Muhammad Abdullah Baig (366718)

DSA (Project) 😊

Project Name:

Shortest Route Finder between main cities in Pakistan.

Description:

This project is a web application that helps users find the shortest route between two locations. It utilizes the Leaflet library for displaying maps and calculating routes. The user can select a start location and an end location from dropdown menus and click the "Find Shortest Route" button to display the shortest route on the map.

Installation:

To run the project locally, follow these steps:

1. Open the project directory.

2. Open the `index.html` file in a web browser.

Usage:

1. Open the web page in a browser after completing the installation steps.

2. Select a start location and an end location from the dropdown menus.

3. Click the "Find Shortest Route" button.

4. The following steps outline how the code works:

- The HTML markup provides a container div for the map and a form with dropdown menus for selecting the start and end locations. It also includes the necessary CSS and JavaScript files.

- The JavaScript code initializes the Leaflet map, sets the map's center and zoom level, and adds a tile layer from OpenStreetMap.

- It creates markers for each location in the connections array and adds them to the map.

- It creates polylines between connected locations to represent the connections on the map.

- The code populates the start and end location select options dynamically based on the markers.

- The `createGraph` function creates a graph representation of the connections using an adjacency list.

- The `calculateDistance` function calculates the distance between two locations (currently set to a fixed value of 1 for simplicity).

- The `getLatLng` function returns the latitude and longitude coordinates for a given location.

- The `findShortestRoute` function is triggered when the "Find Shortest Route" button is clicked. It retrieves the selected start and end locations and applies Dijkstra's algorithm to find the shortest route.

- The `dijkstra` function implements Dijkstra's algorithm to find the shortest path between the source and destination. It uses the graph representation created earlier.

- If a shortest path is found, the code removes previous polylines, creates a new polyline for the shortest route, and fits the map to display the entire route.

Dependencies:

This project depends on the following libraries:

- Leaflet: A JavaScript library for interactive maps. The Leaflet CSS and JavaScript files are included in the project under the `leaflet/dist` directory.

Acknowledgments:

This project was developed using the Leaflet library, which is an open-source project. The map data used in this project is provided by OpenStreetMap contributors.

Leaflet: https://leafletjs.com/

OpenStreetMap: https://www.openstreetmap.org/