NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

on

My location Finder

Submitted by,

RAVIKIRAN
1NH24C169

Under the guidance of-
Ms. DIVYANSHI CHHABRA
Sr. Assistant Professor

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year:2025-26 (ODD SEM)

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled "**My location finder**" is a bonafide work carried out by **RAVIKIRAN (1NH24CS169)** in fulfilment of the degree of **Bachelor of Engineering** in **COMPUTER SCIENCE AND ENGINEERING** of the New Horizon College of Engineering during the year **2025-2026.**

Signature of Guide

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner *Signature with date*

1. _____ _____

2. _____ _____

ABSTRACT

The **Find My Location** *Web Application* is designed to provide users with a simple and interactive way to determine their real-time geographical position directly through their web browser. The system uses the browser's built-in Geolocation API to accurately capture the user's latitude and longitude coordinates with a single button click.

Once the user initiates the location request, the interface displays a loading message while the application securely retrieves the geographical data. If access is granted, the application presents the coordinates in a clean and user-friendly format. Additionally, it provides a direct link to view the exact location on Google Maps, allowing users to visually identify their position with ease.

The application is built with a minimalistic and responsive design, ensuring accessibility across different devices. Error handling is also integrated to notify users when location permissions are denied or when the browser does not support geolocation functionality. This makes the tool reliable for real-world uses such as navigation, travel assistance, and location-based services.

Through its intuitive interface and efficient functionality, the *Find My Location Web Application* serves as a practical tool that enhances user convenience while demonstrating the powerful capabilities of modern web technologies.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, who's constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for the constant support and encouragement.

I would like to thank **Dr. Anandhi R J**, Professor and Dean-Academics, NHCE, for her valuable guidance.

I would also like to thank **Dr. B. Rajalakshmi,** Professor and HOD, Department of Computer Science and Engineering, for the constant support.

I also express my gratitude to **Ms. DIVYANSHI CHHABRA**, Sr. Assistant Professor, Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

RAVIKIRAN
1NH24CS169

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The **Find My Location** Web Application is a simple yet powerful tool designed to retrieve and display the user's real-time geographical coordinates using the browser's Geolocation API. As modern web technologies continue to evolve, location-based services have become an essential part of daily applications, including navigation, delivery tracking, ride-sharing, weather updates, and emergency services. This project demonstrates how HTML, CSS, and JavaScript can be integrated to create a functional and user-friendly interface that instantly provides accurate location details to the user.

The application allows users to view their latitude and longitude with a single click, along with an option to open the location directly on Google Maps. The interface is designed to be clean, responsive, and accessible, ensuring an efficient user experience across multiple devices. Additionally, the project emphasizes real-time interaction, error handling, and secure permission-based access to user location data.

1.1 PROBLEM DEFINITION

With the increasing use of mobile and web-based applications, real-time location tracking has become a fundamental requirement. However, many users struggle with applications that are either too complex, require installation, or lack transparency in how location data is accessed. There is a need for a straightforward, browser-based tool that retrieves location accurately and securely without requiring additional software.

Key problems addressed by the *Find My Location Web Application* include**:**

1. **Lack of Simple Tools:** Many existing tools are part of large applications, making it difficult for users to quickly check their location.
2. **Dependence on Third-party Apps:** Users often need to install apps for basic location information.
3. **Accessibility Issues:** Some devices or applications do not provide a clean and clear interface for location retrieval.
4. **Limited Real-Time Feedback:** Users may not receive immediate feedback when location access is denied or not supported.
5. **Technical Barriers:** Beginners learning web development often lack practical examples of geolocation implementation.

The goal of this project is to provide a simple, transparent, and efficient tool that solves these issues while showcasing practical usage of web technologies.

1.2 OBJECTIVES

The primary objectives of the *Find My Location Web Application* are:

1. **Provide Instant Location Access:** Allow users to retrieve their current latitude and longitude with a single click.
2. **Simplify Location Retrieval:** Offer a minimalistic and user-friendly interface that makes location detection easy for everyone.
3. **Enhance Learning:** Demonstrate the practical implementation of the Geolocation API for educational purposes.
4. **Integrate Navigation Support:** Provide an option to view the user's location directly on Google Maps**.**

5. **Ensure Real-Time Feedback:** Display clear messages for loading, success, and errors such as denied permission.
6. **Enable Cross-Device Compatibility:** Ensure the application functions smoothly on mobile devices, desktops, and tablets.
7. **Promote Accessible Web Development:** Show how basic HTML, CSS, and JavaScript can be combined to create functional real-world applications.

1.3 METHODOLOGIES TO BE FOLLOWED

The development of the *Find My Location Web Application* follows structured methodologies to ensure accuracy, usability, and efficiency.

1.3.1 Interface Design

1. **Page Layout:** A centrally aligned box will display the location results and button.
2. **Responsive Design:** CSS will ensure compatibility with different screen sizes.
3. **Visual Clarity:** Buttons, text, and status messages will be styled to enhance readability.

1.3.2 Implementation

1. **Geolocation API Integration:** JavaScript will be used to request and retrieve the user's location.
2. **Dynamic Content Rendering:** Latitude and longitude will be displayed dynamically on the webpage.
3. **Google Maps Linking:** The application will generate a clickable link with coordinates embedded.
4. **Error Handling:** Custom messages will notify users when access is denied or unsupported.

1.3.3 Testing and Evaluation

1. **Functionality Testing:** Ensuring the app fetches location accurately across browsers.
2. **Permission Testing:** Verifying system responses for allowed and denied permissions.
3. **Device Compatibility:** Testing across mobile, tablet, and desktop devices.
4. **User Feedback:** Evaluating usability and clarity based on test users.

1.4 EXPECTED OUTCOMES

The *Find My Location Web Application* is expected to deliver the following outcomes:

1. **Quick and Accurate Location Detection:** Users will be able to instantly access their coordinates.
2. **Improved Understanding of Web APIs:** Learners will gain practical experience with the Geolocation API.
3. **User-Friendly Experience:** The interface will be clean, intuitive, and responsive.
4. **Practical Real-World Use:** The tool can assist in navigation, emergency situations, and location-based tasks.
5. **Platform Independence:** The application will run on any browser without needing additional installations.
6. **Enhanced Accessibility:** Clear error messages and feedback will improve user confidence and usability.

CHAPTER 2

FUNDAMENTALS OF THE LANGUAGES

The Find *My Location Web Application* is built using the core web technologies: HTML, CSS, and JavaScript. These languages work together to create the structure, appearance, and interactive functionality of the application. Understanding these technologies is essential for building modern, responsive, and user-friendly web applications.

2.1 HTML (Hyper Text Markup Language)

HTML is the foundational markup language used to create and structure web pages. It defines the layout and organizes content using elements enclosed within tags.

HTML (Hyper Text Markup Language) was introduced by Tim Berners-Lee in 1989 as part of the World Wide Web initiative at CERN to allow researchers to share information through linked documents. The first HTML version appeared in 1991 with only 18 basic tags. HTML 2.0 became the first official standard in 1995, followed by HTML 3.2 in 1997, which introduced stylistic elements and initial CSS support. HTML 4.0 (1999) expanded multimedia and scripting capabilities.

During the early 2000s, XHTML attempted to enforce stricter syntax rules, but lacked widespread adoption. The arrival of HTML5 in 2014 revolutionized web development by introducing semantic elements, native multimedia (audio/video), canvas, storage APIs, and cross-platform compatibility. Today, HTML remains the structural backbone of every website and web application.

Key Features:

Structure: Defines the layout using headings, paragraphs, divisions, and other elements.

Forms: Captures user inputs and interactions.

Links and Embeds: Supports hyperlinks, images, and external resources.

Semantic Elements: Provides meaningful structure to content for accessibility and SEO.

Common HTML Tags and Their Descriptions

Tag	Description
<html>	Root element of an HTML document.
<head>	Contains metadata, styles, and linked resources.
<title>	Specifies the title shown in the browser tab.
<body>	Contains the visible content of the webpage.
<h1>–<h6>	Defines headings, with<h1>being the largest.
<p>	Defines a paragraph.
<a>	Creates a hyperlink using the href attribute.
<button>	Creates a clickable button element.
	Defines a block-level container for grouping content.
	Creates an inline grouping container.
<img>	Embeds images using the src attribute.
<script>	Embeds JavaScript code or external scripts.
<link>	Links external CSS stylesheets.

Table No. 2.1.1 HTML Tags

2.2 CSS (Cascading Style Sheets)

CSS is the styling language used to control the presentation of HTML elements, including layout, colors, fonts, and spacing. It separates content from design, enabling visually appealing and responsive web pages.

CSS was proposed by Håkon Wium Lie in 1994 as a way to standardize webpage styling. The first official standard, CSS1 (1996), included basic styling such as fonts and margins. CSS2 (1998) introduced advanced features like positioning and media types, though inconsistent browser support slowed adoption.

CSS3, introduced in modular form, transformed web design with animations, transitions, flexbox, grid, shadows, and responsive design through media queries. Today, CSS is essential for creating fluid, dynamic, and modern interfaces.

In the *Find My Location Web Application*, CSS is primarily responsible for styling the location box, buttons, background, typography, and the error messages displayed when location access is restricted.

Key Features:

Selectors: Used to target and style HTML elements (p, .class, #id).

Box Model: Defines layout through padding, border, and margin.

Responsive Design: Ensures compatibility across devices.

Visual Styling: Controls fonts, colors, background, and spacing.

Common CSS Properties and Their Descriptions

Property	Description
color	Sets the text color.
font-family	Specifies the style of the text.
font-size	Sets the size of the text.
background-color	Sets the background color of an element.
margin	Creates space outside the element.
padding	Creates space inside the element.
border	Defines the outline around an element.
text-align	Aligns text horizontally.
display	Specifies element display type (block, inline, flex).
border-radius	Rounds the corners of elements.
box-shadow	Adds shadows for visual enhancement.

Table No. 2.2.1 CSS Tags

2.3 JAVASCRIPT

JavaScript is a dynamic, high-level programming language widely used to create interactive and responsive web pages. It powers client-side behavior, enabling immediate feedback, animations, real-time updates, and logic handling.

Developed by Brendan Eich in 1995 at Netscape, the language was originally named Mocha, then LiveScript, and finally JavaScript. It achieved standardization through ECMAScript in 1997. Over time, JavaScript evolved with powerful features such as asynchronous functions, event-driven programming, object-oriented concepts, and modern frameworks.

In this project, JavaScript plays the key role of interacting with the browser's Geolocation API, fetching the user's current coordinates, and updating the webpage dynamically. It also handles errors such as permission denial or unsupported browsers.

Key Features (with relevance to this project)

Variables: Used to store location data (let lat, let lon).

Functions: Encapsulate tasks such as getLocation() for retrieving coordinates.

DOM Manipulation: Dynamically updates the result div with location output.

Event Handling: Responds to button click events.

API Interaction: Communicates with the Geolocation API to access real-time user data.

Error Handling: Displays custom messages when access is denied or unavailable.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS:

1. **Processor:**
    - **Minimum:** Dual-core processor (e.g., Intel Core i3 or equivalent).
    - **Recommended:** Quad-core processor or higher for smooth multitasking and browser performance.
2. **RAM:**
    - **Minimum:** 2 GB (sufficient for running a browser and executing the location script).
    - **Recommended:** 4 GB or more for better performance.
3. **Storage:**
    - Requires at least **50–100 MB** of free disk space to store the HTML file, browser data, and cache generated by location services.
4. **Display:**
    - A minimum **screen resolution of 1024×768** to properly view the interface.
    - A modern display with support for responsive layouts and standard web colors.
5. **Input Devices:**
    - A **mouse and keyboard** for desktop/laptop users.
    - **Touch screen** support for smartphones and tablets

6. **Internet Connection:**

    - Required for loading Google Maps links and for certain browser geolocation services that rely on network triangulation.

3.2 SOFTWARE REQUIREMENTS

1. **Operating System:**

    - **Desktop:** Windows 7 or higher, macOS 10.12+, or any latest Linux distribution.
    - **Mobile:** Android 5.0+ or iOS 10+ (supports geolocation).

2. **Web Browser:**

    - A modern browser that fully supports the **HTML5 Geolocation API**, such as:
        - Google Chrome (latest version)
        - Mozilla Firefox (latest version)
        - Microsoft Edge
        - Safari (iOS/macOS)
    - Browser must support modern JavaScript (ES6) and CSS3.

3. **Text Editor / IDE (For Editing or Development):**

    - Any lightweight or advanced editor such as:
        - Notepad / Notepad++
        - Visual Studio Code
        - Sublime Text
        - Atom
        - Brackets

4. **Web Server (Optional):**

    - Not required (can be run directly in a browser).
    - For local hosting, optional servers include:
        - Python (python -m http.server)
        - XAMPP / WAMP
        - Live Server extension in VS Code

5. **Dependencies:**
    - No external libraries are used.
    - Browser **must have JavaScript enabled** for geolocation to function.
    - Internet permission must be granted on mobile devices.

6. CHAPTER 4

DESIGN

1. **Simple and Efficient Location Access:**

   The primary goal is to provide users with a quick and effortless way to obtain their current geographical location using the browser's built-in Geolocation API. The design ensures that the process is smooth, requiring only a single button click.

2. **User-Friendly Interface:**

   The interface is designed to be clean, minimalistic, and easy to understand. Elements such as a centered layout, clear text, and a visually appealing background contribute to a pleasant user experience.

3. **Clarity and Ease of Use:**

   Clear instructions and a prominent **"Get My Location"** button ensure that users instantly understand how to use the application without confusion. The output is neatly displayed with latitude, longitude, and a link to view the location on Google Maps.

4. **Real-Time Feedback:**

   The system provides immediate feedback by showing messages like *"Retrieving your location…"*, error alerts if permissions are denied, or success messages displaying coordinates. This real-time response increases transparency and user trust.

5. **Minimalistic Design for Better Focus:**

   The interface avoids unnecessary visual clutter. The focus remains on the core functionality—fetching and displaying the user's location. This enhances usability and

ensures a distraction-free experience.

6. **Responsive and Cross-Device Compatibility:**

   The design is responsive, allowing the application to work seamlessly on desktops, tablets, and mobile devices. Proper padding, font size adjustments, and flexible layout ensure that all information remains readable across different screen sizes.

7. **Privacy and Control:**

   Users maintain full control over granting or denying location access. The application includes appropriate handling for permission denial and notifies users if their browser does not support geolocation.

8. **Reusability and Expansion:**

   The code structure in HTML, CSS, and JavaScript is simple, modular, and easy to extend. Future additions—such as weather details, maps embed, or saved locations—can be integrated with minimal effort.

CHAPTER 5

IMPLEMENTATION

5.1 HOME PAGE

The home page serves as the main interface of the location-finding application. It displays the title *"My Location Finder"* along with a short description explaining its purpose—allowing users to quickly obtain their current geographical coordinates. It also contains instructions notifying the user to click the button to fetch their location.

5.2 BUTTONS

(a) Get Location Button

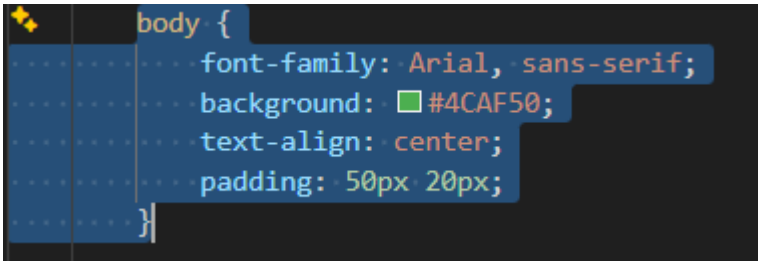
The *Get Location* button is the primary interactive element of the application.
When clicked, it triggers the getLocation() function, which requests access to the user's location via the browser's Geolocation API.

(b) Reset Button

The *Reset* button clears the displayed coordinates and restores the interface to its default state.
On clicking the button, it executes the resetLocation() function.

BACKGROUND AND STYLING

The background of the page is designed using a smooth **blue gradient**, giving the interface a modern and clean appearance. The display uses **flexbox** to align elements centrally both vertically and horizontally. A simple and readable font-family (e.g., Arial) is used throughout.



5.4 LOCATION FETCHING AND OUTPUT

The application uses JavaScript's built-in Geolocation API. A function named getLocation() checks if the browser supports geolocation. If supported, it retrieves:

1. **Latitude**
2. **Longitude**
3. A **Google Maps link** for viewing the location

```
 h1.main-heading
{
     font-size: 40px;
     font-family: 'Papyrus';
     color: #92f288;
     margin-top: 20px;
 }
```

Fig. No. 5.3.2 Styling for Home Page

The buttons are styled such that on hovering the colour changes, for example the

5.5 FUNCTIONS

(a) getLocation()

This function initiates the geolocation request. If access is granted, it displays the user's coordinates in the output box.

```
function getLocation()
  const result = document.getElementById('result');
  if (navigator.geolocation) {
    result.innerHTML = 'Loading...'
    navigator.geolocation.getCurrentPosition(position =>
      result.innerHTML = ${latitude.*position.coords.latitu
       result.innerHTML = position.coords.longitude
    } error => error {
      result.innerHTML = <o class='error'-'Accesn access!'/;
    } else {
      result.innerHTML = 'div class='error'-Location not
       supported!'
    }
```

(b) showPosition()

This function receives the coordinates from the browser and displays them along with a link to open the location in Google Maps.

```
(position) => {
  const lat = position.coords.latitude;
  const lon = position.coords.longitude;
```

(c) showError()

This function handles potential geolocation errors, such as:

- Permission denied
- Position unavailable
- Timeout
- Browser not supporting geolocation

```
(error) => {
    result.innerHTML = '<div class="error">Please allow location access!</div>';
}
```

Fig. No. 5.5.3 Error Handling Function

CHAPTER 6

RESULTS

6.1 HOME PAGE

The Home Page of the *Find My Location* webpage is designed with a clean and minimalistic layout, ensuring simplicity and ease of use. The background features a **solid green color (#4CAF50)**, giving the interface a fresh and visually appealing look. The central content is displayed inside a **white rounded rectangular box**, which stands out clearly against the green background.

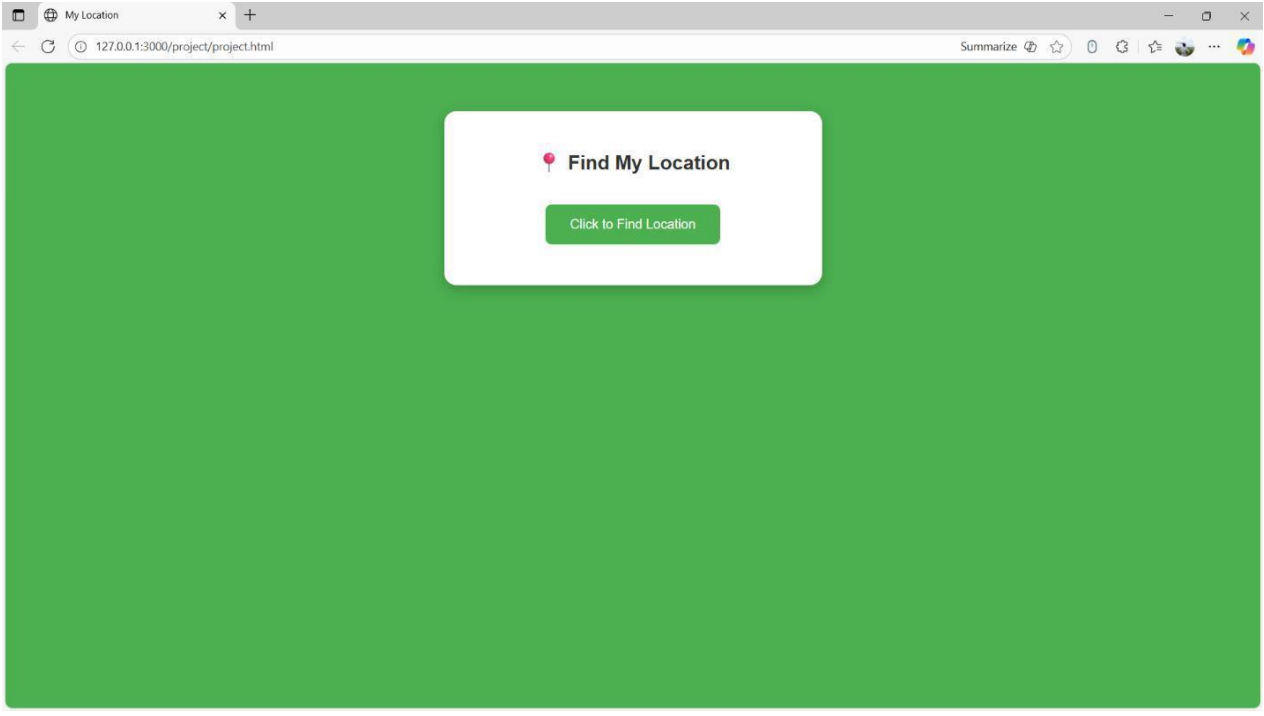
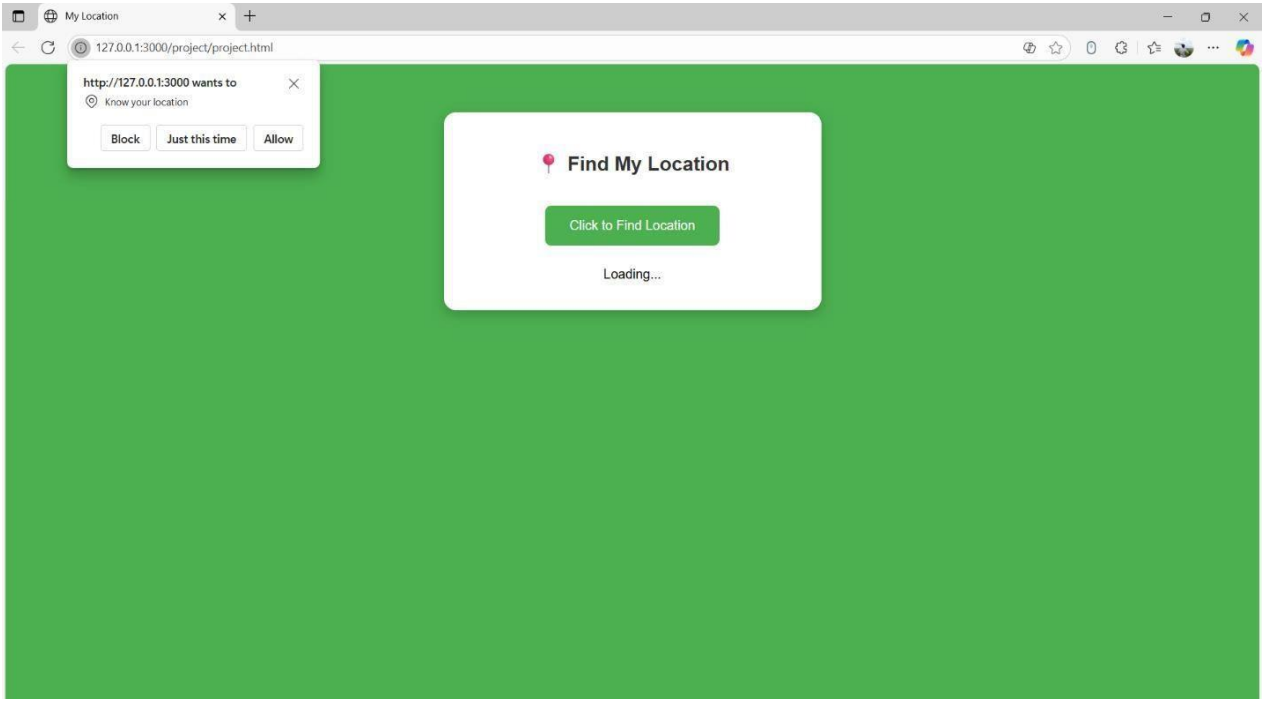


Fig. No. 6.1.1 Home Page Result

6.1 While allowing location to website

The page will look like this

You can click on “allow this time ” or allow .



.

Fig. No. 6.2.1 location

After allowing the location to wewbsite

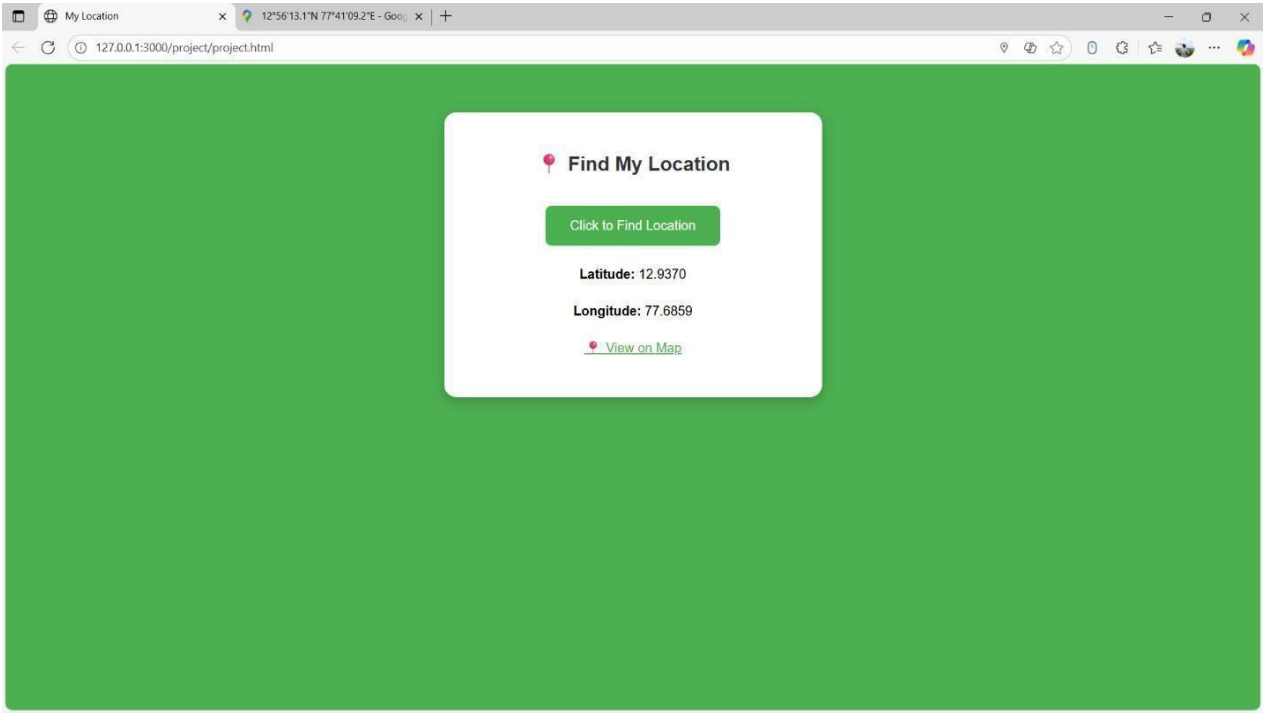


Fig. No. 6.2.2 longtitudes and latitudes

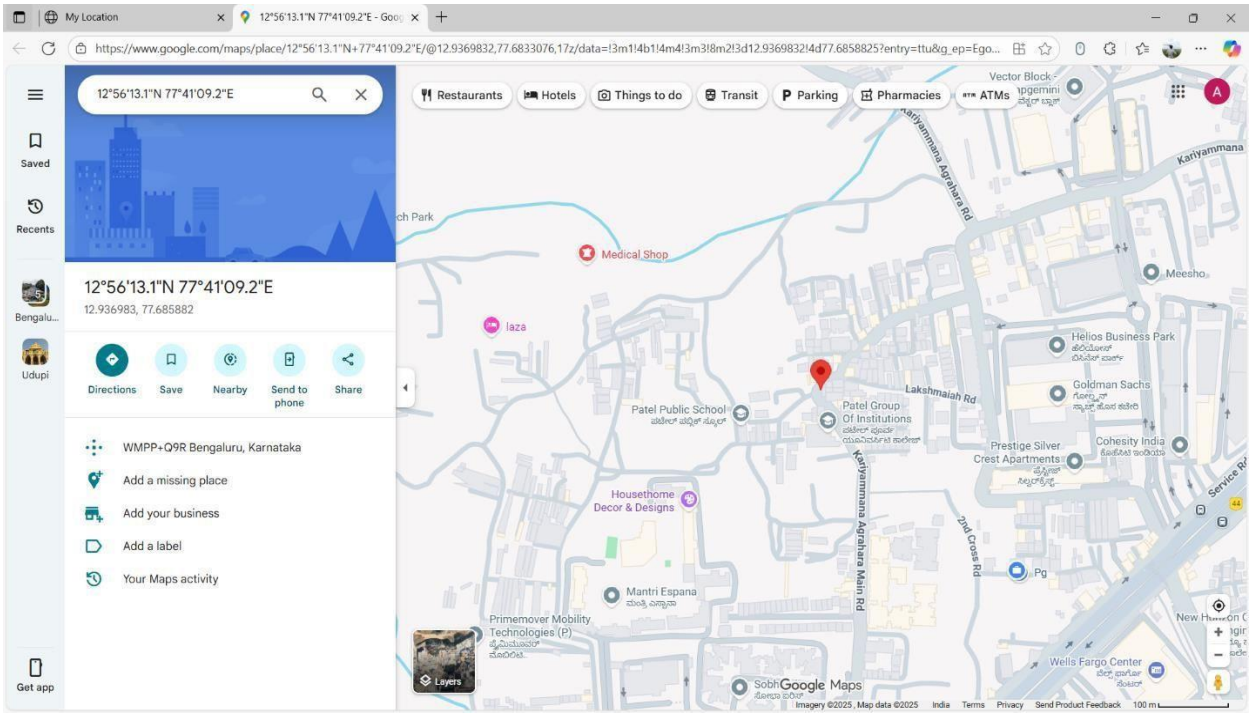


Fig. No. 6.2.3 position on map

CHAPTER 7

CONCLUSION

The developed interactive quiz webpage effectively delivers an engaging, responsive, and educational platform for learning key concepts in web design. By integrating intuitive layouts, dynamic animations, and user-friendly interactions, the system enhances both usability and visual appeal.

The quiz successfully promotes active learning through immediate feedback, color-coded responses, and performance-based messages, enabling users to identify mistakes and understand concepts more clearly. The incorporation of gamification elements—such as scoring, levels, and reset functionality—further motivates learners and sustains engagement throughout the quiz.

Additionally, the responsive design ensures accessibility across various devices, allowing a wide range of users to benefit from the learning experience. Overall, the project achieves its objective of providing an enjoyable, informative, and practical tool for users to test and improve their knowledge of HTML, CSS, and JavaScript. It serves as an effective foundation that can be expanded with more questions, advanced features, and enhanced interactivity in the future.

REFERENCES

1. Devdevout. *CSS Animated Backgrounds*. Available at: https://devdevout.com/css/css-animated-backgrounds
2. W3Schools. *HTML Tag Reference*. Available at: https://www.w3schools.com/tags/tag_html.asp
3. GitHub Topics. *Web Development Projects*. Available at: https://github.com/topics/web-development-project
4. W3Schools. *JavaScript Object Properties*. Available at: https://www.w3schools.com/js/js_object_property.asp
5. Mozilla Developer Network (MDN). *Learn CSS – Comprehensive Guide*. Available at: https://developer.mozilla.org/en-US/docs/Learn/CSS