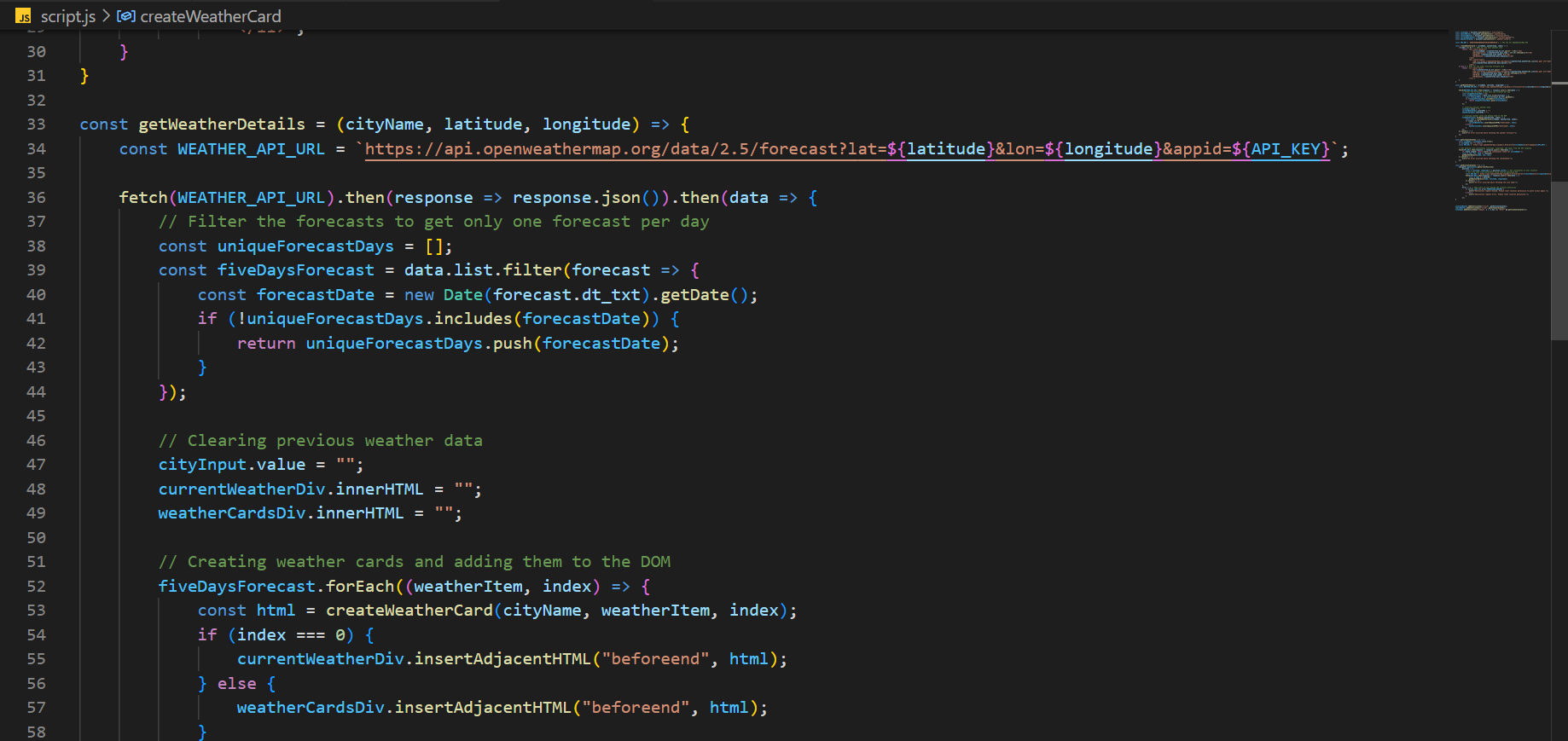


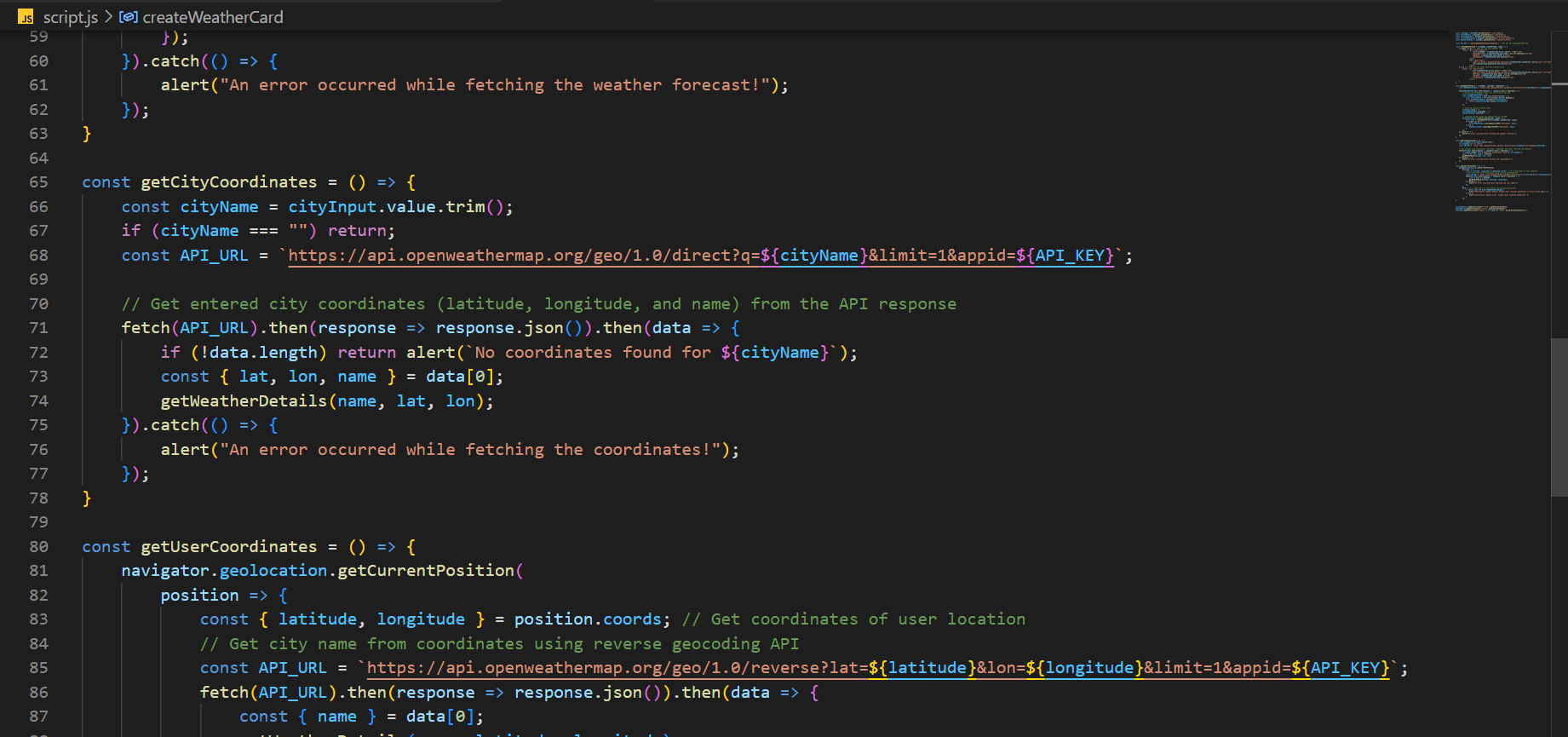


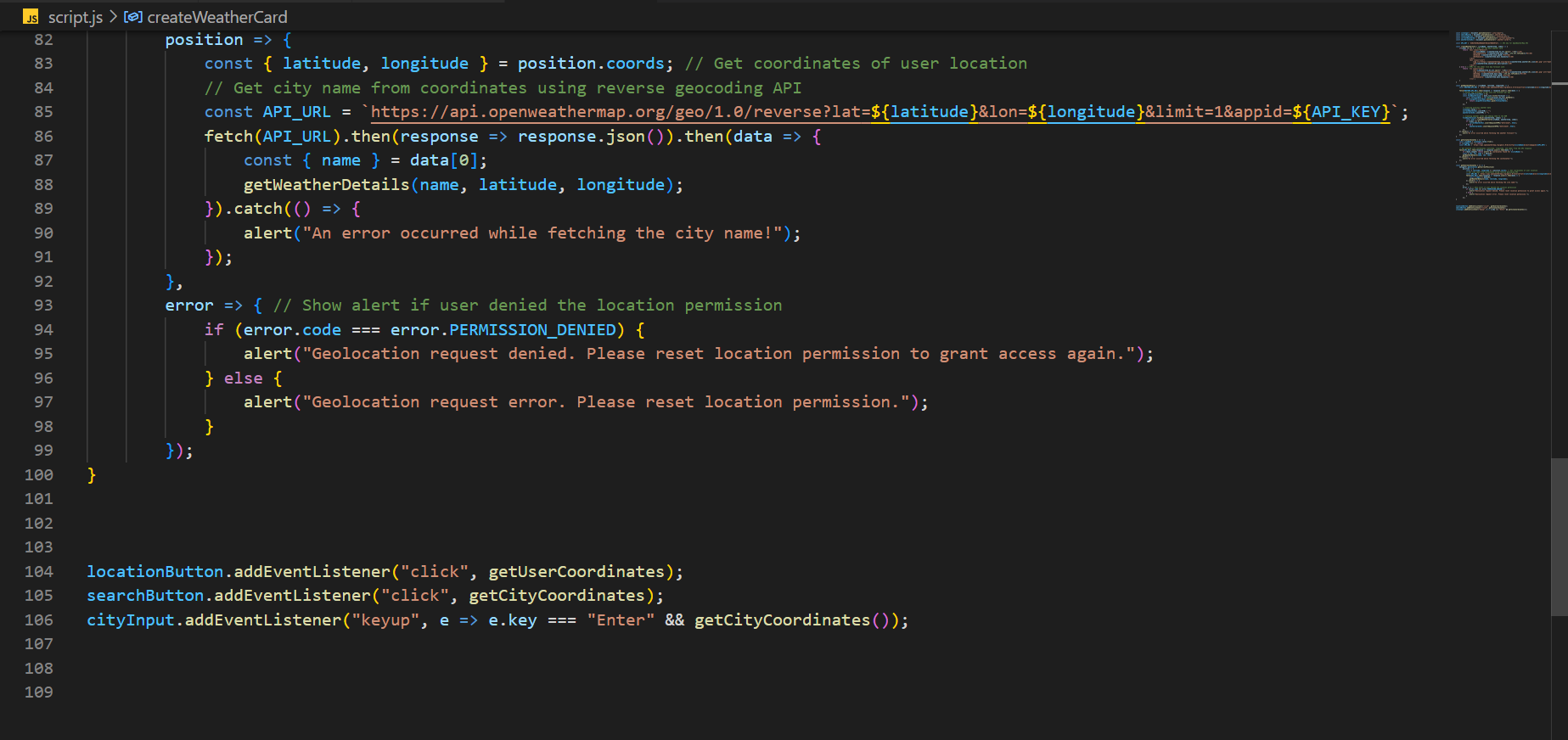
* 1. **JavaScript Code**

This screenshot exhibits the JS code for our Weather App project, illustrating the different functions and events we trigger according to different scenarios.



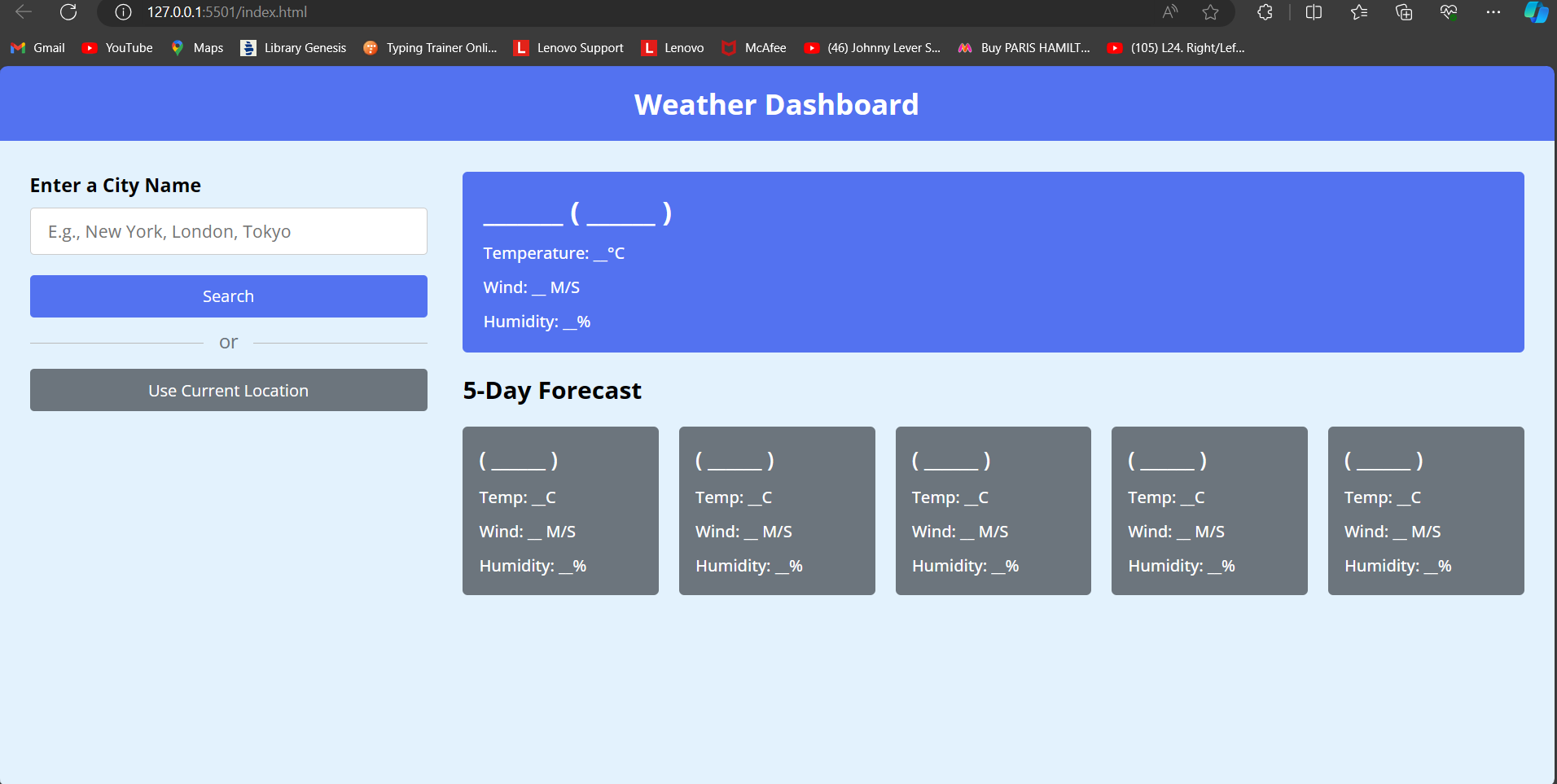




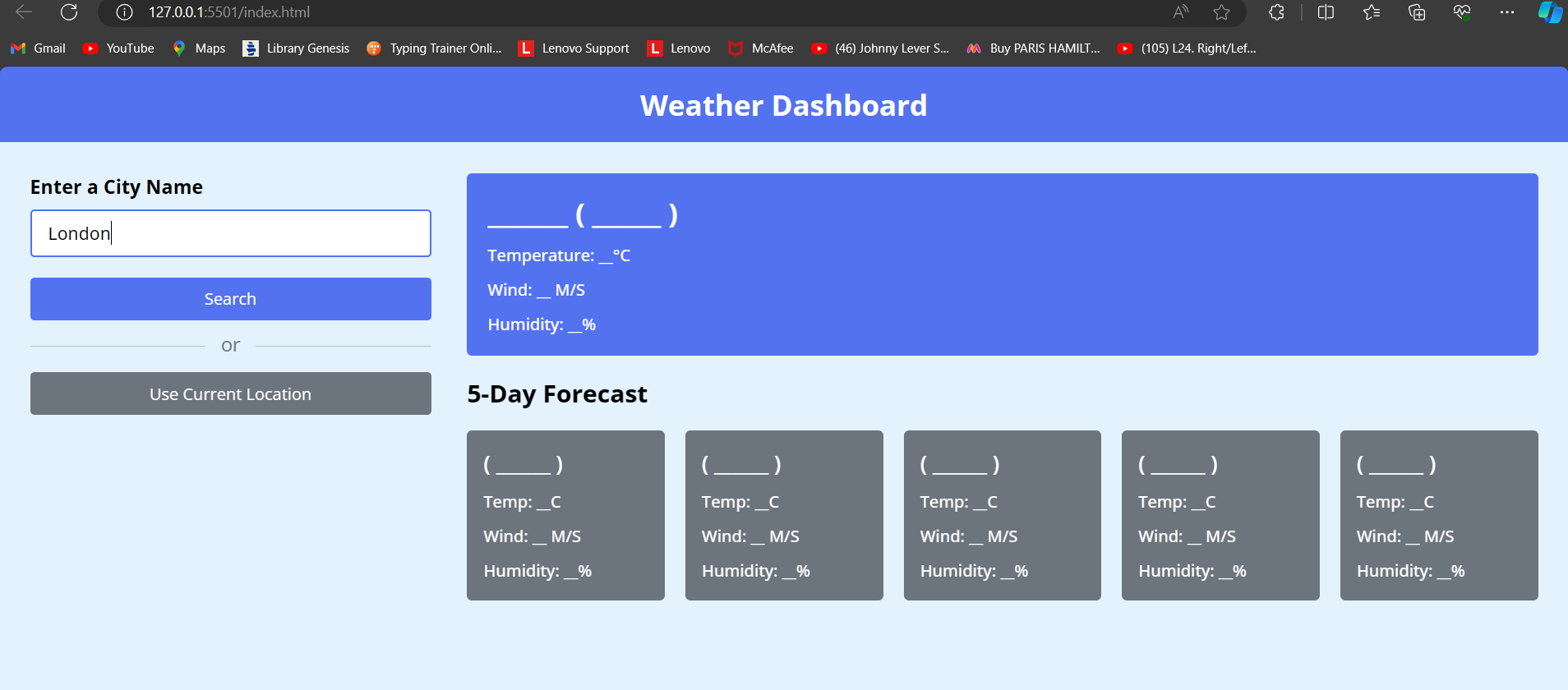


# Results

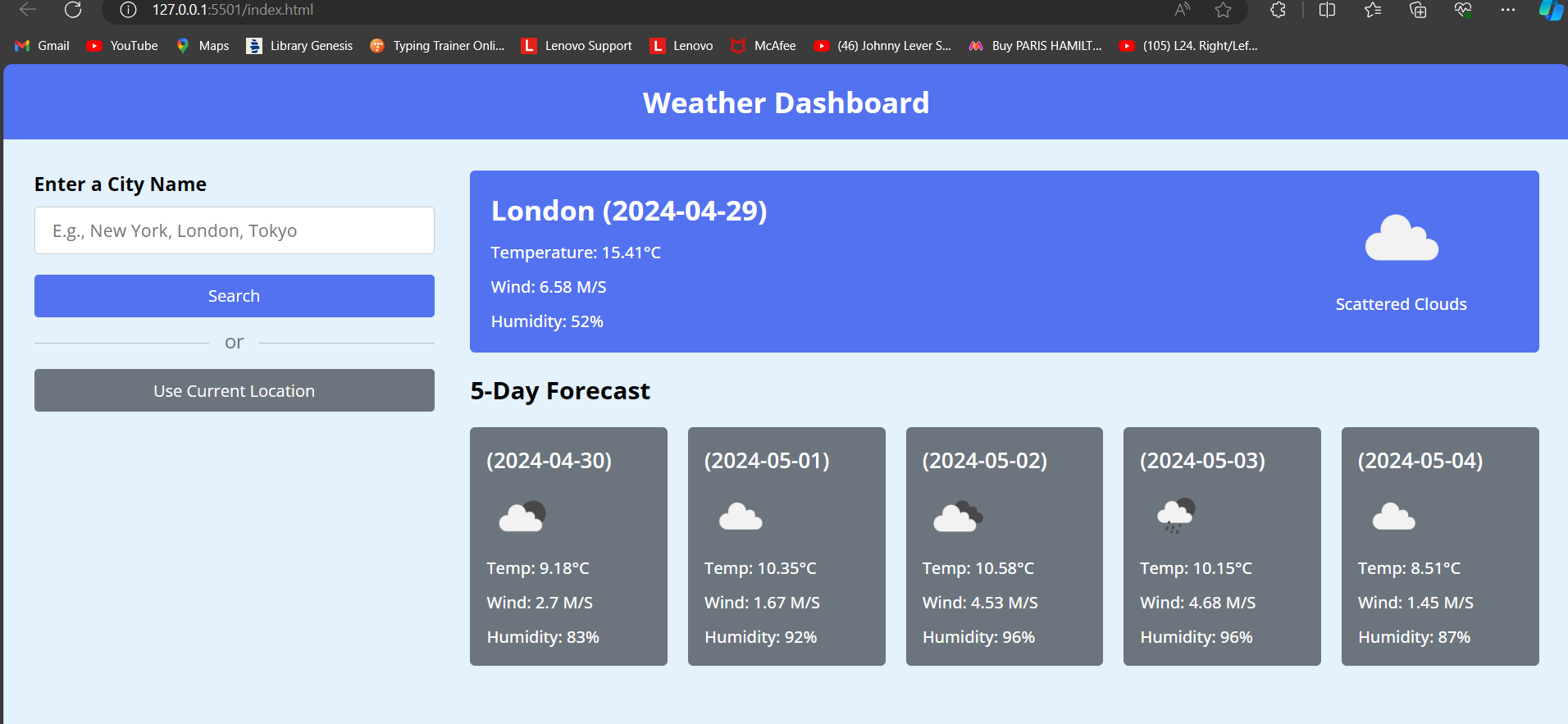
**Project Screenshots for all scenarios:**

****

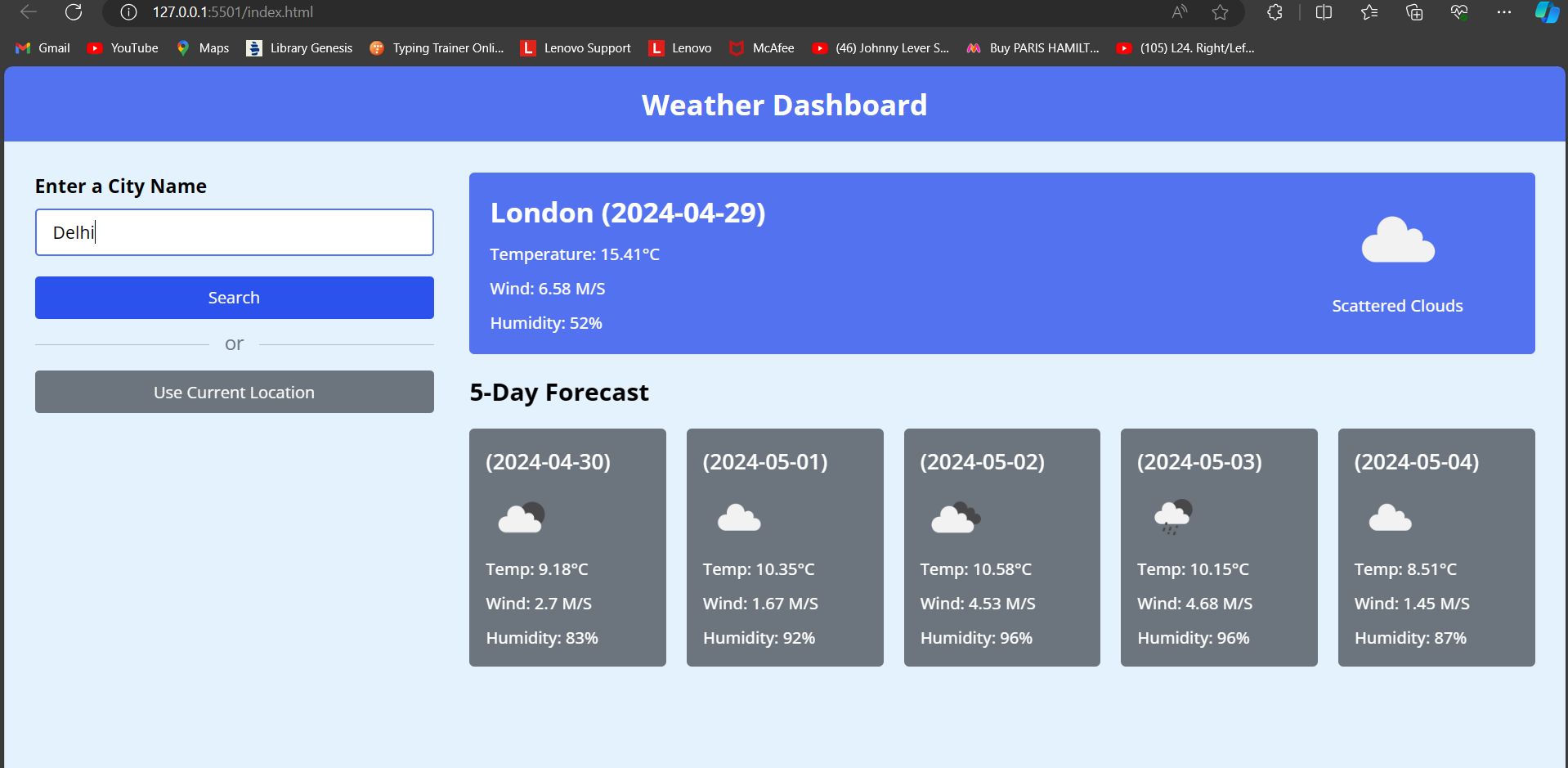
Static View

****

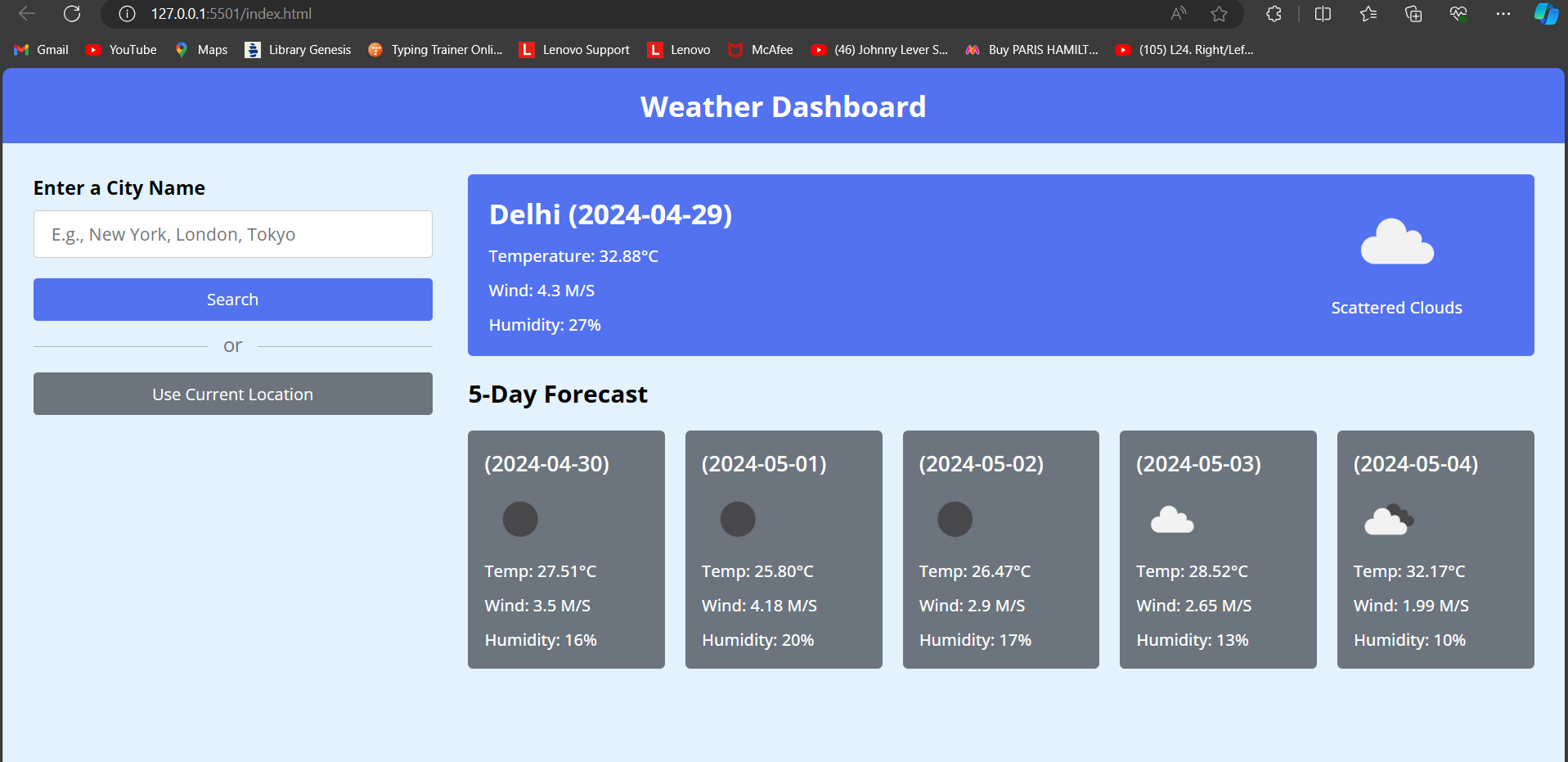
Searching of a city

****

Getting Weather Results

****

Searching for another city

****

Getting Results

# References

* **HTML, CSS, and JavaScript Documentation:**
* **Mozilla Developer Network (MDN) - HTML:** https://developer.mozilla.org/enUS/docs/Web/HTML
* **Mozilla Developer Network (MDN) JavaScript:**

https://developer.mozilla.org/en-US/docs/Web/JavaScript

* **Tailwind Documentation:**
* **Tailwind Official Documentation:**

https://tailwindcss.com/

* **GeeksForGeeks Tailwind Tutorial:**

<https://www.geeksforgeeks.org/tailwind-css/>

* Weather Api :
* [OpenWeatherMap API guide - OpenWeatherMap](https://openweathermap.org/guide)