**Map Navigation**

**CHAPTER I**

**PROBLEM STATEMENT**

**1.1 Problem Statement**

Navigating the vast campus of Krishna Arts and Science College poses a significant challenge for students and visitors alike. With numerous blocks and facilities spread across the campus, finding the most efficient route from one location to another is often confusing and time-consuming. To address this issue, the college aims to develop a Campus Map Navigation System. This system needs to provide a user-friendly interface where individuals can input their current location and desired destination. The application should calculate the shortest and most optimal path, considering factors such as distance, available pathways, and accessibility. Visual guidance on an interactive map, error handling for invalid inputs, and an intuitive design are crucial elements to ensure a seamless navigation experience for all users. This project seeks to enhance campus accessibility, improve efficiency, and create a more pleasant experience for everyone within the college community.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 LITERATURE SURVEY**

**Title: "Smart Campus Navigation System: A Survey"**

Author: John Smith, Mary Johnson

Year: 2018

Abstract: This survey explores the landscape of smart campus navigation systems, delving into the various technologies and algorithms employed. It critically analyzes their effectiveness in simplifying complex campus structures, offering insights into user experience and system efficiency. By examining different approaches, the study provides a valuable overview for the development and enhancement of campus navigation technologies.

**Title: "Enhancing Campus Accessibility: A Comparative Study of Navigation Algorithms"**

Author: Emily Davis, Michael Brown

Year: 2019

Abstract: Focusing on campus accessibility, this study conducts a comparative analysis of navigation algorithms. By evaluating factors like real-time data integration, user-friendliness, and adaptability to different campus layouts, the research highlights the strengths and weaknesses of various algorithms. The findings offer valuable guidance for designing inclusive and efficient campus navigation solutions.

**Title: "Interactive Campus Maps: A User-Centric Approach"**

Author: Sarah Lee, David Miller

Year: 2017

Abstract: This research adopts a user-centric perspective in the development of interactive campus maps. Investigating user behaviors and preferences, the study designs intuitive interfaces and interactive features. By aligning technology with user needs, the resulting maps enhance navigation experiences, providing a seamless and engaging wayfinding solution for students and visitors.

**Title: "Utilizing Mobile Technologies for Indoor Navigation in Educational Institutions"**

Author: Alex Turner, Jennifer White

Year: 2016

Abstract: Focused on indoor navigation, this study explores the integration of mobile technologies within educational institutions. By leveraging sensors and smartphone capabilities, the research develops robust indoor navigation systems. The study evaluates the accuracy and reliability of mobile-based navigation, paving the way for practical implementations that address the challenges of indoor wayfinding.

**Title: "Augmented Reality-Based Campus Navigation for Visually Impaired Individuals"**

Author: Lisa Adams, Robert Harris

Year: 2020

Abstract: Addressing accessibility challenges, this study presents an augmented reality-based solution for visually impaired individuals navigating college campuses. By overlaying auditory and tactile cues through augmented reality interfaces, the research provides an inclusive and empowering navigation experience. The study's innovative approach not only enhances accessibility but also promotes independence and confidence among visually impaired students and staff.

**2.2 REVIEW ON THE EXISTING WORK**

The current state of campus navigation systems predominantly relies on static maps and physical signages, both of which have notable limitations. Static maps, although useful for providing an overall layout, lack the dynamism required to keep up with real-time changes on a campus. Similarly, physical signages can be helpful but are often limited in scope, providing general directions but lacking specificity. This results in inefficiencies, especially for newcomers who may find it challenging to navigate large or complex campuses effectively.

While there are online mapping solutions, many of these primarily focus on outdoor navigation, disregarding the intricate pathways within buildings. This oversight creates difficulties for users trying to locate specific rooms, offices, or facilities within structures. Furthermore, accessibility concerns for visually impaired individuals and those with mobility challenges have not been adequately addressed in most existing systems, leaving a significant portion of the campus population underserved.

The existing work in campus navigation systems, although functional to a certain extent, faces significant challenges in terms of real-time updates, indoor mapping, and accessibility features. There is a pressing need for a more robust and versatile solution that not only provides accurate and real-time navigation both indoors and outdoors but also caters to the diverse accessibility requirements of all campus users. Addressing these limitations is crucial for enhancing the overall campus experience and ensuring inclusivity for everyone within the academic community.

**2. 3 PROPOSED WORK**

To enhance a basic Campus Map Navigation System, several key improvements and additions can be proposed. First and foremost, incorporating a dynamic database backend is essential. This database would store detailed information about each block, including the location of classrooms, labs, restrooms, and other facilities. Integrating this database would enable real-time updates and ensure accuracy in navigation instructions. Implementing an indoor mapping feature would greatly enhance the system's usability. This could involve floor plans for buildings, guiding users not just to the correct block but also to the specific room or office they are looking for. Making navigation within buildings more intuitive and precise. Accessibility features are crucial for ensuring inclusivity. Integrating voice-guided navigation for the visually impaired and optimizing the interface for screen readers are essential steps. Furthermore, incorporating elevator-friendly routes for users with mobility challenges ensures the system caters to diverse user needs.

Error handling and feedback mechanisms should be refined. Providing clear error messages when users input invalid or out-of-range values helps guide them effectively. Additionally, adding a feedback system would enable users to report issues and suggest improvements, fostering continuous enhancement of the system. Security measures should be implemented to protect user data and system integrity. Utilizing secure authentication methods for administrative access and encrypting sensitive user information are paramount. For scalability, the system can be expanded to include features like user profiles, personalized navigation preferences, and integration with mobile applications. This expansion would not only enhance the user experience but also allow for a more customized and efficient navigation process. By incorporating these proposed enhancements, the Campus Map Navigation System could evolve into a robust, user-friendly, and inclusive solution, providing a seamless experience for all users navigating the college campus.

**2.4 TOOLS USED**

The following tools and technologies are utilized in the development process:

**1. HTML (Hypertext Markup Language):**

HTML is used for creating the structure and layout of the web pages. It defines the elements and their hierarchy, allowing for the creation of user interfaces.

**2. CSS (Cascading Style Sheets):**

CSS is employed to style the HTML elements, providing visual enhancements such as colors, fonts, layouts, and responsiveness. It ensures a visually appealing and consistent user interface.

**3. JavaScript:**

JavaScript is utilized for client-side scripting. In the given code, JavaScript is responsible for handling user input, processing values, and redirecting users to specific pages based on their inputs. It facilitates interactivity and dynamic behavior on the web page.

**4. Image Files (bg.png, logo.png, etc.):**

Image files are used for backgrounds, logos, or other visual elements on the web pages. They enhance the aesthetic appeal and visual content of the user interface.

**5. Text Editor or Integrated Development Environment (IDE):**

A text editor or IDE is used to write, edit, and manage the code files. Popular choices include Visual Studio Code, Sublime Text, or any other preferred text editor.

**6. Web Browser:**

Web browsers like Google Chrome, Mozilla Firefox, or Safari are used to preview and test the web pages. Developers use browsers to ensure that the web pages are displayed correctly and function as intended.

**7. Version Control System (Optional):**

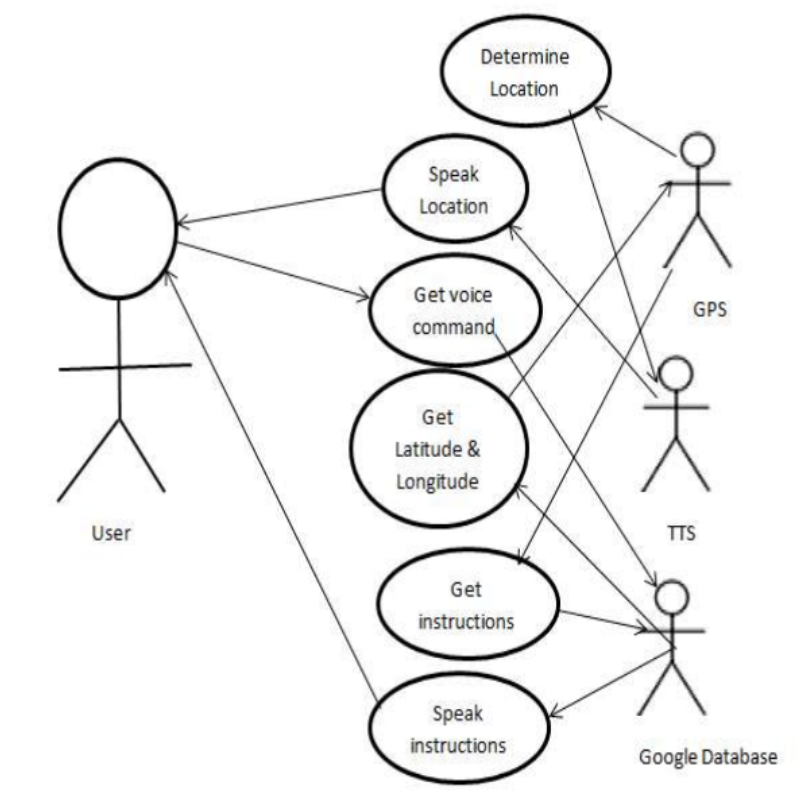
Version control systems like Git can be used to track changes in the code, collaborate with other developers, and maintain a version history of the project. This ensures efficient collaboration and code management, especially in team-based projects.

These tools collectively enable the development, styling, and interactive functionalities of the Campus Map Navigation System

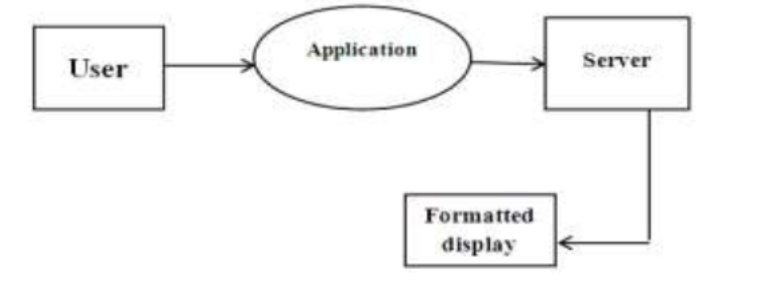
**CHAPTER III**

**DESIGN**

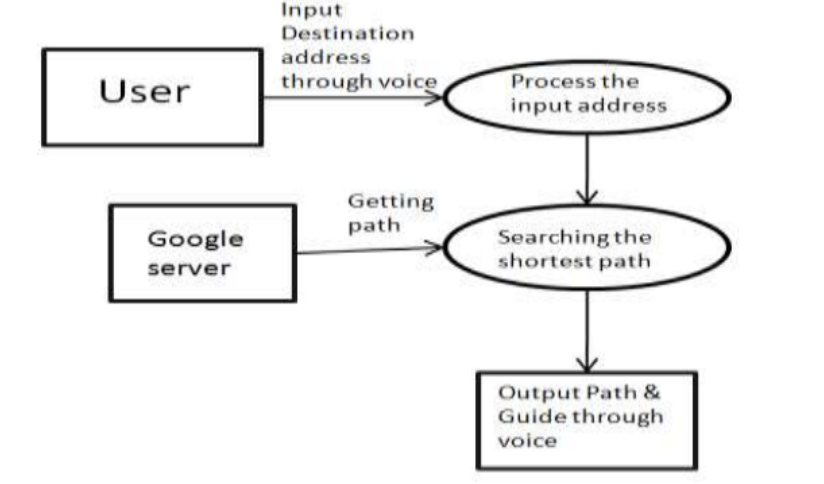
**Use-case Diagram:**

****

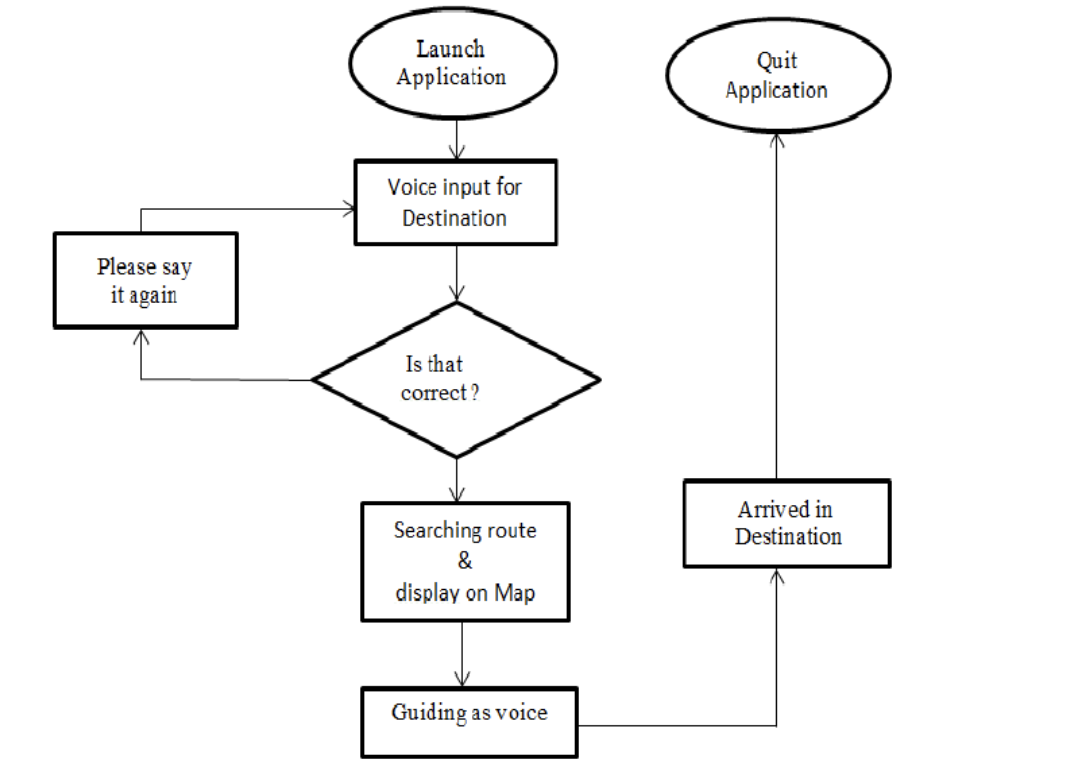
**Data Flow Diagram (DFD) :**

The 0-level DFD of the system **:**

The 1-level DFD of the system :

****

**State Diagram**

****

**CHAPTER IV**

**MODULES**

**4.1 MODULES**

1. Input Module: Accepts user input for source and destination block numbers.

2. Validation Module: Validates the user inputs to ensure they meet the required criteria.

3. Navigation Module: Determines the optimal route based on user inputs and redirects the user to the corresponding page.

4. User Interface Module: Handles the layout, design, and visual elements of the user interface.

5. Background Module: Manages the background image displayed on the webpage.

6. Logo Module: Manages the display of the college logo on the webpage.

7. Top-Right Links Module: Manages the navigation links in the top-right corner of the webpage.

8. Error Handling Module: Manages the redirection to specific error pages based on user inputs or other errors.

**4.2 MODULES DESCRIPTION**

**(i) Module Name: Input Module**

Description: The Input Module is responsible for capturing user input for the source and destination block numbers. It includes the HTML input fields (input1 and input2) where users can enter their starting point (FROM) and destination (TO) block numbers, ranging from 1 to 5. This module initializes the input fields, allowing users to specify their desired navigation route within the acceptable range.

**(ii) Module Name: Navigation Module**

Description: The Navigation Module contains the checkValues() JavaScript function. This function processes the user inputs obtained from the Input Module. It evaluates the entered block numbers and determines the appropriate URL based on predefined conditions. Depending on the source and destination inputs, this module constructs the target page's URL and redirects the user to the corresponding HTML page (e.g., ab.html, ac.html, etc.). It handles various navigation scenarios, guiding users to the correct destination based on their input values.

**(iii) Module Name: Validation Module**

Description: The Validation Module ensures that the user inputs entered in the Input Module are valid and within the acceptable range of 1 to 5. It validates the input values to prevent errors and guide users when they enter incorrect or empty values. This module provides error handling functionality to maintain data integrity and enhance the user experience.

**(iv) Module Name: User Interface Module**

Description: The User Interface Module manages the visual presentation of the application. It includes the HTML and CSS components responsible for creating the layout, input fields, buttons, background image, logo, and navigation links. This module ensures a visually appealing, responsive, and user-friendly interface, enhancing the overall user experience during navigation.

**(v) Module Name: Background Module**

Description: The Background Module is responsible for displaying the background image (bg.png) on the webpage. It ensures the image is appropriately sized, positioned, and repeated (or not repeated) to create an aesthetically pleasing backdrop for the user interface.

**(vi) Module Name: Logo Module**

Description: This module handles the presentation of the college logo (logo.png) on the user interface. It ensures the logo is displayed correctly, with proper dimensions and positioning. The logo provides a visual representation of the institution and enhances the overall branding of the application.

**(vii) Module Name: Top-Right Links Module**

Description: The Top-Right Links Module handles the display and functionality of the navigation links in the top-right corner of the webpage. It includes links to Home (abt.html), Refer (ref.html), and Contact (abt.html) pages. Users can quickly access these pages for additional information or navigation within the application.

**(viii) Module Name: Error Handling Module**

Description: The Error Handling Module handles various error scenarios that might occur during user interactions. It manages the redirection to specific error pages based on invalid inputs or other errors. This module ensures that users receive clear and informative error messages, guiding them to appropriate error pages such as same.html for identical source and destination inputs or exist.html for invalid combinations. It provides a seamless error-handling experience for users.

**CHAPTER V**

**TESTING USING SELENIUM**

**CHAPTER VI**

**CONCLUSION**

In conclusion, the Campus Map Navigation System offers a promising solution to the challenges faced by individuals navigating the vast and intricate layout of Krishna Arts and Science College. By allowing users to input their source and destination block numbers, the system aims to streamline campus navigation. The code demonstrates a functional prototype with modules for user input, validation, navigation logic, user interface, and error handling. However, for a comprehensive and efficient system, future enhancements could focus on real-time data integration, indoor mapping, and accessibility features. Emphasizing user experience, accuracy, and inclusivity is essential in creating a robust campus navigation solution. Overall, this project serves as a foundational step, highlighting the potential for further development and innovation in campus navigation technologies, ultimately enhancing the overall experience for students, staff, and visitors within the college campus.

**APPENDIX**

1. **SAMPLE CODE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>MAP NAVIGATION</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #fafafa;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-image: url('bg.png'); /\* Replace 'your-background-image.jpg' with the actual image file path \*/

background-size: cover;

background-repeat: no-repeat;

background-attachment: fixed;

background-position: center;

}

.container {

background-color: #fff;

border: 1px solid #ccc;

padding: 20px;

width: 300px;

text-align: center;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.logo {

margin-bottom: 20px;

}

.logo img {

border-radius: 50%;

width: 150px;

height: 150px;

object-fit: cover;

}

.input-container {

margin-bottom: 20px;

}

.input-container input {

width: 50%;

padding: 6px;

border: 1px solid #ccc;

border-radius: 3px;

outline: none;

}

.login-button {

background-color: #0095f6;

color: #fff;

border: none;

padding: 10px;

width: 100%;

border-radius: 3px;

cursor: pointer;

font-weight: bold;

}

.login-button:hover {

background-color: #007ee7;

}

.or-separator {

margin: 10px 0;

font-weight: bold;

}

.signup-link {

color: #0095f6;

text-decoration: none;

}

.top-right-links {

position: absolute;

top: 10px;

right: 100px;

display: flex;

gap: 30px;

}

.top-right-links a {

color: #333;

text-decoration: none;

font-weight: bold;

}

</style>

</head>

<body>

<div class="top-right-links">

<a href="abt.html"><button> Home</a></button>

<a href="ref.html"><button>REFER</a></button>

<a href="abt.html"><button>Contact</a></button>

</div>

<div class="container">

<div class="logo">

<a href="abt.html">

<img src="logo.png" alt="chatbot"></a>

</div>

<div class="input-container">

<label for="input1">FROM:</label>

<input placeholder = "Enter 1 to 5" type="text" id="input1"><br><br>

<label for="input2"> TO&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;: </label>

<input placeholder = "Enter 1 to 5" type="text" id="input2"><br><br>

<div class="search-button">

<button class="login-button" onclick="checkValues()">GET NAVIGATE</button>

</div>

<script>

function checkValues() {

var value1 = document.getElementById("input1").value;

var value2 = document.getElementById("input2").value;

var output = "";

if (value1 === "1" && value2 === "2"){

window.location.href = "ab.html?output=" + encodeURIComponent(output);

} else if (value1 === "1" && value2 === "3"){

window.location.href = "ac.html?output=" + encodeURIComponent(output);

} else if (value1 === "1" && value2 === "4"){

window.location.href = "ad.html?output=" + encodeURIComponent(output);

} else if (value1 === "1" && value2 === "5"){

window.location.href = "ae.html?output=" + encodeURIComponent(output);

} else if (value1 === "2" && value2 === "1"){

window.location.href = "ba.html?output=" + encodeURIComponent(output);

} else if (value1 === "2" && value2 === "3"){

window.location.href = "bc.html?output=" + encodeURIComponent(output);

} else if (value1 === "2" && value2 === "4"){

window.location.href = "bd.html?output=" + encodeURIComponent(output);

} else if (value1 === "2" && value2 === "5"){

window.location.href = "be.html?output=" + encodeURIComponent(output);

} else if (value1 === "3" && value2 === "1"){

window.location.href = "ca.html?output=" + encodeURIComponent(output);

} else if (value1 === "3" && value2 === "2"){

window.location.href = "cb.html?output=" + encodeURIComponent(output);

} else if (value1 === "3" && value2 === "4"){

window.location.href = "cd.html?output=" + encodeURIComponent(output);

} else if (value1 === "3" && value2 === "5"){

window.location.href = "ce.html?output=" + encodeURIComponent(output);

} else if (value1 === "4" && value2 === "1"){

window.location.href = "da.html?output=" + encodeURIComponent(output);

} else if (value1 === "4" && value2 === "2"){

window.location.href = "db.html?output=" + encodeURIComponent(output);

} else if (value1 === "4" && value2 === "3"){

window.location.href = "dc.html?output=" + encodeURIComponent(output);

} else if (value1 === "4" && value2 === "5"){

window.location.href = "de.html?output=" + encodeURIComponent(output);

} else if (value1 === "5" && value2 === "1"){

window.location.href = "ea.html?output=" + encodeURIComponent(output);

} else if (value1 === "5" && value2 === "2"){

window.location.href = "eb.html?output=" + encodeURIComponent(output);

} else if (value1 === "5" && value2 === "3"){

window.location.href = "ec.html?output=" + encodeURIComponent(output);

} else if (value1 === "5" && value2 === "4"){

window.location.href = "ed.html?output=" + encodeURIComponent(output);

} else if (value1 === value2 ){

window.location.href = "same.html?output=" + encodeURIComponent(output);

}

else{

window.location.href = "exist.html?output=" + encodeURIComponent(output); }

// Your JavaScript code here

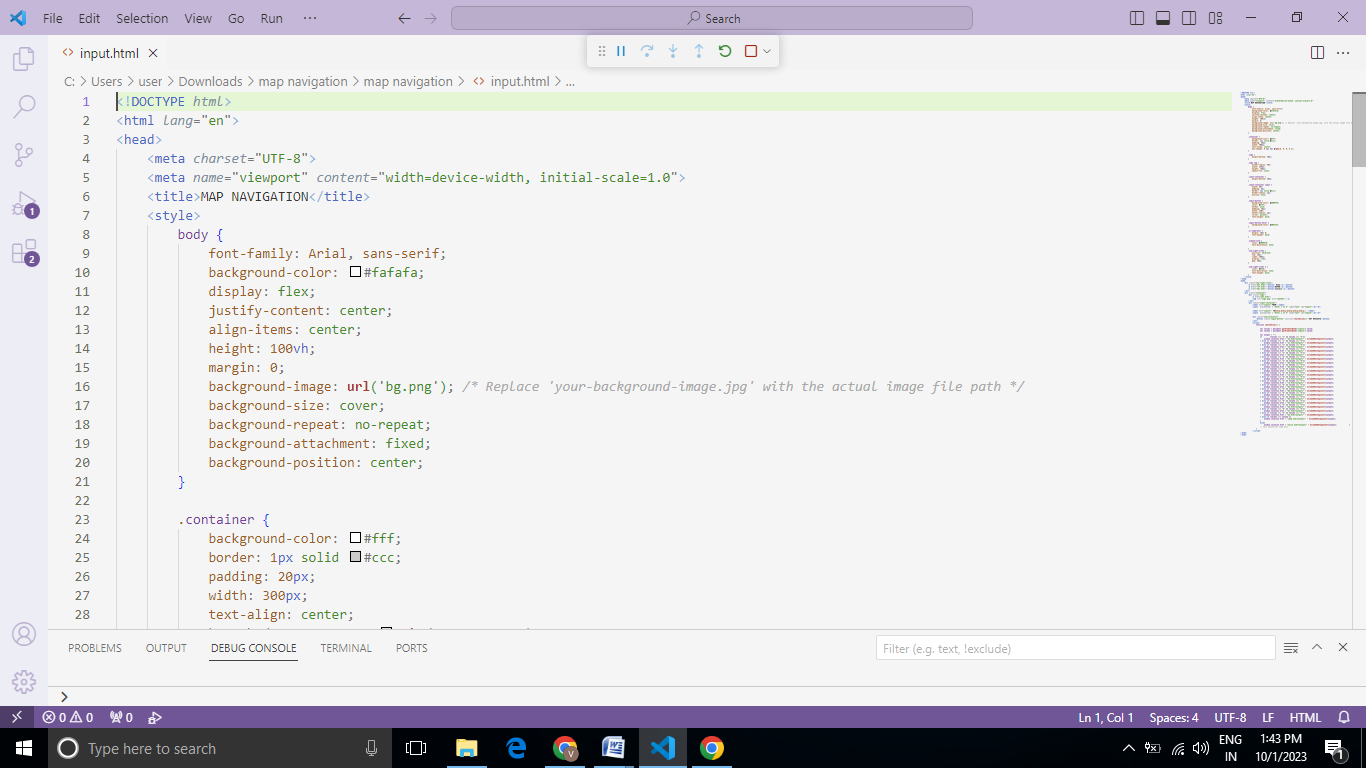
}

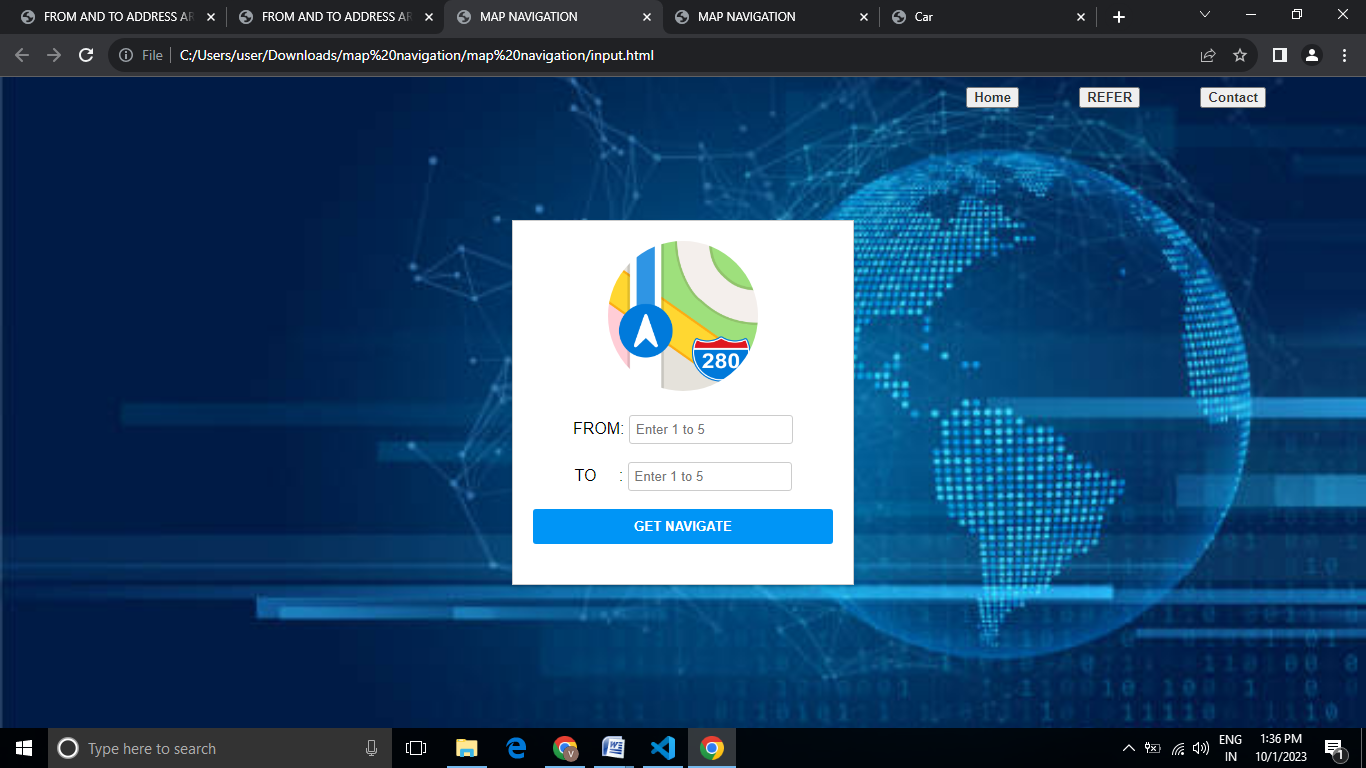
</script>

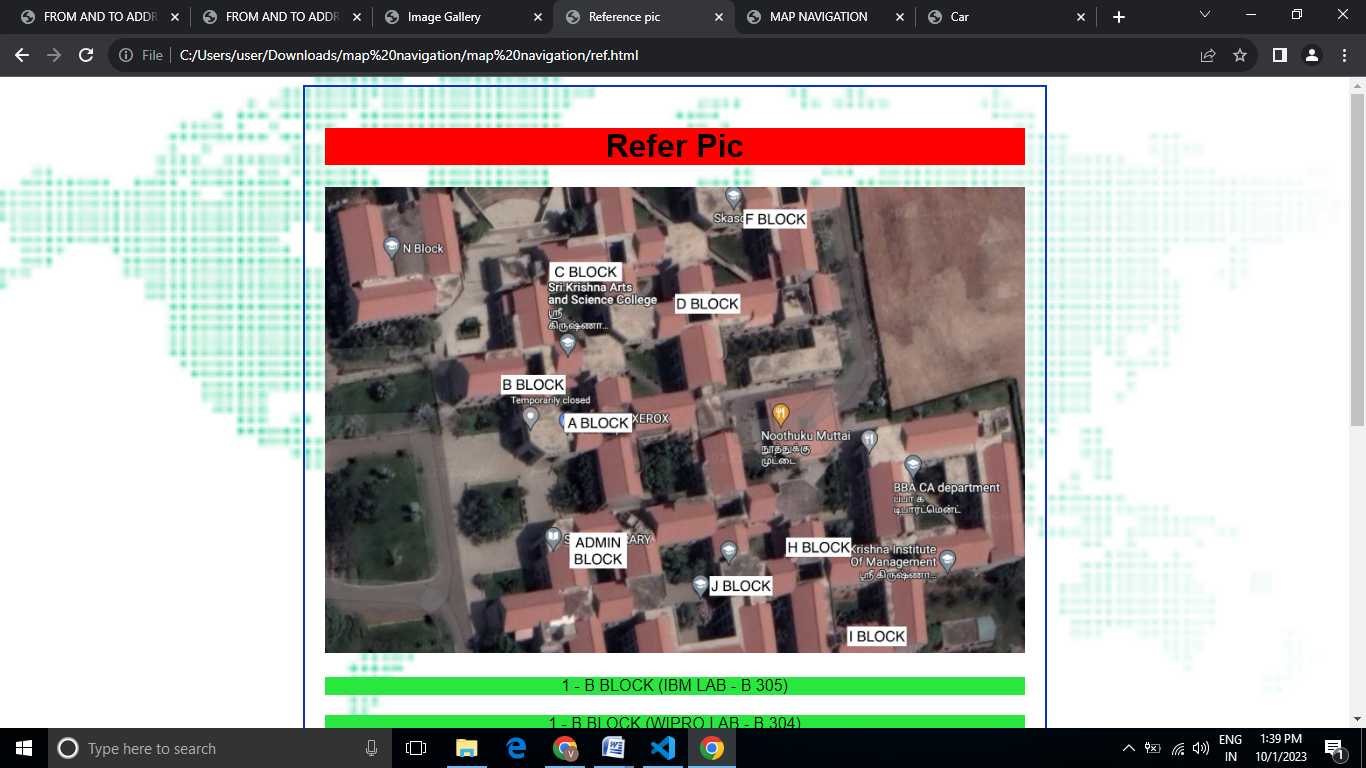
</body>

</html>

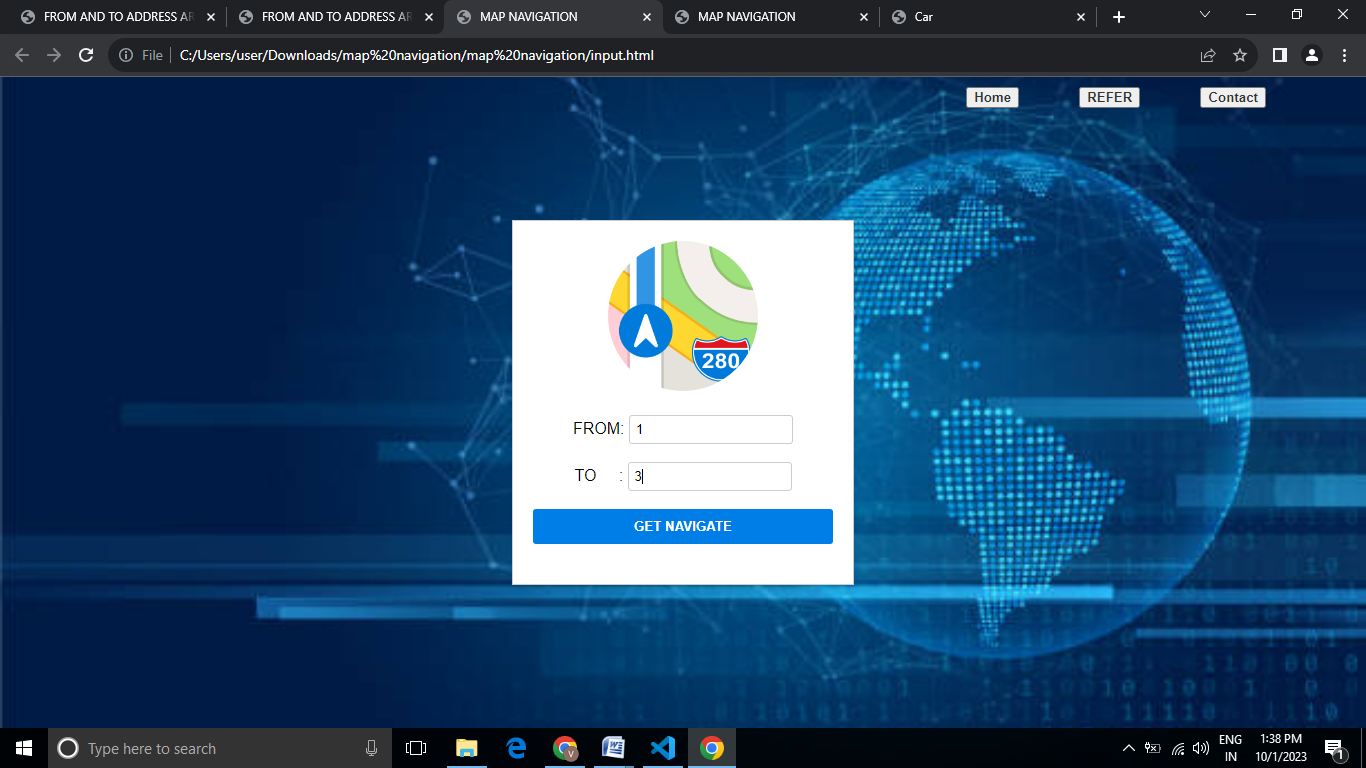
1. **SCREENSHOTS**

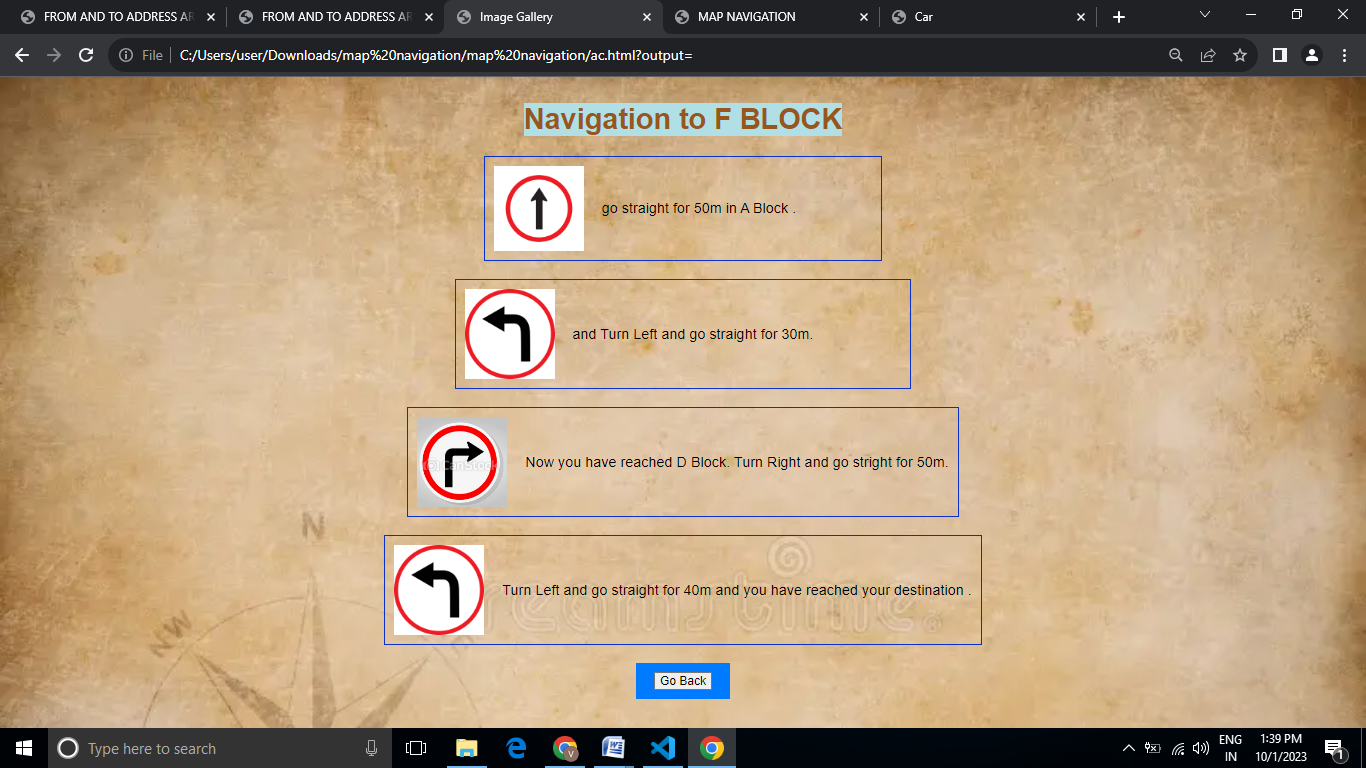
****

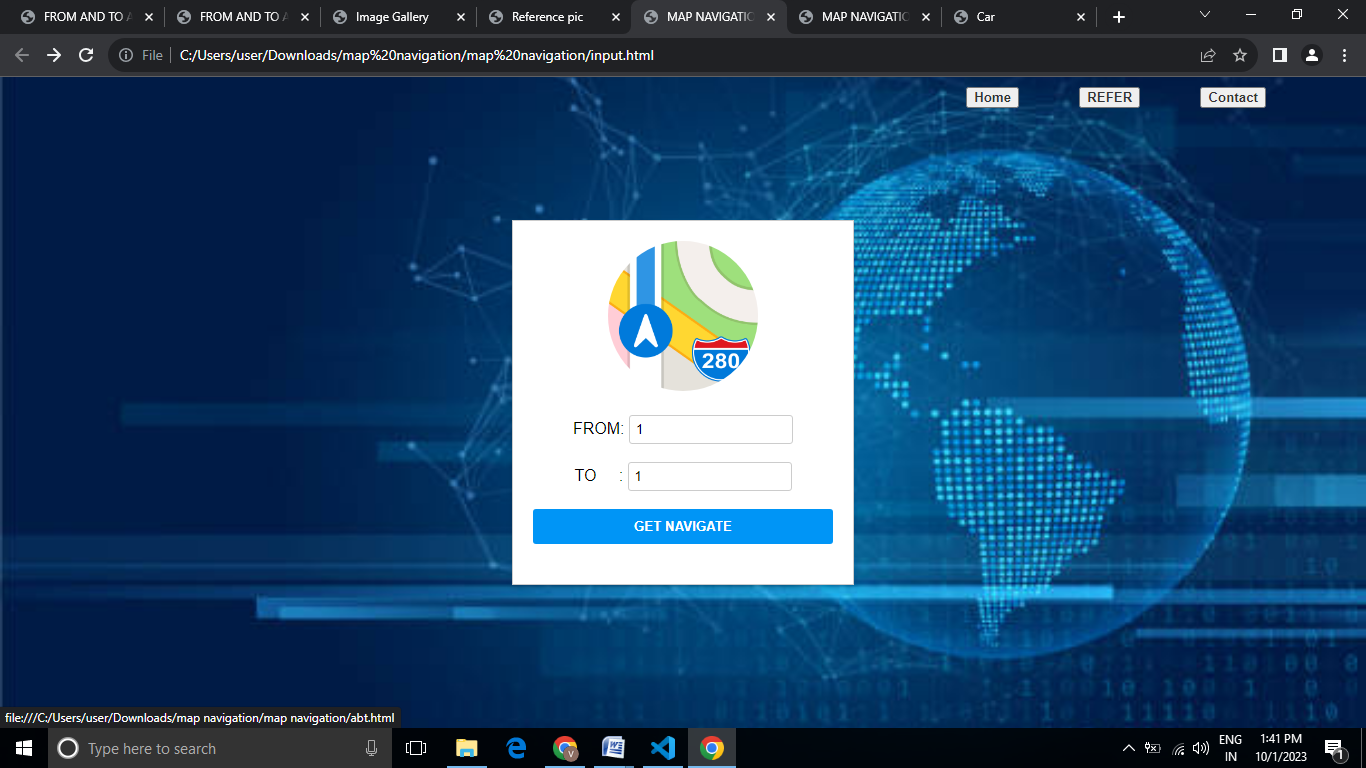
****

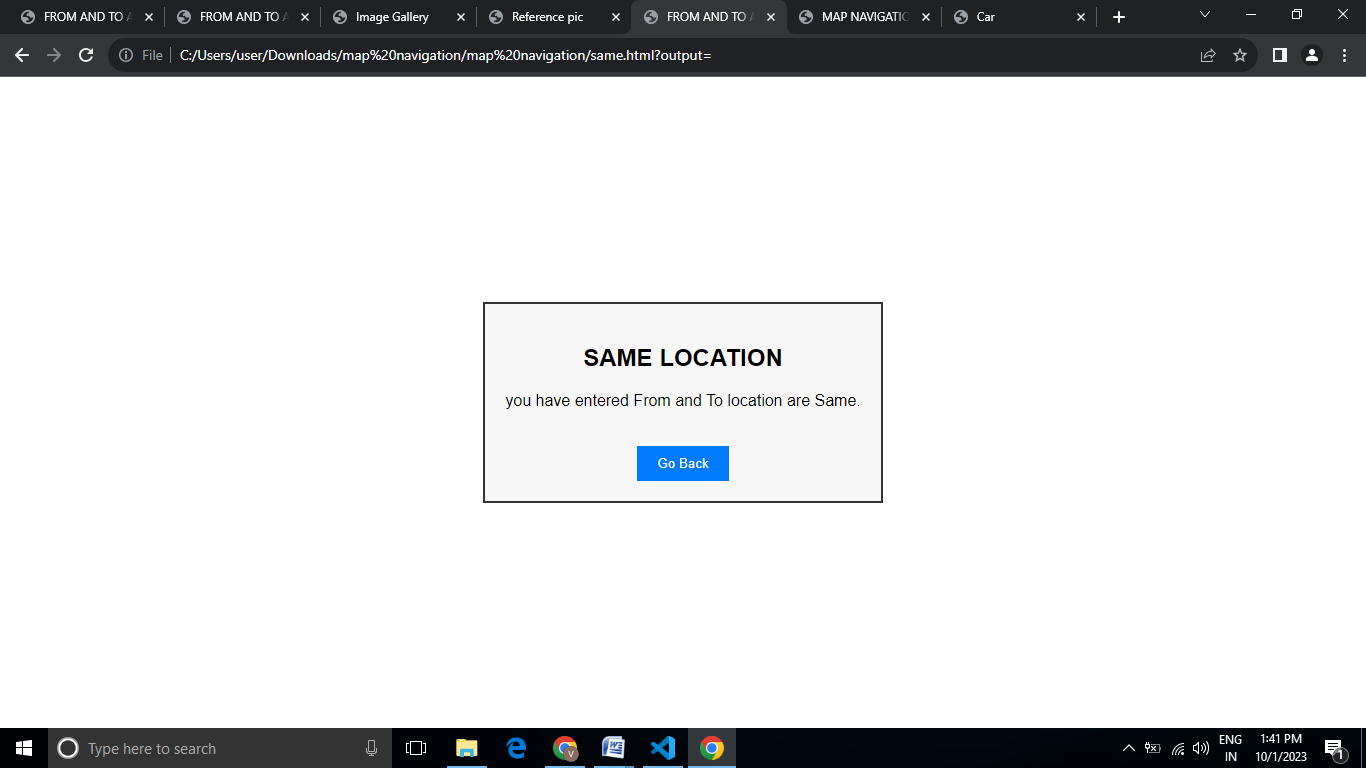
****

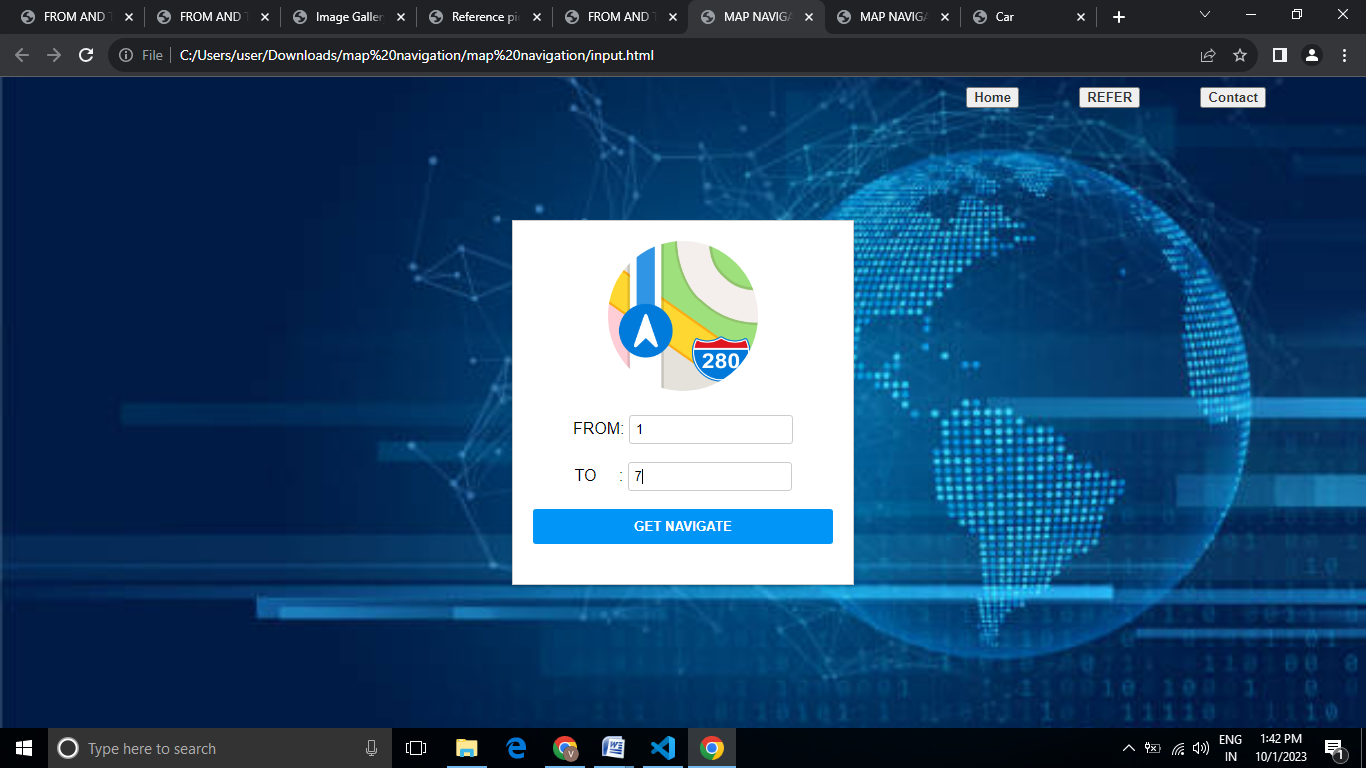
****

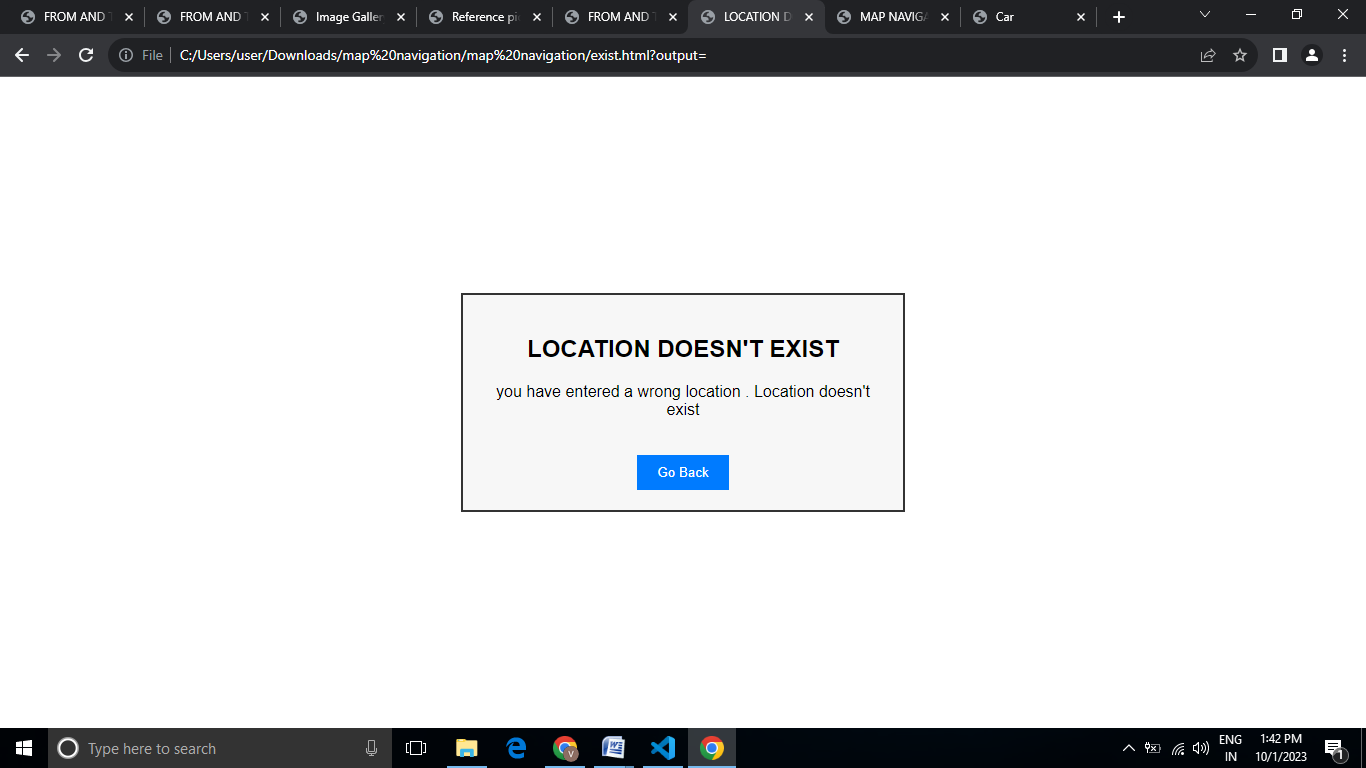
****

****

****

****

****

****