PROJECT REPORT: MAP NAVIGATOR

Project Detail:

- **Project Title: Map Navigator**
- **Project Developer: Tammali Amulya**
- **Project Duration: 6 December 2023 to 6 January 2024**

Abstract:

This project report presents the development of an application titled "Map Navigator" aimed at providing users with a user-friendly interface to find the shortest path between two locations on a map while considering potential hazards along the way. Led by Tammali Amulya, the project duration spans from 6th December,2023 to 22 February 2024

TABLE OF CONTENTS

<u>Main Objective:</u>

The main objective of the project is to facilitate safe and efficient navigation by providing users with the shortest path between two locations on a map while ensuring awareness of potential hazards along the way. By integrating hazard detection and alert mechanisms into the shortest path finding process, users can make informed decisions during their journey.

<u>Functionality Overview:</u>

1.Shortest Path Calculation: • Users input starting and

destination locations.

- Upon clicking the "Find Shortest Path" button, the application calculates and displays the shortest distance and path between the provided locations.
- Dijkstra's algorithm is employed for finding the shortest path efficiently.

2.Potential Hazard Alert: • Hazard information is stored

in a JavaScript object (`hazards`).

- During traversal using breadth-first search (BFS), hazards along the shortest path are detected and highlighted.
- Users are alerted about hazards encountered along the way, ensuring safe navigation.

Components and Implementation:

- **Dependencies:** Utilizes Leaflet.js for map functionality.

- **Input Fields:** Allows users to input starting and destination locations.

- **Output Display:** Displays the shortest distance and path between locations, along with hazard alerts if applicable.

- **Map Display:** Embedded interactive map using Leaflet.js, with markers for cities and polylines for paths.

- **Graph Representation:** Cities and distances represented using a JavaScript object (`exampleGraph`).

- **Styling:** CSS used for styling map containers and HTML elements.

- **Function Definitions:** Defined functions for Dijkstra's algorithm, BFS traversal, displaying path information, highlighting edges, etc.

Project Interface Demonstration:

Project user interface will be as follows.
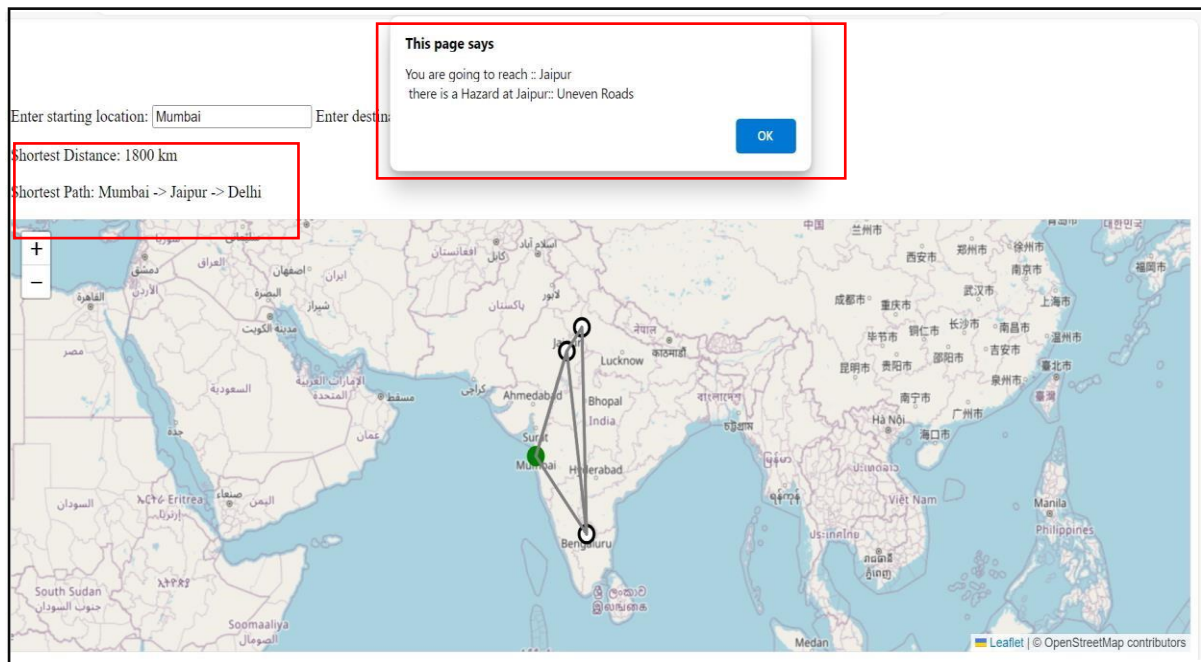
Potential Enhancements:

1.   Error Handling: Implement robust error handling for invalid inputs to enhance user experience.

2.   Performance Optimization: Optimize algorithms for better performance, particularly for large graphs, ensuring faster path calculation.

3.   Additional Features: Consider adding features like alternative route suggestions, user authentication for personalized experiences, and real-time hazard updates for dynamic navigation.

Conclusion:

The project successfully achieves its main objective of providing users with the shortest path between two locations on a map while prioritizing safety through hazard detection and alerts. With potential enhancements, the application can

further enhance user experience and functionality, catering to diverse navigation needs.