**1. Project Title & Problem Statement**

**Project Title:**

**Implementation of Path Finding Algorithm and Visualization through C++ Graphics Library**

**Problem Statement:**

Pathfinding algorithms are essential in various fields such as robotics, game development, and navigation systems. This project aims to develop a C++ program that implements a **pathfinding algorithm** (such as **A\* or Dijkstra’s algorithm**) and **visualizes** the pathfinding process using a **C++ graphics library**.

The objective is to create an **interactive and educational tool** to demonstrate how these algorithms work in real time, helping users understand their **efficiency and behavior**

**2. Data Structures Selection**

To efficiently implement the project, the following **key data structures** will be used:

* **Graph (Adjacency List/Matrix):** Represents the grid or network of nodes (vertices) and connections (edges).
* **Priority Queue (Min-Heap):** Used in A\* and Dijkstra’s algorithms to efficiently select the next best node to explore.
* **2D Array/Vector:** Represents the grid for visualization.
* **Map (Hash Table):** Stores the parent-child relationships for backtracking the final path.
* **Queue:** Used for **Breadth-First Search (BFS)** implementation in an unweighted graph.

**3. Algorithmic Approach**

The project will implement one or more of the following **pathfinding algorithms**:

* **Dijkstra’s Algorithm:** Finds the shortest path from the source node to all other nodes in a **weighted graph**.
* **A\* Algorithm:** An optimized version of Dijkstra’s that uses a **heuristic function** to prioritize paths, making it **faster** in many cases.
* **Breadth-First Search (BFS):** Used for **unweighted graphs** to find the shortest path.

**Algorithm Execution:**

* The algorithm will run on a **grid**, where each **cell** represents a **node**.
* Movement will be **restricted** to **four or eight** directions, depending on implementation choices.

**4. Input & Output Design**

**Input Design:**

The user will specify:

* **Start and goal points** on the grid.
* **Obstacles** (barriers preventing movement).
* **Algorithm selection** (**Dijkstra, A\*, BFS, etc.**).

**Output Design:**

The program will display:

* **Graphical representation of the grid**.
* **Step-by-step pathfinding process** of the selected algorithm.
* **Final shortest path**, highlighted on the grid.
* **Performance metrics**, such as:
  + Number of **nodes explored**.
  + **Execution time**.

**5. Course Concepts Application**

This project integrates **key concepts** from the **Data Structures and Algorithms** course, including:

* **Graph Theory:** Representation and traversal techniques.
* **Heaps and Priority Queues:** Efficient node selection in **A\*** and **Dijkstra’s algorithm**.
* **Algorithm Optimization:** Comparison of different pathfinding strategies.
* **Complexity Analysis:** Evaluating efficiency in terms of **time and space complexity**.
* **Graphics Programming:** Utilizing **C++ graphics libraries** (**SFML, SDL, or OpenGL**) for visualization.

**6. Submission**

This document serves as the **Project Proposal & Conceptual Design** for the **first deliverable**.  
The project will be implemented using **C++** and a suitable **graphics library** (**SFML, SDL, or OpenGL**) for visualization.