Project Deadline: You must submit the project at the time of viva, before the final theory paper.

1. **Pathfinding Visualizer: (Group 1)**

Implement a program that visualizes BFS and DFS algorithms

2. **Pathfinding Visualizer: (Group 2)**

Implement a program that visualizes Dijkstra's pathfinding algorithms

3. **Sorting Algorithm Visualizer: (Group 3)**

Develop a tool that visually demonstrates how different sorting algorithms work. Users can input a list of elements, and the application will show the step-by-step process of algorithms like Insertion Sort and Merge Sort.

4. **Sorting Algorithm Visualizer: (Group 4)**

Develop a tool that visually demonstrates how different sorting algorithms work. Users can input a list of elements, and the application will show the step-by-step process of algorithms like Bubble Sort and Quicksort.

5. **Sudoku Solver: (Group 5)**

Implement a Sudoku puzzle solver that can take an incomplete Sudoku board as input and fill in the missing numbers. Use algorithms such as backtracking and appropriate data structures for efficiently solving Sudoku puzzles.

6. **Memory Matching Game: (Group 6)**

Create a classic memory matching game where the player has to find pairs of matching cards. Utilize data structures like arrays to represent the cards and implement algorithms to shuffle and check for matches.

7. **Tower of Hanoi Solver: (Group 7)**

Develop a Tower of Hanoi game with an automated solver. Allow the player to input the number of disks, and then implement a recursive algorithm with suitable data structures to demonstrate the optimal moves to solve the puzzle.

8. **Sliding Puzzle Solver: (Group 8)**

Build a sliding puzzle game where the player must rearrange numbered tiles to form a specific pattern.

9. **Task Scheduler: (Group 9)**

Create a task scheduling application that prioritizes and manages tasks. Implement queues for efficient task management .

File Explorer: (Group 10)

Build a basic file explorer that can browse, open, and manage files and folders. Employ tree data structures for file organization and add search and file preview features.

Music playlist: (Group 11)

Build a basic music playlist using doubly linked list. Each node would rerresent a music track. The data part of the node can represent the information of the track, previous and next node will represent the previous and next track.

Calendar: (Group 12)

Build a calendar to store number of days and weeks. User can search a day by going to any month and day of the year.