**Semester Project Proposal**

**Smart Navigation System for   
COMSATS University Islamabad, Lahore Campus**

**Group Members**

1. Bilal Sarwar
2. Ali Rafay

**Project Overview**

The Smart Navigation System is a full-stack web-based application built to assist students, faculty, and visitors in efficiently navigating the COMSATS University Islamabad, Lahore Campus. The platform allows users to compute and visualize the shortest or simplest walking paths between buildings such as N Block, Faculty Block, Library, Cafeteria, and more.

The backend logic is implemented in C++, focusing on graph data structures and search algorithms, while the frontend is developed using HTML, CSS, and JavaScript and served through a Node.js Express server. This system bridges high-performance C++ algorithms with a modern, interactive web interface.

**Project Objectives**

* Develop an intelligent routing engine using Data Structures and Algorithms (DSA) in C++.
* Build an interactive web-based frontend to visualize and simulate campus navigation.
* Integrate backend algorithms via a custom API for real-time pathfinding between buildings.
* Allow users to choose between different algorithms (e.g., Dijkstra and BFS) for customized routing.

**Core Features**

**1. Interactive Campus Map**

* A visual HTML-based map interface showing campus buildings as nodes.
* Clickable buildings and animated path visualization on selection.

**2. Smart Pathfinding Engine**

* Calculates the best path between selected locations using:
  + Dijkstra's Algorithm - for shortest physical distance.
  + Breadth-First Search (BFS) - for minimal number of steps.

**3. API Integration via Express Server**

* Requests are handled by a Node.js backend that invokes the C++ executable and returns results in JSON format.

**4. Algorithm Selection**

* Users can select the algorithm they want to use before generating the route.

**5. Real-Time Feedback**

* The frontend displays whether the navigation engine is online.
* Clear error and success messages for better user experience.

**6. Responsive and Clean UI**

* Built entirely with modern HTML and CSS for visual clarity.
* Auto-populating dropdowns for location selection.

**Technical Stack**

**Languages and Frameworks**

* C++ (Core algorithm and graph logic)
* Node.js / Express (API server)
* HTML / CSS / JavaScript (Frontend interface)

**Algorithms Implemented**

* Dijkstra’s Algorithm - Uses a min-heap to calculate the shortest weighted path.
* Breadth-First Search (BFS) - Determines the path with the least number of nodes.

**Data Structures Used**

|  |  |
| --- | --- |
| **Purpose** | **Data Structure** |
| Map Representation | Graph using Adjacency Lists |
| Efficient Lookup | Hash Maps |
| Pathfinding | Queues, Min-Heaps, Vectors |
| Path Reconstruction | Arrays, Stacks |
| Location Data | JSON (via C++ string streams) |

**Project Architecture**

User → index.html → Express API → navigation\_engine.exe → C++ Dijkstra/BFS → JSON Output → Web Interface

**Expected Outcome**

By the end of the semester, the group will deliver a fully functional, tested, and user-friendly web-based smart navigation system. The project will showcase:

* Real-time campus routing capabilities
* Strong integration of C++ algorithms with modern web technologies
* Clean and maintainable code
* Effective demonstration of how DSA concepts apply to real-world navigation challenges

**Deployment Note**

All components can be run locally on a desktop using:

* g++ for C++ compilation
* Node.js for backend
* A modern browser to run the frontend

**Fun Fact:**

The university actually implemented some of these paths in the campus after we made this project.