

# **SHORTEST PATH NAVIGATION SYSTEM**

Submitted in partial fulfilment of the requirements for the award of degree of

**B. Tech (Computer Science and Engineering)**

Submitted to

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA , PUNJAB**



**Submitted By**

**Name – Keshav Singh , Rohit Kumar , Devanshu Tyagi**

**Reg. No. – 12315903,12309807,12307604**

**Faculty Name – Mr. Sudeep Chowhan**

# **SHORTEST PATH NAVIGATION**

## **(DIJKSTRA ALGORITHM)**

**Project Title:** Navigation System Using Graphs

**Subject:** Data Structures & Algorithms

**Submitted By:** Keshav Singh , Rohit Kumar , Devanshu Tyagi

**Roll No:** 12315903 , 12309807 , 12307604

**Guide:** Mr. Sudeep Chowhan

### **Index:**

1. Problem Statement
2. Problem Solving Technique
3. Requirements
4. Code
5. Real-Time Example

## **Problem Statement:**

In real-life applications, cities are connected through roads of different lengths. Finding the shortest route between two locations is an important problem in transportation, delivery services, and travel planning.

The aim of this project is to design and develop a Navigation System that:

- Stores cities and roads with distances
- Allows users to add new road connections
- Finds the shortest path between two cities
- Displays the minimum distance and route
- Saves road data permanently using file handling

This project uses Graph Data Structure and Dijkstra's Algorithm to efficiently compute the shortest path between cities.

## **Problem Solving Technique:**

Data Structures Used:

- Graph (Adjacency List)
- Map for city to index mapping

- Priority Queue (Min Heap)
- Arrays for distance and parent

Algorithm Used:

Dijkstra's Algorithm is used to find the shortest path in a weighted graph with non-negative weights.

Working:

1. Each city is treated as a node
2. Each road is treated as an edge with distance
3. The user selects source and destination
4. Dijkstra's Algorithm finds the shortest distance
5. The path is reconstructed using parent array

Time Complexity:  $O(E \log V)$

Space Complexity:  $O(V + E)$

## **Requirements:**

Hardware Requirements:

- Computer/Laptop
- Minimum 4 GB RAM

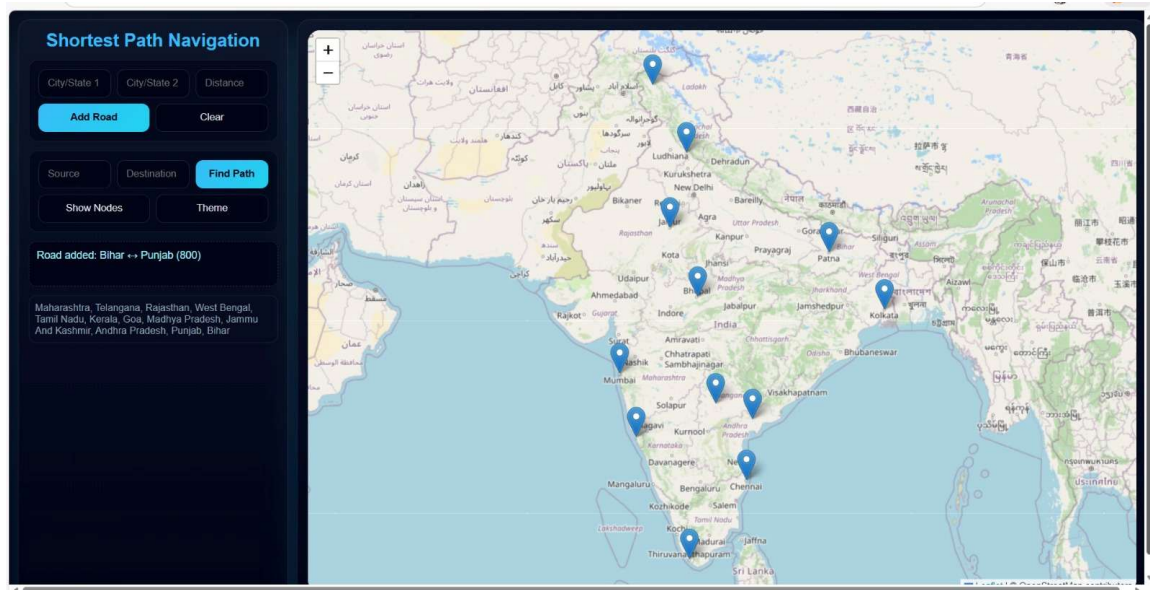
## Software Requirements:

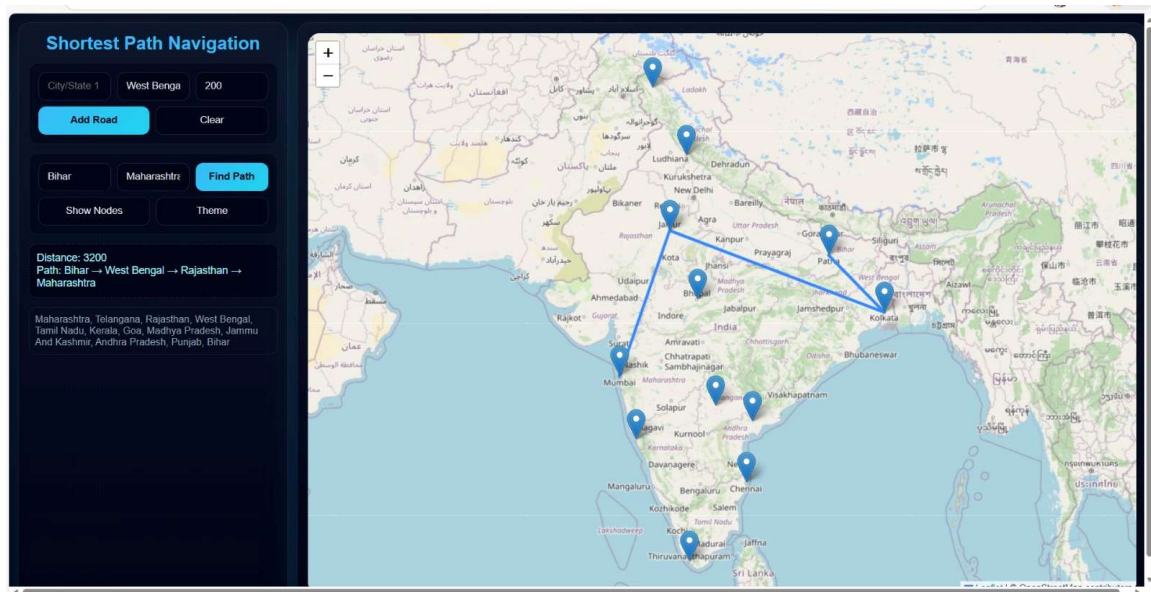
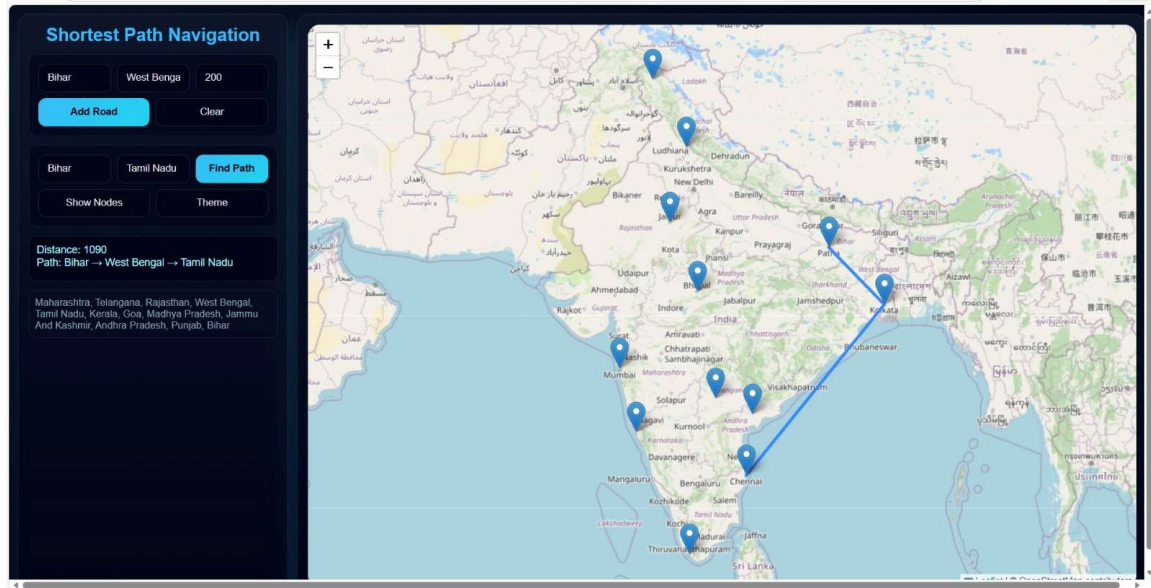
- C++ Compiler (GCC/Clang)
- IDE (VS Code/CodeBlocks/Dev C++)
- Windows/Linux OS

## Concepts Required:

- Graph Data Structure
- Dijkstra's Algorithm
- File Handling
- STL (vector, map, priority\_queue)

## User Interface:





The project is implemented in C++ using:

- Graph representation
- Dijkstra's Algorithm
- File handling for storing roads
- Path reconstruction using parent array

## **Real-Time Example**

This navigation system works similar to real-world navigation apps like Google Maps, Uber, and Swiggy.

Example:

A delivery person wants to travel from City A to City B to City C. Each road has different distances. The system calculates the shortest possible route and displays:

- Minimum distance
- Best path

This helps in saving time, reducing fuel cost, and improving delivery efficiency.

## **Challenges Faced:**

1. Modeling Real-World Locations as a Graph
2. Handling Large Number of Nodes & Edges
3. User Input Errors (Invalid Cities / Case Sensitivity)
4. Showing the Actual Path, Not Just Distance
5. Integrating Algorithm with Map UI
6. Making the Map Look Realistic & Attractive
7. Algorithm Limitations

## **Future Aspects:**

1. Real-Time Traffic Integration
2. Smarter Pathfinding with Heuristics
3. Multi-Criteria Routing (Not Just Distance)
4. Live Road Conditions & Closures