CHAPTER-1

COMPANY PROFILE

Company name: EZ training and technologies Pvt.ltd

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

Locations: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

CHAPTER-2

ABSTRACT

Our Python project focuses on developing a stadium seating management system. It aims to efficiently allocate and reserve seats for various events, optimizing space utilization while accommodating individual and group bookings. The system includes features for seat pricing, real-time availability updates, and user-friendly interfaces for administrators and attendees, aiming to streamline stadium seating management for enhanced event experiences.

This project combines various aspects of software development, including data structures, algorithms, user interfaces, and potentially even web development if you decide to build a web-based interface for it. Here's a general outline of the project:

User authentication, real time updates, seat mapping, user interface, attendance tracking, data security

The project utilizes object-oriented programming principles to create a modular and scalable system. Python's versatility allows for seamless integration of various features, including data storage, user authentication, and real-time updates. The system employs data structures such as lists and dictionaries to represent seating arrangements, enabling efficient retrieval and manipulation of seat information. Additionally, libraries like Flask are employed to develop a web-based interface, facilitating accessibility across different devices and platforms.

One of the key challenges addressed in this project is optimizing seat allocation to maximize occupancy and revenue while ensuring a positive user experience. The system incorporates algorithms for seat recommendation based on factors such as proximity to amenities, group seating preferences, and pricing constraints. Moreover, it implements mechanisms for handling seat reservations, cancellations, and refunds to accommodate dynamic changes in attendance and preferences

CHAPTER-3

INTRODUCTION OF THE PROJECT

- The project includes enhancing the overall attendee experience by providing real-time updates on seating availability, optimizing seat allocation based on various factors, and streamlining ticketing processes through intuitive interfaces.

- A stadium seating management The Stadium Seating Management Python Project aims to revolutionize the management of seating arrangements in stadiums for various events, from sports matches to concerts. By leveraging the power and versatility of Python, this project seeks to automate and optimize processes such as seating allocation, crowd control, and ticketing.

- Key objectives of project aim to streamline the process of organizing and overseeing seating arrangements, ticket sales, and attendance tracking for events held in a stadium or large venue.

- This project typically involves creating a digital platform where users can browse upcoming events, select their preferred seats from an interactive seating chart, purchase tickets securely, and receive confirmation of their bookings. On the administrative side, the system allows event organizers to create and manage events, set seating layouts, monitor ticket sales, and track attendance.

- Key features of such a system include user authentication for secure access, integration with payment gateways for smooth transactions, real-time updates on seat availability, and reporting tools for analyzing event performance.

- Overall, a stadium seating management project aims to enhance the efficiency and convenience of organizing and attending events in large venues, providing a seamless experience for both organizers and attendees.

CHAPTER-4

MODULE DESCRIPTION

Module Name: Stadium Seating System

Description: The Stadium class is designed to manage the seating arrangements and ticket sales for a stadium. It includes functionalities for selling tickets, returning tickets, and checking seat availability for different seating classes (VIP, Upper Class, and Lower Class).

Class: Stadium

Attributes:

capacity: Total capacity of the stadium.

VIP capacity: Capacity of VIP seating.

upper_capacity: Capacity of Upper-Class seating.

lower_capacity: Capacity of Lower-Class seating.

available_seats: Number of available seats in the stadium.

available_vip_seats: Number of available VIP seats.

available_upper_seats: Number of available Upper-Class seats.

available_lower_seats: Number of available Lower-Class seats.

seats: Dictionary containing lists representing seat availability for each seating class.

Methods:

sell_ticket(seat_class): Sell a ticket for the specified seating class.

check_availability(seat_class=None): Check the availability of seats for the specified seating class, or total available seats if no class is specified.

return_ticket (seat_class, seat_number): Return a ticket for the specified seat in the given seating class.

display_menu (): Display the menu of options for interacting with the stadium seating system.

run (): Start the interactive loop for managing ticket sales and seat availability.

This module provides a convenient way to manage seating and ticket sales for a stadium environment. Developers can use it to implement a user-friendly interface for customers to purchase tickets and check seat availability.

CHAPTER-6

ALGORITHM

STEP 1: Start

Import or run the Python script containing the Stadium class.

Create an instance of the Stadium class with the desired capacities (capacity, vip_capacity, upper_capacity, lower_capacity).

Call the run () method of the Stadium instance to start the program.

STEP 2: Display Menu

Inside the run () method, the display_menu () function is called to print the options available

STEP 3: User Input

Prompt the user to enter their choice by displaying a menu with numbered options.

Accept user input for their choice.

STEP 4: Process User Choice

Based on the user's input, execute the corresponding action:

If the choice is '1', sell a VIP ticket.

If the choice is '2', sell an Upper-Class ticket.

If the choice is '3', sell a Lower-Class ticket.

If the choice is '4', return a VIP ticket.

If the choice is '5', return an Upper-Class ticket.

If the choice is '6', return a Lower-Class ticket.

If the choice is '7', check the availability of VIP seats.

If the choice is '8', check the availability of Upper-Class seats.

If the choice is '9', check the availability of Lower-Class seats.

If the choice is '10', check the total availability of all seats.

If the choice is '11', exit the program.

STEP 5: Sell Ticket Method:

When selling a ticket (choices '1', '2', '3'), the sell_ticket () method is called with the corresponding seat class.

STEP 6: Return Ticket Method:

When returning a ticket (choices '4', '5', '6'), the return_ticket () method is called with the corresponding seat class and seat number.

STEP 7: Check Availability Method:

When checking availability (choices '7', '8', '9', '10'), the check_availability () method is called with the corresponding seat class or without any arguments for total availability.

STEP 8: Invalid Choice:

If the user enters an invalid choice, display a message indicating that the choice is invalid and prompt them to enter a valid option.

STEP 9: Exit Program:

If the user chooses to exit (choice '11'), print a message indicating that the program is exiting and break out of the while loop, effectively ending the program.

STEP 10: End

Once the user chooses to exit, the program terminates.

CHAPTER-6

OUTPUTS

CHOICE 1: SELL TICKET(VIP)

```
 Enter your name: sneha
Enter your age: 20
How many VIP seats would you like to book? 4
Tickets sold for VIP seats [1, 2, 3, 4] for sneha (Age: 20)
Ticket sold successfully.
```

CHOICE 2: SELL TICKET (UPPER CLASS)

```
Enter your name: kalyani
Enter your age: 19
How many Upper seats would you like to book? 2
Tickets sold for Upper seats [1, 2] for kalyani (Age: 19)
Ticket sold successfully.
```

CHOICE 3: SELL TICKET (LOWER CLASS)

```
Enter your name: nousheen
Enter your age: 19
How many Lower seats would you like to book? 3
Tickets sold for Lower seats [1, 2, 3] for nousheen (Age: 19)
Ticket sold successfully.
```

CHOICE 4: RETURN TICKET (VIP)

```
Enter VIP seat number to return ticket: 1
Ticket returned for VIP seat 1
Ticket returned successfully.
Enter VIP seat number to return ticket: 2
```

```
Ticket returned for VIP seat 2
Ticket returned successfully.
Enter VIP seat number to return ticket: 5
VIP seat 5 is already available.
Ticket could not be returned.

CHOICE 5: RETURN TICKET (UPPER CLASS)
Enter Upper Class seat number to return ticket: 1
Ticket returned for Upper seat 1
Ticket returned successfully.
Enter Upper Class seat number to return ticket: 2
Ticket returned for Upper seat 2
Ticket returned successfully.
Enter Upper Class seat number to return ticket: 4
Upper seat 4 is already available.
Ticket could not be returned.

CHOICE 6: RETURN TICKET (LOWER CLASS)
Enter Lower Class seat number to return ticket: 1
Ticket returned for Lower seat 1
Ticket returned successfully.
Enter Lower Class seat number to return ticket: 2
Ticket returned for Lower seat 2
Ticket returned successfully.
Enter Lower Class seat number to return ticket: 3
Ticket returned for Lower seat 3
Ticket returned successfully.
```

CHOICE 7: CHECK AVAILABILITY (VIP)

Available VIP seats: 50

CHOICE 8: CHECK AVAILABILITY (UPPERCLASS)

Available Upper-Class seats: 100

CHOICE 9: CHECK AVAILABILITY (LOWER CLASS)

Available Lower-Class seats: 150

CHOICE 10: CHECK TOTAL AVAILABILITY

Total Available seats: 300

CHOICE 11: EXIT

CHAPTER-7

CONCLUSION

In conclusion, the Python-based Stadium Seat Management System represents an efficient solution for automating seat allocation and enhancing user experience within stadium environments. Through a straightforward menu-driven interface, the system facilitates seamless ticket sales, returns, and seat availability checks across different seating classes, ensuring optimal utilization of stadium capacity. While the current implementation offers a solid foundation with basic functionalities, future iterations could explore enhancements such as graphical interfaces, online payment integrations, and real-time updates for a more intuitive and feature-rich experience. Overall, the project lays a strong groundwork for efficient seat management in stadiums, with potential for further customization and expansion to meet evolving requirements and user expectations.