Prepared by ***Muhammad Anas Waqar***

7.	Conclusion	**Pg 49**
	Summarize the key findings, achievements, and challenges encountered during the development of your GUI-based desktop application.	

Project overview:

The project Hotel Management System is a desktop-based application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various rooms, employees, drivers and customers make this system very flexible and convenient. The hotel manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system. Hotel management project provides room booking, staff management and other necessary hotel management features. The system allows the manager to post available rooms in the system.

The Purpose of the whole process is to ease the daily or regular activities of the Hotel Management into an automatic computerized retrievable process. The daily activities include the Room activities, Entering details of the new customer check in, To allocate a room as per the customer need and interest, Recording the checkout time and details, Releasing or Empty of room and to record the process in a computer system for future.

There are 2 modules I have included in this project.

Admin: Admin can add a new room, a new employee and a new driver.

Receptionist: Receptionist can add a new customer and allocate rooms, can check all room details, customers details, employee details, search for a particular room etc.

The application of the Hotel Management System bears the following functions to use by the Administrator:

- Add a new Room
- Add an Employee
- Add a new Customer
- Check room status
- Check all employees details
- Check all Customers details
- Update room status • Update check status etc.

Programming Language Choice:

- **Explain the rationale behind choosing the programming language for your application**

I used Java with the Swing library for the Hotel
Management System, because it is a practical decision and it aligns well with desktop application development. I have given the brief overview of how the choice of Java with Swing contributes to the project below:

Platform Independence:

Java is known for its "write once, run anywhere" (WORA) philosophy. It allows the application to be developed on one platform (e.g., Windows) and run on another (e.g., Linux or macOS) without modification. This portability is advantageous for desktop applications like hotel management systems that may be deployed on diverse operating systems.

GUI Development with Swing:

Java provides Swing, a powerful and flexible GUI toolkit. Swing allows the creation of rich and interactive graphical user interfaces, making it suitable for applications where a visually appealing and user-friendly interface is crucial, such as hotel management systems. **Extensive Standard Library:**

Java comes with a comprehensive standard library that includes utilities for networking, database connectivity (JDBC), and various other functionalities. This simplifies development by providing pre-built modules for common tasks, contributing to the efficiency of the application.

Object-Oriented Programming (OOP):

Java is a robust, object-oriented programming language. The use of OOP principles like encapsulation, inheritance, and polymorphism can contribute to the organization and maintainability of the codebase, especially in larger projects like hotel management systems.

Database Connectivity (JDBC):

Java Database Connectivity (JDBC) is a standard API for connecting Java applications to relational databases. In this hotel management application, the use of JDBC in the Conn class suggests that the application interacts with a database to store and retrieve information.

Community Support and Documentation:

Java has a large and active developer community. Abundant resources, tutorials, and documentation are available, making it easier for developers to troubleshoot issues, seek guidance, and enhance their skills in Java development.

Security Features:

Java has built-in security features, including the ability to run applications in a sandboxed environment. This is crucial for applications that may handle sensitive information, such as hotel management systems, to ensure data integrity and user security.

Scalability:

Java is well-suited for scalable applications. As hotel management systems may need to handle a growing number of rooms, customers, and transactions, Java's scalability is an important consideration.

- **Discuss factors such as ease of development, community support, and compatibility with the chosen database:**

Ease of Development:

Swing is a mature and versatile GUI toolkit for Java. It provides a rich set of components for building interactive user interfaces. Its ease of use and flexibility make it suitable for developing desktop applications.

Compatibility with Java:

Since Swing is part of the Java Standard Edition (SE), it seamlessly integrates with the Java language. This ensures compatibility with your chosen backend, enhancing the overall coherence of your application.

Community Support:

While Swing has been part of Java for a long time, it still enjoys community support. Many resources, forums, and tutorials are available for developers using Swing, making it easier to troubleshoot issues or seek guidance.

Database & Relations:

Database Choice:

I chose mysql for Hotel Management System.

MySQL seamlessly integrates with Java through JDBC, simplifying the connection between the Java-based frontend and MySQL backend for efficient data management. MySQL is an open-source RDBMS, providing cost effectiveness and freedom to modify its source code, aligning well with projects with budget constraints. It is also renowned for its user friendly interface and straightforward setup and it make sure the accessibility for developers and administrators. The large and active MySQL community offers extensive resources, including documentation, forums, and tutorials, facilitating problem-solving and knowledge-sharing. MySQL adheres to ACID principles, ensuring reliable and consistent database transactions, which is imperative for maintaining data integrity in a Hotel Management System. It is designed to scale seamlessly, accommodating both small and large-scale applications, making it ideal for a Hotel Management System poised for growth. Also MySQL is recognized for its high performance, efficiently handling read and write operations, crucial for applications requiring quick data retrieval and updates. MySQL offers multiple storage engines, such as InnoDB and MyISAM, providing flexibility and optimization for different types of data in the Hotel Management System.

Database Relations:

- **List at least four relations in your database schema:** login Table:

Fields: username, password employee Table:

Fields: name, age, gender, job, salary, phone, email, CNIC emplogin Table:

Fields: CNIC, username, password roomlogin Table:

Fields: roomnumber, username, password customeroom Table: Fields: roomnumber,

number empdpt Table:

Fields: CNIC, department room Table:

Fields: roomnumber, availability, cleaning_status, price, bed_type driver Table:

Fields: CNIC, name, age, gender, company, brand, available, location customer Table:

Fields: document, number, name, gender, country, room, checkintime, deposit department Table:

Fields: department, budget

- **Briefly describe each relation and explain how they contribute to the functionality of your application:**

login Table:

The login table manages user credentials, storing username and password combinations. It plays a crucial role in authenticating users, providing access control to the system. employee Table:

The employee table holds information about hotel staff, including details such as name, age, gender, job, salary, phone, email, and a unique identifier, CNIC. This table helps in managing and tracking the hotel's workforce. emplogin Table:

The emplogin table establishes a relationship between employees and their login credentials. It links the employee's CNIC to a username and password from the login table, ensuring secure access for staff members. roomlogin Table:

Similar to emplogin, the roomlogin table associates
specific rooms with login credentials. It links a roomnumber to a username and password from the login table, controlling access to room-related functionalities.

customeroom Table:

The customeroom table manages the relationship between rooms and customers. It links a roomnumber to a number, establishing connections between occupied rooms and customer details.

empdpt Table:

The empdpt table organizes employees based on their departments, linking an employee's CNIC to a specific department. This aids in departmental management and employee assignment.

room Table:

The room table contains details about hotel rooms, including roomnumber, availability, cleaning_status, price, and bed_type. It assists in tracking room statuses, pricing, and bed configurations.

driver Table:

The driver table manages information about drivers, including CNIC, name, age, gender, company, brand, available status, and location. This supports the hotel's transportation logistics.

customer Table:

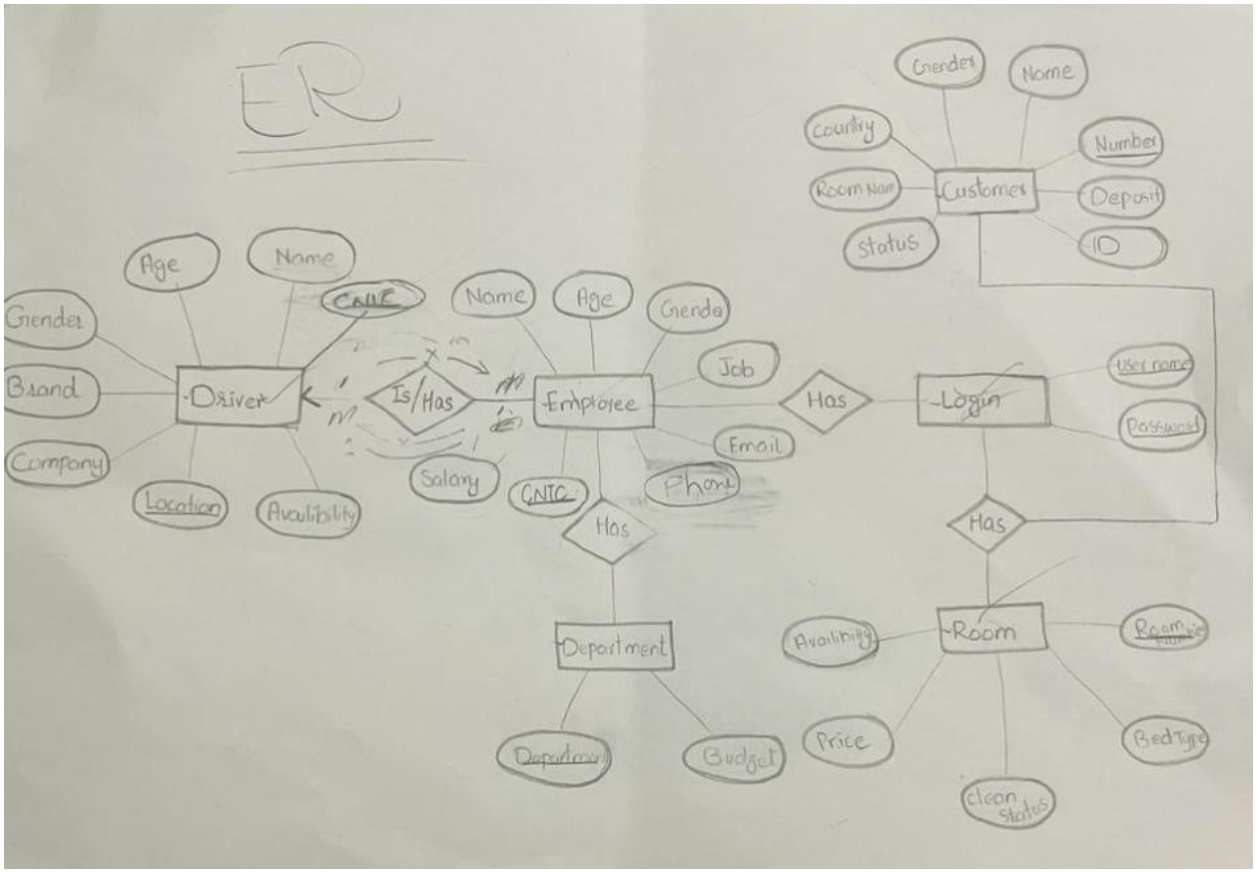
The customer table stores information about hotel guests, including document type, number, name, gender, country, room assigned, checkintime, and deposit. It helps in managing guest accommodations and check-in details.

department Table:

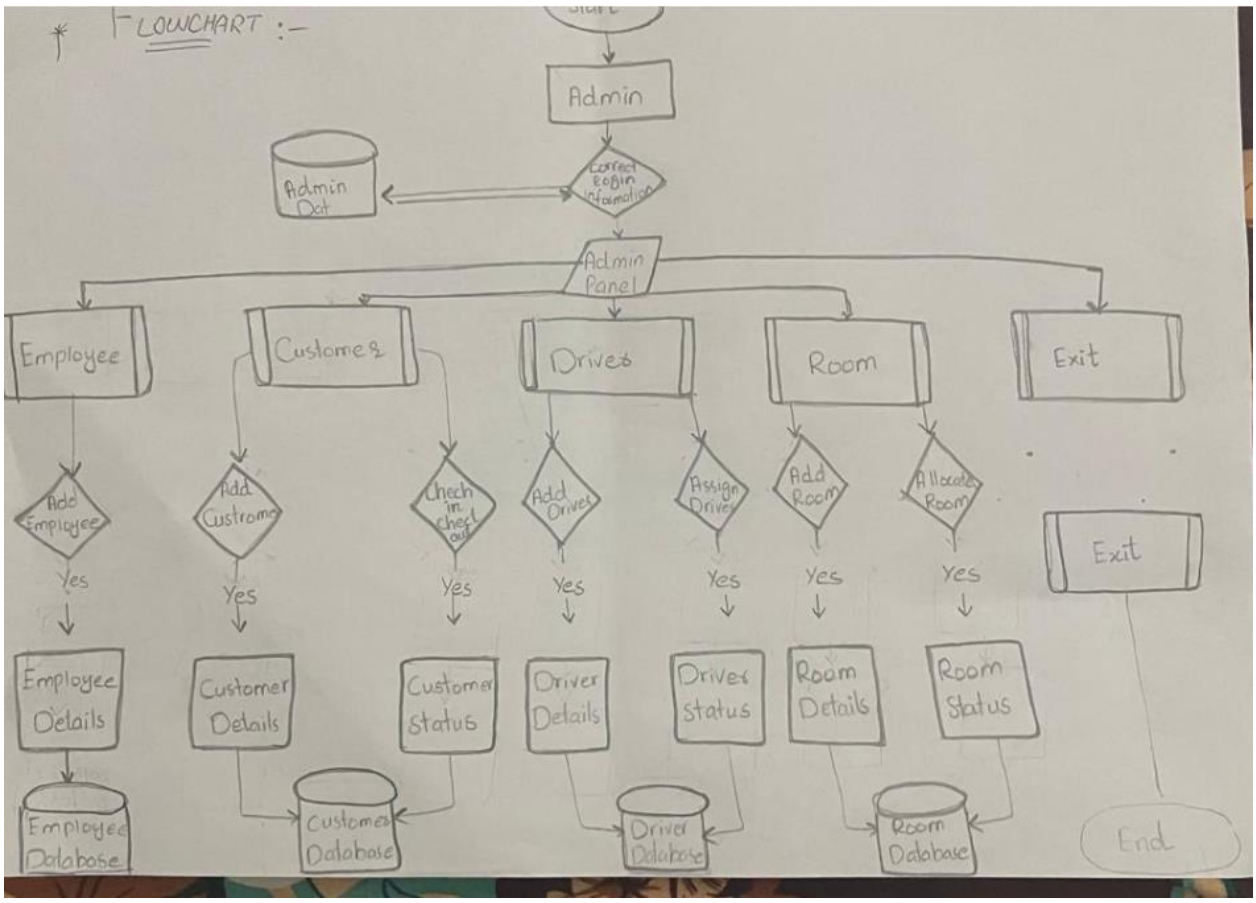
The department table holds details about various hotel departments, such as department name and budget. This assists in budgetary control and overall management of different functional areas within the hotel.

These relations collectively contribute to the comprehensive functionality of the hotel management system, covering aspects ranging from user authentication and employee management to room occupancy, transportation, guest details, and departmental budgets.

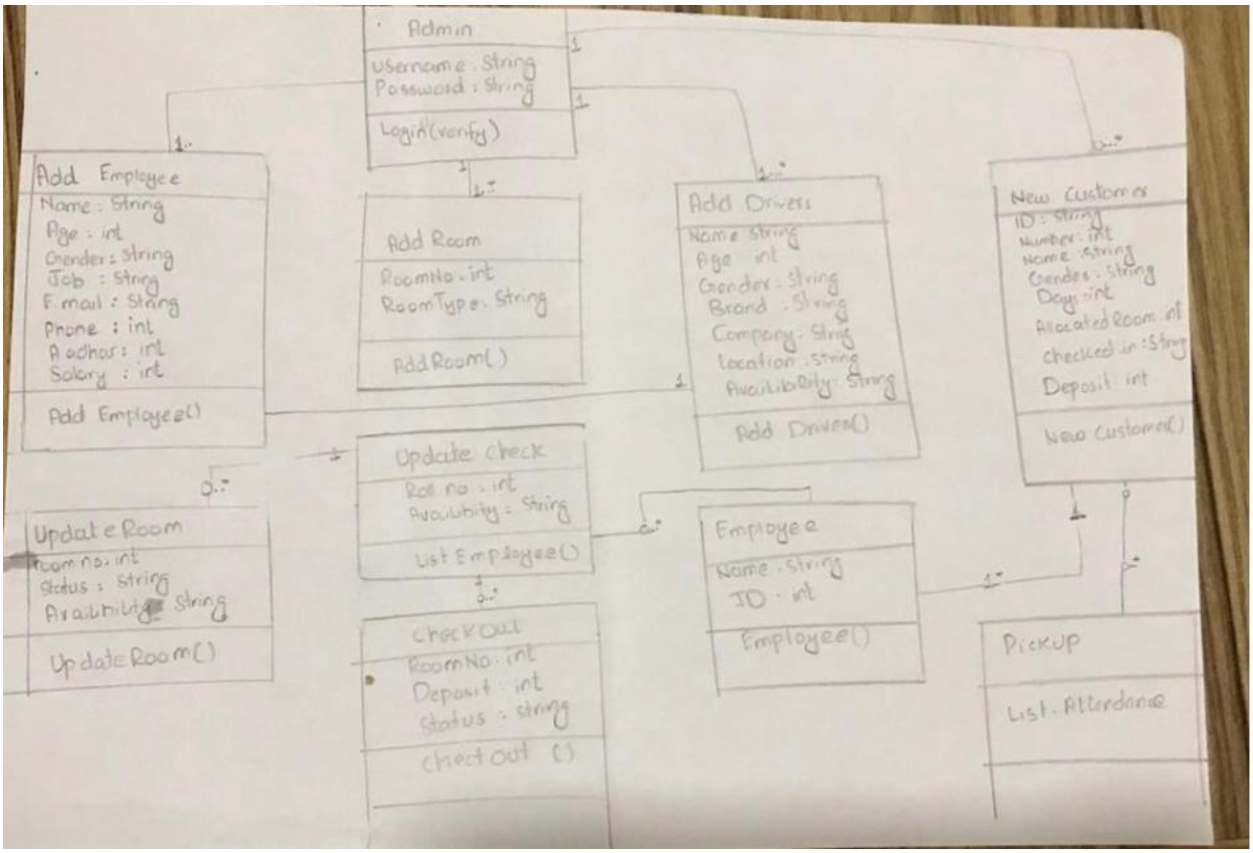
- **Draw ER-Diagram:**



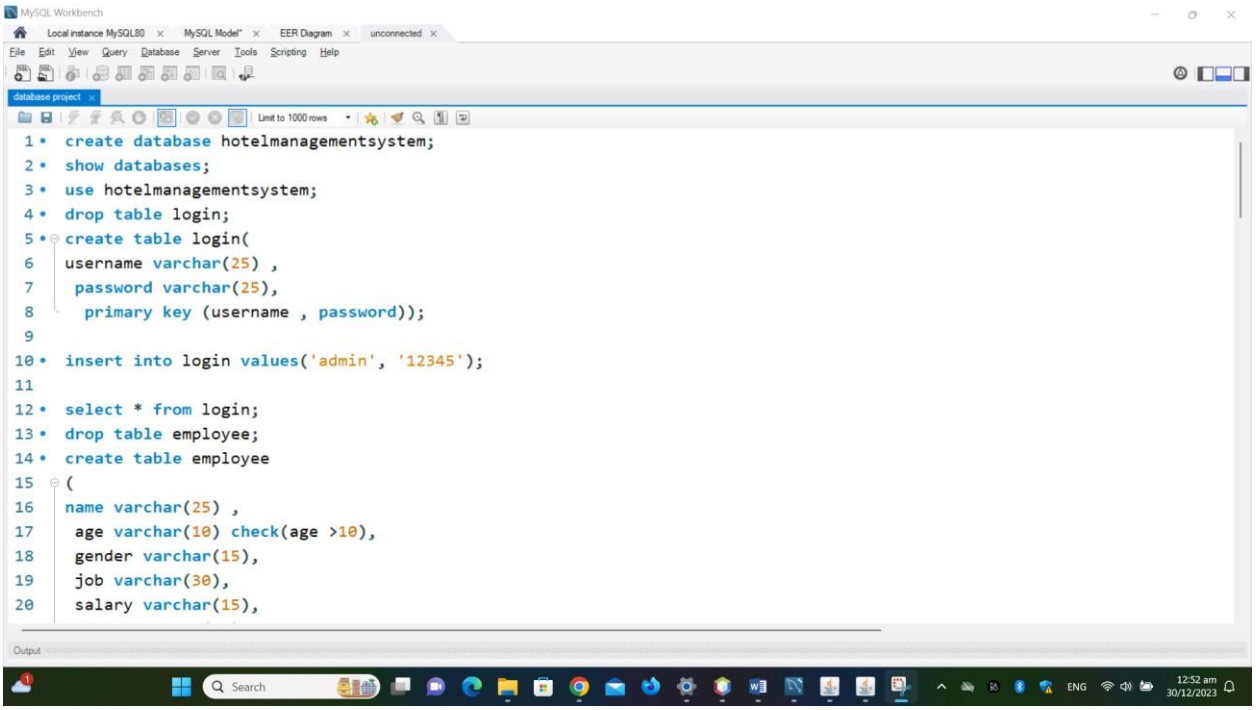
- **FLOWCHAR:**

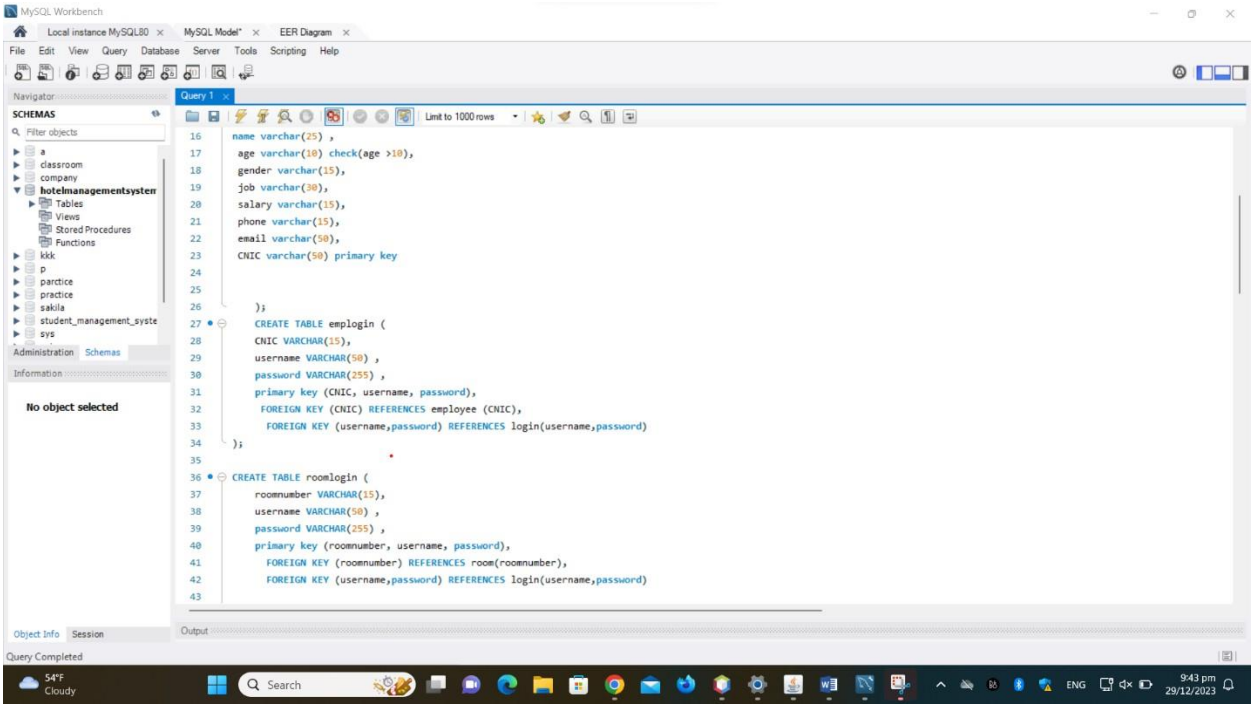


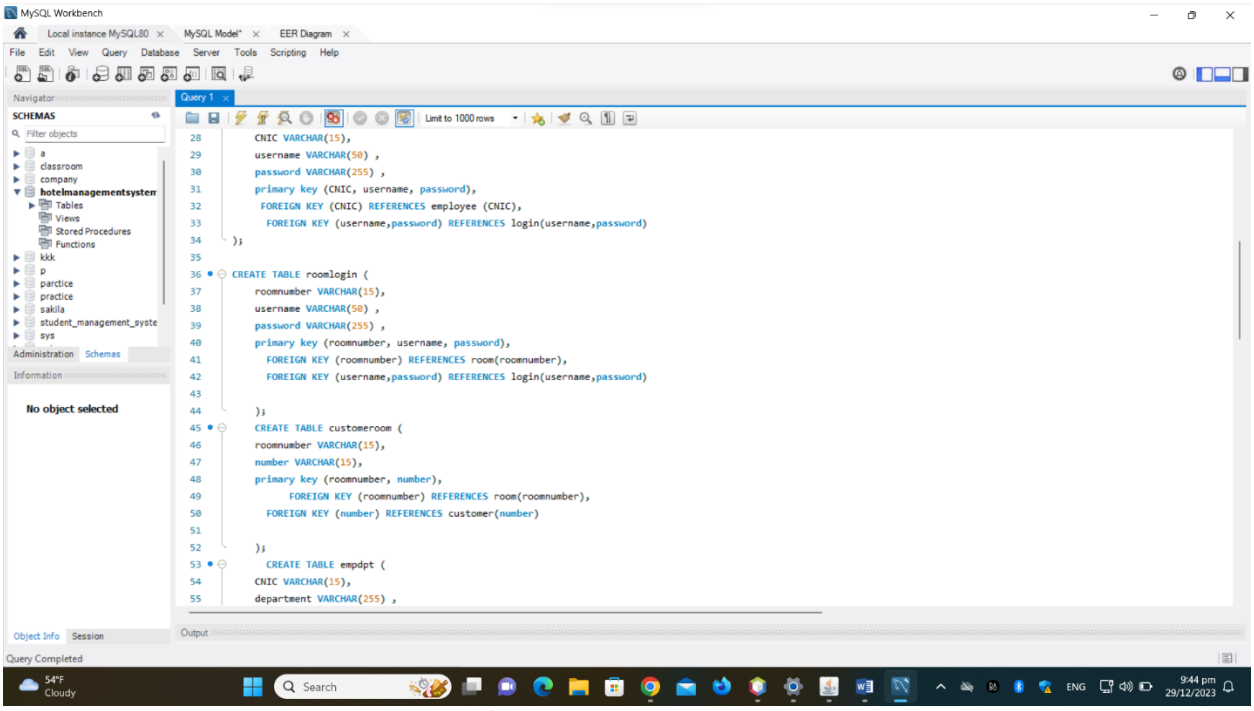
- **OVERVIEW AND COSTRUCTOR CALL TABLE :**

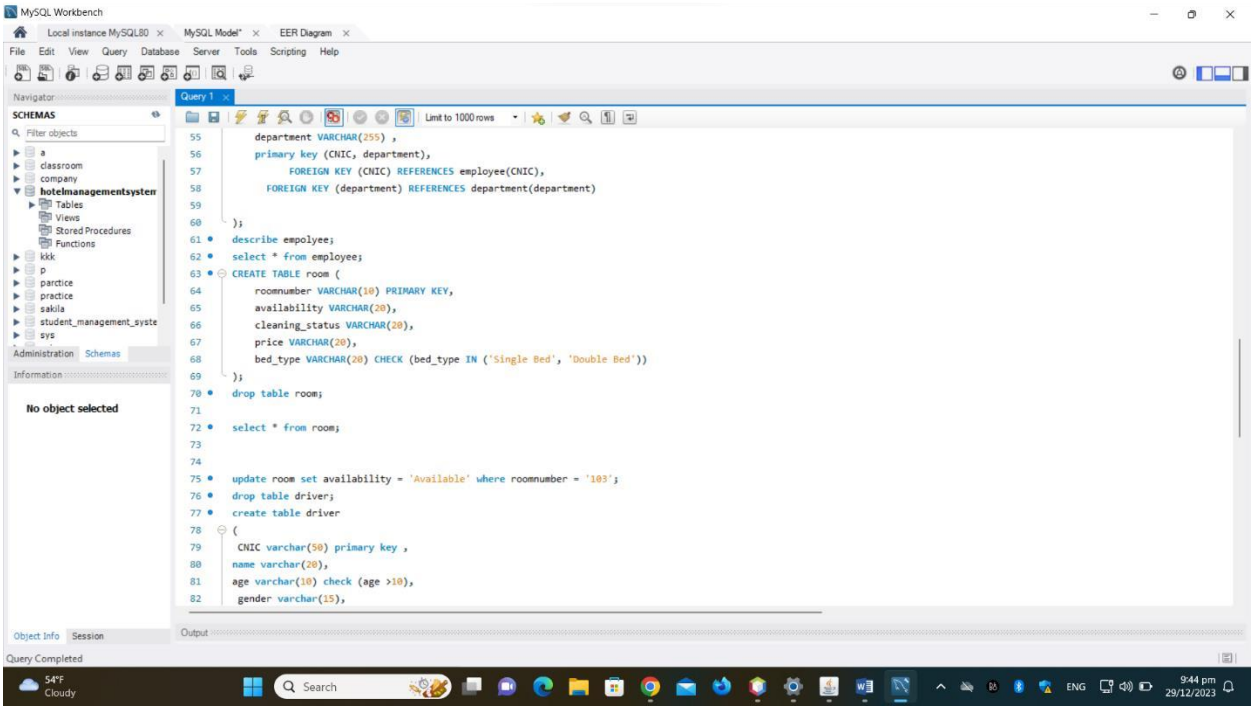


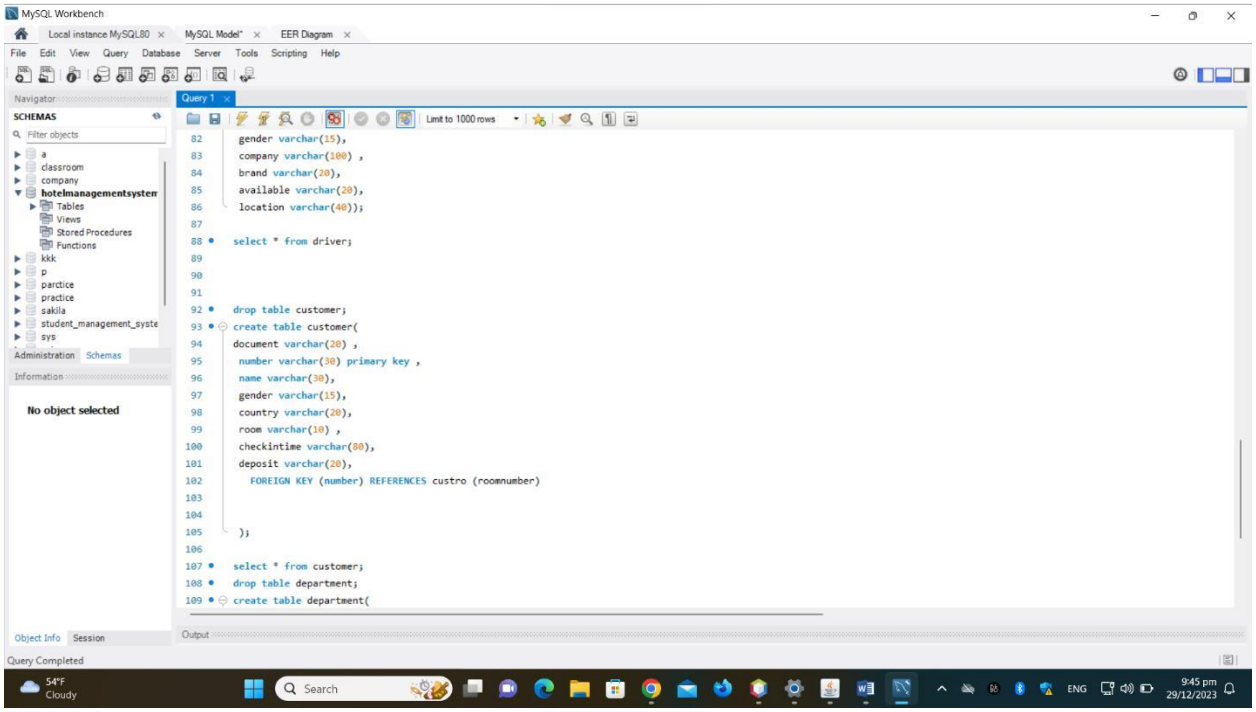
code of relation :

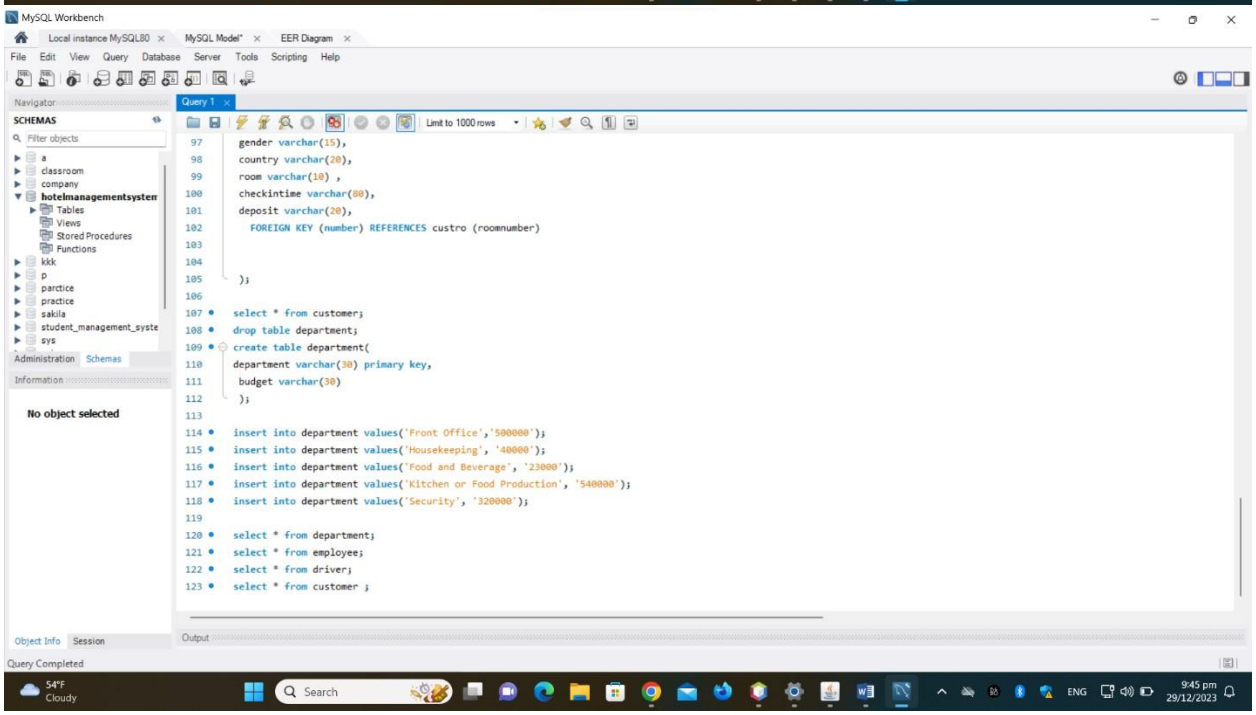



MySQL Workbench
Local instance MySQL80
MySQL Model*
EER Diagram
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
a
classroom
company
hotelmanagementsystem
Tables
Views
Stored Procedures
Functions
kkk
p
parctice
practice
sakila
student_management_syste
sys
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
name varchar(25) ,
age varchar(10) check(age >10),
gender varchar(15),
job varchar(30),
salary varchar(15),
phone varchar(15),
email varchar(50),
CNIC varchar(50) primary key
);
CREATE TABLE emplogin (
CNIC VARCHAR(15),
username VARCHAR(50) ,
password VARCHAR(255) ,
primary key (CNIC, username, password),
FOREIGN KEY (CNIC) REFERENCES employee (CNIC),
FOREIGN KEY (username,password) REFERENCES login(username,password)
);
CREATE TABLE roomlogin (
roomnumber VARCHAR(15),
username VARCHAR(50) ,
password VARCHAR(255) ,
primary key (roomnumber, username, password),
FOREIGN KEY (roomnumber) REFERENCES room(roomnumber),
FOREIGN KEY (username,password) REFERENCES login(username,password)
Output
Query Completed
54°F Cloudy
Search
ENG
9:43 pm 29/12/2023


MySQL Workbench
Local instance MySQL80
MySQL Model*
EER Diagram
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
a
classroom
company
hotelmanagementsystem
Tables
Views
Stored Procedures
Functions
kkk
p
parctice
practice
sakila
student_management_syste
sys
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
CNIC VARCHAR(15),
username VARCHAR(50) ,
password VARCHAR(255) ,
primary key (CNIC, username, password),
FOREIGN KEY (CNIC) REFERENCES employee (CNIC),
FOREIGN KEY (username,password) REFERENCES login(username,password)
);
CREATE TABLE roomlogin (
roomnumber VARCHAR(15),
username VARCHAR(50) ,
password VARCHAR(255) ,
primary key (roomnumber, username, password),
FOREIGN KEY (roomnumber) REFERENCES room(roomnumber),
FOREIGN KEY (username,password) REFERENCES login(username,password)
);
CREATE TABLE customeroom (
roomnumber VARCHAR(15),
number VARCHAR(15),
primary key (roomnumber, number),
FOREIGN KEY (roomnumber) REFERENCES room(roomnumber),
FOREIGN KEY (number) REFERENCES customer(number)
);
CREATE TABLE empdpt (
CNIC VARCHAR(15),
department VARCHAR(255) ,
Output
Query Completed
54°F Cloudy
Search
ENG
9:44 pm 29/12/2023

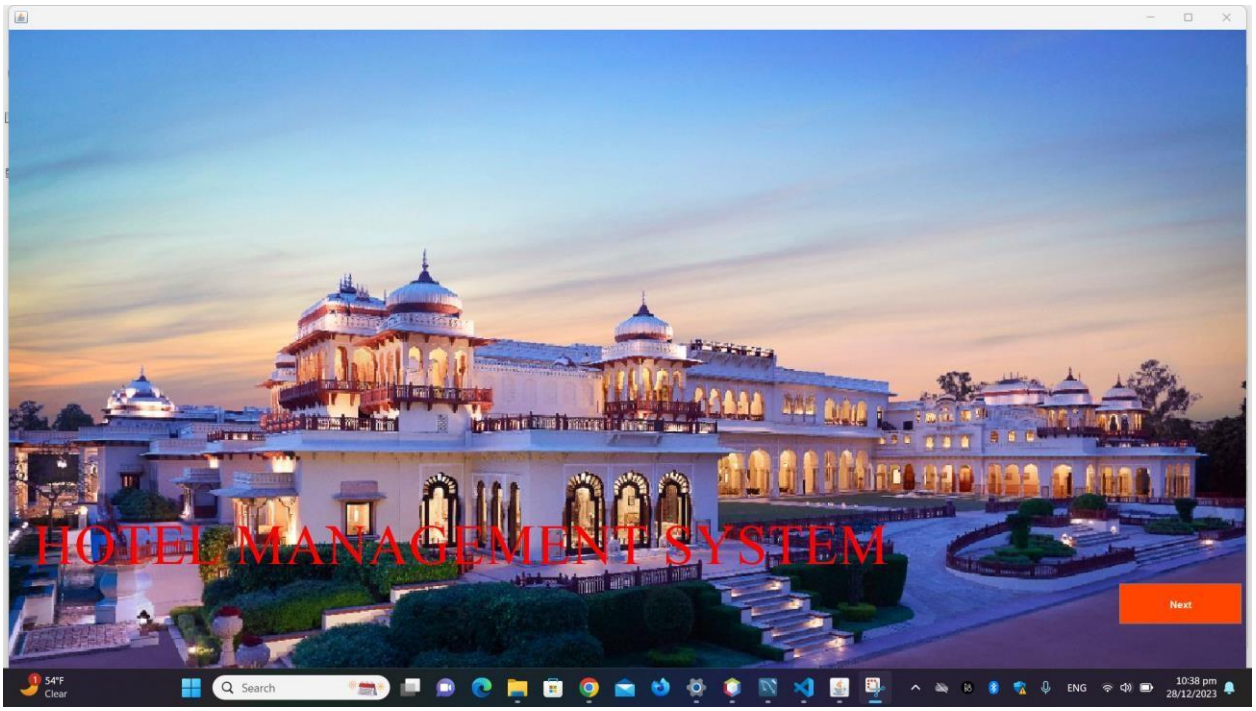

MySQL Workbench
Local instance MySQL80
MySQL Model*
EER Diagram
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
a
classroom
company
hotelmanagementsystem
Tables
Views
Stored Procedures
Functions
kkk
p
parctice
practice
sakila
student_management_syste
sys
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
department VARCHAR(255) ,
primary key (CNIC, department),
FOREIGN KEY (CNIC) REFERENCES employee(CNIC),
FOREIGN KEY (department) REFERENCES department(department)
);
describe empolyee;
select * from employee;
CREATE TABLE room (
roomnumber VARCHAR(10) PRIMARY KEY,
availability VARCHAR(20),
cleaning_status VARCHAR(20),
price VARCHAR(20),
bed_type VARCHAR(20) CHECK (bed_type IN ('Single Bed', 'Double Bed'))
);
drop table room;
select * from room;
update room set availability = 'Available' where roomnumber = '103';
drop table driver;
create table driver
(
CNIC varchar(50) primary key ,
name varchar(20),
age varchar(10) check (age >10),
gender varchar(15),
Output
Query Completed
54°F Cloudy
Search
ENG
9:44 pm 29/12/2023



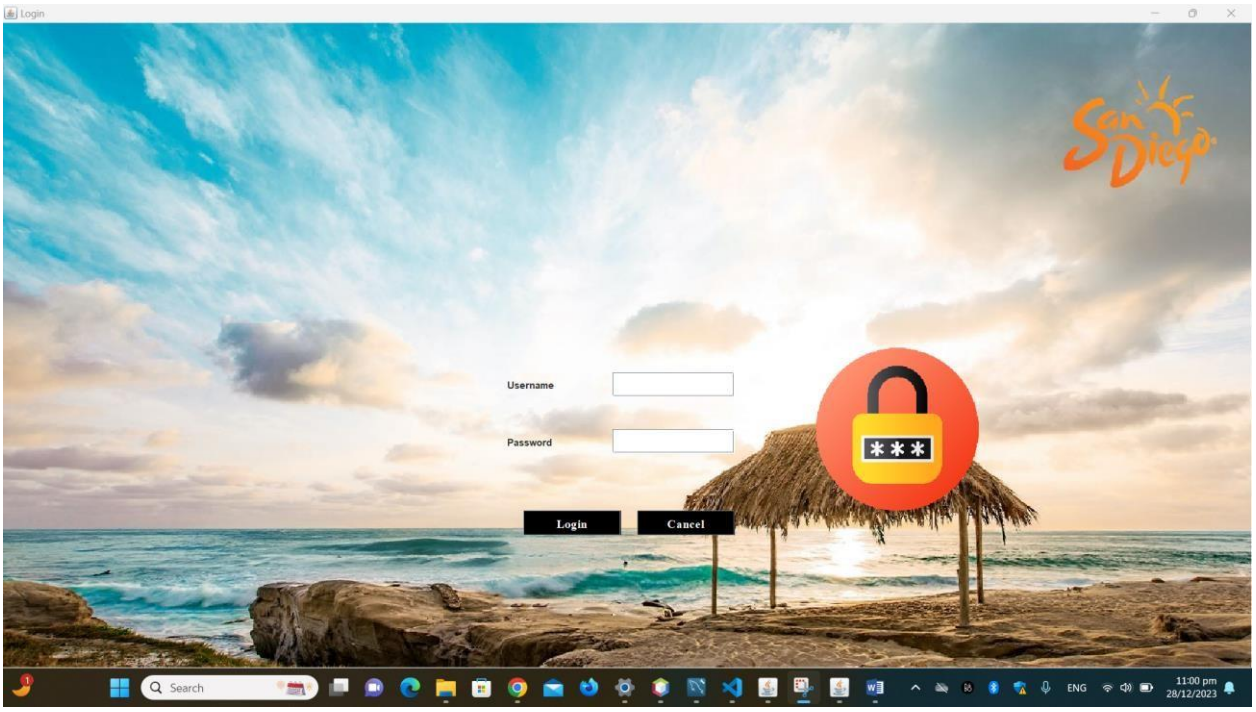


Screen shots :

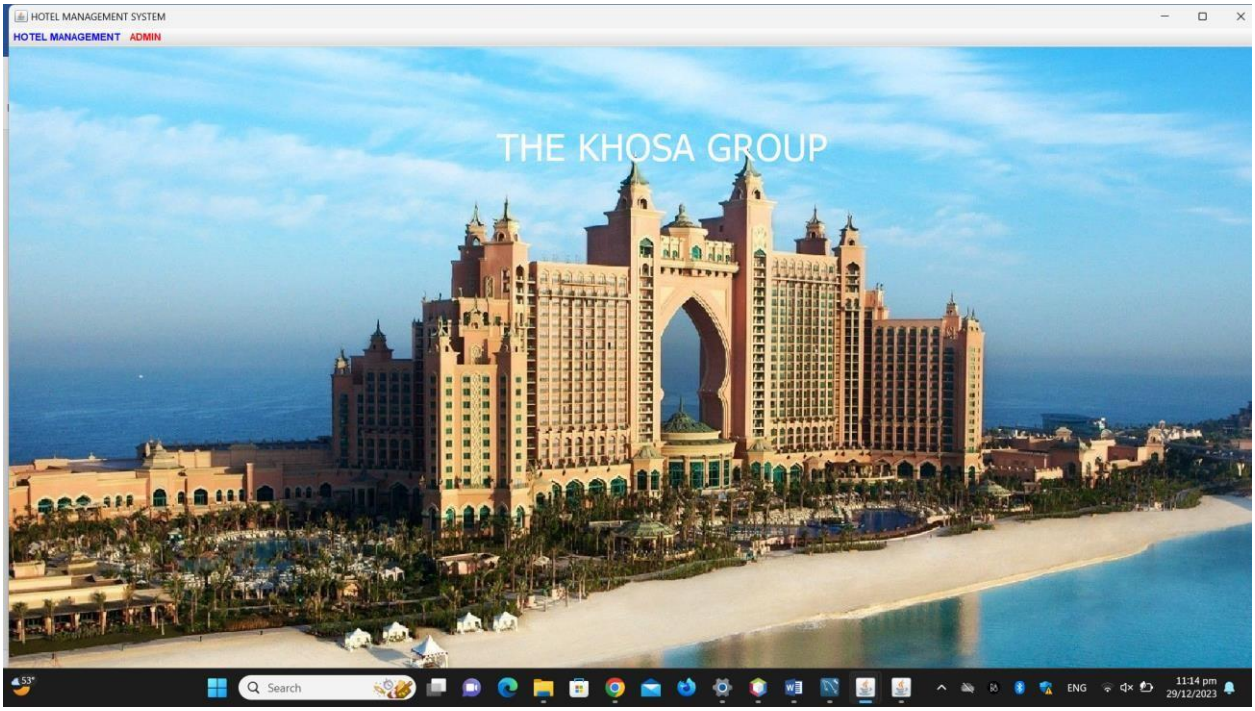
HOTEL MANGEMENT SYSTEM:


HOTEL MANAGEMENT SYSTEM
Next

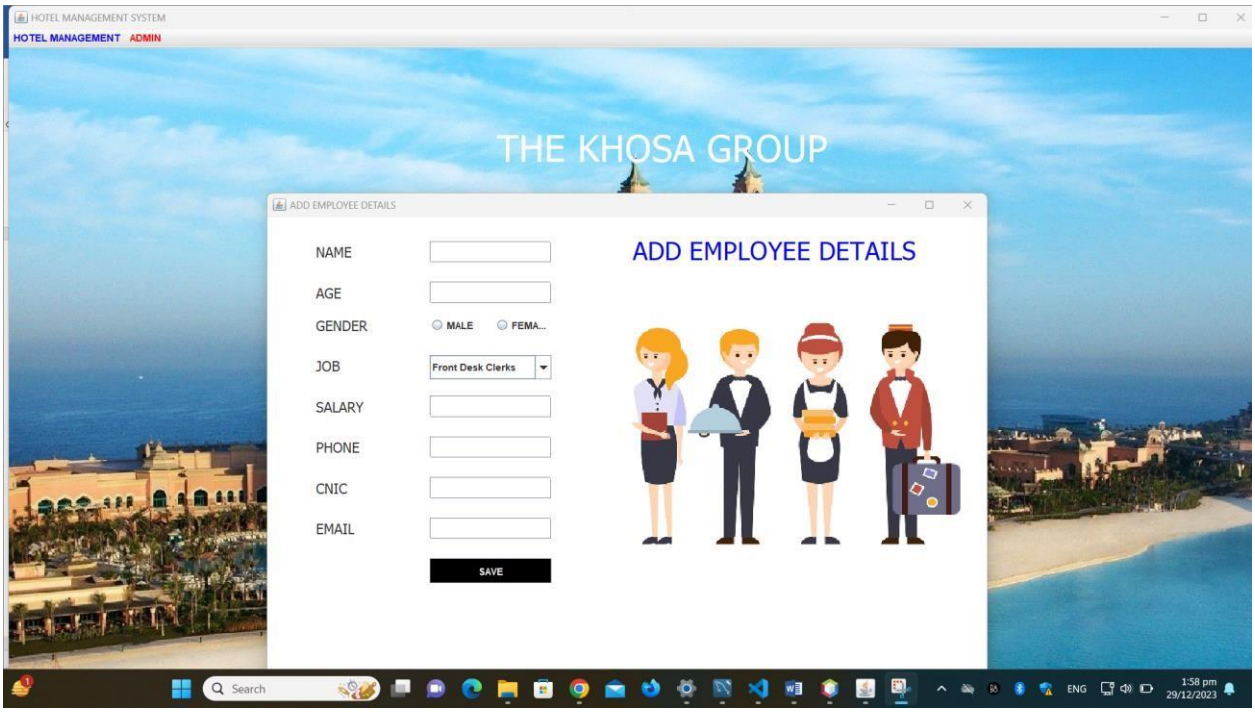
Login


Login
Username
Password
Login
Cancel

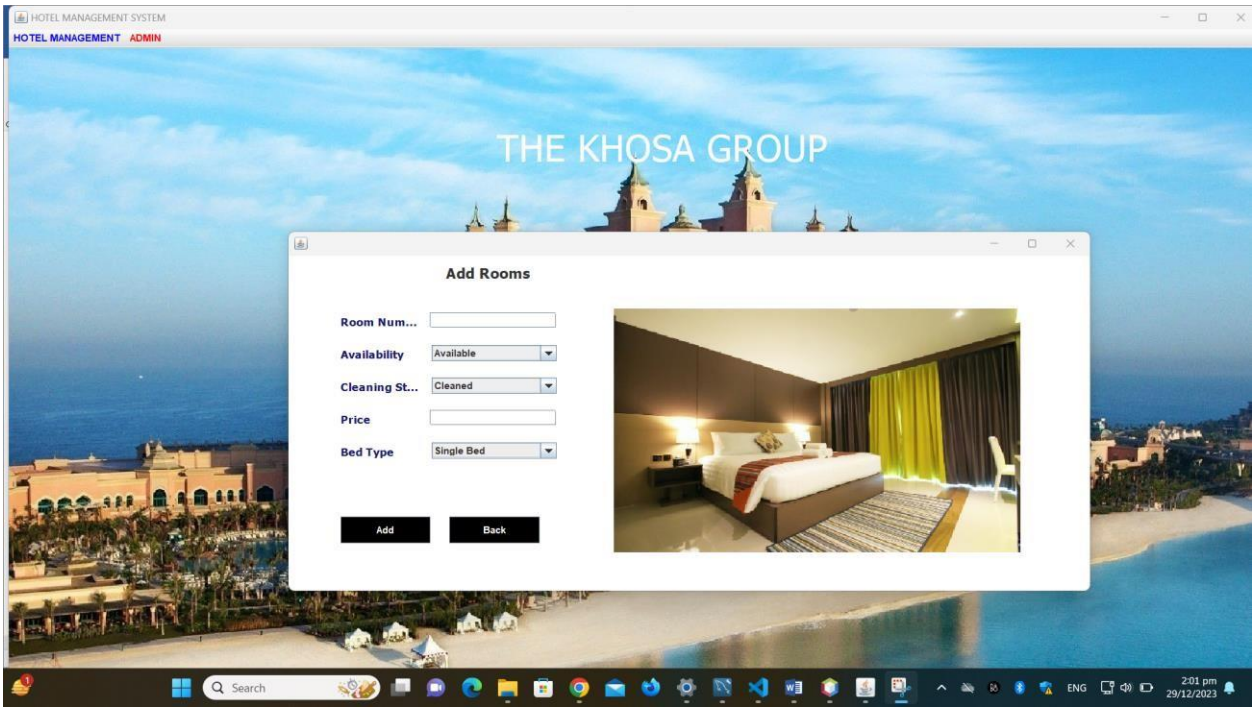
ADMIN & HOTELMANAGEMENT BUTTON



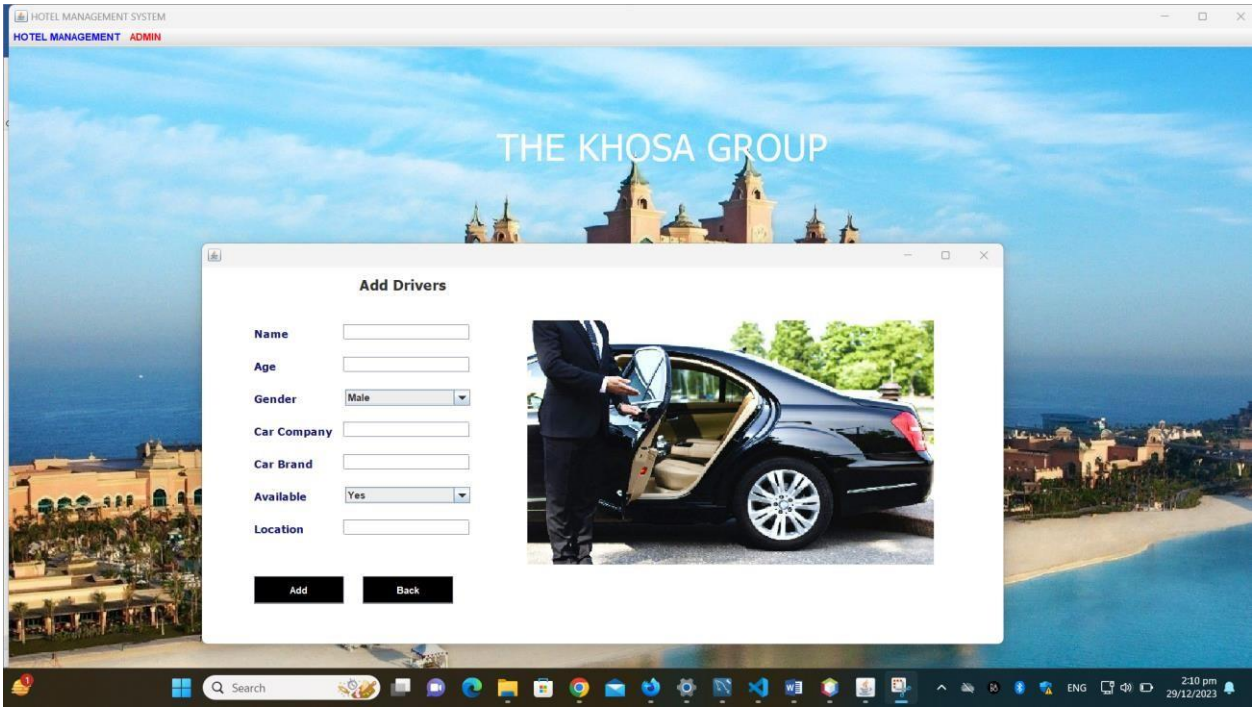
Add Empolyee



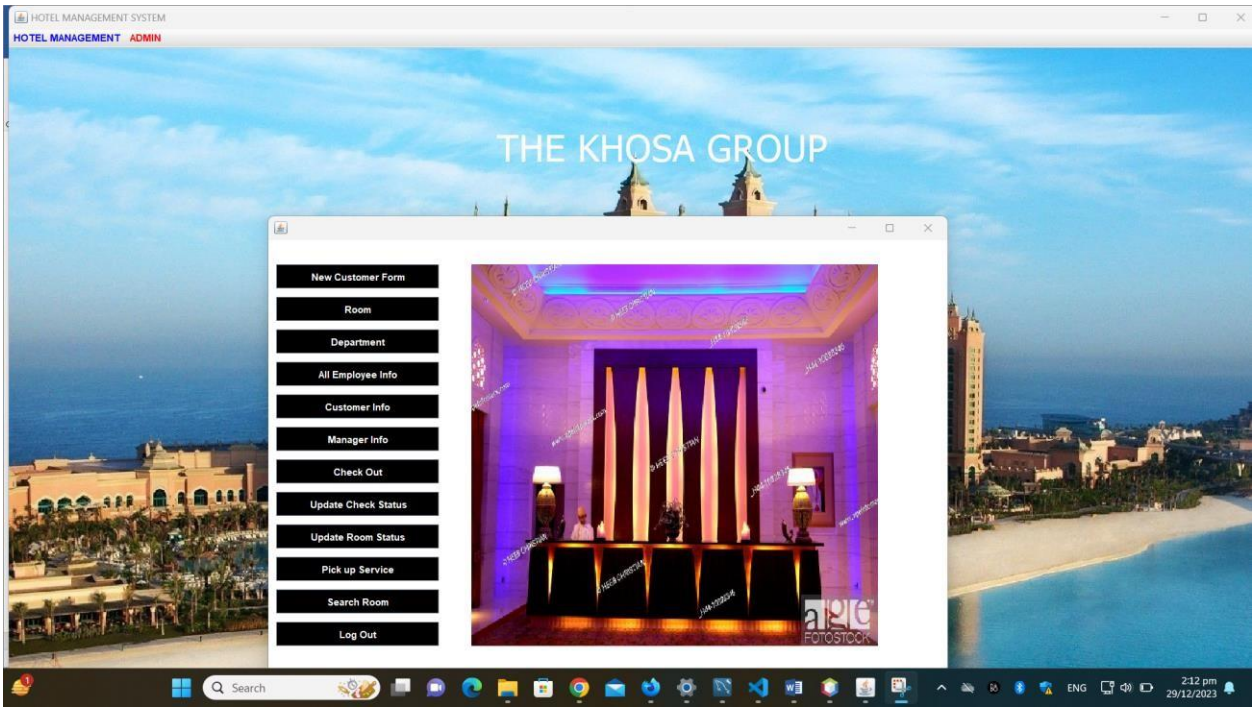
Add rooms



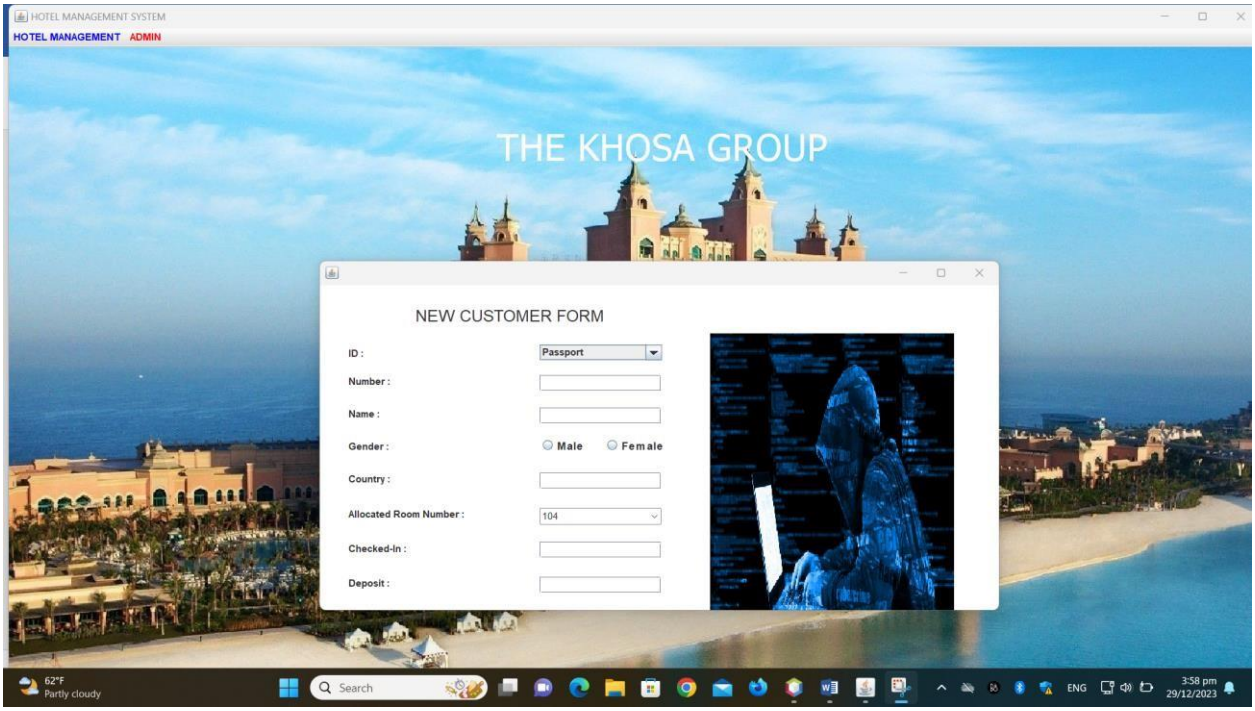
Add drivers



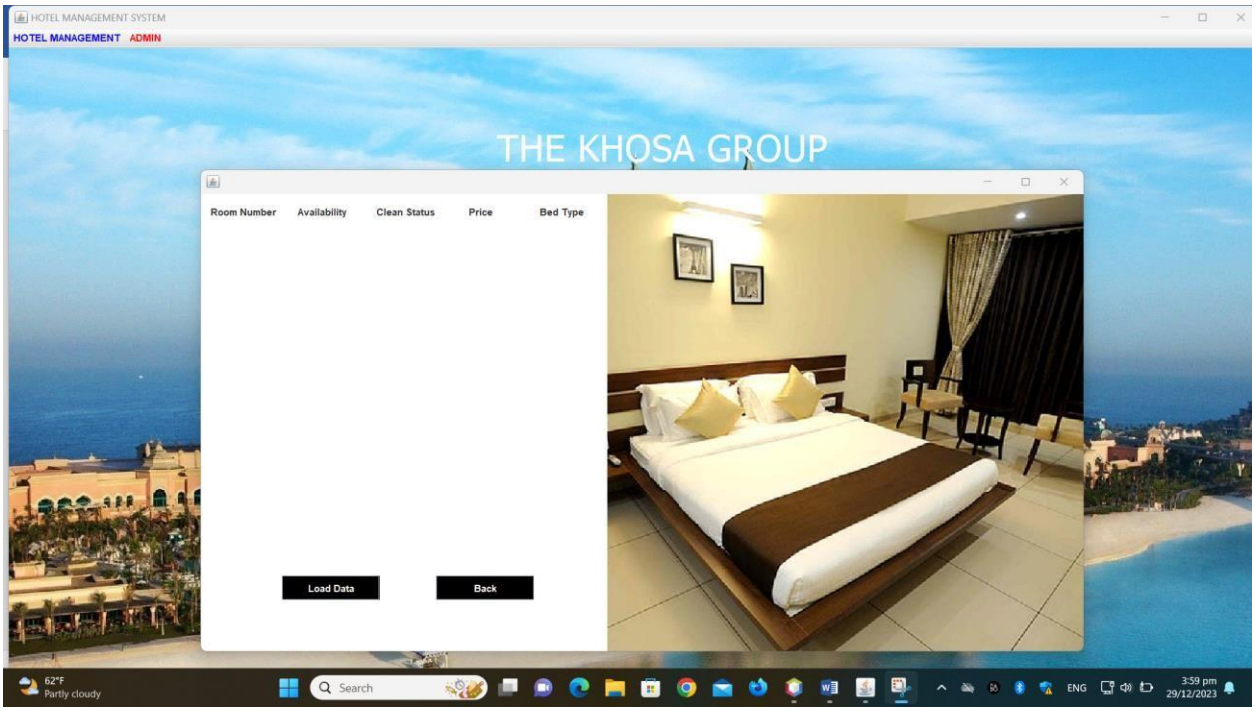
Reception



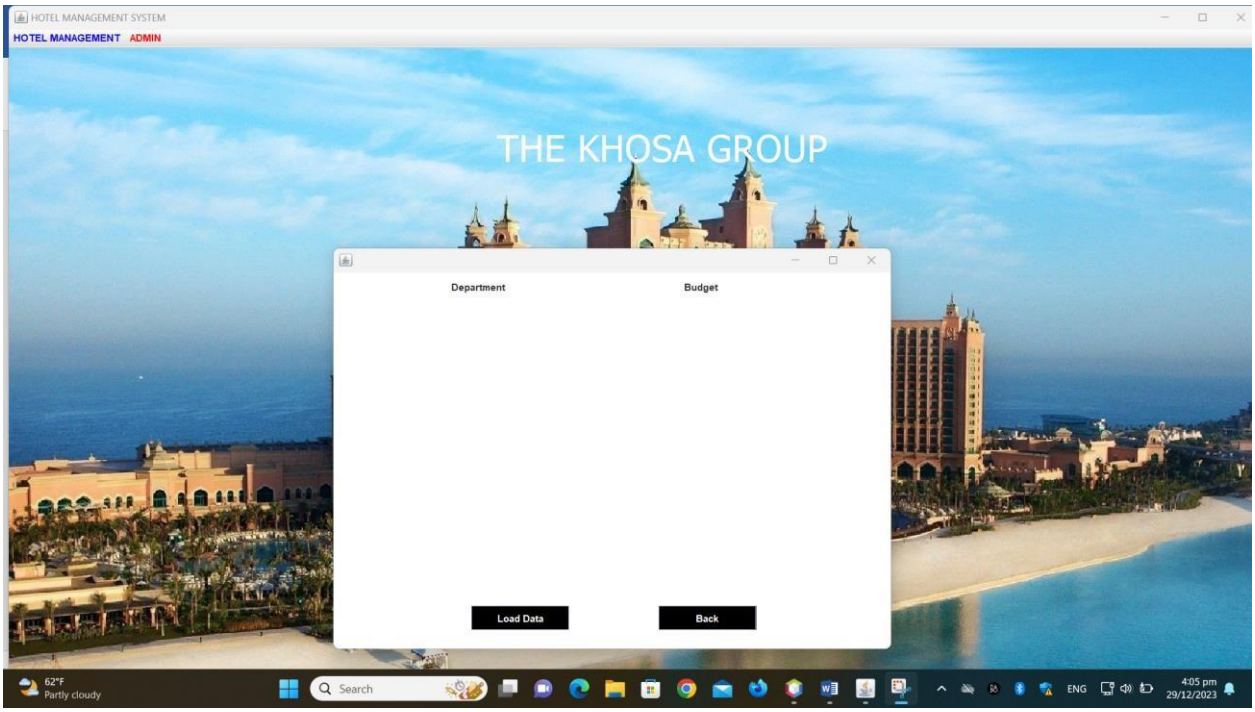
New customer form



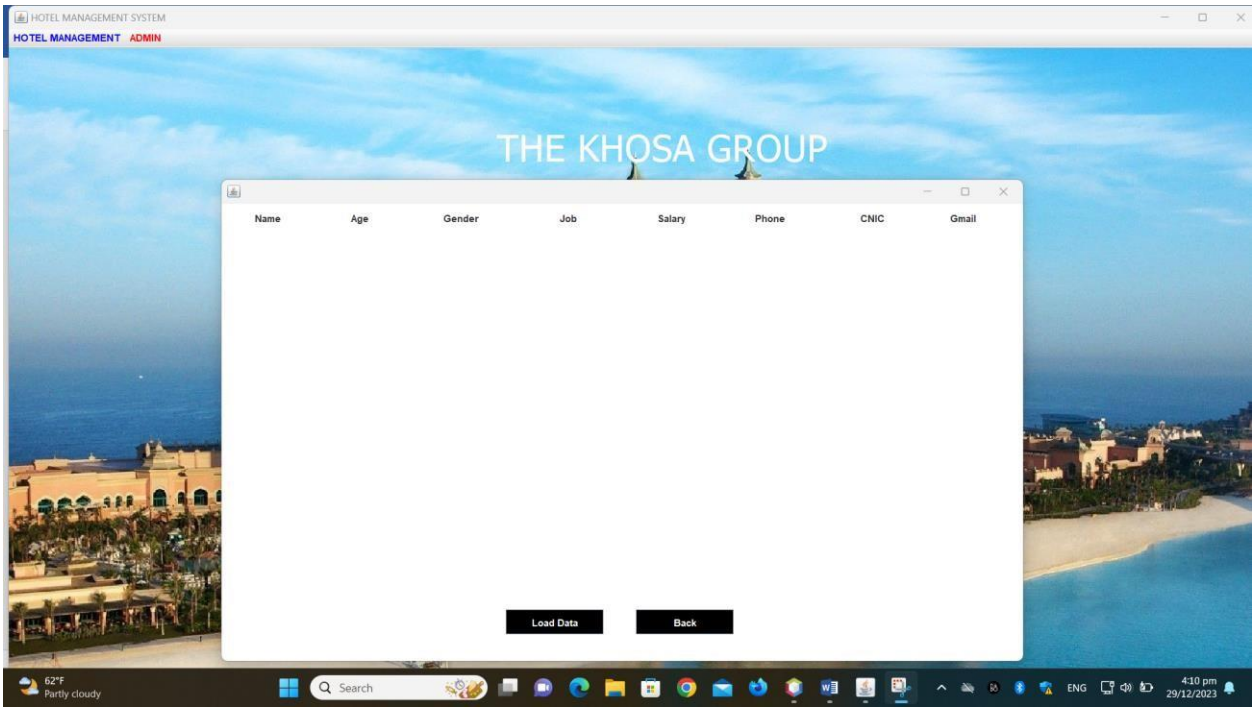
Room Information



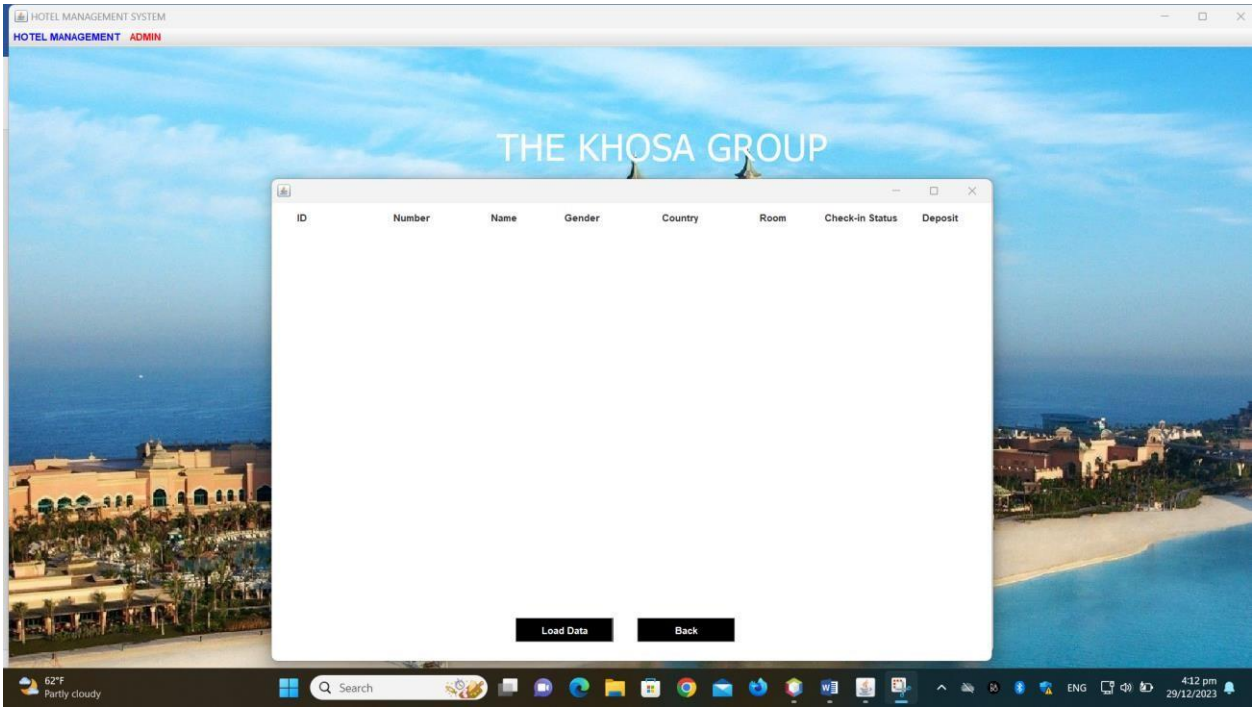
Department information and budget



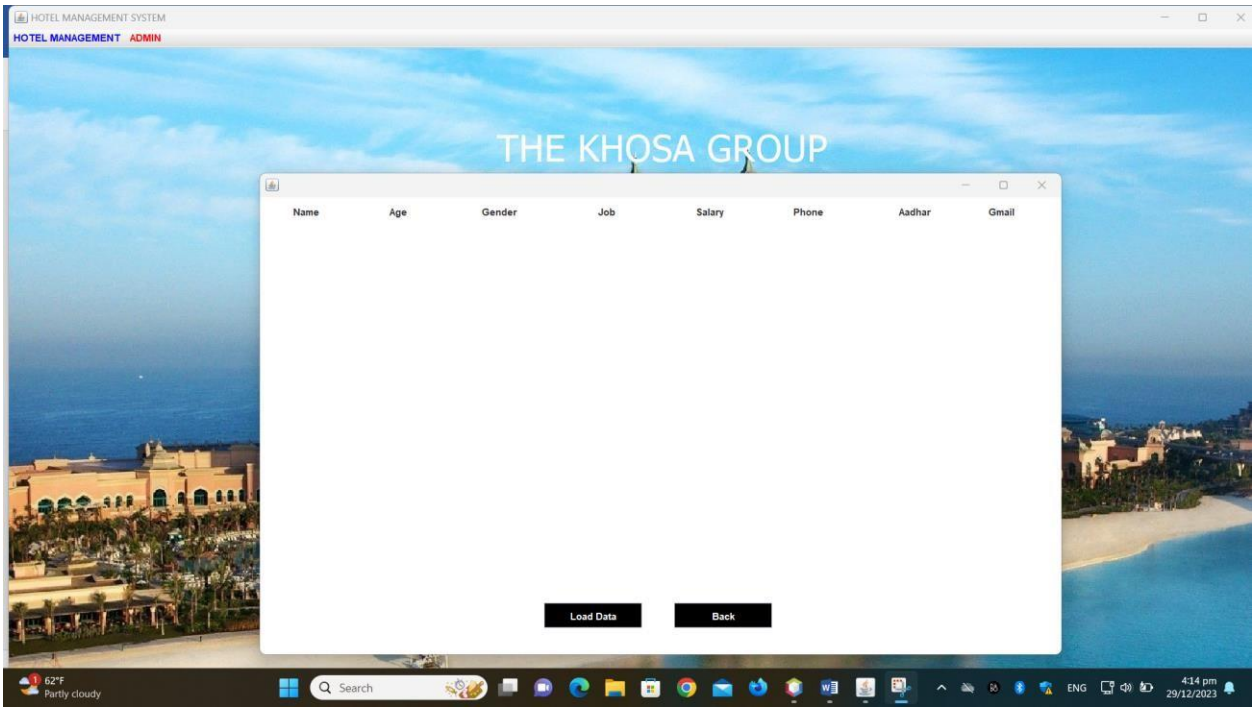
All employee info



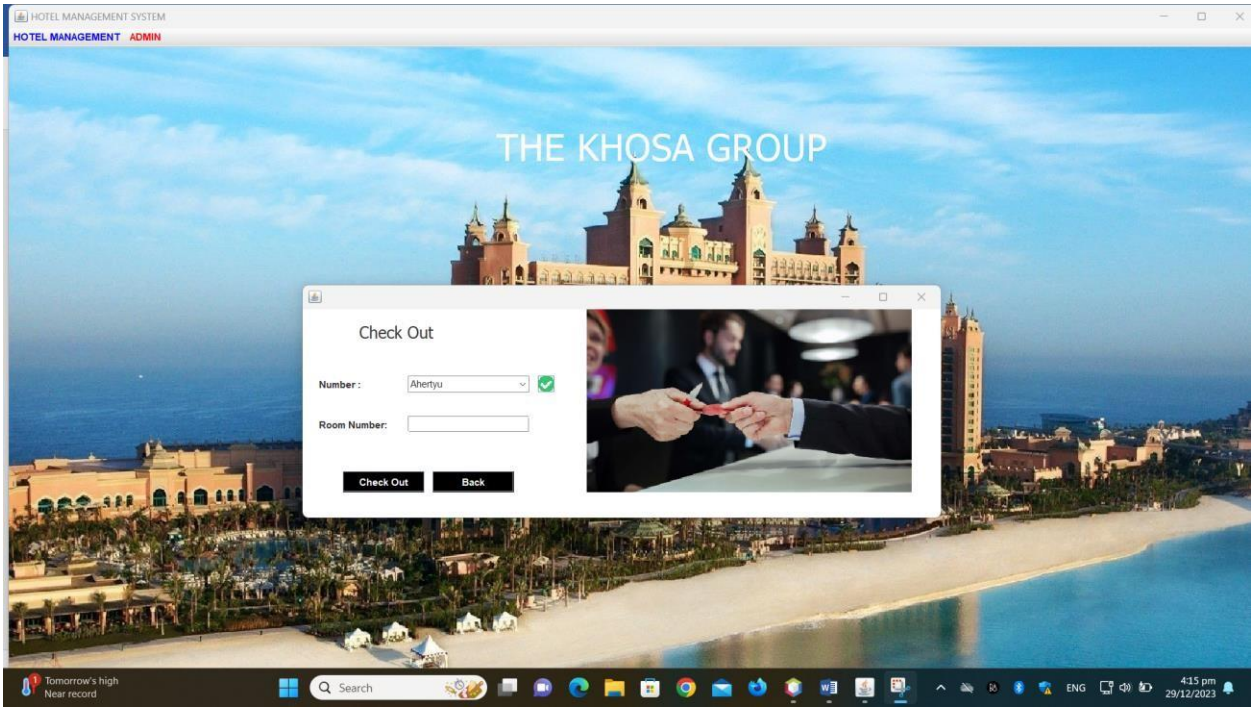
Customer info



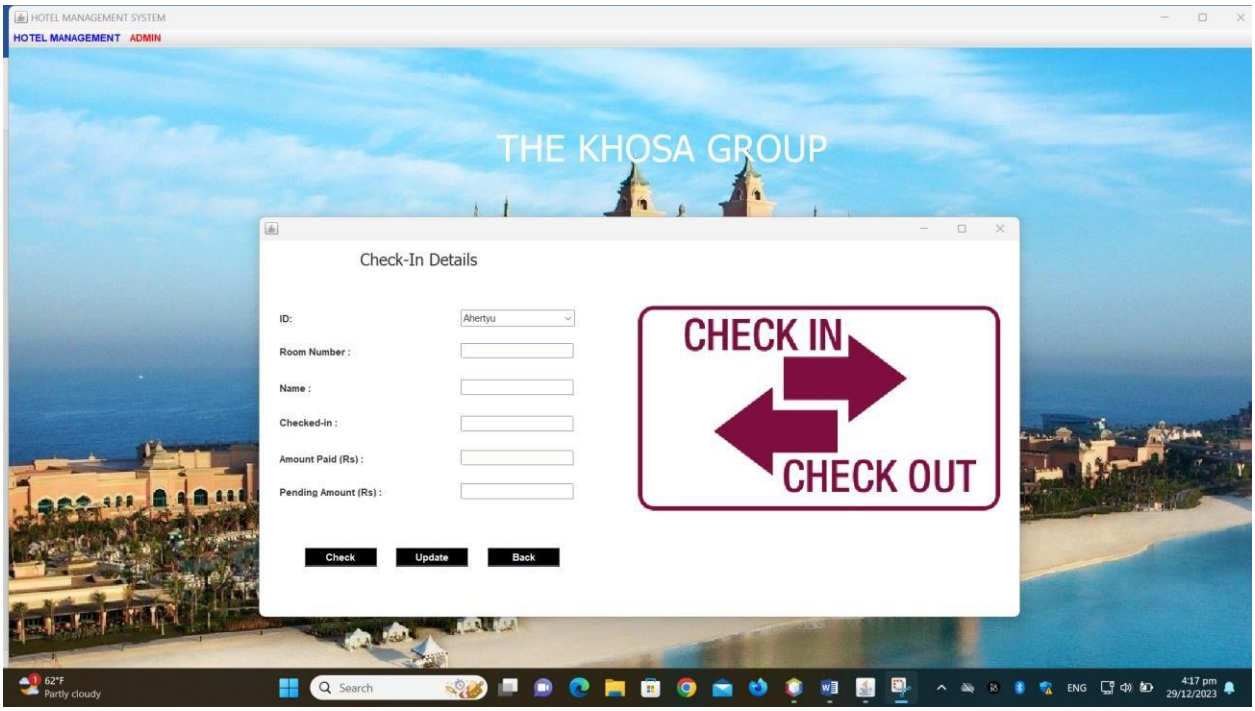
Manager info



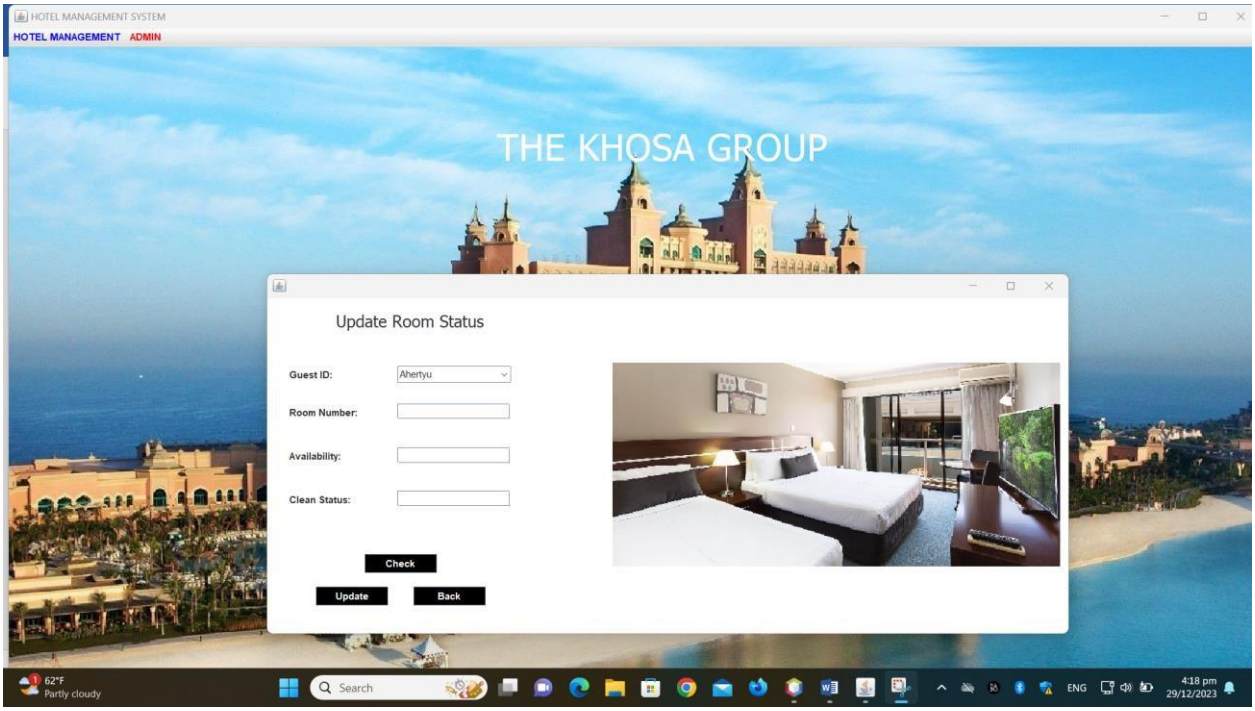
Check out



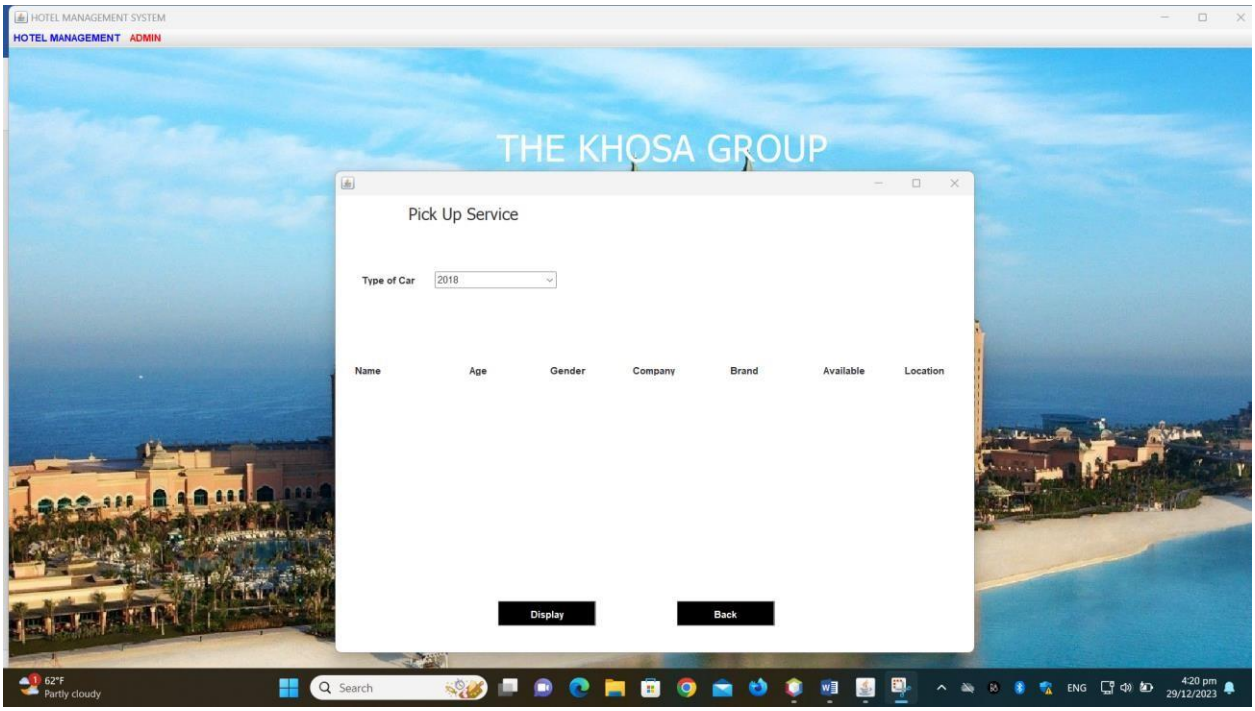
Update check status



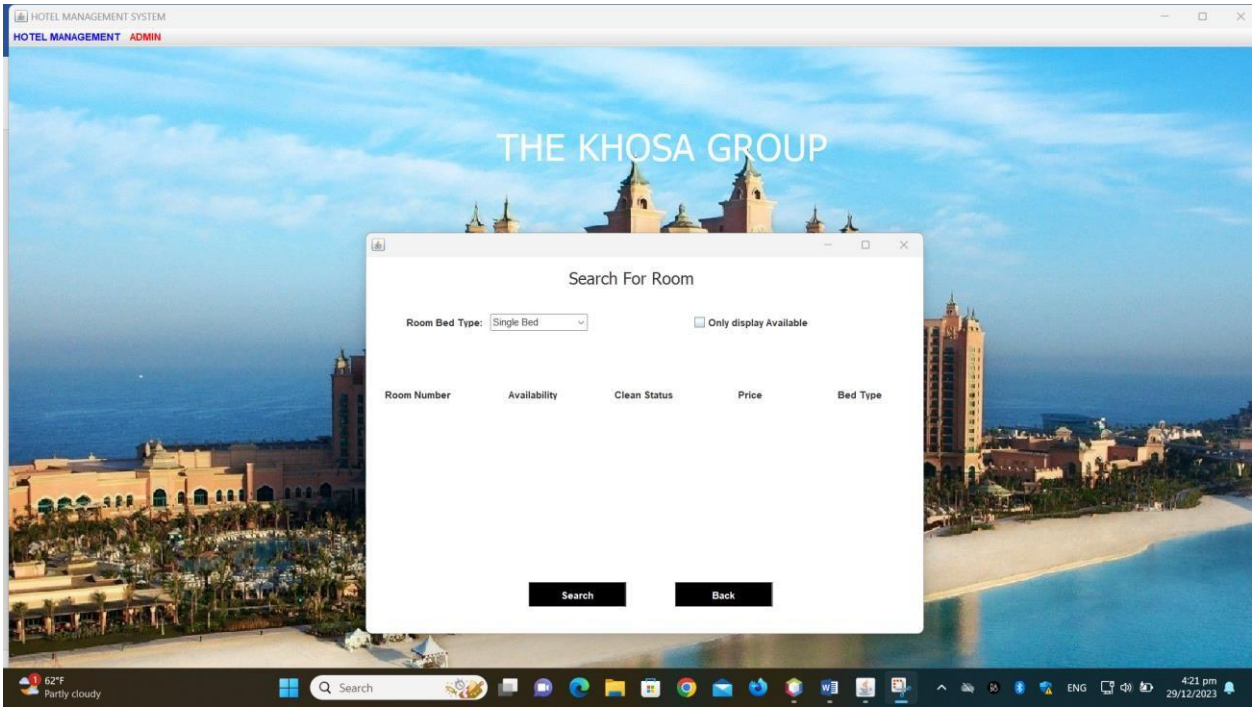
Update room status



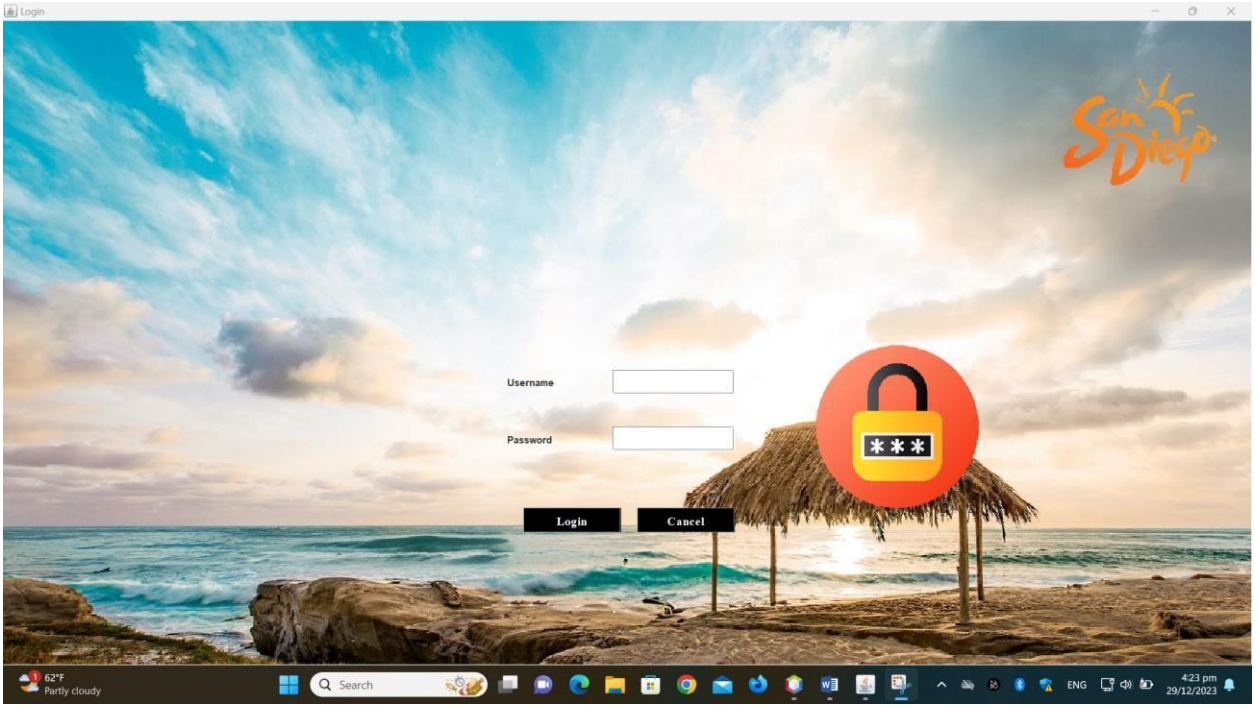
Pick up service



Search room



Log out



Functionality :

- **Outline the core functionality of your GUIbased desktop application:**

User Authentication:

Secure login with username and password.

Role-based access control.

Dashboard:

Key metrics and summaries.

Quick links for module navigation.

Customer Management:

Add, update, and delete customer details.

List all customers.

Room Management:

Add, update, and delete room details.

List all rooms.

Booking and Reservation:

Room booking based on availability.

Check-in, check-out, and modify bookings.

Employee Management:

Add, update, and delete employee details.

List all employees.

Front Desk Operations:

Manage check-ins, check-outs, and inquiries.

Real-time room status updates.

Data Maintenance:

Regular updates for data accuracy.

Backup and recovery mechanisms.

Logout:

Securely end user sessions.

- **Include details on data insertion, deletion, updating, and listing operations:**

User Authentication:

Implement a secure login system with a username and password.Include roles such as Administrator, Data Entry, Supervisory, and Decision Making.Restrict access based on user roles.

Dashboard:

Create an interactive dashboard displaying key metrics and summaries.Include quick links to major functionalities.

Provide a user-friendly interface for easy navigation.

Customer Management:

Insertion:

Allow the addition of new customer details, capturing personal information.

Deletion:

Implement functionality to remove customer records.

Updating:

Enable the modification of customer information.

Listing:

Display a list of all customers with relevant details.

Room Management:

Insertion:

Allow the addition of new room details, specifying room type and charges.

Deletion:

Provide an option to remove room records, managing room availability.

Updating:

Allow modification of room details like type and charges.

Listing:

Display a list of all rooms with their current status (occupied/vacant).

Booking and Reservation:

Room Booking:

Enable customers to book rooms based on availability.

Check-in/Check-out:

Track the status of guests, indicating if they are checked in or out.

Employee Management:

Insertion:

Allow the addition of new employee details, including role and contact information.

Deletion:

Provide an option to remove employee records from the system.

Updating:

Allow modification of employee information.

Listing:

Display a list of all employees with their roles.

Front Desk Operations:

Facilitate front desk activities, including check-ins, checkouts, and inquiries.Provide a user-friendly interface for front desk staff.

User Privilege and Security Control:

Actually I can do it because of exam I skipped for now .I tried but I stuck that's why I removed the roles , Define different user roles with specific privileges.

Ensure that unauthorized access is restricted.

Implement a secure password recovery mechanism.

.

Data Maintenance:

Regularly update and maintain data to ensure accuracy.

Login:

Users log in with their credentials.

Dashboard:

Overview of key metrics.

Navigation:

Access different modules (Customer, Room, Booking, Employee, etc.).

Data Operations:

Insert, delete, update, or list records in each module.

Logout:

Securely log out of the system.

- **Provide a high-level overview of the application workflow:**

User Authentication:

Users log in using their credentials (username and password).

The system authenticates the user and grants access based on their assigned role.

Dashboard:

Upon successful login, users are directed to a dashboard.

The dashboard provides an overview of key metrics, such as room occupancy, pending reservations, and financial summaries.

Quick links on the dashboard allow users to navigate to different modules.

Module Navigation:

Users can access various modules, such as Customer
Management, Room Management, Booking and Reservation, Employee Management, Front Desk
Operations, and Reports.

Customer Management:

Users can add new customer details, update existing information, and delete customer records.

A listing feature displays a comprehensive list of all customers.

Room Management:

Users can add new room details, update room information, and delete room records.

A room listing feature shows the current status (occupied/vacant) of each room.

Booking and Reservation:

Customers can book rooms based on availability.

The system handles reservations and tracks the check-in and check-out status of guests.

Users can modify or cancel bookings.

Employee Management:

Admins can add new employee details, update employee information, and remove employee records.

An employee listing feature displays the roles and contact information of all staff.

Front Desk Operations:

Front desk staff can manage check-ins, check-outs, and customer inquiriesReal-time updates on room status are available.

User Privilege and Security Control:

The system enforces user roles with specific privileges.

Admins can manage user accounts and roles.

Data Maintenance:

Regular data maintenance ensures the accuracy and reliability of information.

Backup and recovery mechanisms are in place to prevent data loss.

Logout:

Users can securely log out of the system, ending their session.

Overall Assessment:

- **Discuss the overall success of your term project:**

The term project, a Hotel Management System, has achieved overall success by effectively meeting its core objectives. The system's functionality, encompassing user authentication, room and customer management, booking processes, and reporting functionalities, has been successfully implemented with a user-friendly interface. The project demonstrated strong technical implementation, adhering to planned timelines and budgets. User feedback and iterative testing played a pivotal role in refining the user experience, resulting in a system that is both efficient and accessible. The team collaboration and project management aspects were effective, overcoming challenges and ensuring a cohesive development process. Looking forward, there is recognition of areas for potential future improvements, reflecting a proactive approach to ongoing development. The documentation is comprehensive,

providing a solid foundation for future maintenance and enhancements. Overall, the project's success is underscored by its functional prowess, positive user experience, and adherence to project management best practices**.**

- **Reflect on challenges faced, lessons learned, and improvements that could be made in future iterations.**

The world is changing rapidly and so is the meaning of the Hotel Management System. Today hotel management is not only confined to hotels but has gone deep into tourism, catering, clubs, etc making it a very paying and an exciting career option.

With the rapid growth of the hotel industry pushed forward by foreign and domestic tourism and business travels, the demand for well trained and quality personnel too has gone up high. India is one of the preferred tourist and travel destinations. Approx 4.4 million tourists visit our country every year. The growth of 20% has been recorded in the tourist and hospitality industry over a few years and more growth is expected in coming years. At present, there are about 200 millions of jobs available in the industry, out of which 20% of the job opportunities are in India.

The Hotel Management System has a lot of enhancement options. In future more features may be added category-wise. It may try to analyze the user behaviour and preferences and accordingly suggest. All concepts can be applied to make the Hotel Management System more efficient.

- **Consider the project's impact on achieving the initial objectives:**

The project, centered around the development of a Hotel Management System, has demonstrated a significant impact in achieving its initial objectives. The system effectively addresses the core goals set at the project's outset, providing comprehensive solutions for customer management, room bookings, and employee oversight within a hotel environment. The successful implementation of user authentication ensures a secure and rolespecific access control mechanism. The project's emphasis on a user-friendly interface facilitates easy navigation and seamless interaction, contributing to a positive user experience. The booking and reservation functionalities align with the objective of streamlining hotel operations, while the robust reporting system aids in decisionmaking processes. The impact is notable in the efficiency gains achieved by automating manual tasks, leading to improved accuracy and timely management of hotel resources. Overall, the project has played a pivotal role in realizing its initial objectives by delivering a functional, efficient, and user-centric Hotel Management System.

Conclusion:

- **Summarize the key findings, achievements, and challenges encountered during the development of your GUI-based desktop application:**

In this project I have tried my best to make user friendly software. This software can be handled by any person who has little bit of idea of computers. In this software I have tried to meet most of the requirements of the present hotel management system including maintaining details of customers, rooms, employees and drivers. I also keep provision to update the details of customers, rooms and employees. In my effort I have tried to make my software all the more user friendly but there may some features which I would like to include in my continuous attempts