







“Ultimate Amusement Park Ride & Ticket Management System ”



Project Overview:

Design a **complete management system** for an amusement park that handles:

-  Ticketing
-  Ride Queues (Multiple rides with priority queue & normal queue)
-  Special Attractions (Using Stack-based access control)
-  Visitor History & Ride Logs

This project mimics **real-life scenarios** like fast-pass tickets, VIPs, and limited-entry attractions!



Core Modules & Features:

1. Ticket Booking System (Queue + Priority Queue)

- Visitors can buy **Regular** or **VIP (Fast-Pass)** tickets.
- Use **two separate queues**:
 - **VIP Queue (High Priority)**
 - **Regular Queue (Normal Priority)**
- Visitors from the VIP queue always get preference over regular visitors.
- Track **ticket numbers & booking time**.

2. 🎢 Ride Management (Advanced Queue System)

- Each ride has its own queue system.
- Rides support:
 - **Add Visitor to Ride Queue**
 - **Process Ride for Next Visitor**
 - **View Queue Status of Each Ride**
- Allow visitors to **switch rides** (move between ride queues).

3. 👤 Haunted House & Escape Room (Stack-based Attractions)

- Limited-entry attractions (only N visitors allowed at once).
- **Stack** ensures:
 - Last visitor to enter will leave first (LIFO).
- Visitors can **exit early** if desired.
- Maintain stack-based access logs.

4. 📅 Visitor History & Ride Logs (Bonus Feature)

- Store history of:
 - All rides visited.
 - Ticket type used.
 - Total waiting time.
- Display reports for each visitor at end.


5. Admin Module (Optional Bonus)

- Admin can:
 - View total tickets sold.
 - Clear all ride queues at closing time.
 - Reset system for next day.
 - Generate logs in text or CSV file.

Key Data Structures:

Feature	Data Structure Used
Ticketing (VIP & Regular)	Priority Queue (2 Queues)
Ride Queues	Queue (FIFO)
Haunted House & Escape Room	Stack (LIFO)
Visitor History	Dictionary / List
Admin Logs	File System





Sample Visitor Flow:

1. Visitor buys **VIP Ticket**  → Gets priority in all queues.
2. Chooses rides → Enters ride queues.
3. Visits Haunted House → Added to Stack (LIFO).
4. Takes rides → Removed from queue after each ride.
5. System keeps logs of:

- Ticket ID
- Ride Names
- Queue Times
- Attraction Visits



Advanced Challenge Ideas (Optional):

-  Simulate Time Delay for Rides & Waiting.
-  Build Command-line Menu for Interactivity.
-  Save Data to File & Reload on Start.
-  Visualize Queue Lengths via ASCII Art.



Learning Outcomes:

- Deep Understanding of **Priority Queues**, **Stacks**, and **Queues**.
- Real-world problem modeling & system design.
- Handling multiple complex data structures together.
- Designing user-friendly & scalable solutions.



Project Extensions (Optional for Extra Credit):

- Web Interface using Flask/Django.
- Graphical Simulation of Ride Queues (Using Pygame/Tkinter).
- Integrate Email Notifications (Simulated).