1. TITLE : Robot Pathfinder Using AI Search Algorithms (BFS, DFS, A*)

Name - Samriti Sharma
Registration Number- 25BCY10160
Course: Build Your Own Project
Institute: Vellore Institute Of Technology, Bhopal.
Year: 2025

2. Introduction:

This project demonstrates an interactive pathfinding robot that navigates a grid-based maze using user inputs. The robot can move manually using keyboard controls, and the system provides AI-powered hints using three classical search algorithms: **Breadth-First Search (BFS)**, **Depth-First Search (DFS)**, and **A* Search**. These algorithms help the user understand how AI finds optimal or exploratory paths in grid environments.

The project combines **game-like interaction** with **AI logic**, making it both educational and engaging.

3, PROBLEM STATEMENT:

To implement an interactive robot navigation system that illustrates how classical AI search techniques are translated into a grid based world for effective decision-making.

4. OBJECTIVES:

To create a robot navigation simulation which is interactive

To implement BFS, DFS, and A* algorithms in the pathfinder
To allow the individuals to navigate manually while getting ai suggestions based on the algorithms
To understand the objective of the algorithms in a practical approach.
To understand how AI makes pathfinding work in robotics and games easier.

5. TOOLS & TECHNOLOGIES USED:

Programming Language: Python
Environment: VS Code
Libraries:

- `collections.deque` (for BFS)
- `heapq` (for A*)
- `os` and `time` (for screen refresh and delays)

Visualization: Emojis for robot, walls, paths, and goal so that it looks good and easy to understand the elements.

6. SYSTEM REQUIREMENTS:

Python 3.8 or above
A terminal/command prompt that supports Unicode emojis
Any standard code editor

7. PROJECT DESCRIPTION:

The robot in is a 5X5 grid where :

- **Walls (⬛)**
- **Free cells (⬜)**
- **Goal (🎯)**
- **Robot (🤖)**

The user can move the robot manually using:

- **W** → Up
- **S** → Down
- **A** → Left
- **D** → Right

Also, the AI gives hints alongside:

B → BFS hint

F → DFS hint

H → A* hint

Each algorithm tells the next suggested move, based on its own logic.

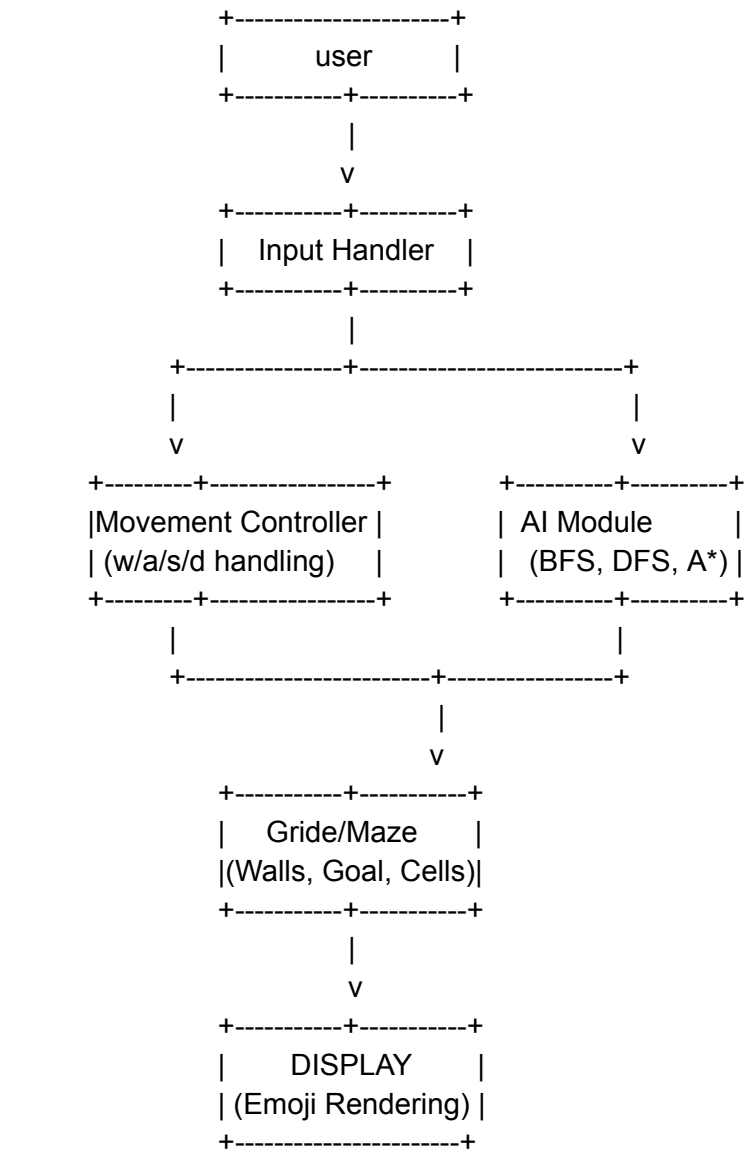
8. FUNCTIONAL REQUIRMENTS:

Manual Movement Module – Move robot using w/a/s/d.
AI Hint Module – BFS, DFS, and A* suggest next move.
Grid Rendering & Collision Module – Displays grid and prevents illegal moves.

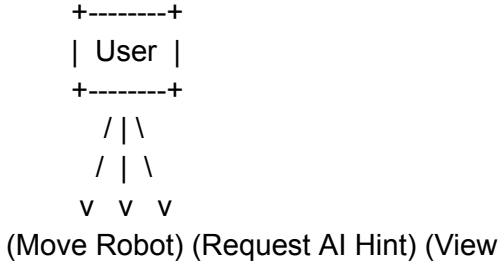
9. NON-FUNCTIONAL REQUIREMENTS:

Usability
Performance
Reliability
Maintainability

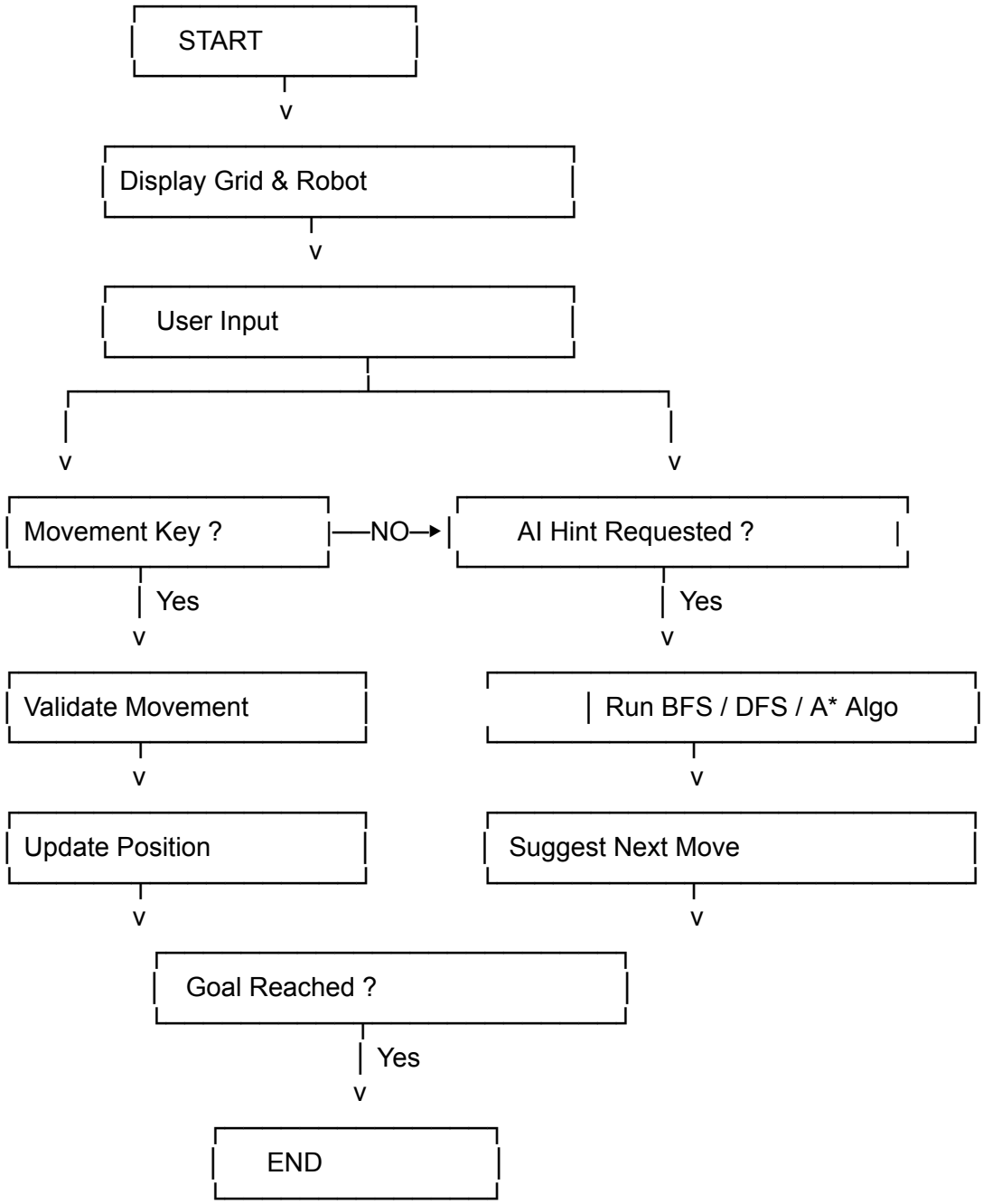
10. SYSTEM ARCHITECTURE



11. USE CASE DIAGRAM:



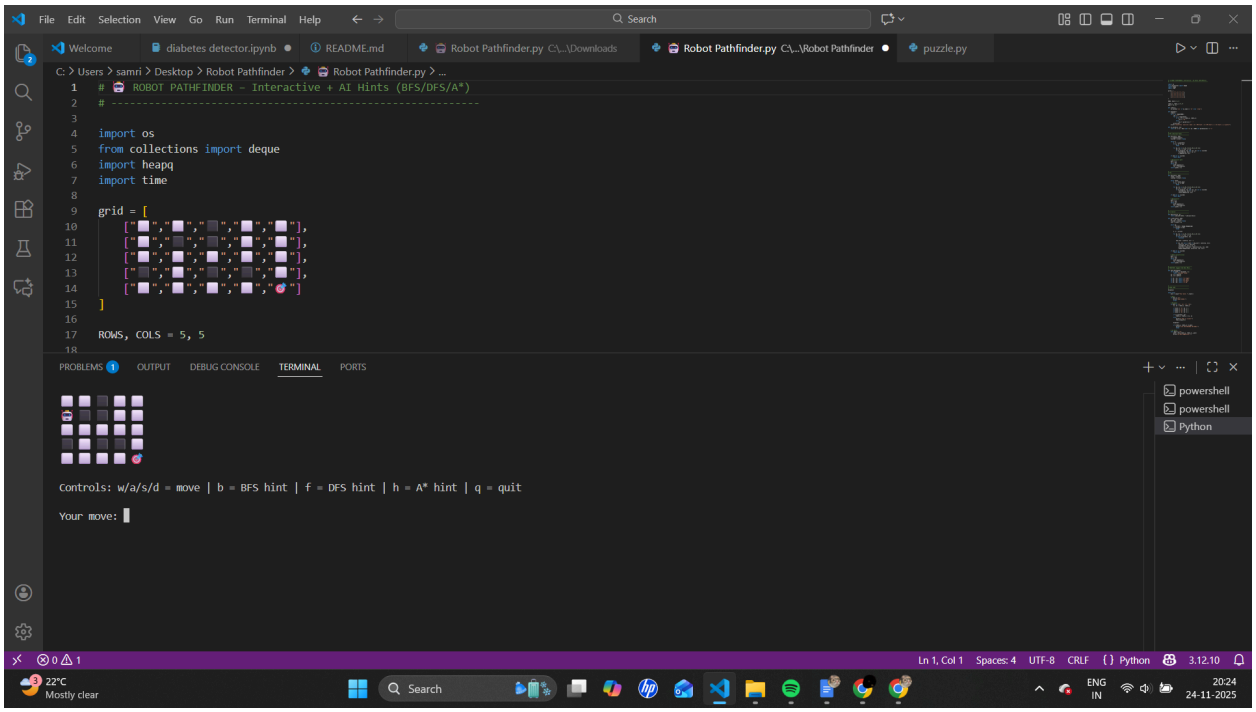
12. FLOWCHART:



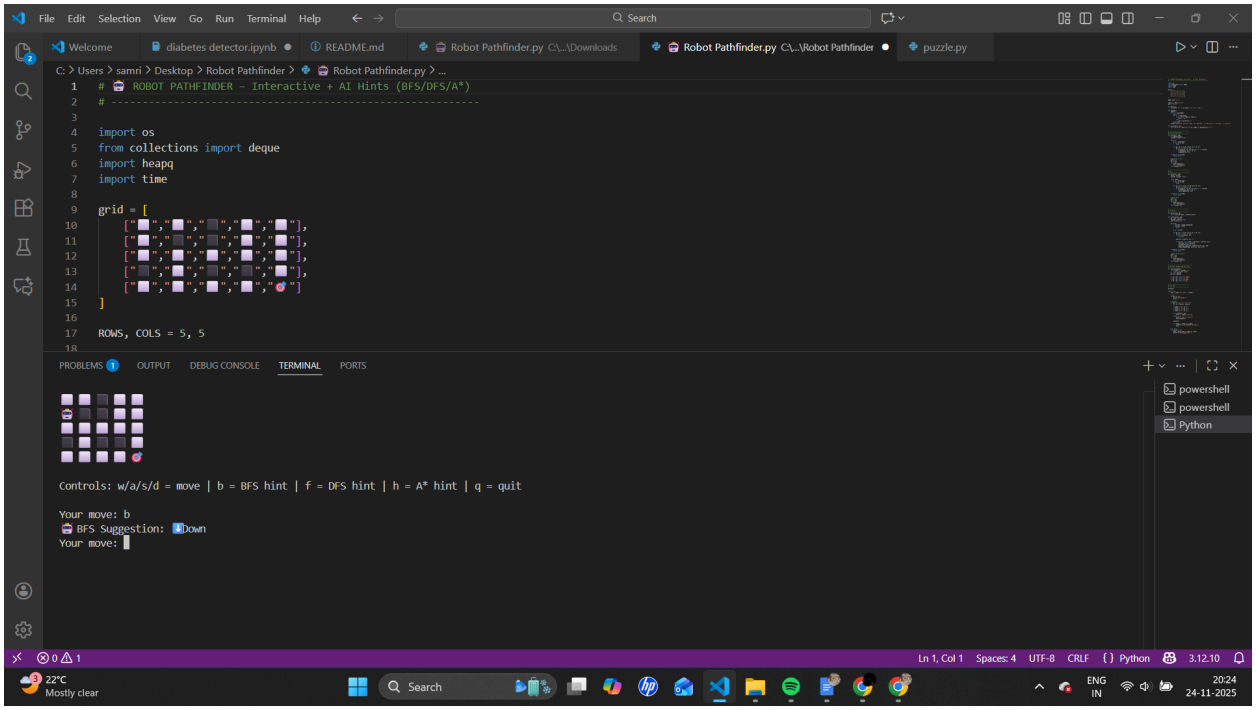
13. SEQUENCE DIAGRAM:

User → System: Input
System → Algorithm: Run BFS/DFS/A*
Algorithm → System: Suggest Move
System → User: Display Suggestion

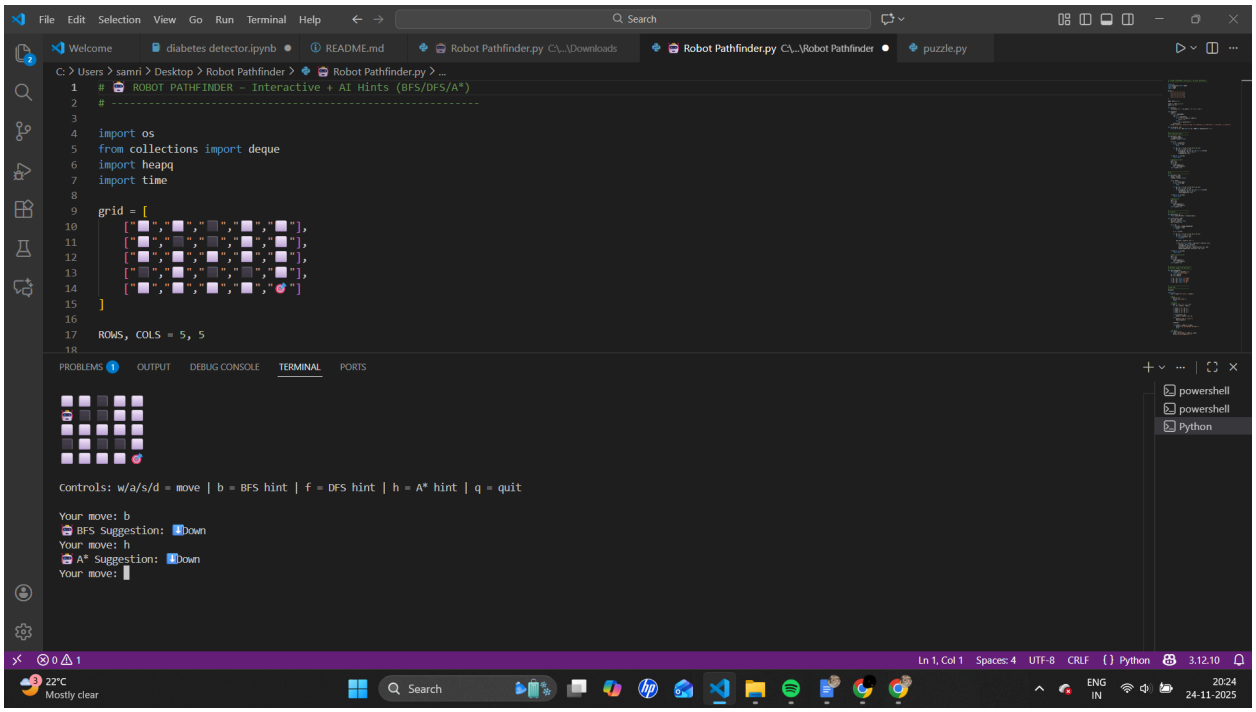
14. RESULTS:



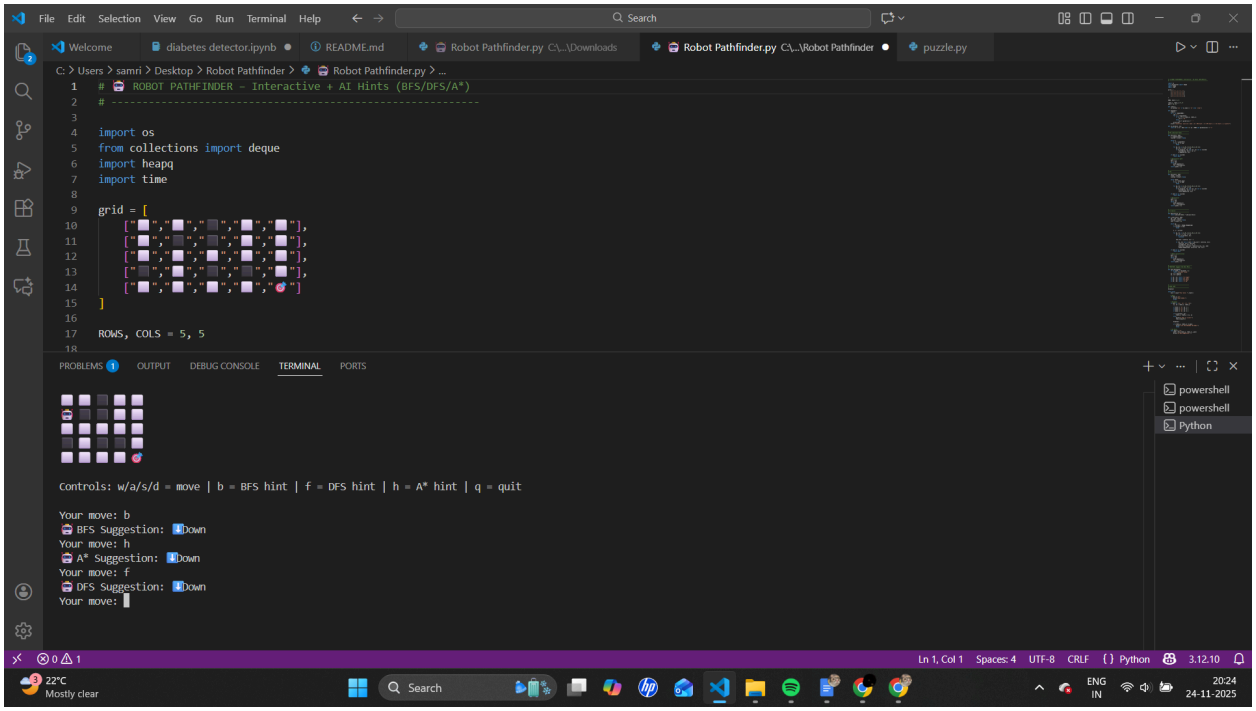
INITIAL GRID.



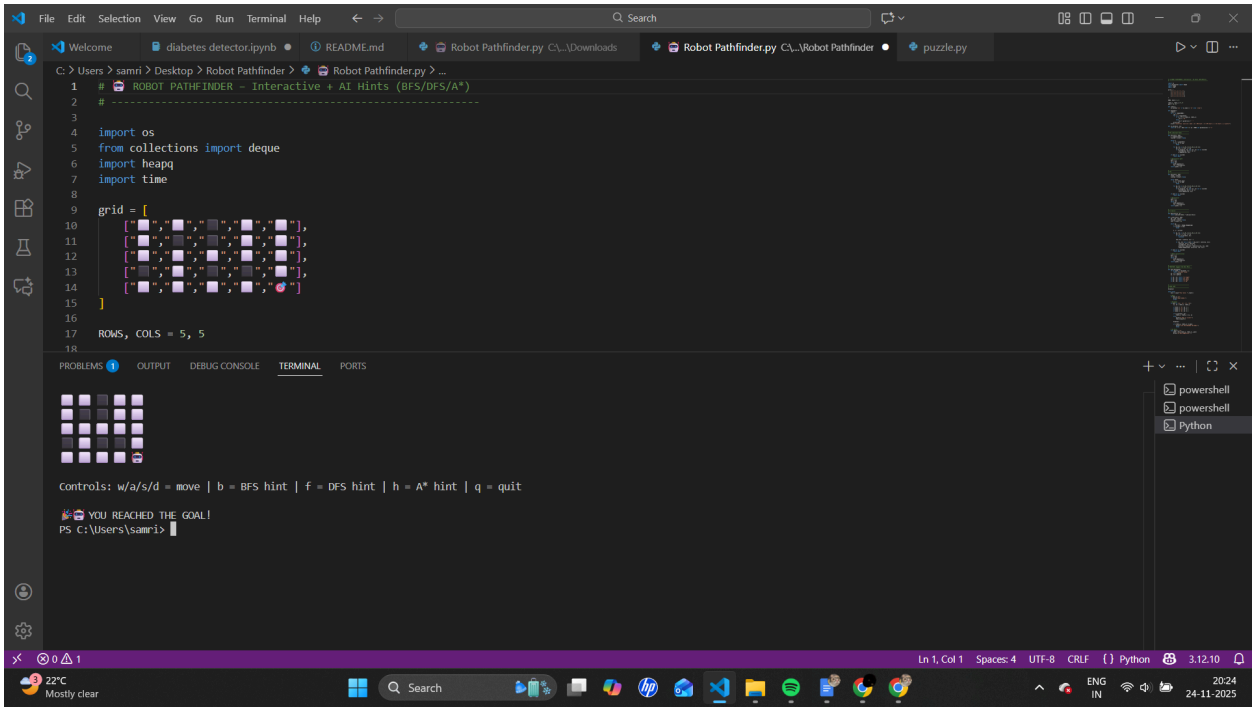
BFS SUGGESTION



A* SUGGESTION



DFS SUGGESTION



GOAL NODE REACHED

12. TESTING APPROACH:

Checking if it show correct output if you collide in the wall or the boundary.
Checking whether all the algorithms give the correct results when asked for hints.

13. CHALLENGES FACED:

Ensuring the grid and the walls and the blockages are visualised properly to avoid confusion.
Debugging the algorithm and the code at times as it is long and complex.
Making sure the interactive flow remains smooth without errors.

14. LEARNINGS AND TAKEAWAYS:

Understood the AI search algorithms in a better way after seeing and implementing them in a practical way.
Learned more coding which was complex, long but equally interesting merged with the AI elements.
Improved my debugging and structuring skills.

15. CONCLUSION:

The Robot Pathfinder effectively demonstrates the utility of classical AI search algorithms in an interactive, real-time robot navigation experience of a grid plane. From user manually directing the robot to AI hints generated from **BFS, DFS, and A***, this system is both engaging and transparent about how the search algorithms function. Furthermore, it successfully illustrates the efficiency of each algorithm: BFS for predetermined pathways, DFS for touching every square available, and A* for best prospective pathway determined in real time.

As for the development of said project, I learned various tangential skills relating to algorithm development, modular code usage, grid management, and user expectation creation. Furthermore, utilizing emojis to represent the robot and control options was a more interactive and personal experience, as well as the AI hints truly functioning as a capacity for suggestion. Thus, this project achieves its goals and is a great starting point for future development down the road, whether with enemy robots, randomly generated mazes, or a more extensive GUI.