**Hotel Management System**

**Overview**

The **Hotel Management System** is a comprehensive software solution designed to manage hotel operations efficiently. Developed as part of a Data Structures and Algorithms (DSA) lab project, this system integrates various functionalities, including room management, employee management, customer handling, and revenue tracking.

The project focuses on modular design, clean code structure, and robust functionality to simulate real-world hotel management systems.

**Features**

**Admin Functionalities**

1. **Hotel Management**:
   * View details of all hotels in the chain.
2. **Room Management**:
   * View all rooms and their statuses.
   * Add and delete rooms dynamically.
   * Update room cleanliness and booking status.
3. **Employee Management**:
   * Add and delete employees.
   * View all employee details.
   * Assign and track employee roles and salaries.
4. **Customer Management**:
   * Check-in and check-out customers.
   * Maintain a detailed log of customer stays.
5. **Feedback Management**:
   * View customer feedback.
   * Delete irrelevant or inappropriate feedback.

**Customer Functionalities**

1. **Feedback Submission**:
   * Log complaints or give feedback for service improvement.
2. **Access Important Contacts**:
   * View essential contact details (e.g., reception, janitor, emergency services).

**Technical Details**

**Programming Language**

* The system is implemented in **C++**, leveraging Object-Oriented Programming (OOP) principles and data structures such as vector for dynamic management.

**Modules**

1. **Hotel Management**:
   * Structs: hotelChain, hotel, room, employee, customer.
   * Features:
     + Manage multiple hotels within a single chain.
     + Handle all aspects of hotel operation, from room assignment to financial reporting.
2. **Room Management**:
   * Tracks availability, cleanliness, and pricing of rooms.
   * Enables dynamic addition and deletion of rooms.
3. **Employee Management**:
   * Stores and retrieves employee details.
   * Manages employee assignments, salaries, and roles.
4. **Customer Log**:
   * Maintains detailed records of all customer stays.
   * Provides insights into occupancy trends and customer preferences.
5. **Feedback Handling**:
   * Stores customer feedback dynamically.
   * Allows administrators to review and manage feedback entries.

**How It Works**

**For Admins:**

1. Log in with a username and password.
2. Access various admin functionalities, such as managing rooms, employees, and customers.
3. Track hotel performance and address customer feedback.

**For Customers:**

1. Submit feedback or complaints.
2. Access important hotel contact details for assistance.

**Sample Use Case**

**Admin Workflow:**

1. Log in as an admin.
2. Select a hotel branch to manage.
3. View available rooms and book a customer into a selected room.
4. Add new rooms or employees as required.

**Customer Workflow:**

1. Log in as a customer.
2. Submit feedback or complaints about their stay.
3. Access essential hotel contacts for support.

**Challenges Solved**

* **Dynamic Data Management**: Efficiently managed dynamic data using STL containers like vector.
* **User Roles**: Implemented separate workflows for admin and customer roles.
* **Error Handling**: Added safeguards for invalid inputs and edge cases.

**Future Enhancements**

* **Database Integration**:
  + Store data persistently instead of relying on in-memory structures.
* **User Interface**:
  + Develop a graphical user interface (GUI) for improved usability.
* **Advanced Analytics**:
  + Include reports on occupancy rates, peak seasons, and customer demographics.

**Credits**

This project was developed by Srijan Kumar as part of the **Data Structures and Algorithms (DSA) lab course** at IIT Guwahati. Special thanks to **Prof. Gaurav Trivedi** for guidance and support.

**How to Run**

1. Clone the repository or copy the source code.
2. Install and run make on terminal.
3. Run the executable (./main).
4. Follow on-screen instructions for admin or customer workflows.