**Project Proposal:**

**Graph Shortest Path Visualization Using Dijkstra's Algorithm**

1. **Project Title:**

Graph Shortest Path Visualization Using Dijkstra's Algorithm

1. **Introduction:**

This project aims to implement and visualize Dijkstra's algorithm for finding the shortest path in a weighted graph. The program takes user input for the number of nodes, edges, and edge weights, then computes the shortest paths from a given source node. Additionally, it generates a Graphviz-compatible DOT file to visualize the graph.

**3. Objectives:**

* Implement Dijkstra's algorithm using a priority queue.
* Allow user input for a weighted, undirected graph.
* Compute and display the shortest path from a source node to all other nodes.
* Generate a DOT file for visualization using Graphviz.
* Enable users to convert the DOT file into an image representation of the graph.

**4. Methodology:**

* **Graph Representation:** The graph is represented using an adjacency list where each node stores edges with weights.
* **Dijkstra’s Algorithm:** Uses a priority queue (min-heap) to efficiently determine the shortest paths.
* **User Input:** The program asks for the number of nodes, edges, and their respective weights.
* **Shortest Path Calculation:** Stores shortest path distances in an array and backtracks to reconstruct paths.
* **Graph Visualization:** Generates a DOT file that can be processed with Graphviz to create an image representation of the graph.

**5. Features:**

* Efficient shortest path calculation using Dijkstra’s algorithm.
* Dynamic graph input from the user.
* Output of shortest distances for each node.
* Exportable graph visualization file (DOT format).
* Compatibility with Graphviz for graphical representation.

**6. Tools & Technologies:**

* Programming Language: C++
* Compiler: g++
* Graph Visualization: Graphviz (DOT format)
* IDE: VS Code (optional)

**7. Expected Outcomes:**

* A fully functional C++ program that computes shortest paths in a weighted graph.
* A DOT file representing the graph structure.
* A visual representation of the graph using Graphviz.
* Understanding of graph algorithms and visualization techniques.

**8. Conclusion:**

This project provides a practical approach to understanding graph theory and shortest path algorithms. The integration with Graphviz makes it more interactive and visually informative, aiding in better comprehension of Dijkstra’s algorithm and its applications.