MyResort

A RESORT BOOKING PLATFORM

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

(MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

ABIN SIMON
Reg No : 24PMC103

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2024

MyResort

A RESORT BOOKING PLATFORM

**A PROJECT REPORT SUBMITTED IN PARTIAL
FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE**

MASTER OF COMPUTER APPLICATIONS

(MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

ABIN SIMON

Reg No : 24PMC103

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2024

A Project Report on

MyResort

A RESORT BOOKING PLATFORM

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

ABIN SIMON

Reg No : 24PMC103

Under the guidance of

Mr Robins A Kattoor
Associate Professor
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2024

PG DEPARTMENT OF COMPUTER APPLICATIONS

Marian College Kuttikkanam Autonomous

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

MyRESORT

is a bonafide record of work done by

ABIN SIMON

Reg. No : 24PMC103

In partial fulfillment of the requirements for the award of Degree of

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2024-2025

Mr Robins A Kattoor
Associate Professor
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

Mr Win Mathew John
Head of the Department
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

External Examiner

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the Almighty for granting me the opportunity to complete my project successfully. It gives me immense pleasure to present this report, and I am indebted to several individuals who have played pivotal roles in this journey.

I would like to express my heartfelt thanks to **Mr Win Mathew John,** Head of the department (MCA), for his support in completing this project successfully.

I extend my heartfelt thanks to the project guide, **Mr Robins A Kattoor,** Associate Professor at PG Department of Computer Application, Marian College Kuttikkanam for her valuable support in completing this project work successfully.

I must also thank other faculty members at the PG Department of Computer Application Studies for their support and encouragement in completing the study successfully.

In conclusion, my heartfelt thanks to everyone who has been a part of this journey, directly or indirectly. Your support and encouragement have been invaluable, and I am truly grateful for the opportunity to undertake this project in such a nurturing academic environment.

ABIN SIMON

ABSTRACT OF MyRESORT

MyResort is a modern resort management system designed to streamline the hospitality experience for both guests and administrators. The platform enables users to seamlessly book rooms, order food, and manage their stay through an intuitive digital interface. Built with the Django framework, MyResort combines a robust backend with a user-friendly frontend to ensure efficient resort operations, secure bookings, and scalable management solutions. The system offers comprehensive features including user registration and authentication, room availability management, integrated food ordering, and automated booking confirmations. Guests can easily browse available rooms, check real-time availability, make reservations, and place food orders through their accounts. The platform maintains detailed booking records and provides guests with access to their reservation history, current bookings, and food order status through a personalized dashboard.

For administrators, MyResort delivers a powerful management dashboard that enables efficient oversight of all resort operations. The admin interface facilitates room management, booking processing, food order tracking, and revenue monitoring. Administrators can view occupancy rates, manage room status, process food orders, and generate detailed revenue reports separating room bookings and food service income. MyResort aims to revolutionize traditional resort management by providing a digital solution that enhances guest satisfaction while optimizing operational efficiency. By focusing on automation, user experience, and administrative control, MyResort is positioned to become a comprehensive solution for modern resort operations, offering a seamless experience for guests while providing powerful management tools for staff. The system represents a significant step forward in digitalizing resort operations, making it an ideal solution for properties seeking to modernize their guest services and management processes.

TABLE INDEX

TABLE INDEX

FIGURE INDEX

Figure Page No.

1. INTRODUCTION

1.INTRODUCTION

1.1 PROBLEM STATEMENT

The traditional resort management processes rely heavily on manual booking, paper-based records, and in-person interactions, leading to inefficiencies, booking conflicts, and revenue tracking challenges. There's a need for a digital solution that automates room reservations, food ordering, and payment processing while providing real-time availability updates. The system should enable customers to book rooms online, order food and allow administrators to track revenue streams, manage bookings, and generate reports, ultimately enhancing both customer experience and operational efficiency.

1.2 PROPOSED SYSTEM

MyResort is a web-based resort management system that digitalizes the entire booking process. It features a user-friendly interface for room reservations, integrated food ordering, and secure online payments. Customers can view room availability, make bookings, and place food orders in real-time through the platform. Administrators get a comprehensive dashboard to monitor room occupancy, food orders, and separate revenue streams. The system generates invoices and provides detailed business analytics for informed decision-making.

1.3 FEATURES OF THE PROPOSED SYSTEM

1. **Registration and Authentication**: The system provides secure user registration and login functionality. Customers can create an account using their email and password, which are encrypted for data security.
2. **Room Management**: Customers can view and book different types of rooms with real-time availability status and pricing information.
3. **Food Order System**: Guests can browse the resort's food menu and place orders, maintaining records of all orders with their status and room numbers.
4. **Booking Management:** Users can make room reservations by selecting dates and receive automatic confirmation.
5. **Admin Dashboard:** Administrators can monitor all resort operations including room bookings, food orders, and separate revenue streams through a comprehensive dashboard. The interface manages room availability, and generates reports for both room bookings and food service operations.

2. FUNCTIONAL REQUIREMENTS

2. FUNCTIONAL REQUIREMENTS

1. Admin Dashboard Management

- Admin can view and manage all room bookings and food orders through the dashboard.
- Admin can monitor separate revenue streams for rooms and food orders.
- Admin has access to view all customer details and booking history.
- Admin can update room availability status and pricing

2. User Registration and Login

- Users can create an account by registering with necessary details like name, email, and password.
- Registered users can log in securely using their credentials.

3. Room Booking System

- The system displays a list of rooms with detailed information, including amenities and prices.
- Users can check and book room availability for specific dates.
- System automatically prevents double bookings for the same dates.
- Users can add pool access to their room booking

4. Food Order Management

- Users can browse the food menu and place orders through their account.
- Users can view their food order history.
- Admin can update the status of food orders

5. *Booking Processing and History*

- Admin can manage customer bookings, including viewing and updating booking status.
- Users can view their complete booking history with room and food order details.

6. *Revenue Tracking*

- System automatically calculates and displays separate revenue for rooms and food.
- Admin can view monthly revenue statistics.
- System maintains records of all financial transactions.

3. NON-FUNCTIONAL REQUIREMENTS

3. NON-FUNCTIONAL REQUIREMENTS

1. *Performance*

- The system can handle a large number of concurrent users without performance degradation.
- Page load times is optimized to ensure fast browsing, aiming for a maximum load time of 5 seconds.

2. *Scalability*

- The platform is scalable, allowing the addition of new products and users without requiring major system reworks.
- The system can support increased traffic as the platform grows, ensuring stability even with a large user base.

3. *Security*

- User data, including personal and payment information, must be encrypted and stored securely.
- The platform is implemented with secure authentication mechanisms

4. *Usability*

- The platform has an intuitive and easy-to-use interface, ensuring that both admin and customers can navigate it without difficulty.
- Clear instructions and helpful tooltips is provided to guide users through registration, room booking, pool access and food order.

5. *Availability*

- The platform is available 24/7, with minimal downtime. Any scheduled maintenance should be communicated to users in advance.

6. *Maintainability*

- The code is modular and well-documented to make future updates and maintenance easier.

7. *Compatibility*

- The platform is compatible with all modern web browsers (Chrome, Firefox, Safari, Edge) to ensure accessibility for all users.
- The platform is optimized for both desktop and mobile use

4. UML DIAGRAMS

4.1 USE CASES

Use Case 1: Customer Registration

Use Case ID: UC1

Actor: Customer

Description: New Customer can create an account.

Preconditions:

- Customer must not have an existing account.

Main Flow:

1. Customer clicks "Register".
2. Customer enters personal details.
3. Customer creates login credentials.
4. System validates information.

Postconditions:

- New Customer account is created.

Use Case 2: Room Booking

Use Case ID: UC2

Actor: Customer

Description: Customer can book rooms for their stay.

Preconditions:

- Customer searches for available rooms.

Main Flow:

1. Admin logs into the system.
2. Customer chooses room type.
3. Customer selects check-in and check-out dates.
4. Customer confirms booking

Postconditions:

- Room is booked and confirmation is sent.

Use Case 3: Food Ordering

Use Case ID: UC3

Actor: Customer

Description: Customer can order food through the system

Preconditions:

- Customer must have an active room booking.

Main Flow:

1. Customer browses food menu.
2. Customer selects food items.
3. Customer confirms order.

Postconditions:

- Food order is placed and confirmation is sent.

Use Case 4: Manage Rooms

Use Case ID: UC5

Actor: Admin

Description: Admin can manage room availability and pricing.

Preconditions:

- Admin must be logged into the system..

Main Flow:

1. Admin logs into the system.
2. Admin navigates to "Room Management" section.
3. Admin can update room availability and pricing.

Postconditions:

- Room information is updated in the system.

Use Case 5: Manage Food Orders

Use Case ID: UC5

Actor: Admin

Description: Admin can view and manage food orders for guests.

Preconditions:

- Admin must be logged into the system.

Main Flow:

1. Admin logs into the system.
2. Admin navigates to Food Management.
3. Admin can add, delete or update food items .

Postconditions:

- Food item is updated.

Use Case 6: Monitor Revenue

Use Case ID: UC6

Actor: Admin

Description: Admin can track revenue from room bookings and food orders.

Preconditions:

- Admin must be logged into the system.

Main Flow:

1. Admin logs into the system.
2. Admin navigates to Dashboard.
3. Admin views separate revenue for rooms and food.

Postconditions:

- Revenue statistics are generated and displayed.

Use Case 7: View Booking History

Use Case ID: UC7

Actor: Customer

Description: Customer can view their booking and order history.

Preconditions:

- Customer must be logged into the system.

Main Flow:

1. Customer accesses their dashboard.
2. Customer views booking history.
3. Customer views food order history.
4. Customer can check current booking status.

Postconditions:

- Customer views their complete history

4.2 USECASE DIAGRAM

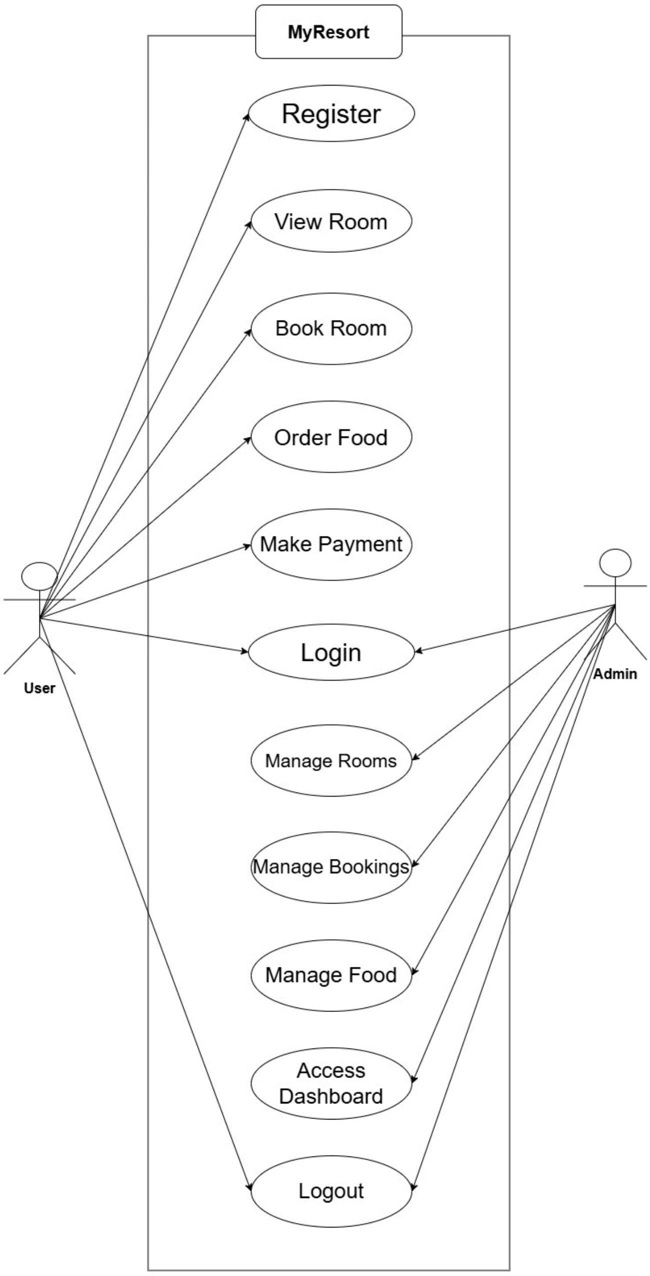


Figure 4.2 Use Case Diagram

4.3 ACTIVITY DIAGRAM OF ADMIN

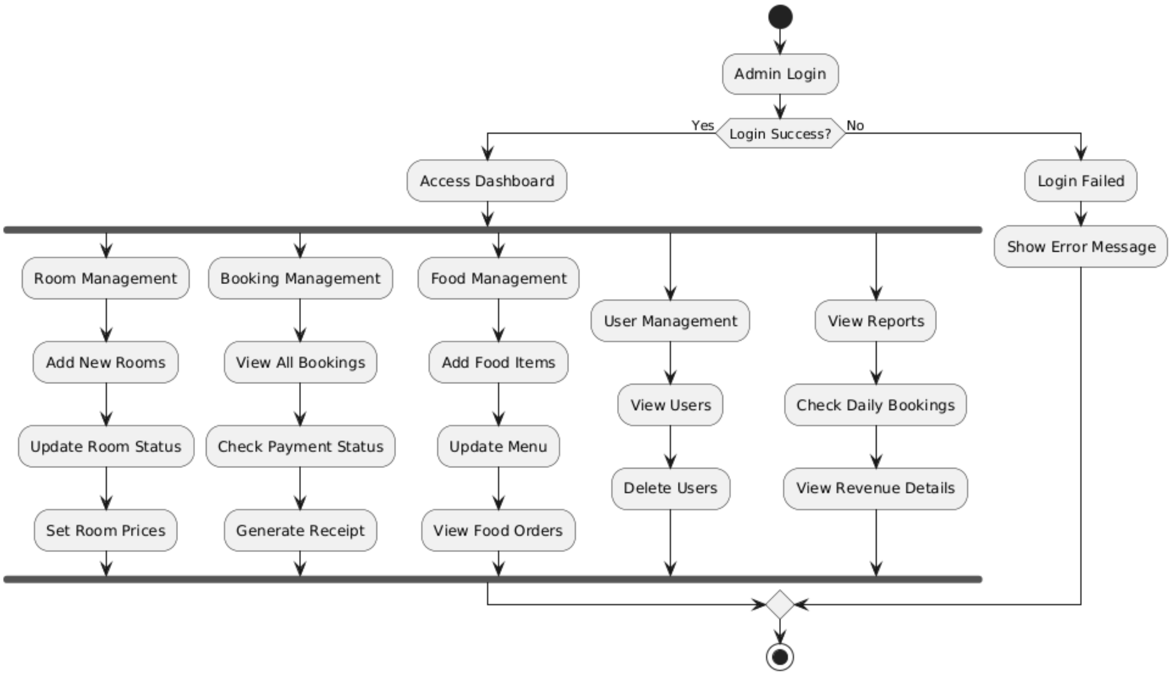


Figure 4.3 Activity Diagram of Admin

4.4 ACTIVITY DIAGRAM OF CUSTOMER

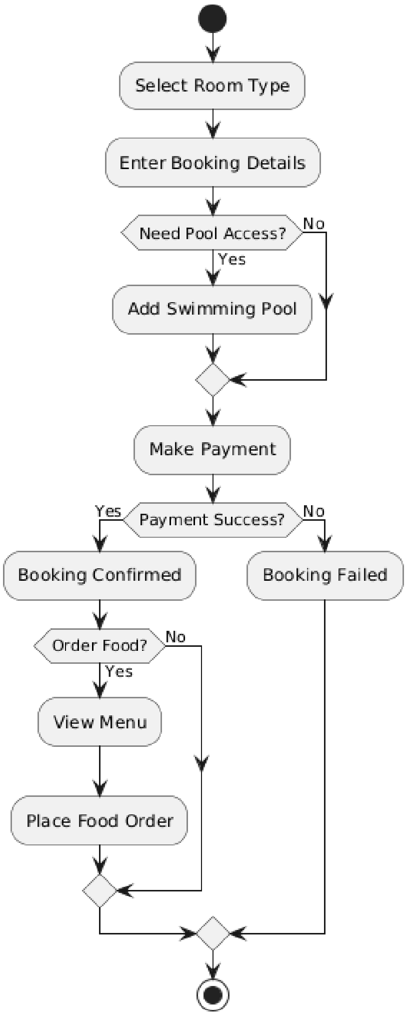


Figure 4.4 Activity Diagram of User

4.5 CLASS DIAGRAM

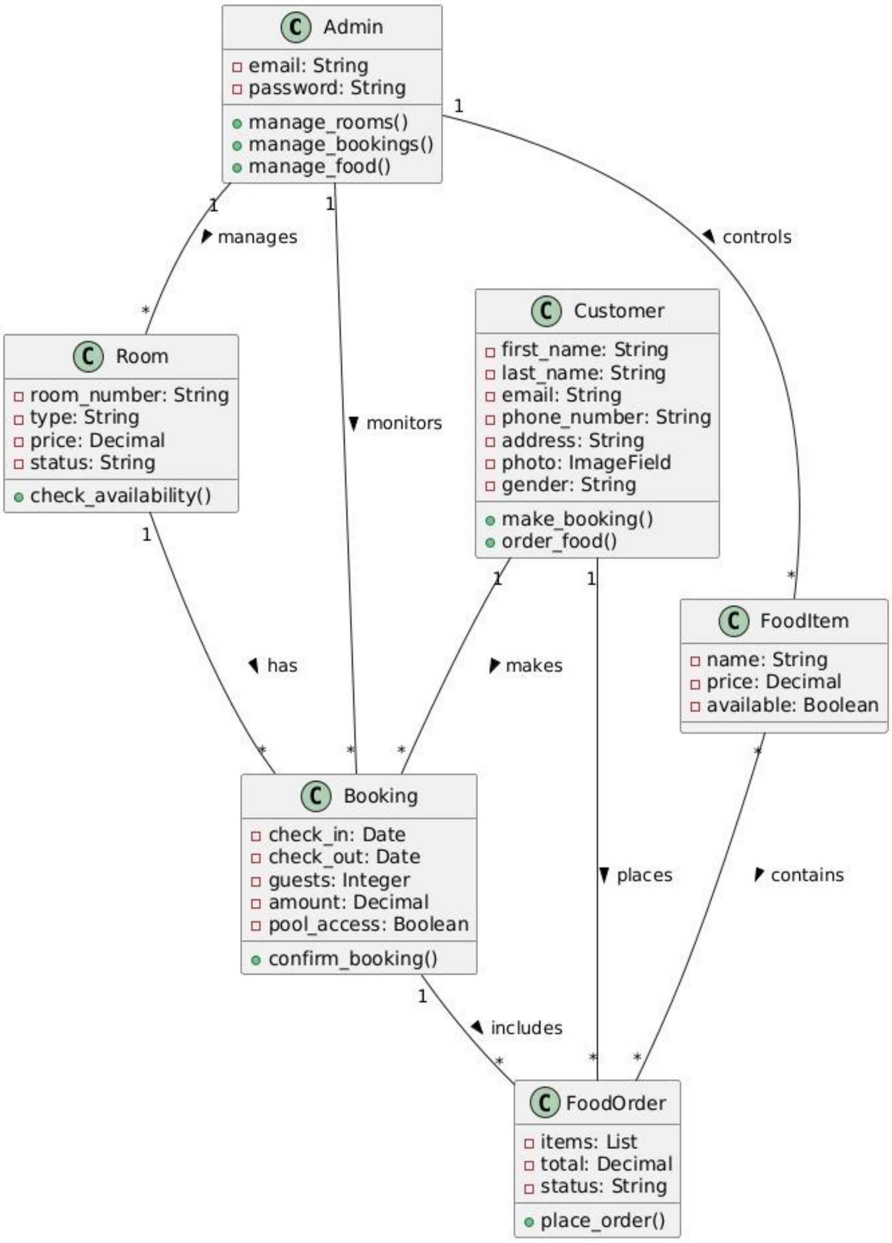


Figure 4.5 Class diagram

4.6 SEQUENCE DIAGRAM

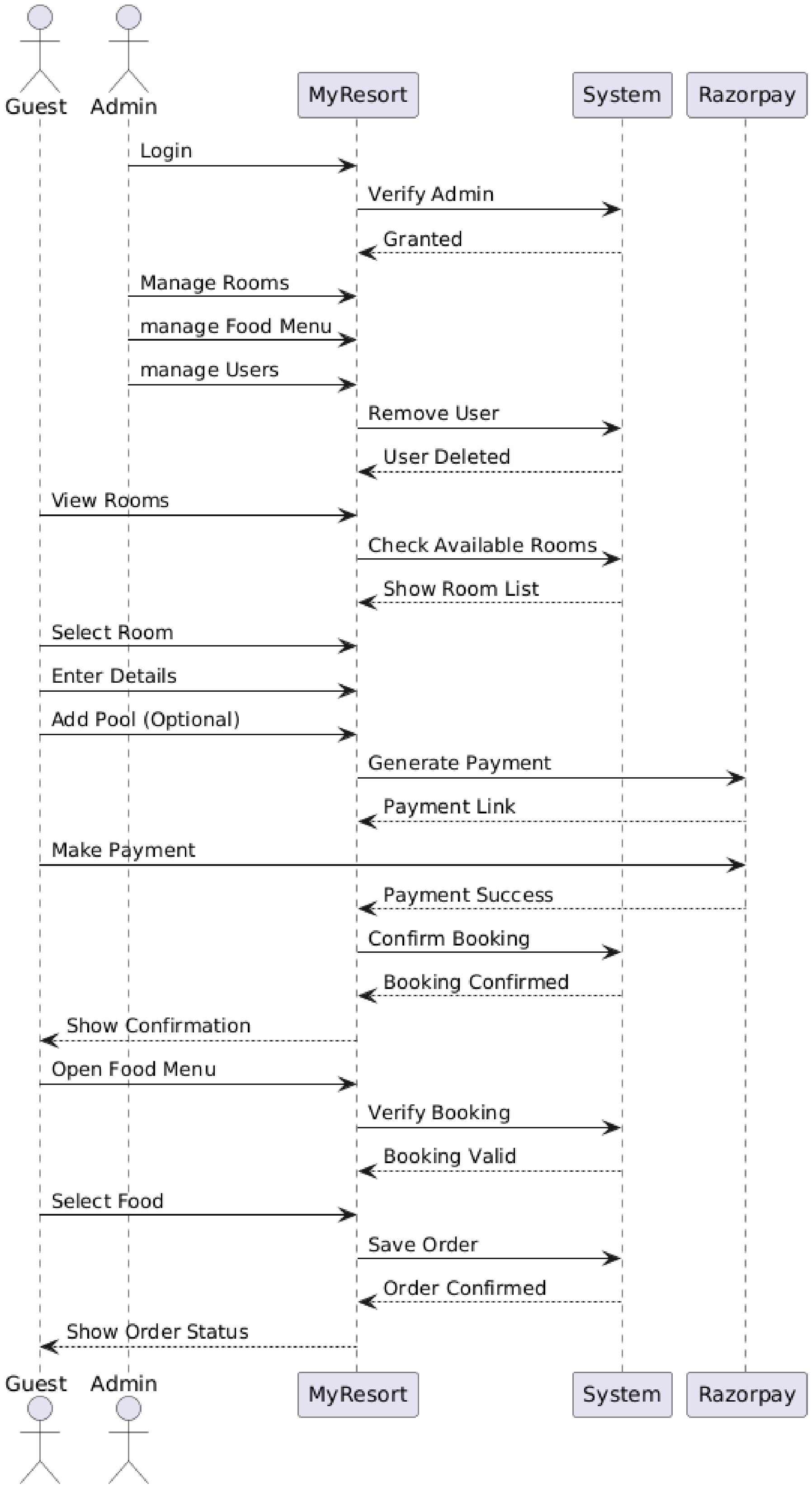


Figure 4.6 Sequence diagram

4.7 ER DIAGRAM

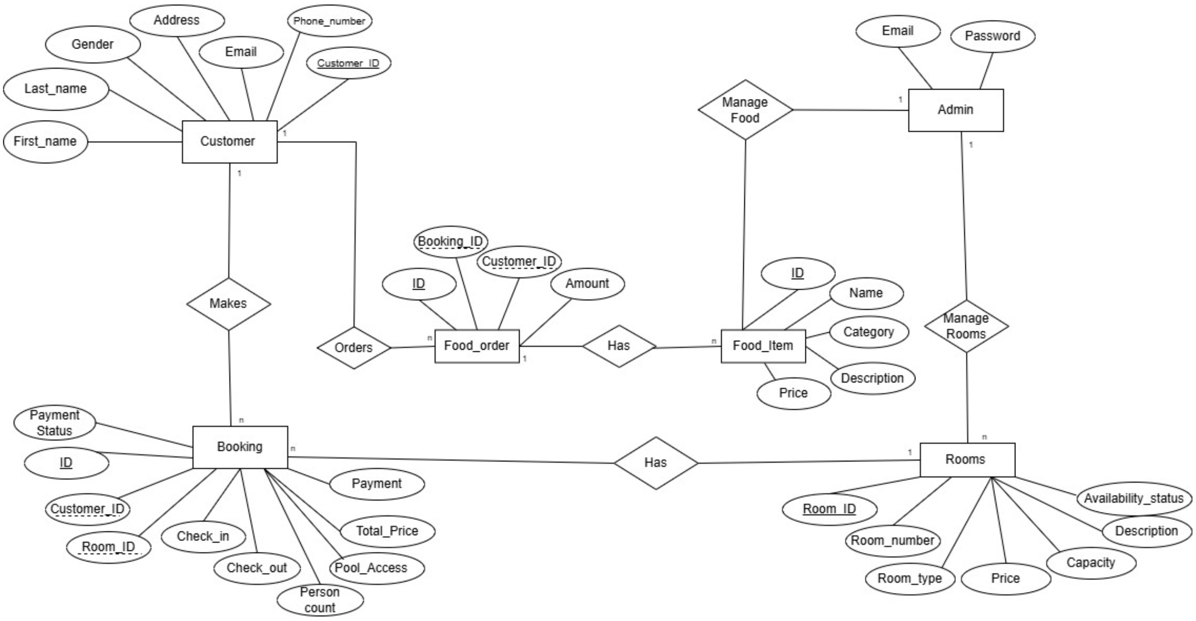


Figure 4.7 ER diagram

5. TEST CASES

5.1 LOGIN PAGE

Sr No.	Test Cases	Feature	Description	Steps To Execute	Expected Results
1	Valid Login	Login	Test with valid credentials	1. Navigate to the Login Page. 2. Enter valid username. 3. Enter valid password. 4. Click on "Login" button.	User should be successfully logged into their account.
2	Invalid Email	Login	Test with invalid Email	1. Navigate to the Login Page. 2. Enter invalid Email 3. Enter valid password. 4. Click on "Login" button.	System should display an error message for an invalid email
3	Invalid Passwo rd	Login	Test with invalid password	1. Navigate to the Login Page. 2. Enter valid username. 3. Enter invalid password. 4. Click on "Login" button.	System should display an error message for an invalid password.
4	Empty Fields	Login	Test with empty fields	1. Navigate to the Login Page. 2. Leave username and password fields empty. 3.Click on "Login" button.	System should prompt the user to fill in both username and password fields.

5.2 REGISTRATION PAGE

Sr No.	Test Cases	Feature	Description	Steps To Execute	Expected Results
1	Valid Registration	Registration	Test with valid user information	1. Navigate to the Registration Page. 2. Enter valid user details (username, email, password). 3. Click on "Register" button.	User should be successfully registered and directed to the login page or a success message should be displayed.
2	Existing Email	Registration	Test with an existing email	1. Navigate to the Registration Page. 2. Enter an email that is already registered. 3. Fill other details. 4. Click on "Register" button.	System should prompt the user that the email is already in use.
3	Weak Password	Registration	Test with a weak password	1. Navigate to the Registration Page. 2. Enter a weak password (less than minimum requirements). 3. Fill other details. 4. Click on "Register" button.	System should display an error message indicating password requirements are not met.
4	Empty Fields	Registration	Test with empty fields	1. Navigate to the Registration Page. 2. Leave one or more fields empty. 3. Click on "Register" button.	System should prompt the user to fill in all required fields.

5.3 ROOM BOOKING

Sr. No.	Test Case	Feature	Description	Steps to Execute	Expected Results
1	Open Room Booking Page	Navigation	Verify that the room booking page is accessible to users	1. 1. Log in to the resort website 2. Navigate to 'Book Room' section	The room booking page opens successfully displaying available rooms
2	Room Booking Successfully	Data Entry	Verify that a new room booking can be made with valid inputs	1. Select check-in and check-out dates 2. Choose room type 3. Select pool access (optional) 4. Fill in guest details 5. Submit booking	Booking is confirmed with a success message and confirmation email is sent
3	Double Booking Prevention	Availability Check	Verify system prevents booking already reserved rooms	1. Select dates for which a room is already booked 2. Attempt to book the same room type	System shows "Not Available" message or hides already booked rooms
4	Pool Access Addition	Optional Service	Verify pool access can be added to room booking	1. Select a room 2. Check the pool access option 3. Complete booking	Pool access is added without any extra cost and appears in booking confirmation
5	Date Selection Validation	Input Validation	Verify proper date selection validation	1. Try to select past dates 2. Try to set check-out before check-in 3. Attempt booking	System prevents invalid date selections with appropriate error messages

6. INPUT AND OUTPUT DESIGN

6.1 INPUT DESIGN

1.Introduction

Input design ensures accurate and user-friendly data collection from resort customers and administrators. This section details the forms, fields, and validation mechanisms implemented in the MyResort platform to provide seamless room booking and food ordering experiences. The input design focuses on minimizing errors and enhancing efficiency in capturing guest and administrative data.

2. Forms and Input Fields

• **Forms:**

- **Registration Form:** Captures essential user details including name, email, password for new user registration.
- **Login Form:** Provides secure login interface for users and administrators.
- **Room Booking Form:** Allows users to select check-in/check-out dates, room type, and number of guests.
- **Food Order Form:** Enables guests to select food items, specify quantities, and enter room number.
- **Admin Dashboard Form**: Allows administrators to manage bookings and update room/food order status

• **Fields:**

- **Text Fields:** Collects name, email, contact through inputs; requests via detailed text areas.
- **Date Fields:** Check-in, check-out dates selected via calendar; bookings auto-tracked for availability.
- **Dropdown Menus:** Room types, food items, booking status, and payments managed through dropdown menus.
- **Numeric Fields:** Handles guest count, room numbers, phone and food quantities with numeric inputs.
- **File Uploads:** Admins can upload room and food images when adding new information.

3. Validation and Error Handling

• Client-Side Validation:

- Utilizes HTML5 attributes like required, pattern, and max length for input constraints.
- Validates guest count against room capacity limits automatically.
- Checks real-time room availability for selected booking dates

• Server-Side Validation:

- Django checks all form submissions for security and completeness.
- Validates room availability before confirming any booking.
- Ensures booking dates are logical and available.
- Verifies room numbers for food orders during submission.

• Error Messages:

- Shows clear errors for invalid or past date selections.
- Displays room availability status when booking unavailable rooms.
- Provides booking confirmation or failure messages instantly.
- Displays specific payment errors with validation prompts.

6.2 OUTPUT DESIGN

1.Introduction

Output design focuses on presenting meaningful information about room bookings, food orders, and resort management in an intuitive and organized manner. It ensures that all outputs, from room availability to booking confirmations, are clear, accurate, and visually appealing.

2. User Interface Elements

- **Dashboard/Overview:** For guests: The dashboard displays current bookings, room details, food orders, and booking history. And admin can access to room occupancy rates, daily food orders, revenue statistics, and booking management
- **Room Listings:** Displays available rooms with high-quality images, room types, pricing, and amenities. Interactive features include date-based availability checking and room type filtering.
- **Booking Details and Invoice:** Users can view detailed booking information including check-in/out dates, room type, and charges. Downloadable invoices containing room charges information.

3. Reports and Summaries

- **User Reports:** Booking history showing past and current room reservations with dates and food order history.
- **Admin Reports:** Monthly revenue reports separating room bookings and food orders.

4. Error and Success Messages

- **Success Messages:** Messages like "Registration Successful", "Booking Successfully", "Payment Processed Successfully" are displayed prominently for user assurance.
- **Error Messages:** Error messages such as "Invalid Login Credentials" or "Room Not Available for Selected Dates", "Payment Processing Failed" are displayed in red with clear instructions on corrective action.

7. SYSTEM IMPLEMENTATION

7.1. Introduction

The MyResort project is a comprehensive resort management system designed to streamline room bookings and food ordering processes. Built using Django for the backend, the project implements HTML, CSS, and Bootstrap for a responsive and user-friendly frontend interface. While SQLite serves as the database to efficiently store and manage resort data, including room information, customer details, bookings, and food orders.

Development Environment Setup:

- **Python Version:** 3.9+
- **Django Version:** 4.0+
- **Database:** SQLite
- **Frontend:** HTML, CSS, JavaScript

7.2. Project Structure

- The MyResort project follows Django's recommended architecture by organizing components into logical apps, ensuring maintainability and scalability. Here's the directory structure:

```
MyResort /
│
├── MyResort /
│   ├── settings.py
│   ├── urls.py
│   └── manage.py
│
├── resort/
│   ├── templates/
│   ├── base.html
│   ├── login.html
│   ├── models.py
├── models.py
│   └── urls.py
│   ├── views.py
│   └── urls.py
│
├── static/
├── css
│   ├── styles.css
```

7.3. Database Design and Models

7.3.1. Customer Table

Field Name	**Data Type**	**Constraints**
customer_id	AutoField	Primary Key
first_name	CharField(50)	NOT NULL
last_name	CharField(50)	NOT NULL
email	EmailField	Unique, NOT NULL
address	TextField	NOT NULL
phone_number	CharField(15)	Unique, NOT NULL
photo	ImageField	NULLABLE
gender	CharField(10)	Choices, NOT NULL

7.3.2. Room Table

Field Name	**Data Type**	**Constraints**
Room_id	AutoField	Primary Key
room_number	CharField(10)	NOT NULL
room_type	CharField(10)	Choices, NOT NULL
price_per_night	DecimalField(10,2)	NOT NULL
capacity	IntegerField	NOT NULL
description	TextField	NOT NULL
availability_status	BooleanField	Default=True
image	ImageField	NULLABLE

7.3.3. Booking Table

Field Name	**Data Type**	**Constraints**
Booking_id	AutoField	Primary Key
customer	ForeignKey(Customer_id)	On Delete CASCADE
room	ForeignKey(Room_id)	On Delete CASCADE
check_in_date	DateField	NOT NULL
check_out_date	DateField	NOT NULL
num_persons	IntegerField	NOT NULL
total_amount	DecimalField(10,2)	NOT NULL
status	CharField(20)	Choices, Default='pending'
created_at	DateTimeField	Auto Now Add
special_requests	TextField	NULLABLE
payment_status	CharField(20)	NULLABLE
Razerpay_paymentId	CharField(100)	NULLABLE
Razerpay_OrderId	CharField(100)	NULLABLE
Razerpay_signature	CharField(100)	NULLABLE

7.3.4. Swimming Pool Booking Table

Field Name	Data Type	Constraints
Pool_id	AutoField	Primary Key
customer	ForeignKey(Customer_id)	On Delete CASCADE
booking	ForeignKey(Booking_id)	Related Name='pool_bookings'
date	DateField	NOT NULL
time_slot	CharField(20)	Choices, NOT NULL
num_persons	IntegerField	Default=1
created_at	DateTimeField	Auto Now Add
status	CharField(20)	Choices, Default='pending'

7.3.5 Food Item Table

Field Name	Data Type	Constraints
Item_id	AutoField	Primary Key
name	CharField(100)	NOT NULL
description	TextField	NOT NULL
price	DecimalField(10,2)	NOT NULL
category	CharField(20)	Choices, NOT NULL
image	ImageField	NULLABLE
is_available	BooleanField	Default=True
created_at	DateTimeField	Auto Now Add
updated_at	DateTimeField	Auto Now

7.3.6 FoodOrder Table

Field Name	Data Type	Constraints
foodOrder_id	AutoField	Primary Key
customer	ForeignKey(Customer_id)	On Delete CASCADE
booking	ForeignKey(Booking_id)	On Delete CASCADE
total_amount	DecimalField(10,2)	NOT NULL
status	CharField(20)	Choices, Default='pending'
special_instructions	TextField	NULLABLE
created_at	DateTimeField	Auto Now Add
updated_at	DateTimeField	Auto Now

7.3.7 FoodOrderItem Table

Field Name	Data Type	Constraints
orderItem_id	AutoField	Primary Key
food_order	ForeignKey(FoodOrder_id)	Related Name='items'
food_item	ForeignKey(Item_id)	NOT NULL
quantity	PositiveIntegerField	Default=1
price	DecimalField(10,2)	NOT NULL
special_instructions	TextField	NULLABLE

7.4. URL Routing and Views

URL routing connects user requests to specific views using Django’s urls.py.

Routing Example:

- /rooms/ - Displays available rooms
- /login/ - Handles user login
- /admin/ - Admin dashboard for managing resort
- /food/ - Food menu and ordering
- /bookings/ - Booking management.

Views:

- Room Views: Display room details, availability, and booking options
- User Views: Handle login, registration, and profile management
- Food Views: Manage food menu and orders
- Admin Views: Monitor bookings, revenue, and resort operations.

HTTP Methods:

- GET: Retrieve room listings, food menu, booking history
- POST: Process room bookings, food orders, user registration
- PUT: Update booking status, food order status

7.5. Templates and Frontend Integration

MyResort uses Django's template engine, with a foundation built on HTML, CSS, and JavaScript for the frontend development. The system's base template (base.html) establishes a consistent layout across all pages, featuring a professional header with navigation options, user authentication status, and a footer containing essential resort information. This base template is extended by specific pages like room_booking.html for reservations, food_menu.html for dining options, and dashboard interfaces for both users and administrators.

7.6. User Authentication and Authorization

Django’s built-in authentication system is used to handle login, logout, and registration. Here are some important features:

- **Login and Logout:** Users can log in using their credentials and log out when needed.
- **Registration:** New users can register with unique emails and strong passwords.

7.7. Forms and Validation

- **Registration Form:** Ensures email uniqueness and strong password validation.
- **Room Booking Form:** Validates dates, guest numbers, availability.
- **Food Order Form:** Validates room number, order quantities.
- **Admin Forms:** Validates booking updates and revenue entries.

7.8. Business Logic and Core Functionality

- **Room Management:** Real-time availability checking and Date conflict prevention.
- **Food Service:** Digital menu management.
- **Administrative Functions:** Revenue monitoring (rooms and food), Booking management and customer data handling.

7.9. Testing and Debugging

- **Unit Testing:** Unit tests are written to test the models, views, and forms to ensure correct functionality. For example, tests ensure that the cart updates correctly when products are added or removed.
- **Debugging:** Django's built-in debug toolbar is used for troubleshooting. Custom error handling in views improves the user experience by providing helpful error messages for common issues (e.g., 404 or 500 errors).

8. FUTURE ENHANCEMENTS

1. *AI-Powered Room Recommendation*

Use AI to suggest rooms tailored to guest preferences by analyzing past booking patterns. Implement smart pricing strategies based on seasonal trends and automate room allocation for optimal usage and personalized guest experiences.

2. *Smart Room Technology Integration.*

Integrate digital room keys via a mobile app, IoT-based controls for AC, lights, and TV, and voice-activated room services. Enable automated temperature and lighting adjustments for enhanced comfort and energy efficiency.

3. *Enhanced Guest Services*

Offer a digital concierge chatbot, automate housekeeping and in-room service requests, collect feedback through a digital system, and build personalized guest profiles for tailored experiences.

4. *Virtual Resort Experience.*

Provide 360° virtual room tours, AR-based resort navigation, a virtual concierge, and an interactive resort map with real-time updates for an immersive and user-friendly guest experience.

5. *Mobile App Development*

With the increasing trend of mobile shopping, developing a mobile application for the platform will significantly enhance the user experience. A mobile app can include features like push notifications for promotions and order updates, in-app payments, and easy access to browsing and purchasing.

9. CONCLUSION

CONCLUSION

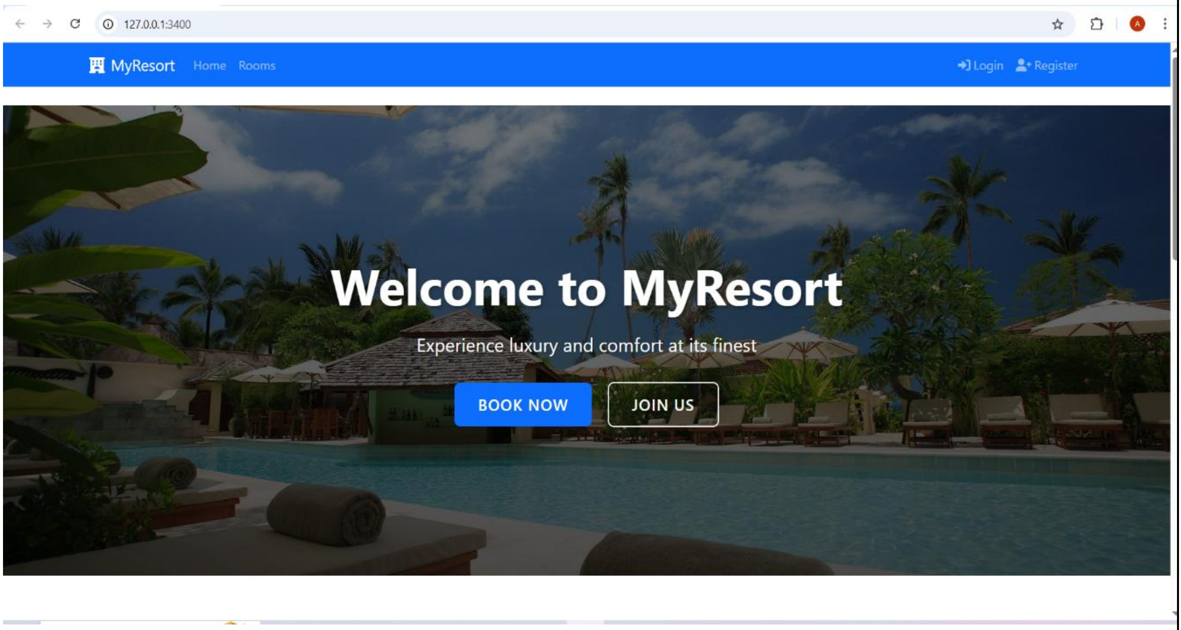
The MyResort management system is designed to streamline resort operations by providing an efficient and user-friendly platform for both guests and administrators. Through its integrated room booking system, food ordering capabilities, and comprehensive admin dashboard, the platform successfully bridges the gap between traditional resort management and modern digital solutions. The system effectively manages room reservations, food orders, and administrative tasks while maintaining a seamless user experience for guests and staff alike. The platform's core strengths lie in its ability to handle real-time room availability, process food orders efficiently, and provide detailed revenue tracking for management. By automating crucial processes like booking confirmations, room status updates, and food order management, the system significantly reduces manual workload and minimizes human error. The administrative dashboard provides valuable insights into resort operations, enabling data-driven decision-making for better resource allocation and service improvement.

The project is built with scalability and flexibility in mind, allowing for seamless integration of new features and adaptation to growing business needs. Whether it's expanding room categories, enhancing food service capabilities, or implementing new guest services, MyResort is positioned to grow alongside the business. Through this digital transformation, MyResort demonstrates how technology can enhance traditional hospitality services, creating a more efficient and enjoyable experience for both guests and staff. The system stands as a testament to the successful integration of technology in the hospitality sector, paving the way for future innovations in resort management.

ANNEXURE

Screenshots:

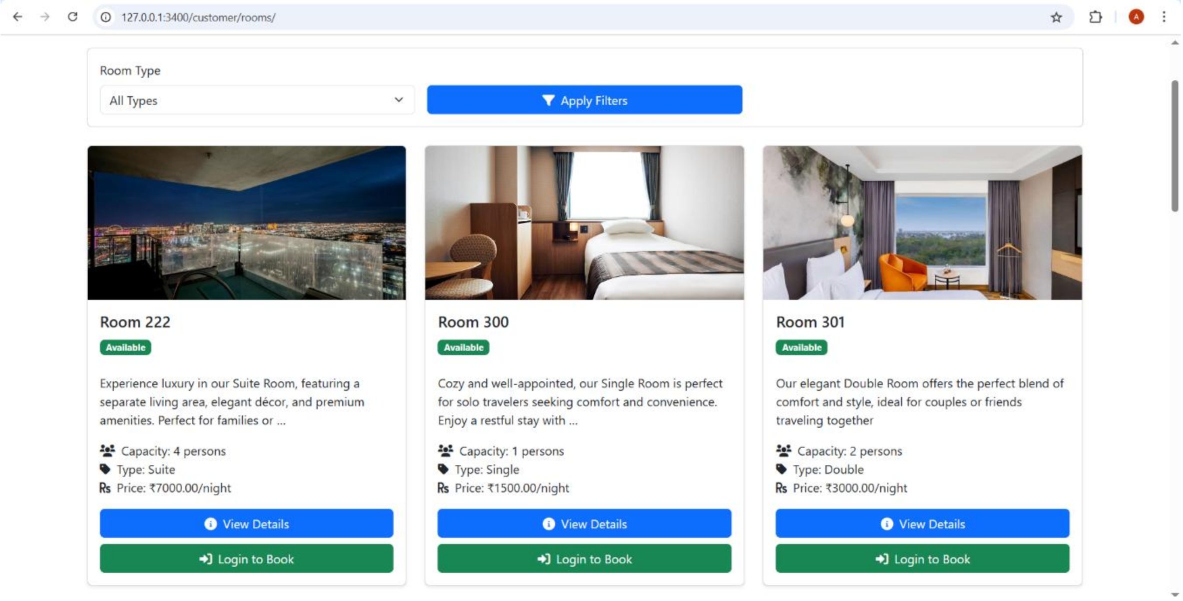
Home Page



Welcome to MyResort

Experience luxury and comfort at its finest

Rooms Page



Room Type: All Types

Room 222

Available

Experience luxury in our Suite Room, featuring a separate living area, elegant décor, and premium amenities. Perfect for families or ...

Capacity: 4 persons

Type: Suite

Price: ₹7000.00/night

Room 300

Available

Cozy and well-appointed, our Single Room is perfect for solo travelers seeking comfort and convenience. Enjoy a restful stay with ...

Capacity: 1 persons

Type: Single

Price: ₹1500.00/night

Room 301

Available

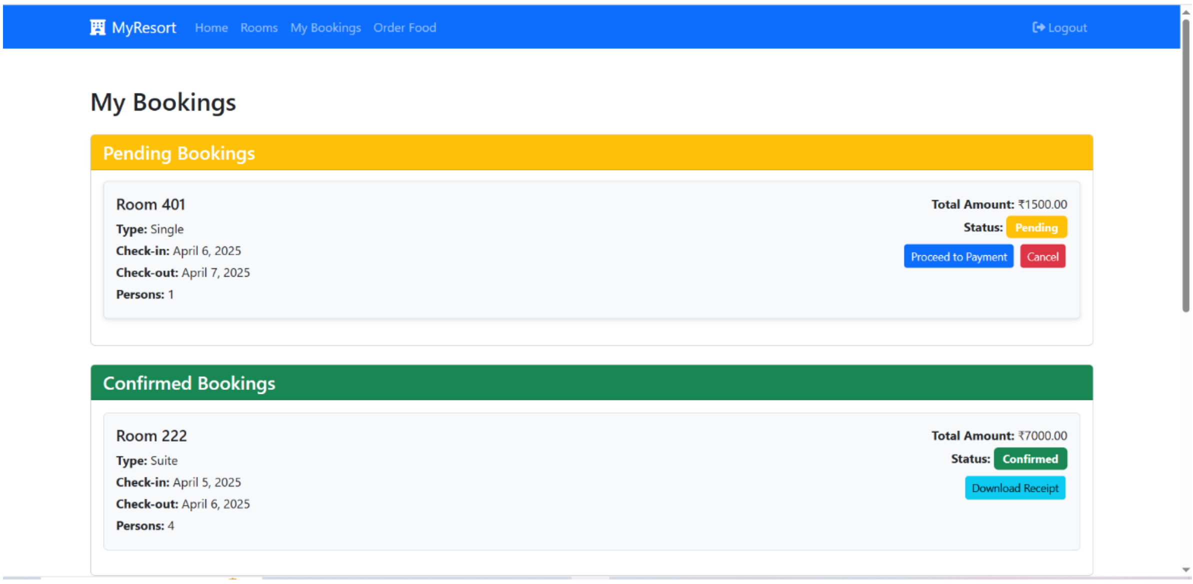
Our elegant Double Room offers the perfect blend of comfort and style, ideal for couples or friends traveling together

Capacity: 2 persons

Type: Double

Price: ₹3000.00/night

Room Booking History Page



My Bookings

Pending Bookings

Room 401

Type: Single

Check-in: April 6, 2025

Check-out: April 7, 2025

Persons: 1

Total Amount: ₹1500.00

Status: Pending

Confirmed Bookings

Room 222

Type: Suite

Check-in: April 5, 2025

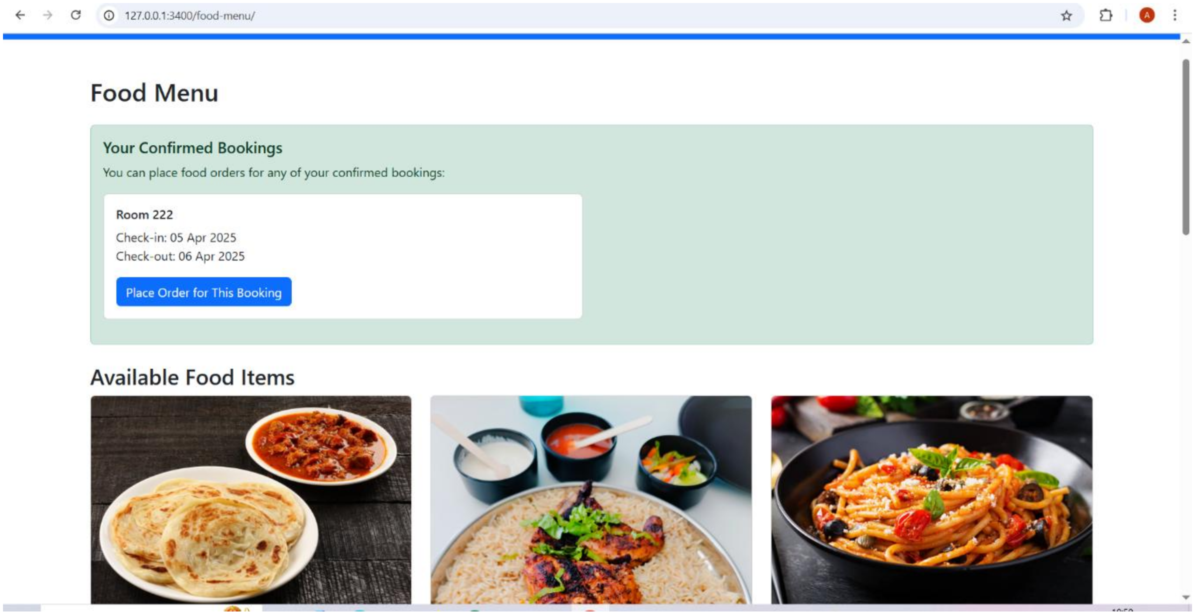
Check-out: April 6, 2025

Persons: 4

Total Amount: ₹7000.00

Status: Confirmed

Food Order Page



Food Menu

Your Confirmed Bookings

You can place food orders for any of your confirmed bookings:

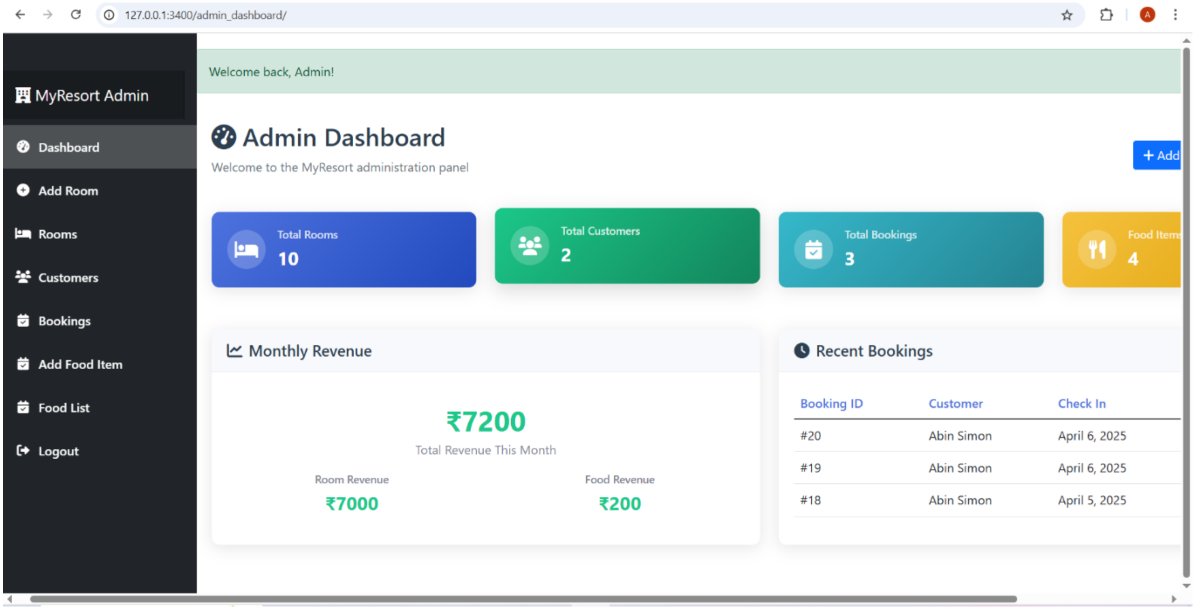
Room 222

Check-in: 05 Apr 2025

Check-out: 06 Apr 2025

Available Food Items

Admin Page



Welcome back, Admin!

Admin Dashboard

Welcome to the MyResort administration panel

Total Rooms 10

Total Customers 2

Total Bookings 3

Food Items 4

Monthly Revenue

₹7200

Total Revenue This Month

Room Revenue ₹7000

Food Revenue ₹200

Recent Bookings

Booking ID	Customer	Check In
#20	Abin Simon	April 6, 2025
#19	Abin Simon	April 6, 2025
#18	Abin Simon	April 5, 2025