

MAP NAVIGATION

CO1, CO2, CO3 S3

PROBLEM STATEMENT: Navigating from one location to another can be complex due to traffic conditions, route availability, and geographic constraints. Traditional maps offer static information and lack real-time adaptability. This project aims to develop a dynamic map navigation system that uses geolocation data and algorithms to suggest optimal paths. The challenge lies in integrating real-time data and user inputs to generate accurate and efficient routes. Factors such as distance, travel time, and road conditions must be considered. The system should provide interactive and user-friendly navigation support for various use cases including walking, driving, and public transport.

AIM: The aim of this project is to implement a map navigation system using programming tools and geolocation data. It seeks to help users find optimal routes between locations, enhancing travel efficiency and user experience.

OBJECTIVE:

- 1.To collect and process geolocation data for route mapping.
- 2.To identify key parameters affecting route selection such as distance and traffic.
- 3.To develop a navigation algorithm that suggests optimal paths.
- 4.To evaluate the system's accuracy and responsiveness.
- 5.To provide a reliable tool for users to navigate efficiently

DESCRIPTION : This project focuses on building a map navigation system using programming tools and geolocation APIs. It involves processing location data to determine routes between two points. The system will use algorithms like Dijkstra's or A* for pathfinding and may integrate APIs such as Google Maps for real-time data.

The goal is to assist users in finding the best possible route based on current conditions. The system will be interactive, allowing users to input start and end locations and receive visual and textual directions. It enhances travel planning and reduces uncertainty in navigation.

ALGORITHM:

1. Collect geolocation data using APIs or predefined coordinates.
2. Preprocess data to identify nodes (locations) and edges (paths).
3. Implement a pathfinding algorithm (e.g., Dijkstra's or A*) to calculate shortest routes.
4. Integrate real-time traffic or distance data if available.
5. Display the route visually on a map interface.
6. Provide step-by-step navigation instructions.
7. Evaluate accuracy by comparing predicted vs. actual travel paths

PROGRAM :

```
import folium
```

```
import geopy.distance
```

```
# Define coordinates for two locations start =
```

```
(13.0827, 80.2707) # Chennai
```

```
end = (13.6288, 79.4192) # Tirupati
```

```
# Calculate distance distance =
```

```
geopy.distance.geodesic(start, end).km print(f'Distance
```

```
between locations: {distance:.2f} km")
```

```
# Create map
```

```
map_nav = folium.Map(location=start, zoom_start=7)
```

```
# Add markers folium.Marker(start, tooltip='Start:
```

```
Chennai').add_to(map_nav) folium.Marker(end, tooltip='End:
```

```
Tirupati').add_to(map_nav)
```

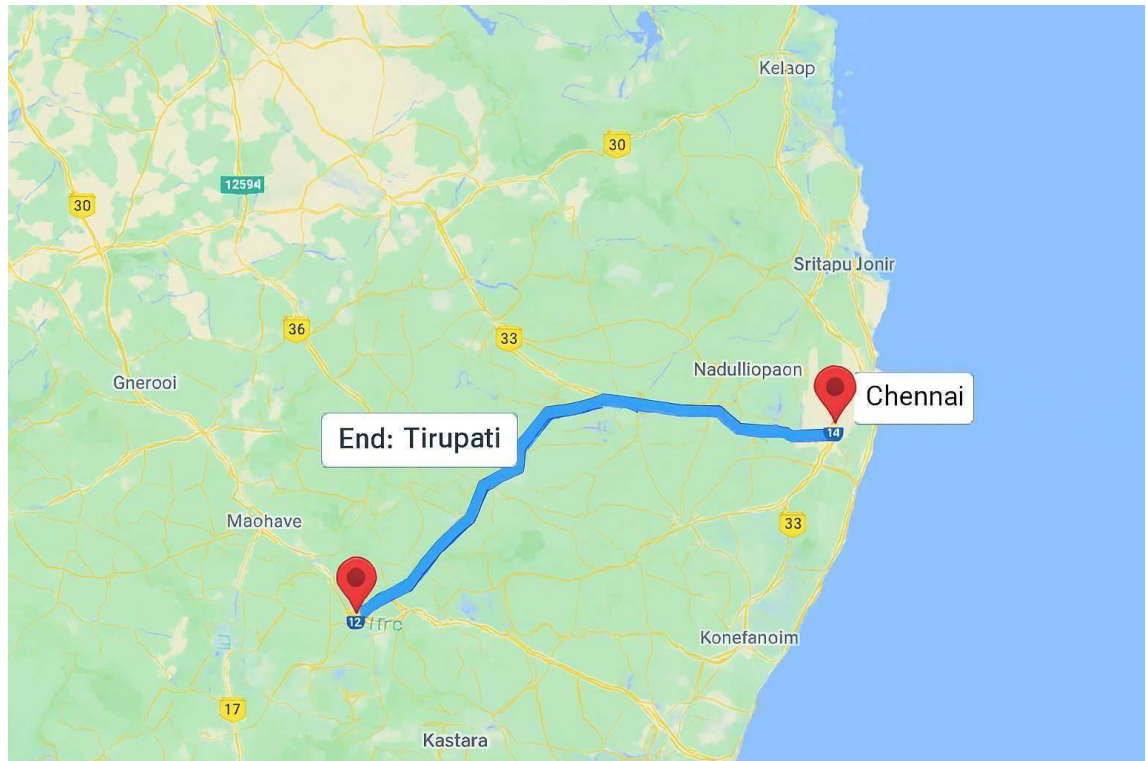
```
# Draw route line
```

```
folium.PolyLine([start, end], color='blue', weight=2.5,  
opacity=1).add_to(map_nav)
```

```
# Save map
```

```
map_nav.save("map_navigation.html")
```

OUTPUT:



Displays a map with start and end markers.

Shows a line connecting the two locations.

CONCLUSION:

The implementation of map navigation demonstrates how geolocation data and algorithms can be used to build efficient routing systems. It provides users with visual and textual guidance for travel planning. While basic in structure, the system can be enhanced with real-time traffic data and multi-modal transport options. Future improvements may include voice navigation, dynamic rerouting, and integration with mobile platforms.