MINI PROJECT REPORT

ON



Submitted by

**PRANEETH WARAN K**

**NC.SC.U4CSE24222**

For

23CSE113-User Interface Design

II Semester

B.Tech. CSE

School of Computing

Amrita Vishwa Vidyapeetham, Nagercoil

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Chapter Name** | **Page No.** |
| 1. | Introduction | **1** |
| 2. | Problem Statement | **2** |
| 3. | Objectives | 3 |
| 4. | System Design | 4 |
| 5. | Modules of project | 7 |
| 6. | Code | 9 |
| 7. | Output Screenshots | 16 |
| 8. | Application of the project | 18 |
| 9. | Limitations of the project | 20 |
| 10. | Bibliography (5 lines) | 22 |
| |  |  | | --- | --- | | 11. |  | | GITHUBLINK | 23 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



**INTRODUCTION**

This project allows users to retrieve their current location using the Geolocation API in their browser.

It displays both the user's public IP address and their real-time geographic coordinates (latitude and longitude).The IP address shown are v4 format(octal format)

If permission is granted, it shows the exact position on a Google Map embedded in the page.

Features:

Fetch user's public IP address via an API.

Get real-time location using the Geolocation API.

Show location accuracy.

Embed a Google Map based on live coordinates.

**Technologies Used:**

HTML, CSS, JavaScript

IP API (ipify,OpenCage)

Geolocation API

Google Maps (embedded iframe)

**1**

**PROBLEM STATEMENT**

**Title**: Real-Time Location Detection in Web Browsers

**Problem**:  
In today’s digital world, many web applications need to access and display the user’s current location for services like maps, local recommendations, and personalized experiences. However, detecting a user's exact location in real-time using a web browser poses challenges such as inconsistent accuracy, lack of device GPS, and restricted access due to privacy permissions.

To develop a simple web-based solution that retrieves and displays a user’s current geographical location using the browser's Geolocation API. The application should:

* Retrieve and display the user's public IP address.

**2**

**OBJECTIVES**



The main objectives of this project are:

1. **To retrieve and display the user’s public IP address**  
   Using a third-party API to fetch and show the user's IP for informational purposes.
2. **To detect the user's real-time geographical location**  
   Utilizing the browser's built-in Geolocation API to access the device’s GPS coordinates (latitude and longitude).
3. **To display the user's location on a map**  
   Embedding Google Maps with the user’s coordinates for a visual representation of their current position.
4. **To ensure user privacy and handle permissions properly**  
   Prompting the user for location access, and handling all potential errors such as denial of permission or unavailable location data.
5. **To enhance user experience with a clean, responsive interface**  
   Designing the page with modern styling (CSS) to make it simple, intuitive, and responsive on various screen sizes.

6**.To provide fallback options or messages in case of failure**  
Showing meaningful error messages when location services fail, time out, or are not supported.

3



SYSTEM DESIGN

The system is a lightweight web application that detects and displays the user's IP address and current geographical location using the browser’s APIs and third-party services. It is fully client-side and does not require a backend server.

| **Component** | **Description** |
| --- | --- |
| **HTML** | Provides the structure of the web page (buttons, output containers). |
| **CSS** | Styles the page to ensure a clean, modern, and responsive UI. |
| **JavaScript** | Handles logic for geolocation, API calls, and DOM updates. |
| **Geolocation API** | Browser API that fetches real-time location coordinates (requires user permission). |
| **IP API (ipify.org,OC)** | Public API used to retrieve the user’s public IP address. |
| **Google Maps Embed** | Displays a live map based on the user’s coordinates. |

**4**

**Workflow / Data Flow**

1. **User opens the page**
   * The application automatically fetches and displays the user’s IP address using https://api.ipify.org.

User clicks the "**Get Location**"button

* + JavaScript uses the Geolocation API to request permission.
  + If granted, the device’s GPS provides the latitude, longitude, and accuracy.

1. **Displaying Location**
   * The coordinates and accuracy are shown on the page.
   * A Google Maps iframe is loaded with the real-time position.
2. **If Geolocation Fails**
   * Appropriate error messages are shown (e.g., Permission Denied, Unavailable).
   * Optionally, IP-based geolocation can be used as a fallback.

5

WORK FLOW DESIGN

User Action

[Web Page UI]

[Fetch IP via ipify API]

Display IP Address on Page

Click "**Get Location**"

[Browser's Geolocation API]

Get Latitude & Longitude

Display Coordinates & Embed Google Map

If Error

Show Fallback/Error Message

**6**



MODULES OF PROJECT

**1. User Interface Module**

* **Purpose**: Displays the layout of the page, including buttons and output containers.
* **Technologies Used**: HTML + CSS

**Features**:

* + "Get Location" button
  + IP Address display area
  + Real-time location display (latitude, longitude, accuracy)
  + Embedded Google Map
  + Error/info messages

**2. IP Address Module**

* **Purpose**: Fetches and displays the user's public IP address.
* **Technologies Used**: JavaScript, fetch() API, ipify API

**Features**:

* + Calls https://api.ipify.org?format=json
  + Handles success or failure of API call
  + Displays the IP in the UI

**3. Geolocation Module**

* **Purpose**: Accesses the user’s real-time geographic location from the device.

**Technologies Used**: JavaScript,navigator.geolocatio

**7**

**Features**:

* + Retrieves latitude, longitude, and accuracy and addresses
  + Continuously watches location (watchPosition)
  + Handles geolocation errors

**4. Google Map Embed Module**

* **Purpose**: Visually represents the user’s location using Google Maps.
* **Technologies Used**: HTML iframe + coordinates from the Geolocation Module

**Features**:

* + Dynamically generates Google Maps URL
  + Embeds map centered on current user location

**5. Error Handling Module**

* **Purpose**: Provides informative feedback if something goes wrong.
* **Technologies Used**: JavaScript

**Features**:

* + Handles permission denial, timeouts, and device/location errors
  + Updates the UI with human-readable error message

8



CODE

**The following code included with the css**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Location</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            box-sizing: border-box;

            background-color: #f3f4f6;

            color: #333;

            display: flex;

            flex-direction: column;

            align-items: center;

            padding-top: 50px;

        }

        h2 {

            margin-top: 20px;

            font-size: 1.5rem;

            color: #4a5568; }

**9**

        p {

            margin: 5px 0;

            font-size: 1rem;

        }

        button {

            padding: 10px 20px;

            border: none;

            border-radius: 8px;

            background-color: #4299e1;

            color: white;

            font-size: 1rem;

            cursor: pointer;

            transition: background-color 0.3s;

            margin-bottom: 20px;

            margin-top: 100px;

        }

        button:hover {

            background-color: #3182ce;

        } #locationInfo {

            max-width: 600px;

            width: 90%;

            background: white;

            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);border-radius: 8px;padding: 20px; text-align: center;

        }

**10**

        iframe {

            width: 100%;

            height: 300px;

            border: 0;

            border-radius: 8px;

            margin-top: 15px;

        }

    </style>

</head>

<body>

    <button id="getSystemLocationBtn">Get Location</button>

    <div id="locationInfo">

        <p>Waiting for location...</p>

    </div>

    <script>

        let watchID = null;

        // Display user's IP address

        fetch('https://api.ipify.org?format=json')

            .then(response => response.json())

            .then(data => {

                document.getElementById('locationInfo').innerHTML = `<p>Your IP Address: ${data.ip}</p>`;

            })

            .catch(error => {

                console.error("Error fetching IP address:", error);

            });

        // Get system location on button click

        document.getElementById('getSystemLocationBtn').addEventListener('click', function () {

            if (navigator.geolocation) {

                if (watchID !== null) {

                    navigator.geolocation.clearWatch(watchID);

                }

                watchID = navigator.geolocation.watchPosition(showSystemLocation, showError, {

                    enableHighAccuracy: true,

                    timeout: 5000,

                    maximumAge: 0

                });

                document.getElementById('locationInfo').innerHTML += "<p>Tracking your location...</p>";

            } else {

                document.getElementById('locationInfo').innerHTML += "<p>Geolocation is not supported by this browser.</p>";

            }

        });

        // Show system location and fetch address

        function showSystemLocation(position) {

            const lat = position.coords.latitude;

            const lon = position.coords.longitude;

            const accuracy = position.coords.accuracy;

            document.getElementById('locationInfo').innerHTML = `

                <p>Your IP Address is still displayed above.</p>

                <h2>Real-Time System Location:</h2>

                <p>Latitude: ${lat}</p>

                <p>Longitude: ${lon}</p>

                <p>Accuracy: ±${accuracy} meters</p>

                <iframe

                    src="https://www.google.com/maps?q=${lat},${lon}&z=15&output=embed"

                    allowfullscreen>

                </iframe>

                <p><strong>Fetching address...</strong></p>

            `;

            // Reverse geocoding using OpenCage API

            fetch(`https://api.opencagedata.com/geocode/v1/json?q=${lat}+${lon}&key=5f3cac373c3542f698f207cbb1851b95`)

                .then(response => response.json())

                .then(data => {

                    const address = data.results[0]?.formatted || "Address not found.";

                    document.getElementById('locationInfo').innerHTML += `<p><strong>Address:</strong> ${address}</p>`;

                })

                .catch(error => {

                    console.error("Error fetching address:", error);

                    document.getElementById('locationInfo').innerHTML += `<p>Unable to fetch address.</p>`;

                });

        }

        // Handle location errors

        function showError(error) {

            let message = '';

            switch (error.code) {

                case error.PERMISSION\_DENIED:

                    message = "User denied the request for Geolocation.";

                    break;

                case error.POSITION\_UNAVAILABLE:

                    message = "Location information is unavailable.";

                    break;

                case error.TIMEOUT:

                    message = "The request to get user location timed out.";

                    break; 

                case error.UNKNOWN\_ERROR:

                    message = "An unknown error occurred.";

                    break;

            }

            document.getElementById('locationInfo').innerHTML += `<p>${message}</p>`;

        }

    </script>

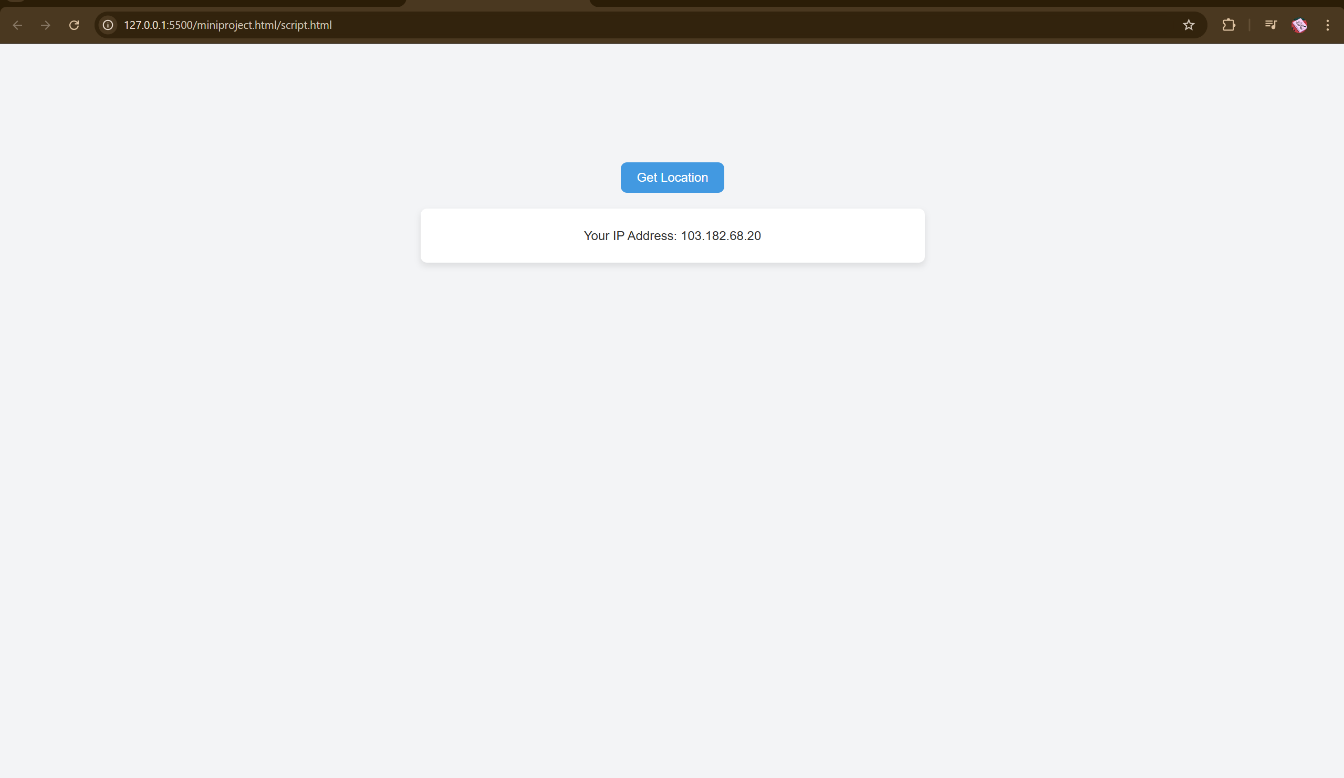
</body>

</html>

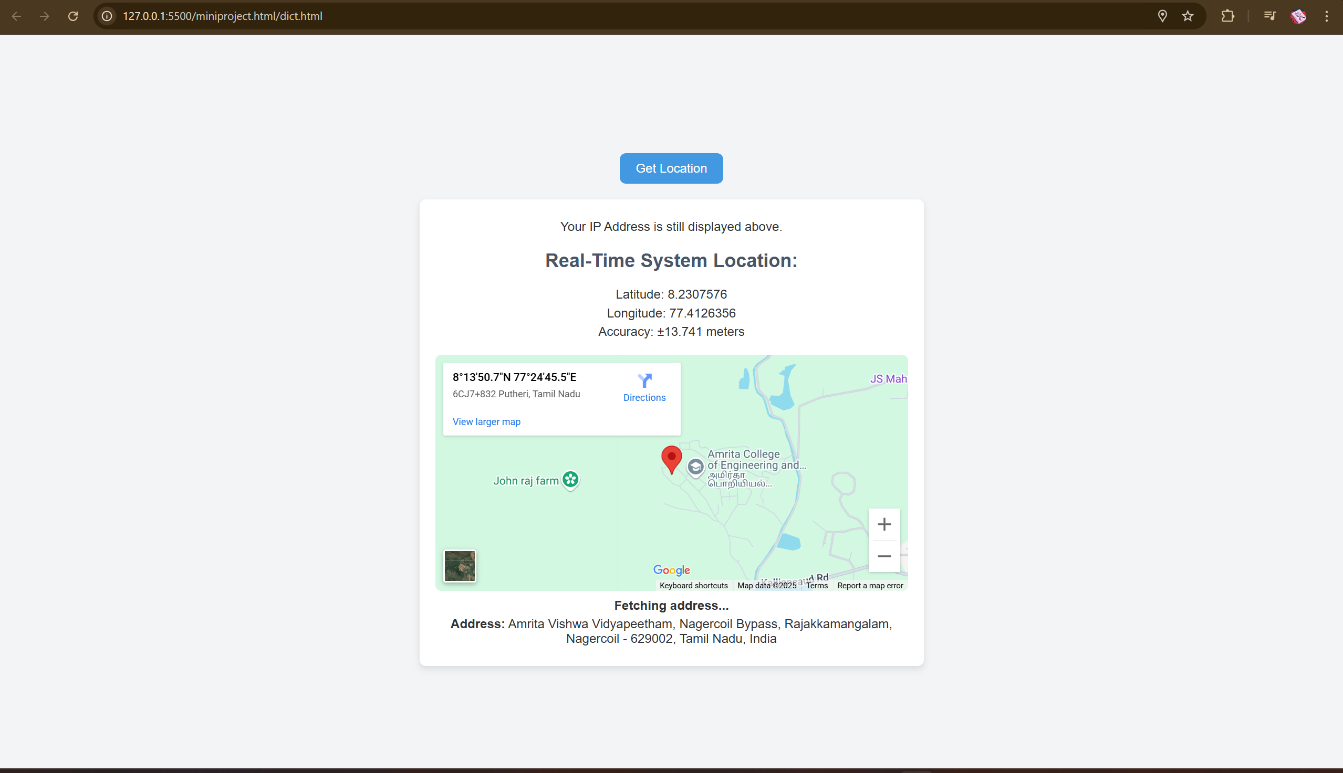
OUTPUT SCREENSHOTS

ON CONNECTING WITH AVVP CONNECT WIFI

IP ADDRESSES



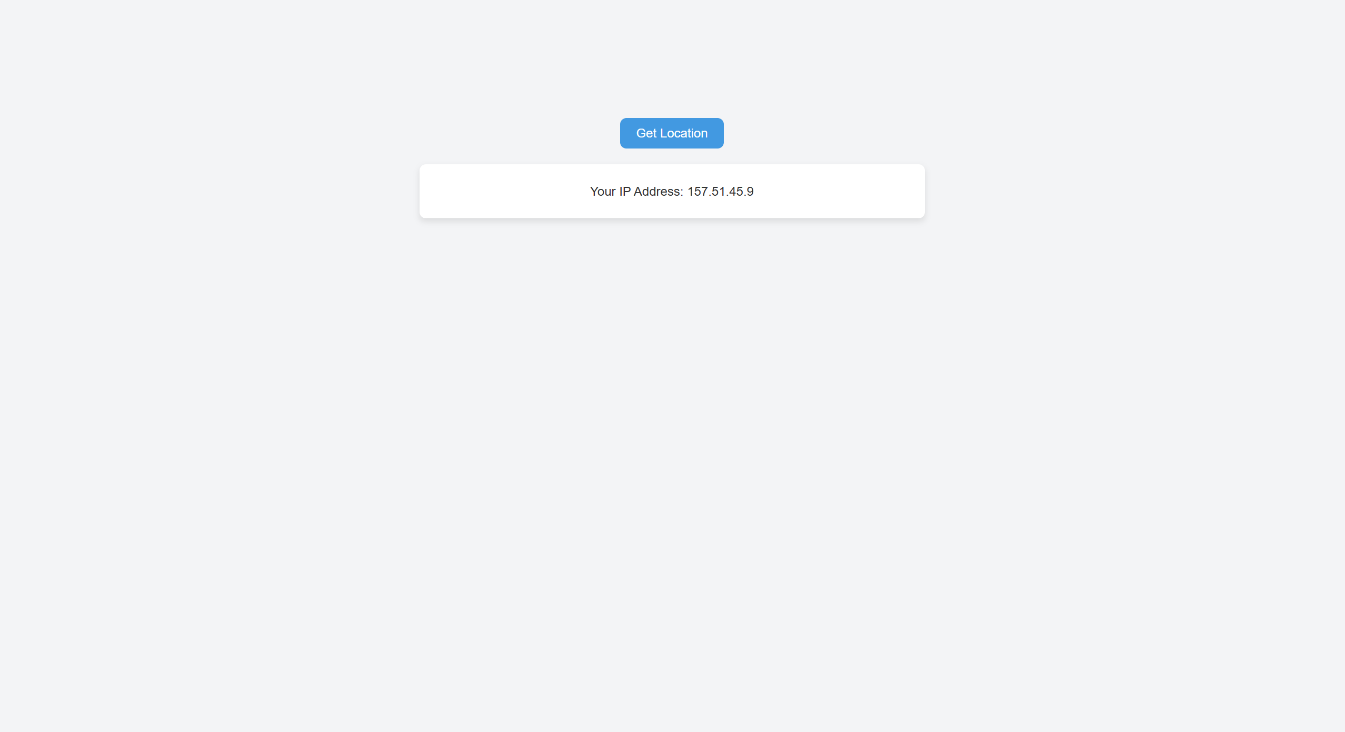
LOCATION:



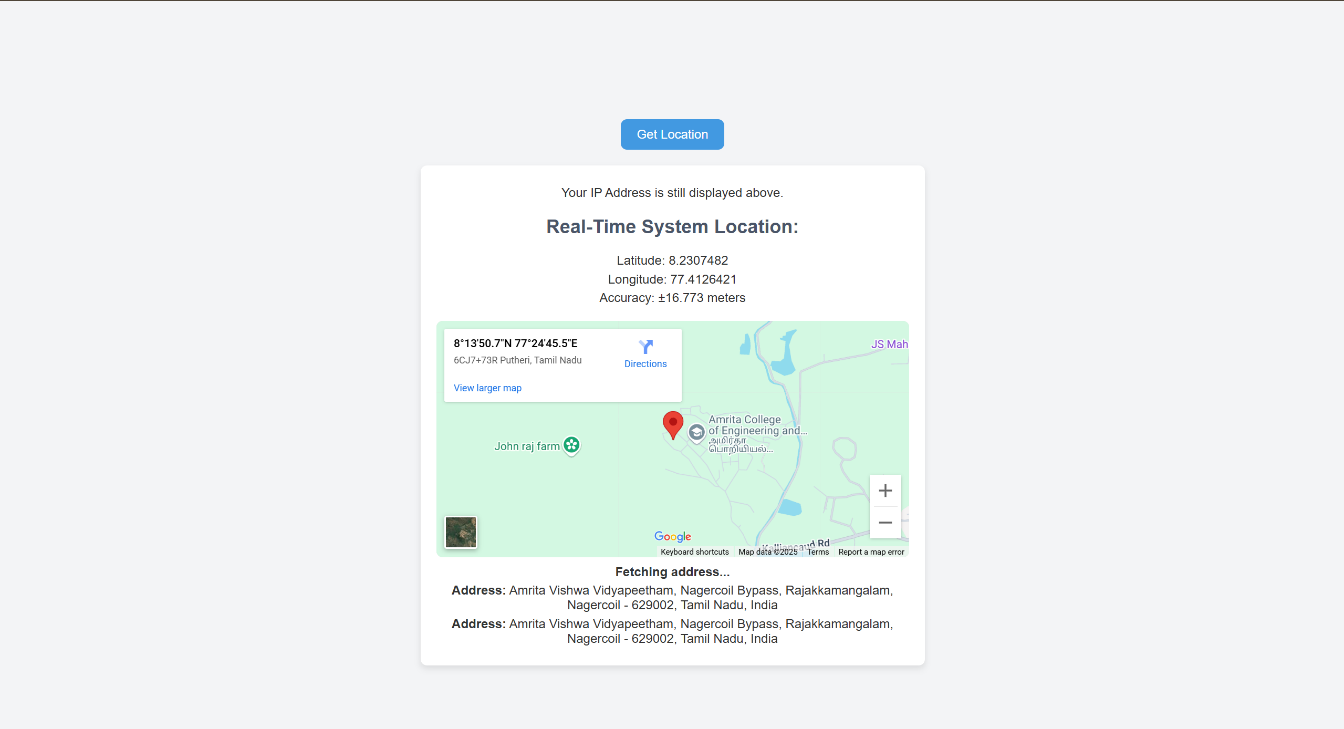
**16**

ON CONNECTION WOITH DIFFERENT WIFI

IP ADDRESS



LOCATION



**17**



APPLICATION OF THE PROJECT

**1.Location-Based Services**

Can be used as a base for building applications like:

* + Nearby restaurant or hospital finders
  + Local event listings
  + Weather apps showing local forecasts

**2.Travel & Navigation Tools**

* Can be integrated into:
  + Travel booking platforms to detect user location for pickup/drop suggestions
  + Navigation helpers or maps-based services

**3. E-commerce & Delivery Systems**

* Useful in:
  + Auto-detecting user address for product delivery
  + Showing availability of services based on region

**18**



**4. Security & Access Control**

* Can be used in web apps to:
* Restrict or grant access based on geographic location
* Trigger alerts if user location is suspicious or outside allowed areas

**5. Web Analytics & User Insights**

* Can help website owners:
  + Understand where their users are accessing from
  + Customize content and offers based on user location

LIMITATIONS

**1. User Permission Dependency**

* The Geolocation API requires **explicit permission** from the user.
* If permission is denied, the system cannot access location data.

**2. Accuracy Depends on Device & Network**

* Location accuracy is limited by:
  + Device capabilities (no GPS = less accurate)
  + Wi-Fi or cellular data quality
  + Browser settings and permissions

**3. IP Address is not equal to Exact Location**

* The IP-based location is **approximate** and can be **regionally inaccurate**.
* It's not suitable for pinpointing precise physical locations.

**20**



**4.Offline Limitation**

* The app requires an **active internet connection** to:
  + Fetch IP address
  + Display Google Maps
  + Load styling and run APIs

**5. No Backend or Data Storage**

* There's no server-side logic or database integration.
* User location data isn't stored or processed beyond display.

21

BIBLIOGRAPHY

1. ipify.org, third party API Open Cage-Free IP Address API, <https://www.ipify.org>
2. Google Maps Embed API -https://developers.google.com/maps/documentation/embed
3. W3Schools -HTML, CSS, JavaScript Tutorials, <https://www.w3schools.com>
4. Stack Overflow ,YouTube– Community discussions on geolocation and API integration, <https://stackoverflow.com>
5. Greeks for Greeks -what is an IP address https://www.geeksforgeeks.org/what-is-an-ip-address/

22



GITHUB LINK

https://github.com/PRANEETHWARANK/USERINTERFACE-USEREXPIRENCE.git

**23**