Project Title: “Maze Solving with Dijkstra Algorithm”

A. Team Members.

1. **Durgesh Singh (IT18098)**: Specialization in Web Development.
2. **Swarn Singh Warshaneyan (IT23139)**: Specialization in Game Development.

B. Project Overview.

This project visualizes Dijkstra's algorithm in a 2D grid environment to demonstrate its efficiency in finding the shortest path between two points. The system combines a visually appealing user interface with an interactive maze-solving feature.

C. Role Contributions.

1. Durgesh Singh.

- **Algorithm Implementation**: Developed the core pathfinding functionality using Dijkstra's algorithm.
- **Dynamic Path Drawing**: Ensured that the shortest path is dynamically updated and visualized during execution.
- **Backtracking and Error Handling**: Incorporated mechanisms to reconstruct the optimal path and display appropriate messages for blocked paths or missing inputs.
- **Documentation**: Handled detailed explanations of the implementation process and results in the project report.

2. Swarn Singh Warshaneyan.

- **UI/UX Design**: Created the canvas design and intuitive user interface using HTML, CSS, and Bootstrap. This included adding image/color representations for walls, paths, start, and endpoints.
- **Interactive Features**: Implemented user interactions like setting start and end points, creating walls, and randomizing grid elements.
- **Visual Animations**: Developed smooth animations to depict the algorithm's progress and the spider-web effect for better user engagement.
- **Presentation**: Prepared and designed the PowerPoint slides to convey the technical and design elements of the project effectively.

D. Implementation Details.

1. **Algorithm Workflow**:
    - **Initialization**: Distance array (dist) and priority queue are used to manage traversal.
    - **Traversal**: Nodes are visited based on the minimum distance. The algorithm ignores walls and processes valid neighbors.
    - **Visualization**: A dynamic canvas updates to showcase the traversed path and optimal solution.
2. **User Interactions**:
    - Set start/end points.
    - Create or randomize walls.
    - Visualize the pathfinding process with clear animations.
3. **UI Features**:
    - Designed with accessibility and ease of use in mind.
    - Color-coded grid for clear differentiation between walls, paths, start, and endpoints.
4. **Error Handling**:
    - Alerts the user if the pathfinding process is triggered without valid start and end points.
    - Displays a message when no valid path exists.

E. Achievements.

- Seamlessly combined web development and game development elements.
- Provided an educational and visually engaging demonstration of Dijkstra's algorithm.
- Achieved efficient collaboration through role-based division and pair programming.

F. Challenges.

- Initial disagreement on project topic preferences was resolved by mutual consensus on level of shared interest.
- Initial disagreement on project role preferences was resolved by mutual consensus on past role experiences.
- Balancing algorithm performance with visual appeal was a key technical challenge.