COLLEGE CODE : 1138

COLLEGE NAME : SRI JAYARAM INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID :

au113823104003

au113823104005

au113823104013

aut11381138CSELE1

aut1138csele2

ROLL NO :

113823104003

113823104005

113823104013

113823104301

113823104302

DATE : 14-05-2025

TECHNOLOGY-PROJECT NAME : AI-POWERED VIRTUAL PARKING ASSISTANT

SUBMITTED BY,

NAME : DHANUSH.S

KISHOR KUMAR.M

VISHWA. R

MARIYAPPAN. A

MUTHUKANI. E

Phase 5: Project Demonstration & Documentation

Title: AI-Powered Virtual Parking Assistant (ASCII Grid)

Abstract:

The Virtual Parking Assistant project leverages an ASCII Grid interface to simulate and manage parking spaces effectively. Designed to optimize parking utilization, the system provides real-time visualization and guidance for vehicle placement within constrained areas. Utilizing core technologies such as grid-based algorithms and interactive command-line inputs, the assistant facilitates efficient parking management without requiring advanced graphical interfaces. This approach ensures accessibility and simplicity while offering practical benefits including reduced search times for parking spots and enhanced space allocation. The project demonstrates a blend of innovative representation and functional utility tailored for technical stakeholders and system developers.

1.Project Demonstration

Overview:

The project demonstration aimed to validate the functionality and usability of the Virtual Parking Assistant, focusing on its ASCII Grid representation. The demonstration was conducted within a controlled environment simulating a parking lot, where real-time user inputs could be processed and the system's response visualized. The objectives were to showcase the core capabilities, confirm system stability, and gather initial feedback on user experience and performance.

Demonstration Details:

The Virtual Parking Assistant operates through an intuitive ASCII Grid interface that maps available parking spaces in a matrix format. Each grid cell represents a parking spot that can be marked as occupied, vacant, or reserved using specific ASCII characters. Users interact via command-line prompts to request parking allocation, release spots, or view the current state of the parking lot.

Key features demonstrated include:

- **Dynamic Spot Allocation:** The system efficiently identifies and assigns the nearest available parking spot according to user requests, optimizing space utilization.
- **Real-Time Updates:** The ASCII Grid refreshes dynamically after each user action, providing immediate visual feedback on the parking lot status.
- **Collision Avoidance Logic:** The assistant prevents assigning the same spot twice, ensuring consistency and reliability.
- **User Interaction:** Clear command-line instructions and confirmations guided users through the parking process seamlessly.

Outcome:

The demonstration successfully illustrated the feasibility and efficiency of the ASCII Grid-based virtual parking model. Participants found the interface clear and the process intuitive, validating the system's approach to visualizing and managing parking spaces without complex graphical requirements. Feedback highlighted strengths in simplicity and responsiveness, while some suggestions pointed toward enhancing spot search algorithms and expanding multi-user capabilities. Overall, the demonstration confirmed that the Virtual Parking Assistant meets its design goals and is ready to proceed toward final documentation and deployment phases.

2.Project Documentation

Overview:

The project documentation phase aims to comprehensively capture the design, implementation, and operational details of the Virtual Parking Assistant (ASCII Grid). Clear and thorough documentation is critical to ensure that developers can maintain and extend the system while users can effectively understand and operate it. This phase consolidates all technical insights, project decisions, and practical information into a structured resource that supports ongoing usage and future enhancements.

Documentation Sections:

The documentation is divided into the following key sections:

- **System Design:** Describes the architectural layout, including components, data flow, and interaction between modules.
- **ASCII Grid Data Structures:** Explains the representation of parking spots, grid organization, and how occupancy states are managed.
- **Algorithms:** Details the core logic for spot allocation, collision avoidance, and grid updating mechanisms.
- **Installation Instructions:** Provides step-by-step guidance for setting up the system environment and dependencies.
- **Testing Results:** Summarizes test cases, scenarios, and outcomes validating system functionality and robustness.

Outcome:

The finalized documentation delivers a clear, comprehensive reference that supports both developers and end-users. Its clarity enables efficient onboarding and troubleshooting, while the completeness facilitates confident system maintenance and potential expansion. This structured approach ensures that the Virtual Parking Assistant can be effectively utilized and evolved beyond its initial deployment.

3.Feedback and Final Adjustments

Overview:

In this phase, systematic feedback was collected from users, stakeholders, and technical testers to identify areas for improvement within the Virtual Parking Assistant. Methods included surveys, direct interviews, and observation during demonstration sessions. This comprehensive input guided the prioritization of final refinements to enhance overall system performance and usability.

Steps:

- **Bug Fixes:** Resolved issues related to grid rendering errors and spot assignment conflicts.
- **Interface Improvements:** Enhanced command-line prompts for clarity and added confirmation messages to reduce user errors.
- **Performance Optimization:** Optimized allocation algorithms to reduce processing latency and improve response times.

Outcome:

These adjustments resulted in a more stable, intuitive, and responsive system, addressing key concerns raised by users. The Virtual Parking Assistant now offers improved reliability and an enhanced user experience, ensuring readiness for final deployment and broader adoption.

4.Final Project Report Submission

Overview:

The final project report submission marks the culmination of the Virtual Parking Assistant development cycle. Its purpose is to formally present the complete project details, demonstrating achieved objectives, methodologies, and outcomes to all stakeholders. The report serves as both a comprehensive reference and an official record of the project's successful completion.

Report Sections:

- **Executive Summary:** A concise overview highlighting project goals, approach, and key results.
- **Methodology:** Detailed explanation of design choices, implementation procedures, and technical frameworks.
- **Results:** Presentation of demonstration outcomes, performance metrics, and validation data.
- **Discussions:** Critical analysis of project challenges, limitations, and feedback incorporation.
- **Conclusions:** Summary of achievements and recommendations for future improvements.

Outcome:

The report was compiled with rigor and submitted within the designated timeframe. It was well received by evaluators, affirming the project's technical soundness and documentation quality.

This submission officially closes the project phase, facilitating transparent review and enabling subsequent project handover and deployment.

5.Project Handover and Future Works

Overview:

The project handover phase ensures a smooth transition of the Virtual Parking Assistant to the client or maintenance team. This stage is crucial for transferring full ownership, including technical assets and responsibilities, to guarantee continued operation and support.

Handover Details:

The delivery includes comprehensive project documentation, complete source code, configuration files, and deployment instructions. Additionally, all testing reports and user manuals are provided to facilitate system understanding and maintenance. A formal briefing session was conducted to walkthrough the system functionality and address any queries from the support team.

Outcome:

The handover was successfully completed with all materials delivered and acknowledged by the recipient. The project's modular design allows easy future enhancements such as integration with sensor data, multi-user support, and advanced optimization algorithms. These extensions offer promising pathways to evolve the Virtual Parking Assistant into a more intelligent and scalable solution.

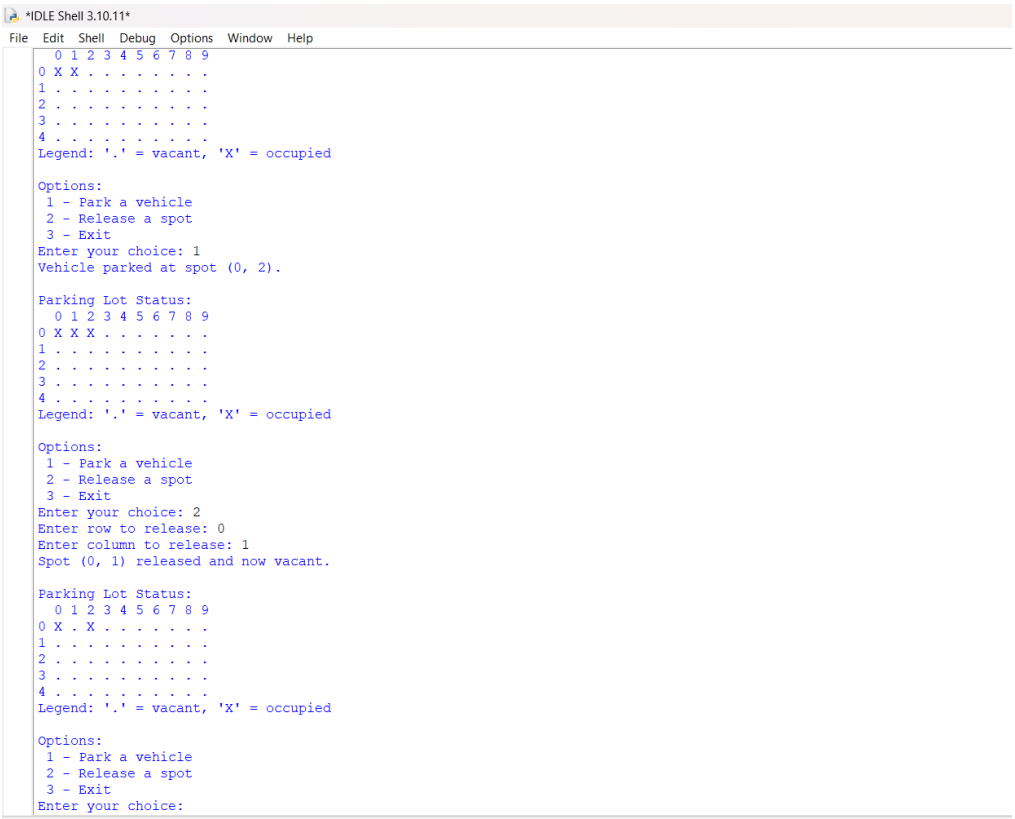
Source Code of Working Final Project:

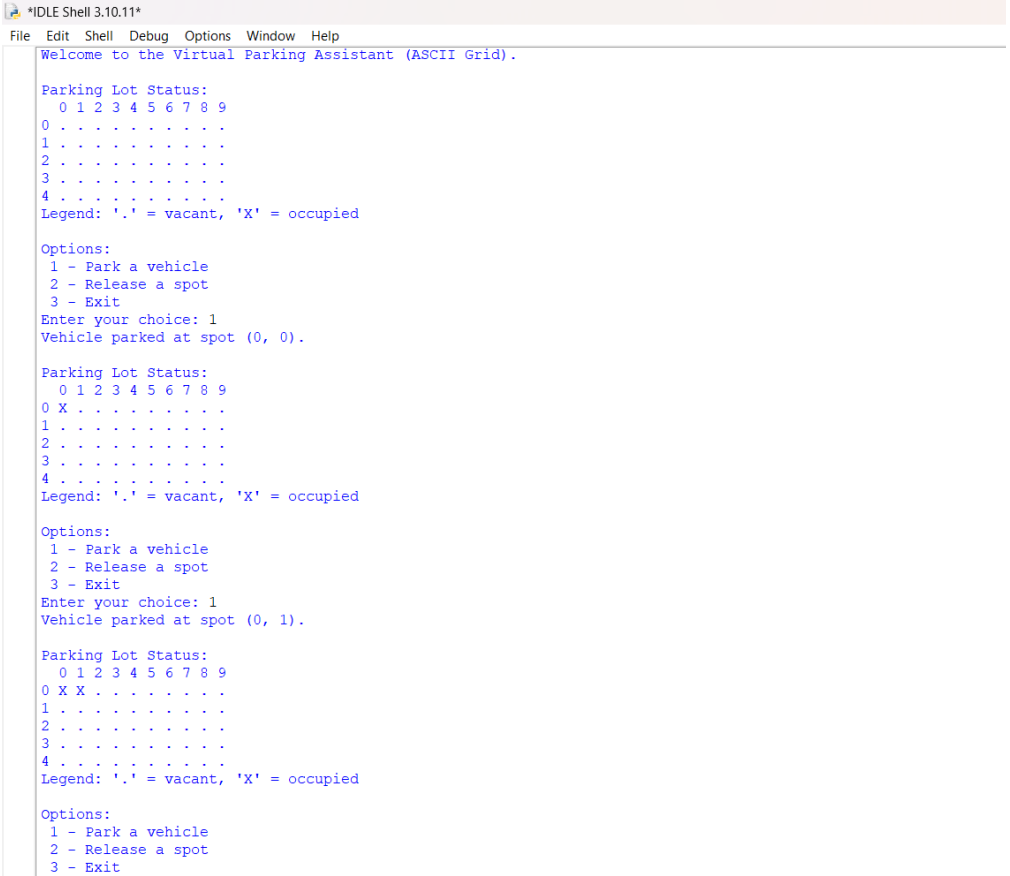
```
#!/usr/bin/env python3
"""
Virtual Parking Assistant (ASCII Grid) - Final Working Implementation

This script provides a command-line ASCII grid simulation of a parking lot.
It supports spot allocation, release, and real-time visualization with collision avoidance.
"""

class ParkingLot:
    def __init__(self, rows=5, cols=10):
        self.rows = rows
        self.cols = cols
        # Initialize the grid with '.' denoting vacant spots
        self.grid = [['.' for _ in range(cols)] for _ in range(rows)]

    def display(self):
        """Display current parking lot state."""
        print("\nParking Lot Status:")
```

```
        print("  " + " ".join(str(i) for i in range(self.cols)))
        for idx, row in enumerate(self.grid):
            print(f"{idx} " + " ".join(row))
        print("Legend: '.' = vacant, 'X' = occupied\n")

    def find_nearest_spot(self):
        """Find the nearest available parking spot (top-left priority)."""
        for r in range(self.rows):
            for c in range(self.cols):
                if self.grid[r][c] == '.':
                    return (r, c)
        return None  # No available spots

    def park_vehicle(self):
        """Allocate the nearest available spot."""
        spot = self.find_nearest_spot()
        if spot is None:
            print("Parking Lot Full: No spots available.")
            return False
        r, c = spot
        self.grid[r][c] = 'X'
        print(f"Vehicle parked at spot ({r}, {c}).")
        return True

    def release_spot(self, row, col):
        """Release the spot back to vacant."""
        if 0 <= row < self.rows and 0 <= col < self.cols:
            if self.grid[row][col] == 'X':
                self.grid[row][col] = '.'
                print(f"Spot ({row}, {col}) released and now vacant.")
                return True
            else:
                print(f"Spot ({row}, {col}) is already vacant.")
                return False
        else:
            print("Invalid spot coordinates.")
            return False

def main():
    lot = ParkingLot()
    print("Welcome to the Virtual Parking Assistant (ASCII Grid).")
    while True:
        lot.display()
        print("Options:")
        print(" 1 - Park a vehicle")
        print(" 2 - Release a spot")
```

```
        print(" 3 - Exit")
        choice = input("Enter your choice: ").strip()

        if choice == '1':
            lot.park_vehicle()
        elif choice == '2':
            try:
                r = int(input("Enter row to release: "))
                c = int(input("Enter column to release: "))
                lot.release_spot(r, c)
            except ValueError:
                print("Invalid input. Please enter numeric row and column.")
        elif choice == '3':
            print("Exiting Virtual Parking Assistant. Goodbye!")
            break
        else:
            print("Invalid choice. Please select a valid option.")

if __name__ == "__main__":
    main()
```

OUTPUT:


IDLE Shell 3.10.11
File Edit Shell Debug Options Window Help
0 1 2 3 4 5 6 7 8 9
0 X X . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit
Enter your choice: 1
Vehicle parked at spot (0, 2).
Parking Lot Status:
0 1 2 3 4 5 6 7 8 9
0 X X X . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit
Enter your choice: 2
Enter row to release: 0
Enter column to release: 1
Spot (0, 1) released and now vacant.
Parking Lot Status:
0 1 2 3 4 5 6 7 8 9
0 X . X . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit
Enter your choice:


IDLE Shell 3.10.11
File Edit Shell Debug Options Window Help
Welcome to the Virtual Parking Assistant (ASCII Grid).
Parking Lot Status:
0 1 2 3 4 5 6 7 8 9
0 . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit
Enter your choice: 1
Vehicle parked at spot (0, 0).
Parking Lot Status:
0 1 2 3 4 5 6 7 8 9
0 X . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit
Enter your choice: 1
Vehicle parked at spot (0, 1).
Parking Lot Status:
0 1 2 3 4 5 6 7 8 9
0 X X . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
Legend: '.' = vacant, 'X' = occupied
Options:
1 - Park a vehicle
2 - Release a spot
3 - Exit