

# DSA Project Presentation

Group Members:

Pranshu Desai (B24CS1057)

Shardul Diwate (B24CS1028)

Vansh Rathi (B24CM1065)

Katravath Kurumurthy (B24CM1035)



# Campus Navigation System

Goal: Find the minimum travelling distances between two points on campus.

01.

The campus is modeled as a weighted undirected graph

02.

Nodes represent buildings/locations.

03.

Edges represent paths with distance weights.



# DSA Concepts Used:

- Graph Representation : Adjacency List
- Priority Queue (Min-Heap): To select the nearest unvisited node.
- Hash Map: To store distances and parent relationships.
- Dijkstra's Algorithm: To find the shortest path efficiently.
- Time Complexity:  $O(E \log V)$ .



# 01.

The campus is represented as a weighted graph with nodes and edges.

# 02.

Each node corresponds to a campus building or landmark.

# 03.

Edges connect nodes with weights representing distances in meters.

# System Design

# 04.

The user inputs a source and destination location.

# 05.

Dijkstra's algorithm computes the shortest path and minimum distance.

# 06.

The result (path and distance) is displayed textually and visually on the graph.

# Dijkstra's Algorithm

- 
- 01 Start from the source node with distance = 0.
  - 02 Visit the unvisited node with the smallest distance.
  - 03 Update its neighbors' distances.
  - 04 Repeat until the destination is reached.



# Code Implementation:

## Core Functions:

- addEdge(u, v, w) - Creates a bidirectional edge.
- dijkstra(source, destination) - Computes the shortest path.

## STL Used:

- unordered\_map
- vector
- priority\_queue

## Code Snippet:

```
while (!pq.empty()) {
    auto [d, node] = pq.top(); pq.pop();
    for (auto &nbr : adj[node]) {
        if (dist[node] + nbr.second < dist[nbr.first]) {
            dist[nbr.first] = dist[node] + nbr.second;
            parent[nbr.first] = node;
            pq.push({dist[nbr.first], nbr.first});
        }
    }
}
```

# Conclusion and Future Scope

## Conclusion:

- Implemented a working Campus Navigation System using Dijkstra's Algorithm.
- Demonstrated usage of graphs, priority queues, and adjacency lists.
- Achieved efficient shortest route computation.

## Future Scope:

- Add GUI using real IITJ map.
- Integrate GPS for real-time navigation.
- Implement A\* Algorithm for optimization.



# Thank You