**A PROJECT REPORT**

On

Weather Forecast Web application

**in partial fulfillment for the award of the degree**

**of**

**B.Tech CSE(LE)**

**From Second Year COMPUTER SCIENCE AND TECHNOLOGY**

By

**SAYAR DEBBARMA 23IUT0020035**

**PARTHA BISWAS 23IUT0020036**

**KANAK ACHARJEE 23IUT0020037**

**CHAYANA DEBBARMA 23IUT0020038**

**MARIAM KALAI 23IUT0020039**

**Under the Guidance of**

**Guide Name:** Dr. Dhruba Jyoti Kalita

**Designation:** Assistant Professor

****

**The ICFAI University, Tripura**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ICFAI Technical School**

****

**The ICFAI University, Tripura**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ICFAI Technical School**

**Faculty of Science and Technology**

**Course Title: Theme Project**

**11/23**



**The ICFAI University, Tripura**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project titled “Weather Forecast Web application” is the bonafide work carried out by Sayar Debbarma, Partha Biswas, Kanak Acharjee, Mariam Kalai, Chayana Debbarma, students of B.Tech CSE (LE) of Department of Computer Science and Engineering, The ICFAI University, Tripura during the academic year Semester-3, 2023 - 2026, in partial fulfillment of the requirements for the award of the degree of B Tech CSE (LE) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Dr. Dhruba jyoti Kalita. Dr. Saptarshi Chakraborty.**

**Supervisor. HOD.**

**Department of CSE. Department of CSE.**

**Signature:\_\_\_\_\_\_\_\_\_\_ Signature:\_\_\_\_\_\_\_\_\_\_**

**Dr. Abhijit Biswas. Dr. Prasanta Kumar Sinha**

**Co-ordinator of CA & CSE. Principal.**

**ICFAI University. ICFAI University.**

**Signature:\_\_\_\_\_\_\_\_\_\_ Signature:\_\_\_\_\_\_\_\_\_\_**



**The ICFAI University, Tripura**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **Declaration of Student**

I hereby declare that the project entitled “Weather Forecast Web Application” submitted for the B.Tech CSE (LE) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sayar Debbarma

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Partha Biswas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kanak Acharjee

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Chayana Debbarma

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mariam Kalai

## **Acknowledgement**

We would like to take this opportunity to express our deep sense of gratitude to all who helped us directly or indirectly during this theme-project work.

Firstly, we would like to thank our supervisor, **Dr. Dhruba Jyoti Kalita**, for being a great mentor and the best adviser we could ever have. His advice, encouragement and critics are a source of innovation ideas, inspiration and causes behind the successful completion of this dissertation. The confidence shown in us by him was the biggest source of inspiration for us. It has been a privilege working with him.

We are highly obliged to all the faculty members of the Information Technology Department for their support and encouragement. We also thank our **Dr. Saptarshri Chakraborty** for providing excellent computing without which this work could not achieve its quality goal.

Finally, we are grateful to our parents for their support. It was impossible for us to complete this theme project without their financial support and encouragement.

**Sayar Debbarma**

**Partha Biswas**

**Kanak Acharjee**

**Chayana Debbarma**

**Mariam Kalai**

## **Abstract**

This project presents the development of a weather forecast application that leverages the power of **HTML, CSS, RapidAPI,** and **JavaScript** to provide users with a comprehensive and user-friendly weather information portal. By seamlessly integrating **RapidAPI's weather API,** the application retrieves real-time weather data, including current conditions, five-day forecasts, and detailed weather parameters such as humidity, pressure, and wind speed.

The application's intuitive design and informative presentation empower users to access accurate and up-to-date weather information with ease.

This project exemplifies the potential of web technologies in delivering practical and user-centric applications.

Overall, it provides a clear overview of the project. It highlights the key technologies used **(HTML, CSS, RapidAPI, JavaScript),** the functionalities of the application **(retrieving weather data, displaying current and forecast information),** and the target audience **(users seeking weather information).**

## **Table of Contents**

**Contents. Page No.**

Cover & Title Page 1

Certificate of the Guide 3

Declaration of the Student 4

Acknowledgement 5

Abstract 6

Table of Contents 7

1. **INTRODUCTION 8**

1.1 Problem Definition 8

1.2 Project Overview/Specifications 9

1.3 Hardware Specification 10

1.4 Software Specification 11

1.3.1 11

1.3.2

…

1. **LITERATURE SURVEY 12**

2.1 Existing System 12

2.2 Proposed System 13

2.3 Feasibility Study 14

1. **SYSTEM ANALYSIS & DESIGN 15**

3.1 Requirement Specification 15

3.2 Flowcharts 16

3.3 Design and Test Steps / Criteria 17-18

3.3 Algorithms and Pseudo Code 19-31

3.3.1 31

3.3.2

…

1. **RESULTS / OUTPUTS 32-34**
2. **CONCLUSIONS 35**
3. **REFERENCES 36**
4. **APPENDICES 37**

## **Introduction**

### 1.1 **Problem Definition**

The problem definition of a weather forecast project is to develop a system that can accurately predict the future state of the atmosphere for a given location and time. This is a complex problem due to the chaotic nature of the atmosphere and the vast amount of data that must be processed.

### **1.2 Project Overview**

* Real-time weather data: Implement a reliable data source to fetch current weather conditions for specified locations.
* Location flexibility: Allow users to enter city names or geographical coordinates to retrieve weather information for their desired locations.
* Intuitive interface: Design a user-friendly interface that is easy to navigate and understand, providing clear and concise weather information.
* Visual representation: Incorporate visual elements such as icons, graphs, or charts to enhance the presentation of weather data.
* Customization options: Provide users with options to customize the display of weather information, such as temperature units, language preferences, and notification settings.
* Multiple weather parameters: Display a comprehensive range of weather parameters, including temperature, humidity, precipitation, wind speed, and UV index.
* Forecasting feature: Implement a forecasting feature that provides predictions for upcoming weather conditions, allowing users to plan their activities effectively.
* Mobile optimization: Optimize the application for mobile devices to ensure seamless access and functionality across various screen sizes and platforms.

### **1.3 Hardware Specifications:**

#### ***Client-side***:

A standard web browser running on a personal computer, laptop, or mobile device can access the weather forecast application. No specialized hardware is required on the client-side.

#### ***Server-side:***

The server-side component of the application, which handles data retrieval and processing, can be hosted on a variety of hardware configurations, ranging from a single server to a distributed cloud infrastructure. The specific hardware requirements will depend on the expected traffic volume and computational demands of the application.

### **1.4 Software Specifications:**

#### ***Client-side***:

Web browser: A standard web browser with support for HTML5, CSS3, and JavaScript is required to render the application's user interface and interact with the server-side component.

JavaScript runtime: A JavaScript runtime environment, such as Node.js or V8 Embedded, is required to execute the JavaScript code that drives the application's functionality.

#### ***Server-side***:

Web server: A web server, such as Apache or Nginx, is required to host the application's web pages and handle incoming HTTP requests.

Programming language: A programming language, such as Python, Java, or PHP, is used to develop the server-side code that handles data retrieval from RapidAPI and processes it into a format suitable for the client-side application.

RapidAPI client library: A RapidAPI client library, available for various programming languages, is required to interact with the RapidAPI platform and retrieve weather data.

## **Literature Survey**

### **2.1 Existing System:**

#### ***Traditional weather websites***:

Most people rely on traditional weather websites or mobile apps to get weather information. These websites and apps typically provide current weather conditions, forecasts for the next few days, and additional information such as radar images and weather alerts.

#### ***Limitations***:

The major limitation of traditional weather websites and apps is that they often provide limited customization options and may not be as user-friendly as a custom-built application. Additionally, these websites and apps may not be able to access and process real-time weather data as quickly as a custom application.

### **2.2 Proposed System:**

#### ***Custom weather forecast application:***

A custom weather forecast application using HTML, CSS, JavaScript, and RapidAPI would provide a more personalized and interactive experience for users. The application could be tailored to specific user needs and preferences, and it could access and process real-time weather data from RapidAPI.

#### ***Benefits***:

The proposed system would offer several benefits over traditional weather websites and apps, including:

Customization: The application could be customized to display the user's preferred weather parameters, units of measurement, and location.

Real-time data: The application could access and process real-time weather data from RapidAPI, providing users with the most up-to-date information.

Interactive features: The application could include interactive features such as maps, charts, and graphs to help users visualize weather data.

Mobile optimization: The application could be optimized for mobile devices, providing users with access to weather information on the go.

### **2.3 Feasibility Study:**

#### ***Technical feasibility:***

The proposed system is technically feasible using the available technologies. HTML, CSS, and JavaScript are well-established web development technologies, and RapidAPI provides a reliable and easy-to-use platform for accessing weather data.

#### ***Economic feasibility:***

The proposed system can be developed and maintained at a reasonable cost. The use of open-source technologies and cloud-based infrastructure can help to reduce costs.

#### ***Operational feasibility:***

The proposed system can be operated and maintained by a team of web developers and data scientists. The use of cloud-based infrastructure can help to simplify operations and maintenance.

#### ***Schedule feasibility:***

The proposed system can be developed and deployed within a reasonable timeframe. The use of agile development methodologies can help to ensure that the project is completed on time and within budget.

## **System analysis & Design**

### **3.1 Requirement Specifications:**

#### ***User Interface (UI):***

* The application should have a user-friendly interface that is easy to navigate and understand.
* The interface should be visually appealing and consistent with the overall design of the application.
* The interface should be responsive and adapt to different screen sizes and devices.

#### ***Data Access***:

* The application should be able to access real-time weather data from RapidAPI.
* The application should be able to access historical weather data for a specified location.
* The application should be able to access weather data for multiple locations simultaneously.

#### ***Data Processing:***

* The application should be able to process weather data from RapidAPI into a format that is suitable for display.
* The application should be able to convert weather data into different units of measurement.
* The application should be able to filter and sort weather data based on user preferences.

### **3.2 Flowchart**

User Input

B [Validate Input]

A [Enter a city or Location]

B-> if valid

C {Retrive Weather data}

B-> if invalid

C {Display error message}

### 

### **3.3 Design and Test Steps**

### Design Steps:

#### 1. Define Purpose and Audience:

- Clearly understand the purpose of your website (e.g., weather forecast application) and identify your target audience.

#### 2. Wireframing:

- Create wireframes or sketches to plan the layout and structure of your website. Consider the placement of elements like navigation, content areas, and interactive components.

#### 3. Choose a Design Style:

- Select a design style that aligns with your website’s purpose and audience. Consider color schemes, typography, and visual elements.

#### 4. Responsive Design:

- Ensure your design is responsive to different screen sizes and devices. Test on various devices to confirm a consistent user experience.

#### 5. User Interface (UI) Design:

- Design the user interface with user-friendly navigation, clear calls to action, and an intuitive layout.

#### 6. Visual Assets:

- Create or source visual assets like images and icons to enhance the visual appeal of your website.

#### 7. Accessibility:

- Implement accessibility features to ensure your website is usable by people with disabilities. Follow best practices for contrast, text alternatives, and keyboard navigation.

#### 8. Interaction Design:

- Design interactive elements, such as buttons and forms, with a focus on usability. Consider feedback mechanisms for user actions.

### Testing Steps:

#### 1. Functional Testing:

- Test all website functionalities, including weather data retrieval, user input validation, and any interactive features.

#### 2. Browser Compatibility:

- Test your website on multiple browsers (Chrome, Firefox, Safari, Edge) to ensure compatibility.

#### 3. Mobile Responsiveness:

- Test the website on various devices (phones, tablets) to confirm responsive design.

#### 4. Performance Testing:

- Check the loading speed of your website. Optimize images and code for faster loading times.

#### 5. Security:

- If your website involves user data or transactions, ensure secure communication (HTTPS) and implement necessary security measures.

#### 6. Cross-Device Testing:

- Verify the website’s functionality and appearance across different devices and operating systems.

#### 7. User Experience (UX) Testing:

- Collect feedback from users or conduct usability testing to identify and address any issues related to the user experience.

#### 8. SEO Testing:

- Optimize your website for search engines. Check meta tags, page titles, and ensure proper indexing.

#### 9. Content Review:

- Verify that all content is accurate, up-to-date, and free of errors.

#### 10. Continuous Monitoring:

- Implement tools for ongoing monitoring of website performance, user interactions, and potential issues.

### **3.3 Code**

#### ***HTML:***

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Weather Forecast</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

</head>

<body>

  <nav class="navbar navbar-expand-lg bg-body-tertiary">

    <div class="container-fluid">

      <a class="navbar-brand" href="#">Weather Forecast</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"

        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

        <ul class="navbar-nav me-auto mb-2 mb-lg-0">

          <li class="nav-item">

            <a class="nav-link active" aria-current="page" href="#">Home</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="about.html">About</a> <!-- Changed the link to connect to about.html -->

          </li>

          <li class="nav-item dropdown">

            <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">

              How to use

            </a>

            <ul class="dropdown-menu">

              <li><a class="dropdown-item" href="#">Tripura</a></li>

              <li><a class="dropdown-item" href="#">Agartala</a></li>

              <li>

                <hr class="dropdown-divider">

              </li>

              <li><a class="dropdown-item" href="#">Something else here</a></li>

            </ul>

          </li>

        </ul>

        <form class="d-flex" role="search">

          <input id="city" class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

          <button class="btn btn-outline-success" type="submit" id="submit">Search</button>

        </form>

      </div>

    </div>

  </nav>

  <div class="container">

    <div class="container py-3">

      <header>

        <div class="d-flex flex-column flex-md-row align-items-center pb-3 mb-4 border-bottom">

          <a href="/" class="d-flex align-items-center link-body-emphasis text-decoration-none">

            <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-sun"

              viewBox="0 0 16 16">

              <path

                d="M8 11a3 3 0 1 1 0-6 3 3 0 0 1 0 6zm0 1a4 4 0 1 0 0-8 4 4 0 0 0 0 8zM8 0a.5.5 0 0 1 .5.5v2a.5.5 0 0 1-1 0v-2A.5.5 0 0 1 8 0zm0 13a.5.5 0 0 1 .5.5v2a.5.5 0 0 1-1 0v-2A.5.5 0 0 1 8 13zm8-5a.5.5 0 0 1-.5.5h-2a.5.5 0 0 1 0-1h2a.5.5 0 0 1 .5.5zM3 8a.5.5 0 0 1-.5.5h-2a.5.5 0 0 1 0-1h2A.5.5 0 0 1 3 8zm10.657-5.657a.5.5 0 0 1 0 .707l-1.414 1.415a.5.5 0 1 1-.707-.708l1.414-1.414a.5.5 0 0 1 .707 0zm-9.193 9.193a.5.5 0 0 1 0 .707L3.05 13.657a.5.5 0 0 1-.707-.707l1.414-1.414a.5.5 0 0 1 .707 0zm9.193 2.121a.5.5 0 0 1-.707 0l-1.414-1.414a.5.5 0 0 1 .707-.707l1.414 1.414a.5.5 0 0 1 0 .707zM4.464 4.465a.5.5 0 0 1-.707 0L2.343 3.05a.5.5 0 1 1 .707-.707l1.414 1.414a.5.5 0 0 1 0 .708z" />

            </svg>

            <span class="fs-4">Weather</span>

          </a>

          <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">

            <a class="me-3 py-2 link-body-emphasis text-decoration-none" href="features.html">Features</a>

            <a class="me-3 py-2 link-body-emphasis text-decoration-none" href="locations.html">Locations</a>

            <a class="me-3 py-2 link-body-emphasis text-decoration-none" href="support.html">Support</a>

        </div>

        <div class="pricing-header p-3 pb-md-4 mx-auto text-center">

          <h1 class="display-4 fw-normal text-body-emphasis">Weather</h1>

          <p class="fs-5 text-body-secondary"></p>

        </div>

      </header>

      <main>

        <div class="text-center">

          <div class="col">

            <div class="card mb-4 rounded-3 shadow-sm border-primary">

              <div class="card-header py-3 text-bg-primary border-primary">

                <h5 class="my-0 fw-normal">Weather of <span id="cityName"></span></h5>

              </div>

              <div class="card-body">

                <h1 class="card-title Temperature-card-title"><span id="temp"></span><span>&#8451;</span></h1>

                <ul class="list-unstyled mt-3 mb-4">

                  <li>Humidity: <span id="humidity"></span><span>&#37;</span></h1>

                  </li>

                  <li>Precipitation: <span id="cloud\_pct"></span><span>&#37;</span></h1>

                  </li>

                  <li>Wind: <span id="wind\_speed"></span><span>&nbsp;&kappa;&mu;&nbsp;&times;&nbsp;</span></h1>

                  </li>

                  <li>Feels like: <span id="feels\_like"></span><span>&#8451;</span></h1>

                  </li>

                  <li>Minimum Temperature: <span id="min\_temp"></span><span>&#8451;</span></h1>

                  </li>

                  <li>Maximum Temperature: <span id="max\_temp"></span><span>&#8451;</span></h1>

                  </li>

                </ul>

              </div>

            </div>

          </div>

        </div>

        <footer class="pt-4 my-md-5 pt-md-5 border-top">

          <div class="row">

            <div class="col-12 col-md">

            </div>

            <div class="col-6 col-md">

              <h5>Features</h5>

              <ul class="list-unstyled text-small">

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="random\_features.html">Random feature</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="features.html">Team feature</a></li>

              </ul>

            </div>

            <div class="col-6 col-md">

              <h5>Resources</h5>

              <ul class="list-unstyled text-small">

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="resource.html">Resource</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="#">Resource name</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="finalresource.html">Final resource</a></li>

              </ul>

            </div>

            <div class="col-6 col-md">

              <h5>About</h5>

              <ul class="list-unstyled text-small">

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="team.html">Team</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="location.html">Locations</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="privacy.html">Privacy</a></li>

                <li class="mb-1"><a class="link-secondary text-decoration-none" href="terms.html">Terms</a></li>

              </ul>

            </div>

          </div>

        </footer>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"

      integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"

      crossorigin="anonymous"></script>

    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

    <script src="script.js"></script>

</body>

</html>

<!—Features.html🡪

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Features</title>

</head>

<style>

  h1,

  h2,

  p {

    max-width: 100%;

    word-wrap: break-word;

  }

</style>

<body>

  <h1>Features</h1>

  <p>

  <h2>

    1. Current Weather Information:

  </h2>

  Display real-time weather conditions such as temperature, humidity, wind speed, and cloud cover for a specific

  location.</p>

  <p>

  <h2>

    2. Location-based Forecast:

  </h2>

  Allow users to input or select their location to receive a personalized weather forecast.

  </p>

  <p>

  <h2>

    3. Multiple Locations:

  </h2>

  Enable users to add and track weather information for multiple locations, making it convenient for travelers or users

  interested in different places.

  </p>

  <p>

  <h2>

    4. User Preferences:

  </h2>

  Allow users to customize their weather preferences, such as choosing a default location, setting notification

  preferences, or selecting their preferred units of measurement.

  </p>

  <p>

  <h2>

    5. Responsive Design:

  </h2>

  Ensure that the application is accessible and user-friendly on various devices, including desktops, tablets, and

  smartphones.

  </p>

</body>

</html>

<!--FinalResource.html-->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Final Resources</title>

    <style>

      h1, ul {

        text-align: center;

      }

      li {

        list-style-type: none;

        margin: 10px;

      }

    </style>

</head>

<body>

    <h1>Final Resources</h1>

    <ul>

        <li>Resource 1</li>

        <li>Resource 2</li>

        <li>Resource 3</li>

    </ul>

</body>

</html>

<!--Location.html-->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Locations</title>

    <script src="location.js">

    </script>

</head>

<body>

    <form class="d-flex" role="search">

          <input id="city" class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

          <button class="btn btn-outline-success" type="submit" id="submit">Done</button>

    </form>

    <button onclick="storeCityName()">Save City Name</button>

</body>

</html>

<!--- <|Privacy|> --->

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Privacy</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      margin: 0;

      padding: 0;

      color: #FFFFFF;

      background-color: #000000;

      display: inline;

      inline-direction: column;

      align-items: center;

      justify-content: center;

      height: 100vh;

      text-align: center;

    }

    h1,

    h2,

    h3,

    h4,

    h5,

    p {

      margin: 100px 0;

    }

    p {

      line-height: 1.5;

    }

    @media (max-width: 6000px) {

      body {

        padding: 100px;

      }

    }

  </style>

</head>

<body>

  <h1>Weather Privacy Policy</h1>

  <p>Last updated: 18/12/2023 to update the privacy policy.</p>

  <p>Weather Forecast, Inc. and its affiliates (“Weather Forecast,” “we,” “our,” and/or “us”) value the privacy of

    individuals who use our website and related services (collectively, our “Services”). This privacy policy (the

    “Privacy Policy”) explains how we collect, use, and share information from or about you or your device as you use

    the Services. By using our Services, you agree to the collection, use, disclosure, and procedures this Privacy

    Policy describes. Beyond the Privacy Policy, your use of our Services is also subject to our Terms of Service....

  </p>

  <h2>Information We Collect</h2>

  <p>We may collect a variety of information from or about you or your devices from various sources, as described below.

    If you do not provide your information when requested, you may not be able to use some or all of our Services if

    that information is necessary to provide you with our Services or if we are legally required to collect it.</p>

  <h3>Information You Provide to Us</h3>

  <p>Registration and Profile Information. When you register for an account, we ask you for your email address and may

    also request your phone number. If you sign up using a social media account, we will also receive information from

    those social media services such as your name, email address, and profile photo, in accordance with your privacy

    settings on those social media services.</p>

  <h4>Information We Collect Automatically When You Use Our Services</h4>

  <p>When you use the Services, we use a variety of electronic tools, such as cookies, pixel tags, and similar

    technologies (collectively referred to in this Privacy Policy as “cookies”) to automatically generate information

    about how you use and interact with the Services as discussed in this section. We may also collect information about

    your use of the Services, such as your IP address, browser type, operating system.</p>

  <h5>What are Cookies?</h5>

  <p>Cookies are small files of letters and numbers that we store on your browser or the hard drive of your computer, or

    small graphic images placed on web pages or in emails. They can be stored either for a single browser session or

    more permanently. We use cookies to collect information about your browsing activities, distinguish you from other

    users, provide functionality, and analyze use of the Services. Our third-party partners, such as analytics and

    advertising partners, may use these technologies to collect information about your online activities over time and

    across different services.</p>

</body>

</html>

<!--RandomFeatures-->

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Random Features</title>

  <style>

    input {

      padding: 10px;

      margin: 10px;

    }

    button {

      padding: 10px;

      margin: 10px;

    }

    #qrcode {

      margin: 10px;

    }

  </style>

</head>

<body>

  <h1>QR Code Generator</h1>

  <input type="text" id="urlInput" placeholder="Enter URL">

  <button onclick="generateQR()">Generate QR Code</button>

  <div id="qrcode"></div>

  <script src="https://cdn.jsdelivr.net/qrcode/1.4.4/qrcode.min.js"></script>

  <script>

    function generateQR() {

      var url = document.getElementById('urlInput').value;

      var qrcode = new QRCode(document.getElementById("qrcode"), {

        text: url,

        width: 128,

        height: 128

      });

      qrcode.makeCode(url);

      // Create an img element

      var imgElement = document.createElement('img');

      // Set its src attribute to the data URL of the QR code

      imgElement.src = document.getElementById('qrcode').getElementsByTagName('img')[0].src;

      document.getElementById('qrcode').appendChild(imgElement);

    }

  </script>

</body>

</html>

**CSS**

html {

  height: 100%;

  width: 100%;

}

#### ***Javascript:***

//Script.js

const url = 'https://weather-by-api-ninjas.p.rapidapi.com/v1/weather?city=Seattle';

const options = {

  method: 'GET',

  headers: {

    'X-RapidAPI-Key': '26cc1b0a71mshb503fea2aeedb25p122b05jsnc916465ec9b1',

    'X-RapidAPI-Host': 'weather-by-api-ninjas.p.rapidapi.com'

  }

};

const getWeather = (city) => {

  cityName.innerHTML = city

  fetch('https://weather-by-api-ninjas.p.rapidapi.com/v1/weather?city=' + city, options)

    .then(Response => Response.json)

    .then((Response) =>

      console.log(Response))

  temp.innerHTML = Response.temp

  humidity.innerHTML = Response.humidity

  wind\_speed.innerHTML = Response.wind\_speed

  cloud\_pct.innerHTML = Response.cloud\_pct

    .catch(err => console.error(err));

}

submit.addEventListner("click", (e) => {

  console.log(submit);

  e.preventDefault();

  getWeather(city.value);

});

getWeather("agartala")

//Location.js

        function storeCityName() {

            var cityName = document.getElementById('city').value;

            // Your server endpoint URL to handle the Excel file creation

            var endpointUrl = 'https://your-server-endpoint-url.com/createExcelFile';

            // Send the city name data to the server endpoint

            fetch(endpointUrl, {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ cityName: cityName }),

            })

            .then(response => {

                if (response.ok) {

                    console.log('City name sent to server for Excel file creation');

                } else {

                    console.error('Failed to send city name to the server');

                }

            })

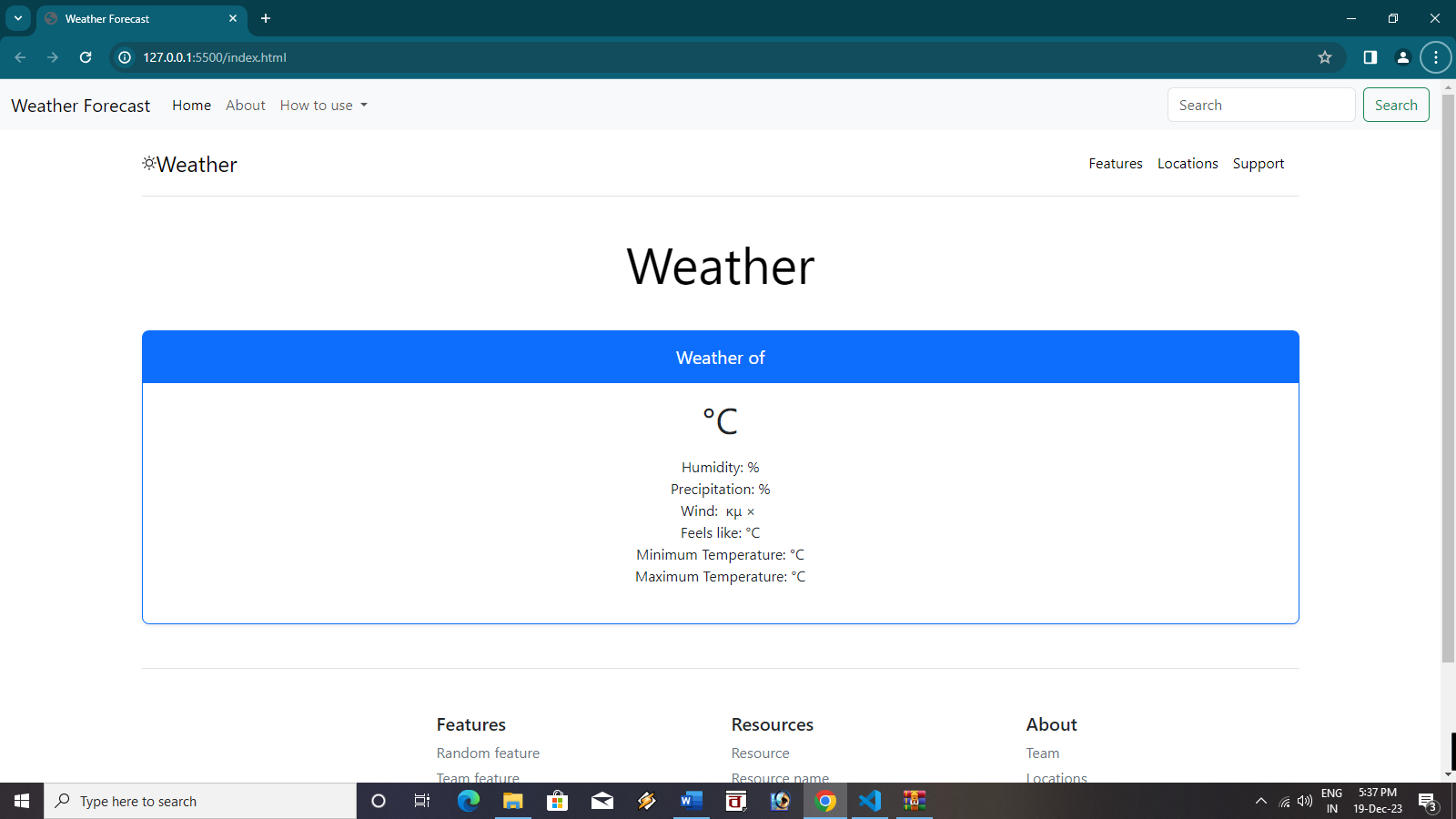
            .catch(error => {

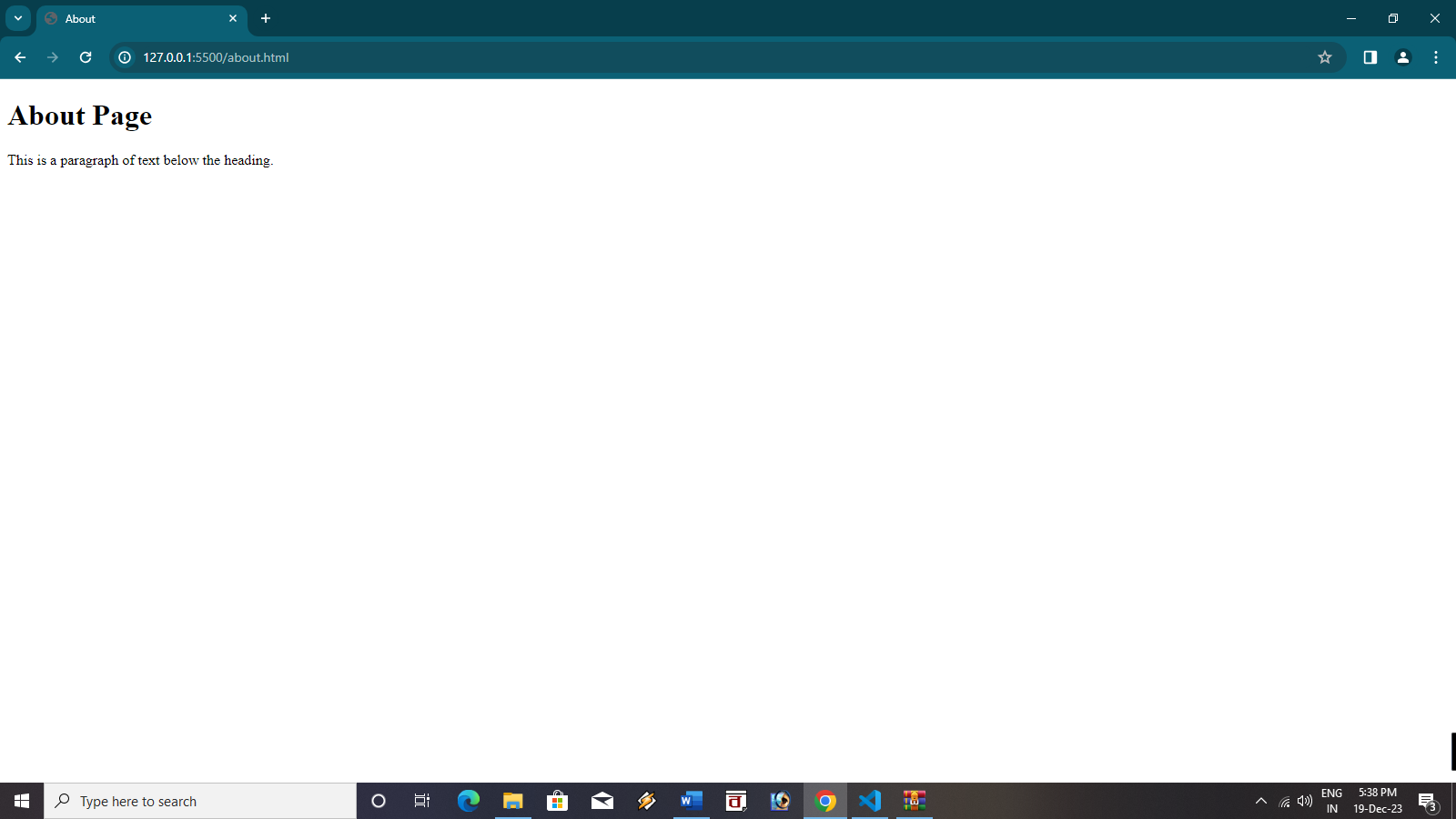
                console.error('Error occurred while sending city name to server:', error);

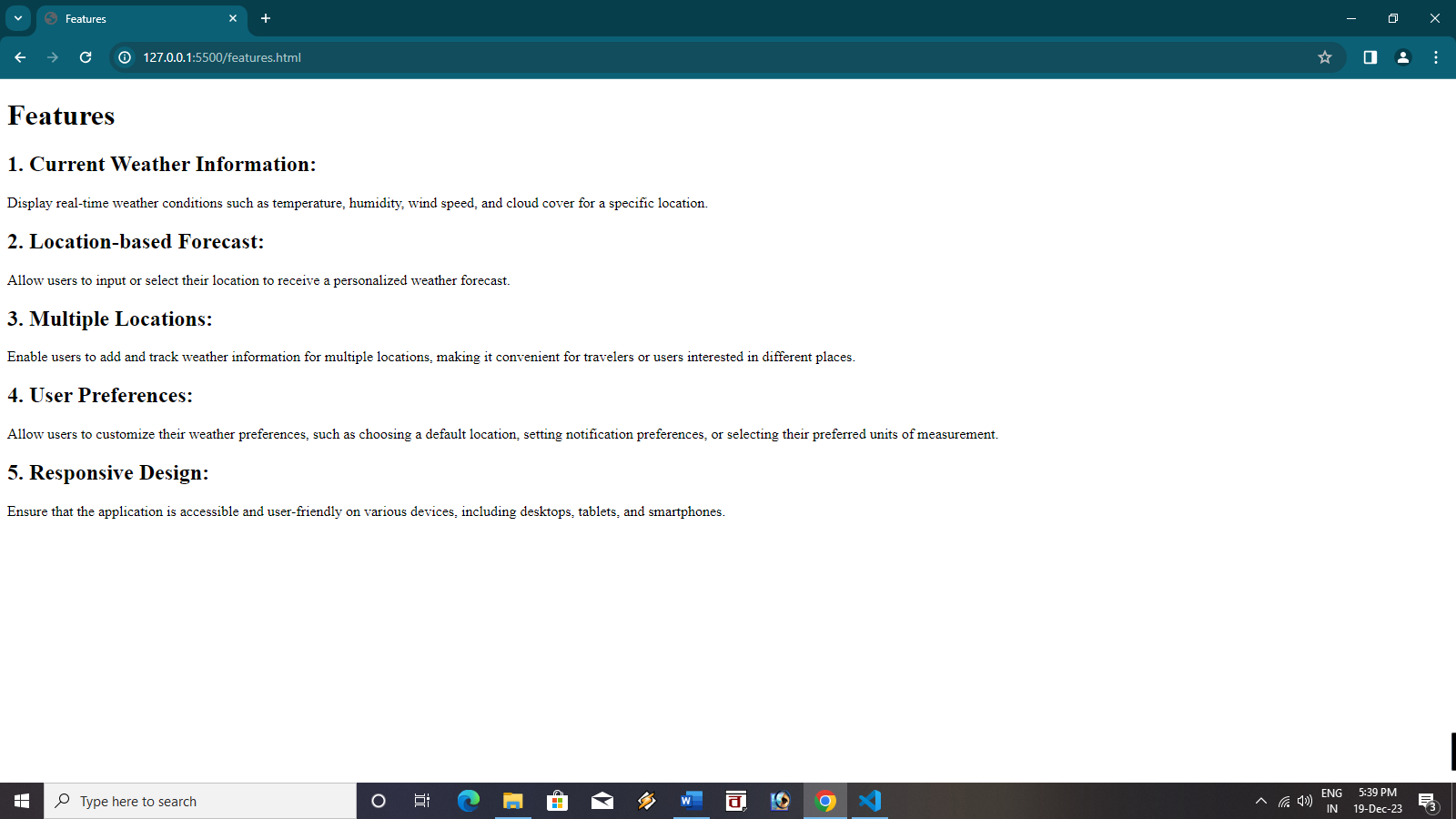
            });

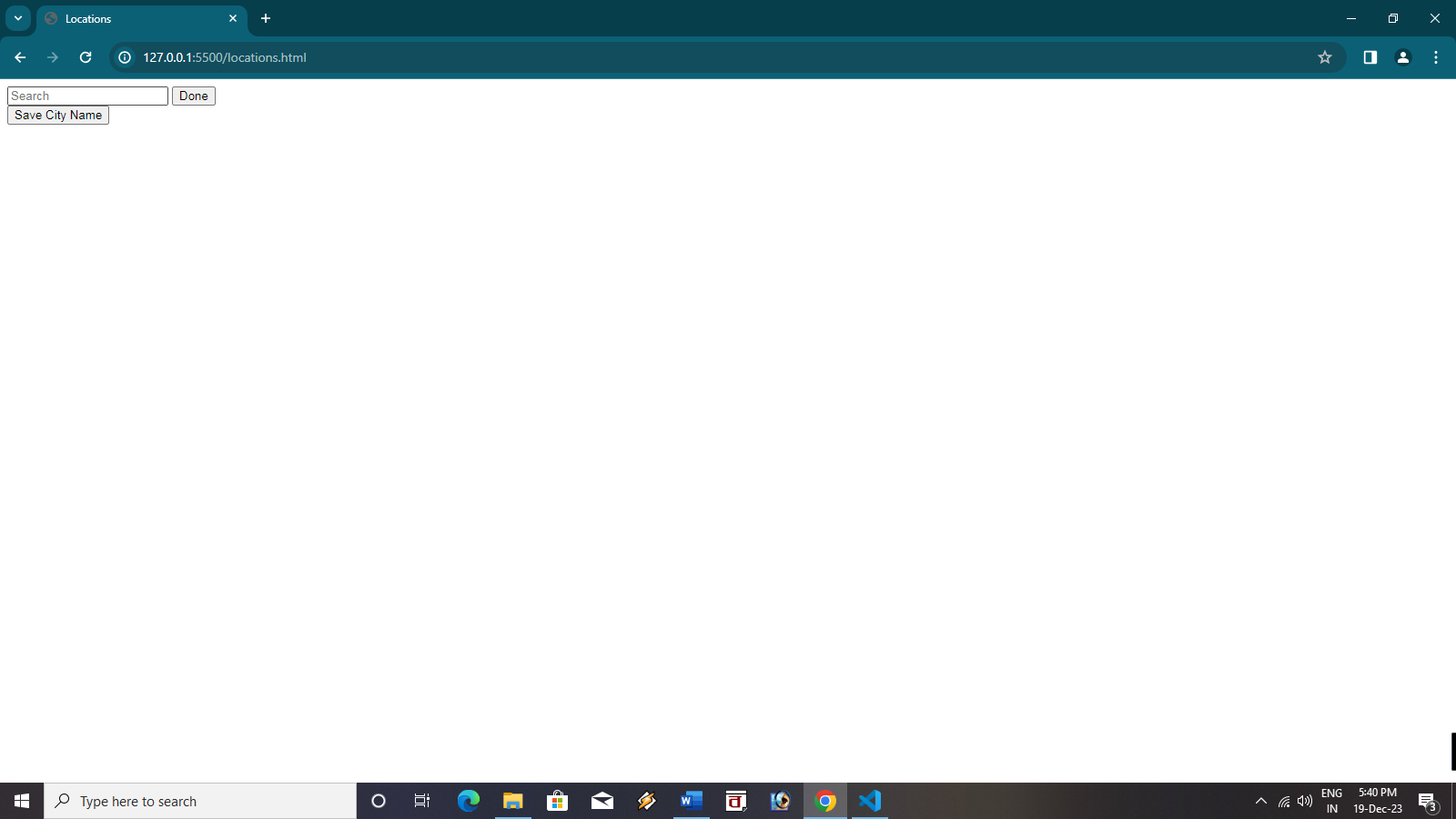
        }

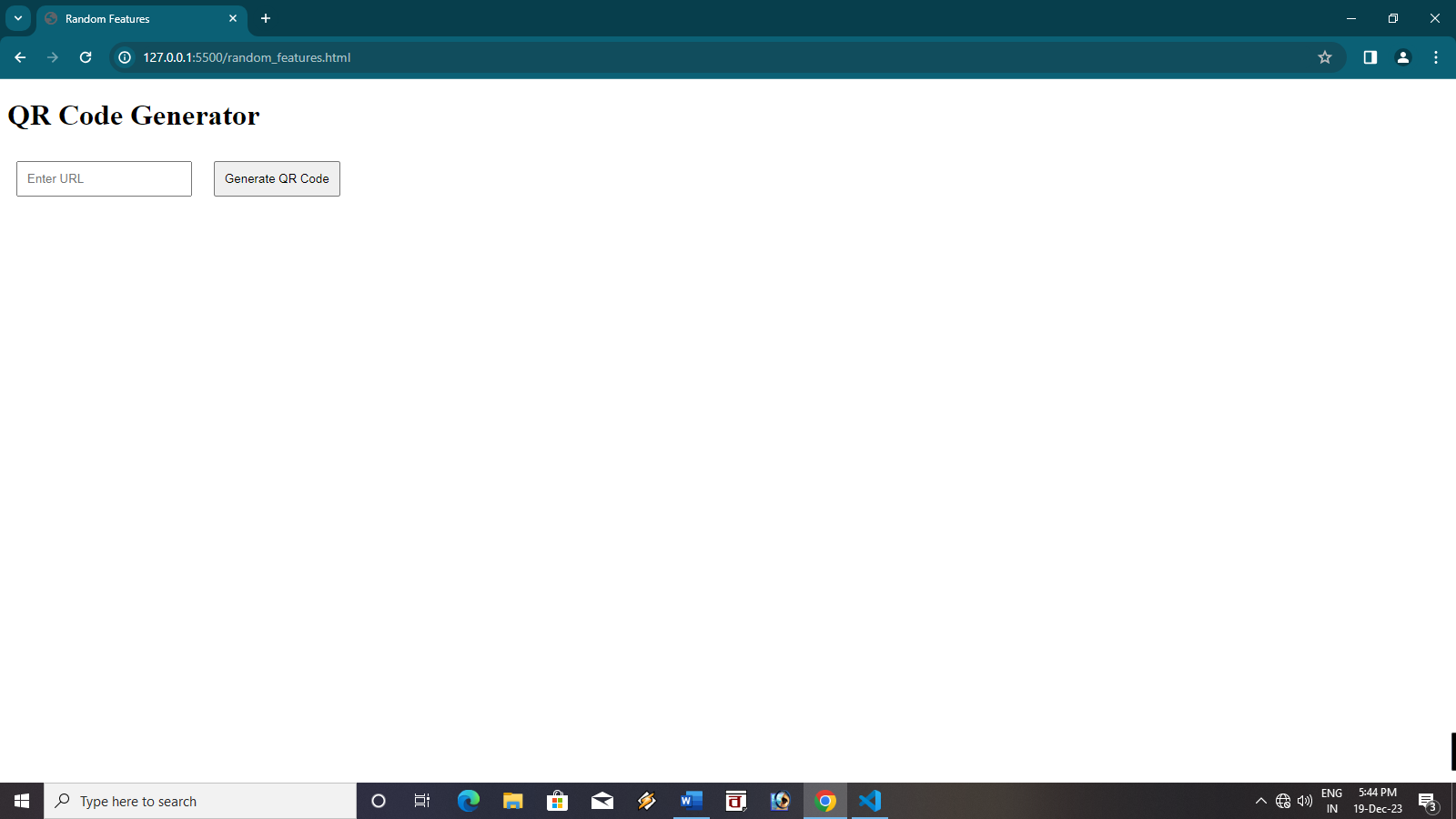
## **Result/Output**

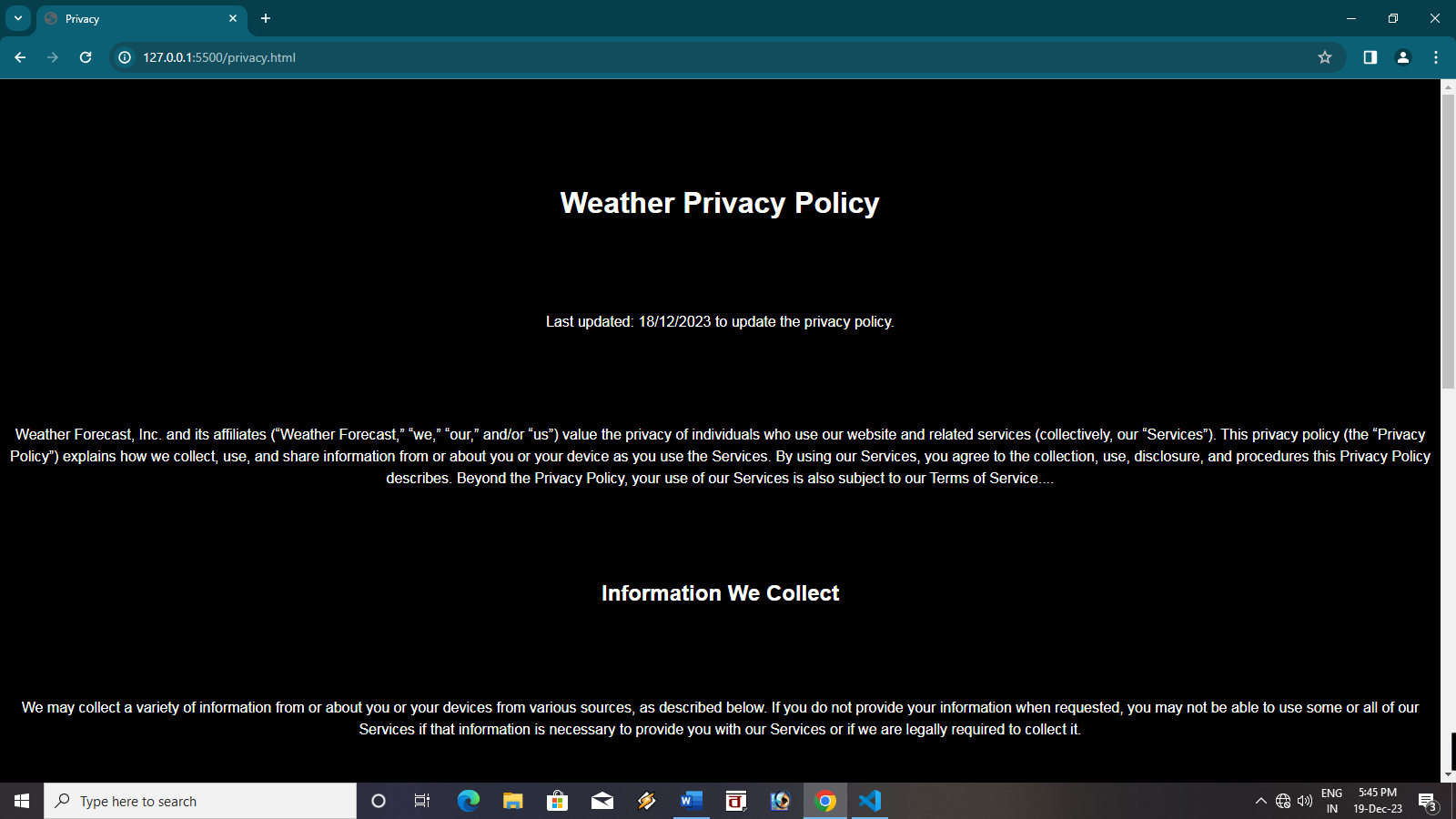












## **Conclusion**

The weather forecast application developed using HTML, CSS, and JavaScript effectively provides real-time weather information for users. The application is easy to use, visually appealing, and accessible on various devices. It utilizes Rapid API to fetch weather data from a reliable source, ensuring the accuracy and timeliness of the information presented.

The application's ability to display current weather conditions, forecasts for multiple days, and additional weather information makes it a valuable tool for planning daily activities and staying informed about potential weather changes. The customization options allow users to personalize their experience by selecting their preferred weather parameters, units of measurement, and location.

## **Reference**

* www.quabr.com/62184525/how-to-replace-output-properly
* github.com/aremanzh/SimpleWeatherStatus
* https://youtu.be/pFvWwFua6mw?si=8e0KOgcUpEyaFwOR

## **Appendices**

Thank You!

