Hotel reservation Asp.net web app (Bookify)

BY:Strugglers Team

- **Team:**
- ١-محمد اشرف عبدالفتاح عباس(leader)
- ٢-وليد السيد مرسي عبدالفتاح السبكي
- ٣-عبدالرحمن ضياء عزيز حافظ
- ٤-عبدالرحمن مصطفي احمد محمود
- ٥- محمود رمضان فوزي عبدالله
-

TABLE OF CONTENT

01 PROJECT PREVIEW

02 PROJECT GOALS

03 Requirements

04 System Design

05 Implementation

06 Evaluator

07 CONCLUSION

01

PROJECT PREVIEW

- Bookify is a web-based hotelreservation system built with ASP.NET Core
- MVC using clean design patterns for scalability and maintainability.
- It allows customersto search rooms, make secure bookingswith ---
- payments, andview their bookinghistory.
- Meanwhile, hotel staff can manage rooms and reservations through a
- powerful Admin Dashboard ,delivering a seamless and professional
- booking experience.

03

PROJECT GOALS

Simplify Hotel Booking Process:

To provide users with a fast, intuitive, and secure platform to search, book, and manage hotel reservations online.

GOAL 1

Enhance User Experience

To design a user-friendly interface with clear navigation, responsivelayout, and seamless interaction across all devices.

GOAL 2

Ensure System Scalability and Maintainability

To build the system using ASP.NET Core MVC and clean architecture principles, allowing for future scalability and easier maintenance.

GOAL 3

Requirements Gathering:

02

1-Stakehoder Analysis:

oCustomer (User):

—Searches for available rooms.

—Views room details(price, features).

—Books a room and pays securelyonline.

—Tracks previous bookings through a personal account.

Requirements Gathering:

02

1-Stakehoder Analysis:

- **Admin (Hotel Staff):**

—Manages rooms and room types.

—Monitors all bookings.

—Controls user permissions.

Requirements Gathering:

02

1-Stakehoder Analysis:

oSystem :

—Integrates securely and smoothly with the system.

—Records all financial transactions accurately.

Requirements Gathering:

1-Stakehoder Analysis:

02

oSystem:

—Integrates securely and smoothly with the system.

—Records all financial transactions accurately.

• User Stories:

-As a customer, I want to sea rch for available rooms so that I can book one that fits my needs.

-As a customer, I want to view room details so that I can decide which room to book.

-As a customer, I want to pay online securely so that my booking is confirmed immediately.

-As an admin, I want to add and manage room types so that the hotel offerings stay up to date.

-As an admin, I want to view all customer bookings so that I can manage hotel occupancy.

As a manager, I want to generate reports about reservations so that I can make informed business decisions.

Functional Requirements:

—User Management: Register, login, and logout using ASP.NET Identity, Role- based access.

—Room management: Add, edit, and delete room types, manage room availability.

Booking management: Search,reservation cart, confirmbooking with payment, booking history.

—Admin Dashboard: mananage rooms and bookings, view customer detai\s, generate booking reports.

—Notifications: Real-time success/failure messages, email confirmations.

Non-Functional Requirements:

— **Performance:** System responds within 3 seconds.
— **Security:** Encrypted data, secure Stripe payments, authentication & authorization.
— **Usability:** Intuitive, responsive UI with Bootstrap. Reliability & Availability: 99%
uptime, atomic booking transactions.
— **Scalability:** Supports growth without redesign.

Use Case

Actors

1-Customer:

- End user who registers, logs in, searches for rooms, and makes reservations.
- Can also view, modify, or cancel bookings, and make online payments.

2-Admin:

- Responsible for managing hotel operations such as adding rooms, updating availability, and viewing reports.
- Can manage users, bookings, and system settings.

3-System:

- Automatically validates user data, processes payments, and updates booking status and room availability in real-time.

Use Case

Main Use Cases

1. Customer

- Register / Login: Customer creates an account or logs in to access the system.
- Search Rooms: Search for available rooms based on date, price, and room type.
- Make Booking: Select a room and confirm reservation.
- Make Payment: Complete payment securely through the system.
- View Booking History: Review past and current bookings.
- Cancel Booking: Cancel a confirmed booking before the stay date.
- Update Profile: Edit personal details such as contact info or password.

2. Admin

- Manage Rooms: Add, update, delete room details and price
- View All Bookings: Access a list all customer reservations.
- Manage Users: View or edit customer information.
- Generate Reports: Get insights bookings, payments, and roo occupancy.
- Manage System Settings: Cont room availability and booking ru

Use Case

Main Use Cases

3. System:

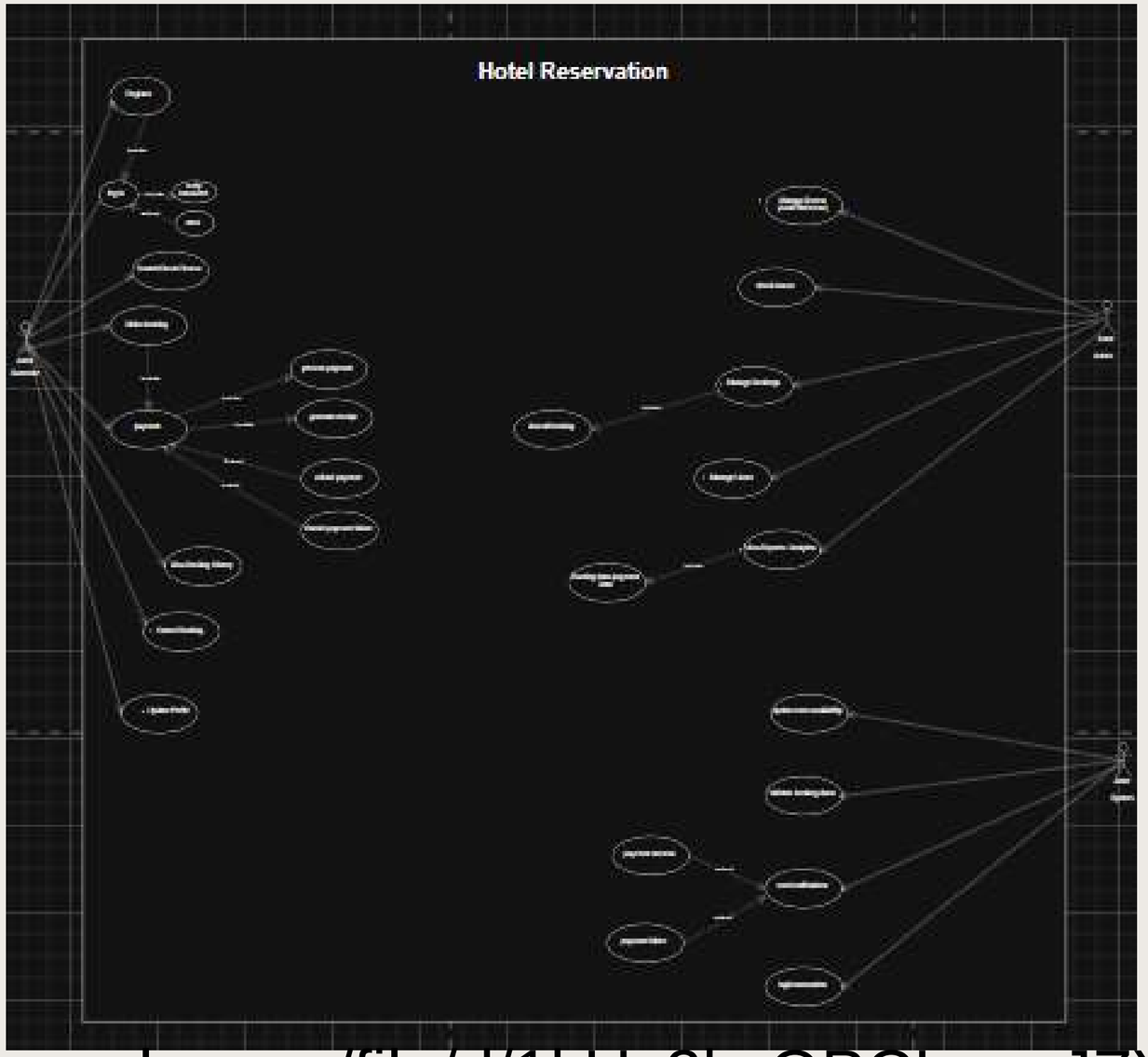
- Process Payment: Validate and process online transactions securely.
- Update Room Availability: Automatically adjust room status after bookings or cancellations.
- Send Notifications: Notify customers of booking confirmations, payments, or cancellations.
- Generate Invoice: Create digital receipts after successful payment.
- Validate Login / Registration: Ensure secure authentication and data validation.

Use Case

Summary

This use case diagram represents the complete workflow of a hotel reservation web application. Customers can search and book rooms, admins can manage operations, and the system handles automation like payments and room availability — ensuring a smooth and efficient hotel booking experience.

Use Case(image)



https://drive.google.com/file/d/1kUc2kzQPCLvg_JZYceixhmCwwKLj_Llx/view

DFD

Data Flow Diagram (DFD) Description

The Bookify Hotel Reservation System DFD (Level 0) illustrates the overall data flow between the system and its external entities. It shows how information moves between users, administrators, managers, and the payment gateway through a centralized web-based system responsible for handling hotel reservations, payments, and reports.

DFD

External Entities and Their Interactions

- ### 1. User

- Inputs to the System:
- Registration details to create a new account.
- Search queries for available rooms.
- Booking requests including selected room and stay dates.
- Payment confirmations for completed reservations.
- Outputs from the System:
- Confirmation of registration and booking.
- Room list based on search criteria.
- Previous reservations and booking reports.

DFD

External Entities and Their Interactions

- **2. Admin**
- Inputs to the System:
- Room and type details for room management.
- User permissions and management data.
- Outputs from the System:
- Updated room management data.
- Booked rooms and reservation details for monitoring and decision-making.

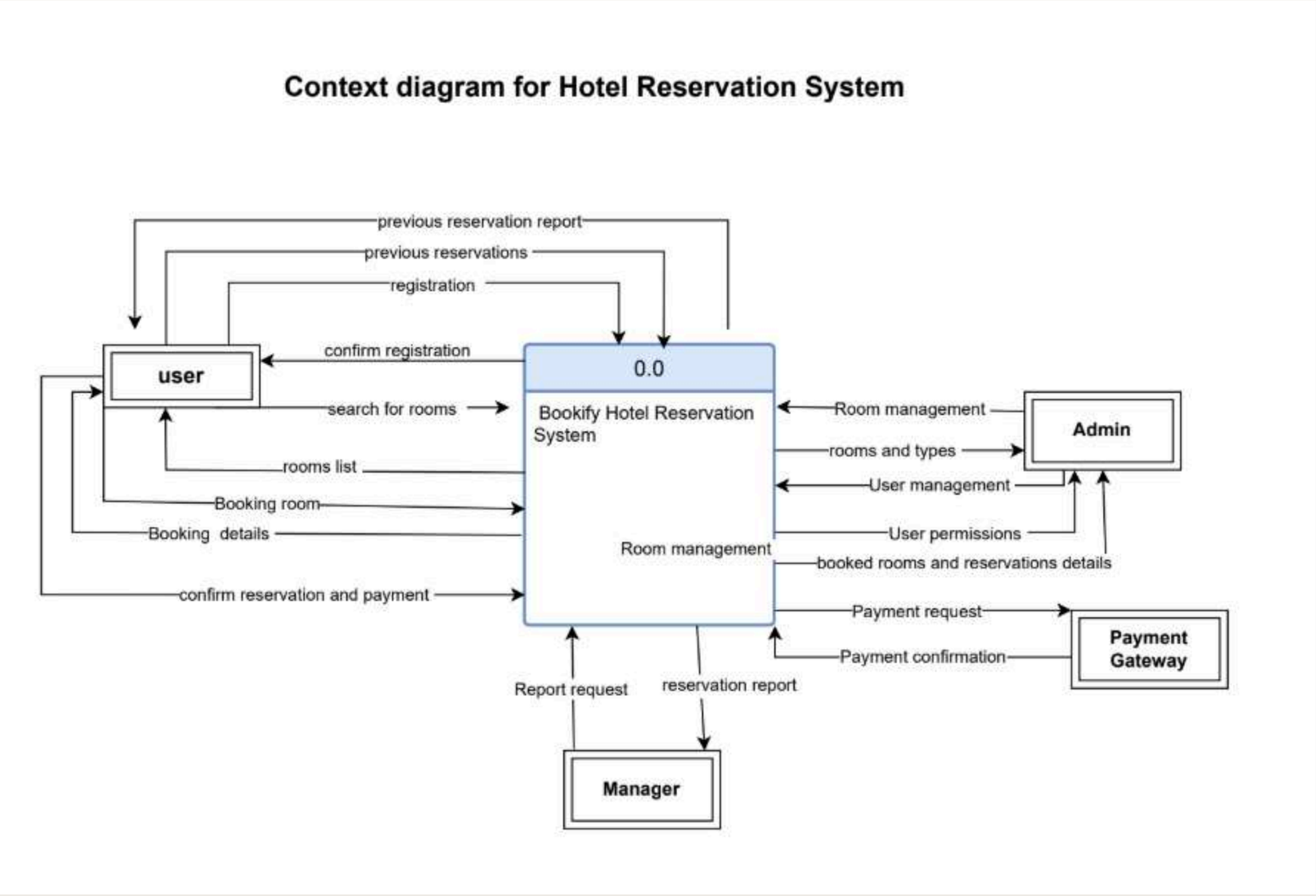
DFD

External Entities and Their Interactions

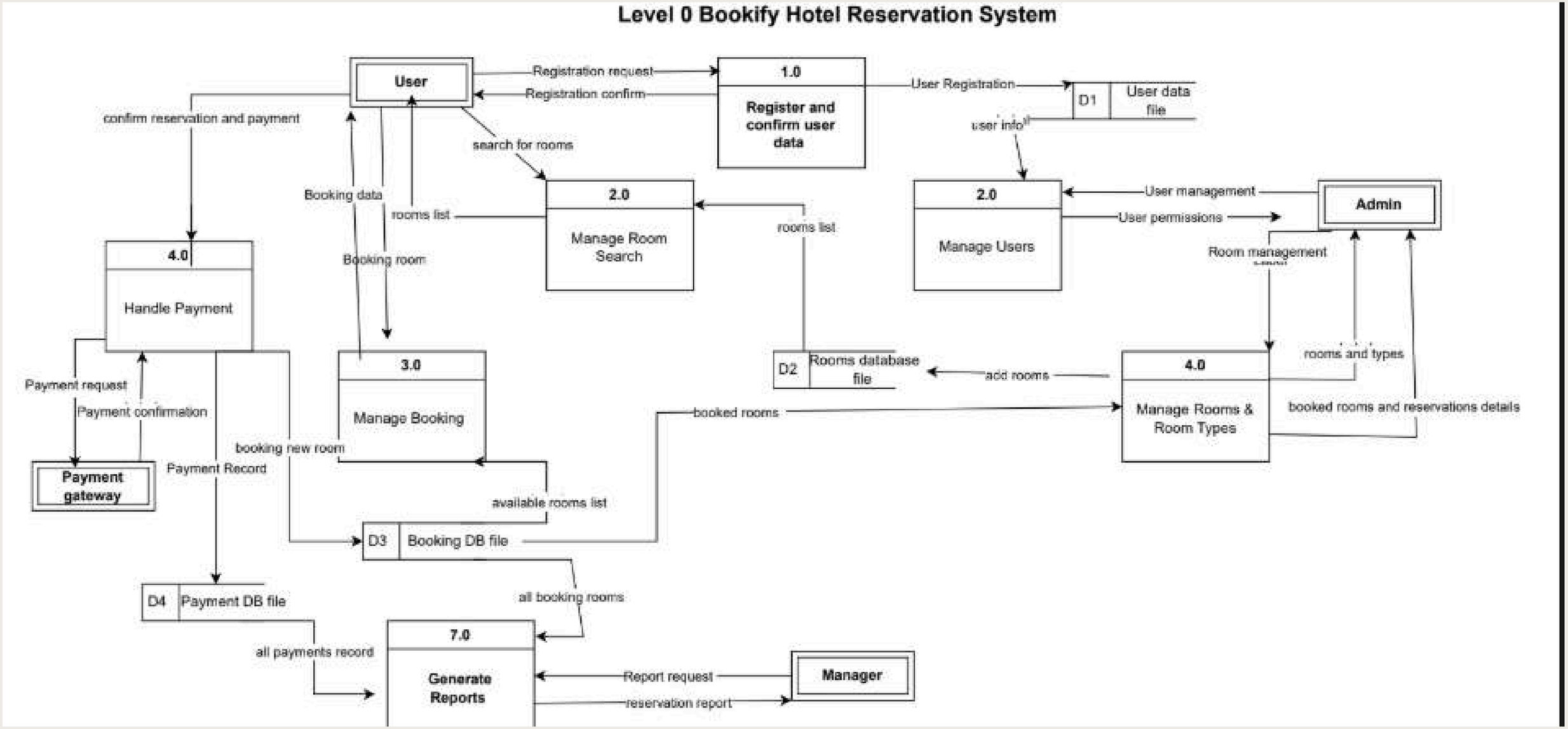
3. Manager

- Inputs to the System:
- Report requests to review system or booking performance.
- Outputs from the System:
- Reservation reports and summaries to assist in management decisions.

DFD(image)


Context diagram for Hotel Reservation System
previous reservation report
previous reservations
registration
confirm registration
user
0.0
Bookify Hotel Reservation System
search for rooms
rooms list
Booking room
Booking details
confirm reservation and payment
Room management
rooms and types
User management
User permissions
Room management
booked rooms and reservations details
Admin
Payment request
Payment confirmation
Payment Gateway
Report request
reservation report
Manager

DFD(image)


Level 0 Bookify Hotel Reservation System
User
Registration request
Registration confirm
1.0
Register and confirm user data
User Registration
D1
User data file
user info
confirm reservation and payment
search for rooms
Booking data
rooms list
2.0
Manage Room Search
rooms list
2.0
Manage Users
User management
User permissions
Admin
Room management
4.0
Handle Payment
Booking room
3.0
Manage Booking
D2
Rooms database file
add rooms
4.0
Manage Rooms & Room Types
rooms and types
booked rooms and reservations details
booked rooms
Payment request
Payment confirmation
booking new room
Payment Record
Payment gateway
available rooms list
D3
Booking DB file
D4
Payment DB file
all booking rooms
all payments record
7.0
Generate Reports
Report request
reservation report
Manager

ERD

Overall ERD Description

This ERD outlines the database structure for a Hotel Reservation and Management System. The system is designed to handle user accounts, room inventory, booking processes, payment transactions, and role-based permissions for staff and customers.

The diagram shows a classic relational database model for a hotel, connecting users to their reservations, payments, and assigned roles.

ERD Entity Descriptions

1. User Management:

- user: This is the core table for storing all user accounts (both customers and staff).
- Key Attributes: user_id, Email, Phone_Number, Password_Hash (crucial for secure login), first_name, Name.
- Purpose: Authenticates users and stores their personal contact information.

2. Room & Inventory Management:

- Room: Manages the physical rooms in the hotel.
- Key Attributes: Room_id, Floor, Room_Num (Room Number), Status (e.g., Available, Occupied, Cleaning).
- Relationships: Linked to a Room-Type via Room_Type_id.
- Room-Type: Defines the categories of rooms available (e.g., Standard, Deluxe, Suite).
- Key Attributes: name, Price_Per_Day. This allows for flexible pricing without hard-coding it into each room.

3. Booking & Reservation Core:

- Reservation (likely represented by "Ressenvation card" and "Booking_id"): This is the central entity that records a booking.
- Key Attributes: Booking_id, CheckIn_date, CheckOut_date, card_id (possibly a folio number).
- Relationships: Connects a user to a Room for a specific date range.

ERD Entity Descriptions

4. Financial Transactions:

- transaction_Reference: Handles all payment activities.
- Key Attributes: pay_id, payment_Date, Amount, Payment_Methode (e.g., Credit Card, Cash).
- Purpose: Creates a record for every payment made, which can be linked to a reservation or a checkout bill.

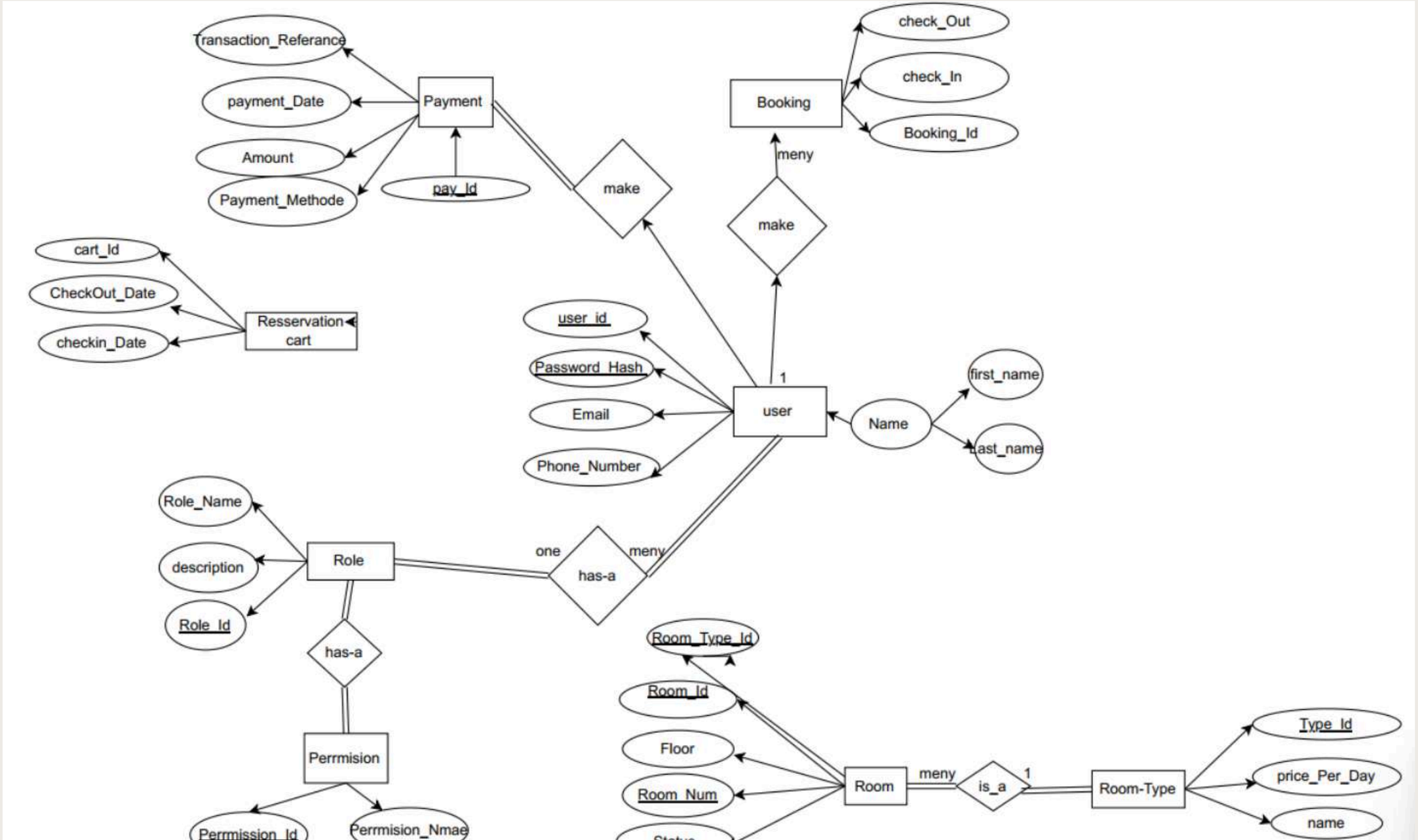
5. Checkout Process:

- check_Out: Manages the final settlement when a guest leaves.
- Key Attributes: checking_id, money (likely the final amount due).
- Relationships: Linked to a Booking_id to finalize the reservation and process any final payments.

6. Access Control & Security:

- Role: Defines different levels of system access (e.g., "Admin", "Receptionist", "Customer").
- Key Attributes: Role_id, role_name, description.
- Permission: Defines specific actions that can be performed within the system (e.g., "create_booking", "view_reports").
- Key Attributes: Permission_id, permission_Nmap (likely the name of the permission).
- The Relationship: The has-a relationship between Role and Permission is a classic many-to-many relationship. This means one role (e.g., "Admin") can have many permissions, and one permission (e.g., "edit_room") can be assigned to many roles. This is typically implemented with a junction table.

ERD(image)


Transaction_Referance
payment_Date
Amount
Payment_Methode
Payment
pay_Id
make
check_Out
check_In
Booking_Id
Booking
meny
make
cart_Id
CheckOut_Date
checkin_Date
Resservation cart
user_id
Password_Hash
Email
Phone_Number
user
1
Name
first_name
last_name
Role_Name
description
Role_Id
Role
one
meny
has-a
has-a
Perrmision
Perrmission_Id
Perrmision_Nmae
Room_Type_Id
Room_Id
Floor
Room_Num
Room
meny
is_a
1
Room-Type
Type_Id
price_Per_Day
name

Schema

Managing users, roles and permissions

User

User_ID	FirstName	LastName	Email	PhoneNumber	PasswordHash

Role

Role_ID	RoleName	Description

Permission

Permission_ID	Permission_Name

User_Role

(intermediate table for the Many-to-Many relationship between User and Role)

User_ID	Role_ID

Role_Permission

Permission and Role Many-to-Many Intermediate Relationship Table

Role_ID	Permission_ID

Reservations and Payments Management

Booking

Booking_ID	User_ID	Room_ID	Check_in	Check_out

Payment

Pay_ID	Booking_ID	Amount	PaymentDate	PaymentMethod	TransactionRefrence

ReservationCart

Cart_ID	User_ID	Room_ID	Checkin_Date	Checkout_Date

Room management and types

Room

Room_ID	RoomNumber	Floor	Status	RoomType_ID

RoomType

Type_ID	Name	PricePer_Day

Schema Relationship

User (User ID) → Booking (User ID)

User (User ID) → User Role (User ID)

User (User ID) → ReservationCart (User ID)

Role (Role ID) → User Role (Role ID)

Role (Role ID) → Role Permission (Role ID)

Permission (Permission ID) → Role Permission (Permission ID)

Booking (Booking ID) → Payment (Booking ID)

Room (Room ID) → Booking (Room ID)

Room (Room ID) → ReservationCart (Room ID)

Room Type (Type ID) → Room (RoomType ID)

UI/UX

Overall UI/UX Description

- Bookify presents a modern, clean, and user-centric hotel booking experience. The design focuses on visual appeal, intuitive navigation, and a seamless booking flow from discovery to confirmation.

UI/UX

Design System & Visual Identity

- Color Scheme: Clean and minimalist (likely whites, neutrals, with accent colors for buttons and highlights)
- Typography: Clear, readable fonts with hierarchical sizing for headings, subheadings, and body text
- Imagery: High-quality room photos and elegant visuals to showcase luxury and comfort
- Layout: Card-based design for rooms, consistent spacing, and a balanced visual structure

UI/UX

Key User Flows & Screens

1. Homepage / Landing Page

- Hero Section: "Find Your Perfect Stay" with a search widget (room type, guests, dates)
- Value Proposition: Brief welcome message with a "Book Now" call-to-action
- Featured Rooms: Grid of room cards with images, descriptions, prices, and "View Details" buttons

UI/UX

2. About Us Page

- Brand storytelling: Mission, Values, and Journey
- Focus on trust-building and emotional connection with users

3. Rooms Listing Page

- Display of all room types in a clean grid
- Each room includes:
- Image
- Name
- Brief description
- Price per night
- "View Details" & "Book Now" buttons

UI/UX

4. Booking / Checkout Flow

- Step 1: Guest Details
- Form for personal information (pre-filled if logged in)
- Special requests text area
- Step 2: Payment Information
- Payment method selection (Credit Card / PayPal)
- Card details form (secure input fields)
- Step 3: Booking Summary
- Clear breakdown of costs
- Final review before confirmation
- "Confirm Booking" button

5. Shopping Cart

- Displays selected rooms with dates and guest counts
- Edit/remove functionality
- Order summary with subtotal, taxes, and total
- "Confirm Booking" action

UI/UX

6. Authentication Pages

- Login: Simple form with email/password, remember me, and forgot password options
- Sign Up: Registration form with name, email, and password fields

7. Contact Page

- Contact form with name, email, and message fields
- Location details and contact information

UI/UX

UX Strengths & Notable Features

1.Progressive Disclosure:

- Simple search on homepage → detailed room views → booking form
- Step-by-step checkout process reduces cognitive load

2.Visual Hierarchy:

- Clear pricing information
- Prominent call-to-action buttons
- Consistent card design for room listings

3.User Convenience:

- Guest details pre-filling capability
- Multiple payment options
- Special requests field for personalization

4.Trust Elements:

- Detailed booking summary with cost breakdown
- Professional and polished visual design
- Comprehensive contact information

5.Responsive Design:

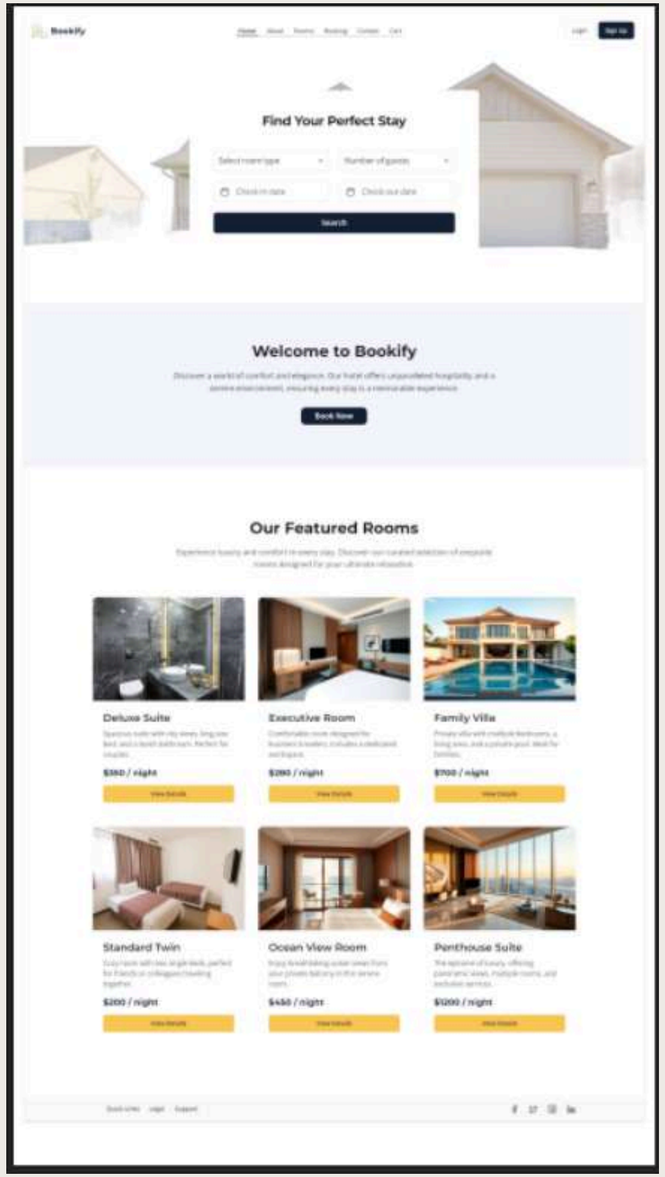
- Layout adapts well to different screen sizes (inferred from structure)

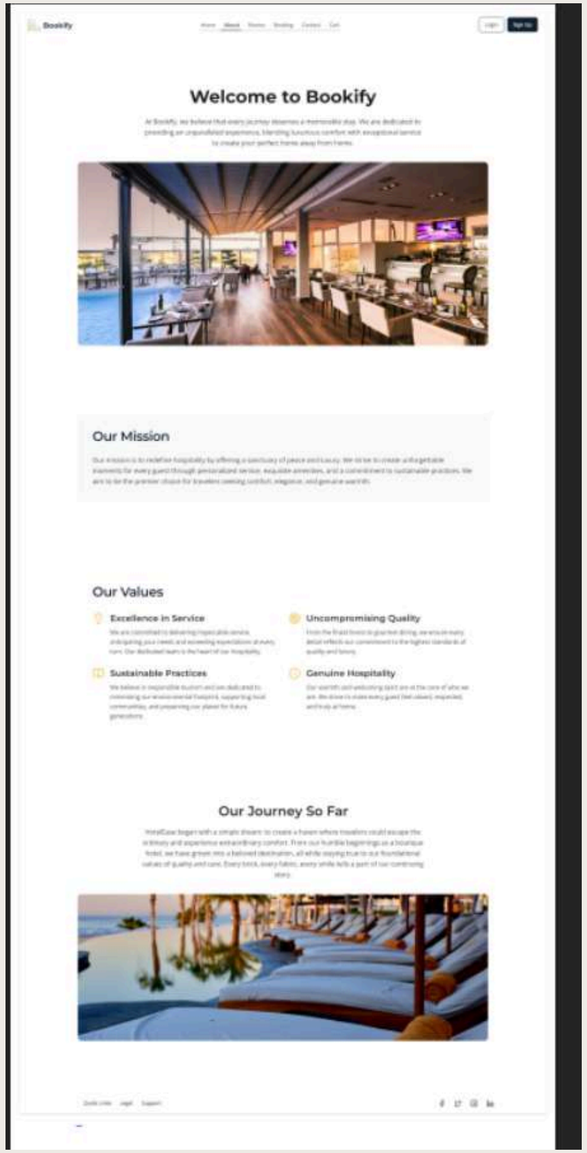
UI/UX

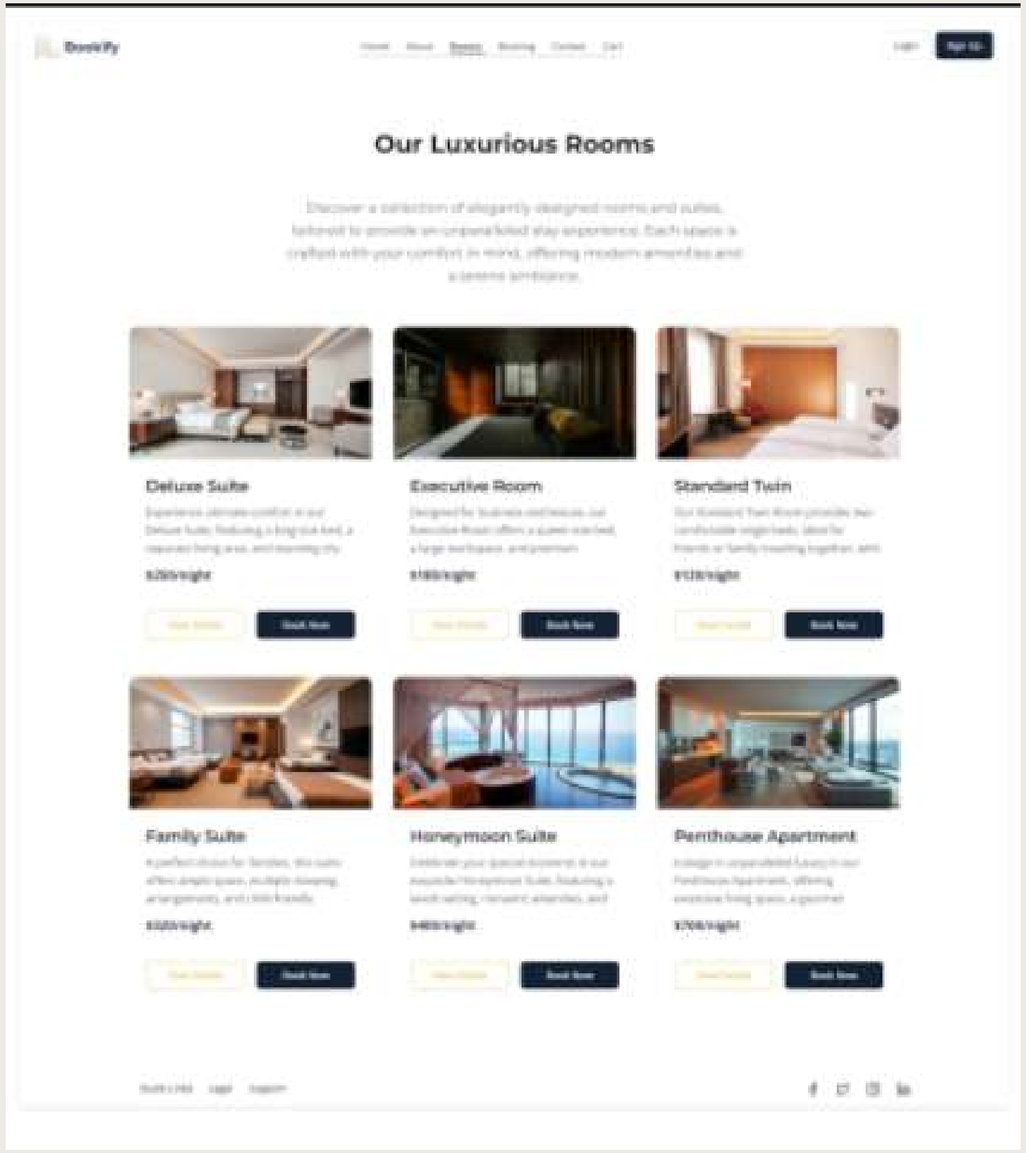
- **User Journey Map!!**

1. Discover: Land on homepage → browse featured rooms
2. Explore: View room details → compare options
3. Select: Choose room → add to cart or book directly
4. Book: Enter details → payment → confirmation
5. Manage: Access bookings through user account

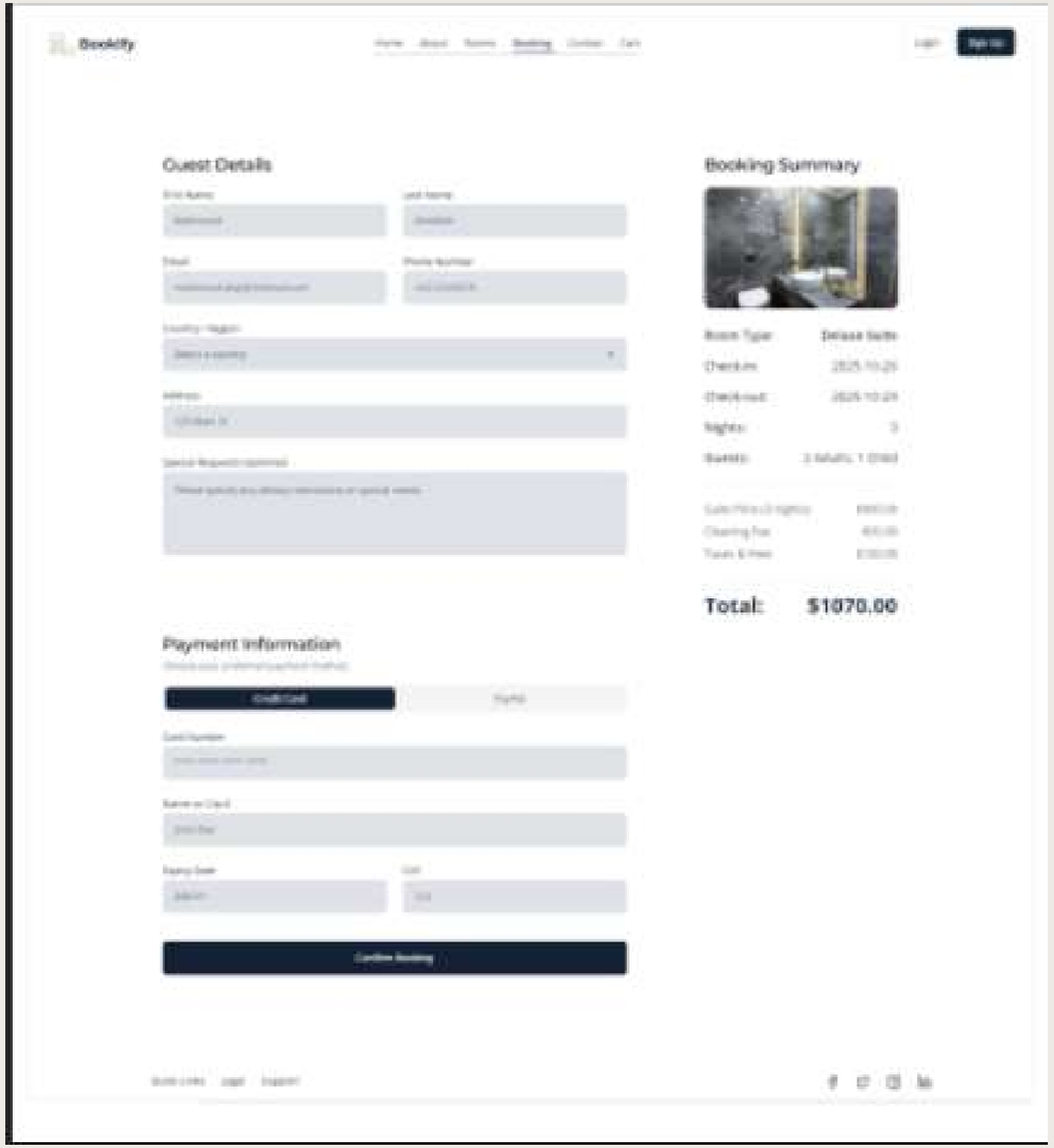
UI/UX design

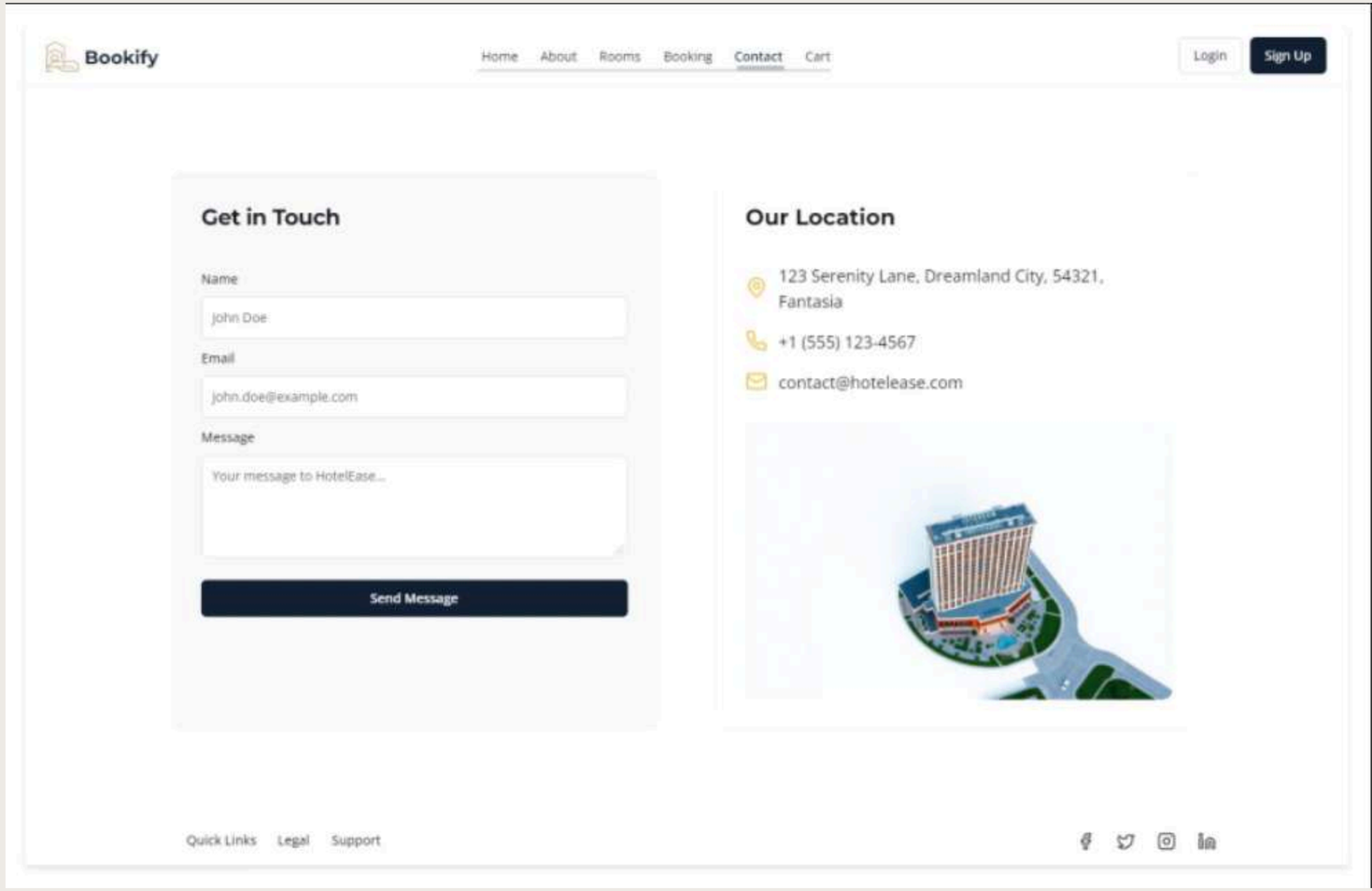

Find Your Perfect Stay
Welcome to Bookify
Our Featured Rooms
Deluxe Suite
Executive Room
Family Villa
Standard Twin
Ocean View Room
Penthouse Suite


Welcome to Bookify
Our Mission
Our Values
Excellence in Service
Uncompromising Quality
Sustainable Practices
Genuine Hospitality
Our Journey So Far

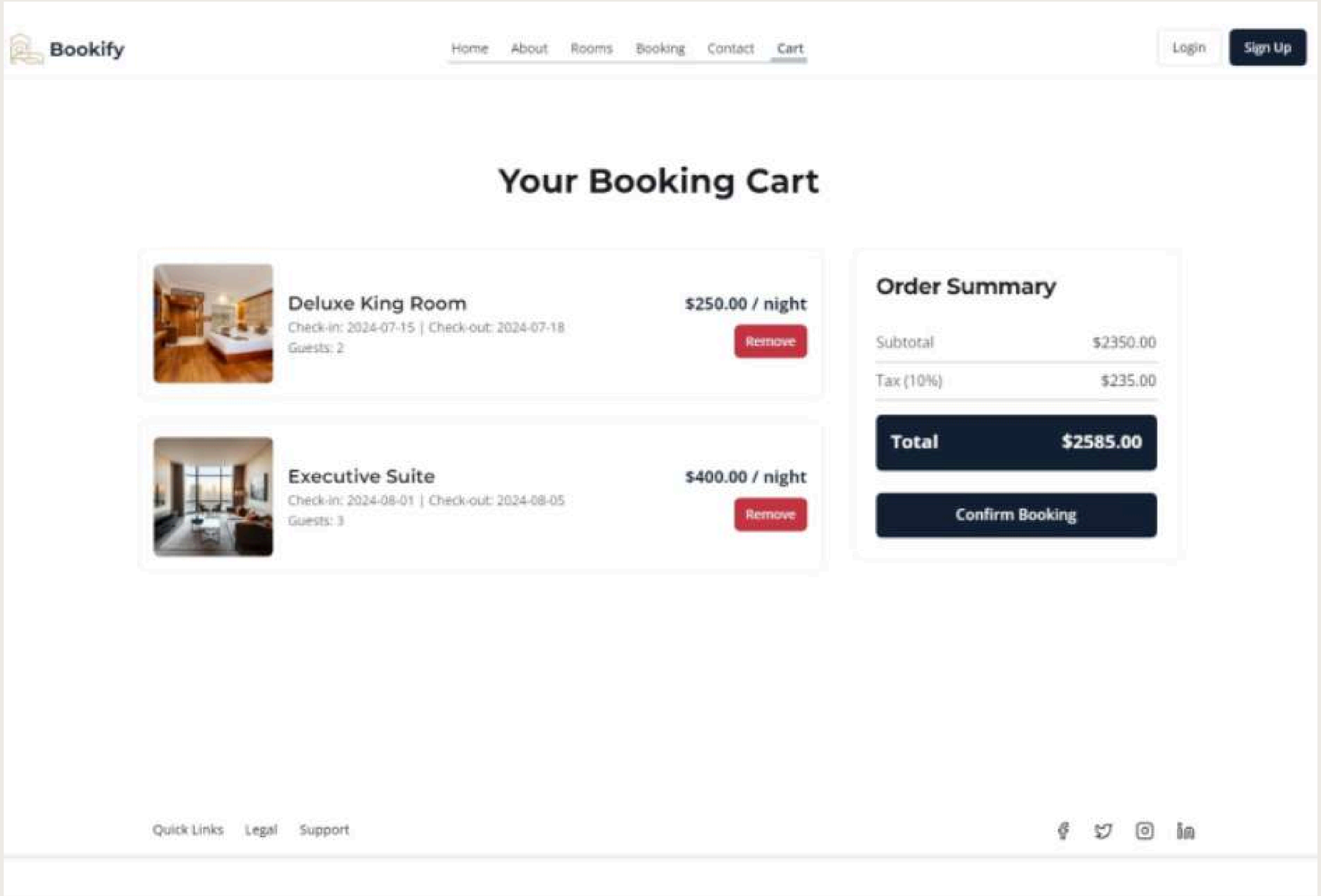

Our Luxurious Rooms
Deluxe Suite
Executive Room
Standard Twin
Family Suite
Honeymoon Suite
Penthouse Apartment

UI/UX design

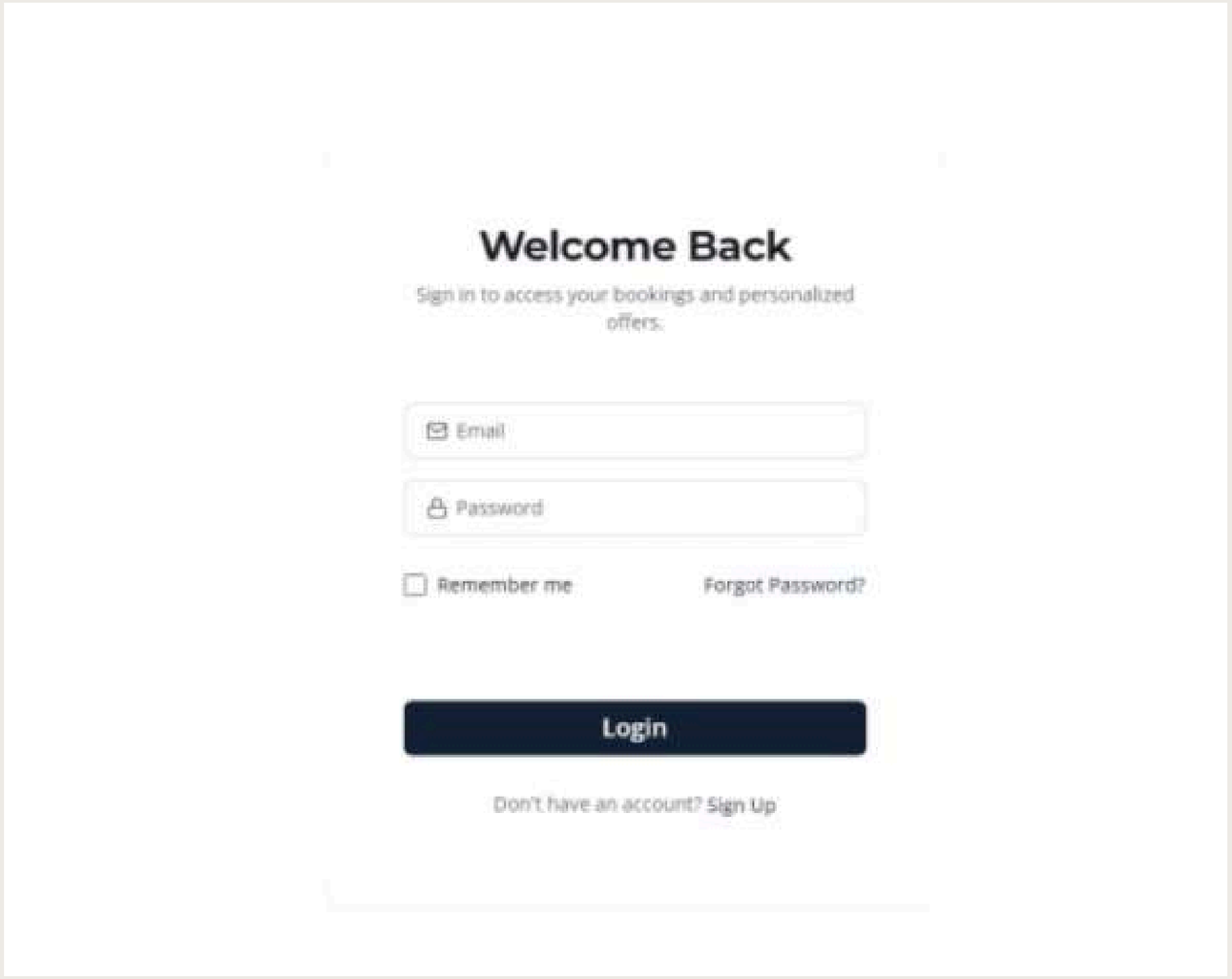

Guest Details
Booking Summary
Total: $1070.00
Payment Information

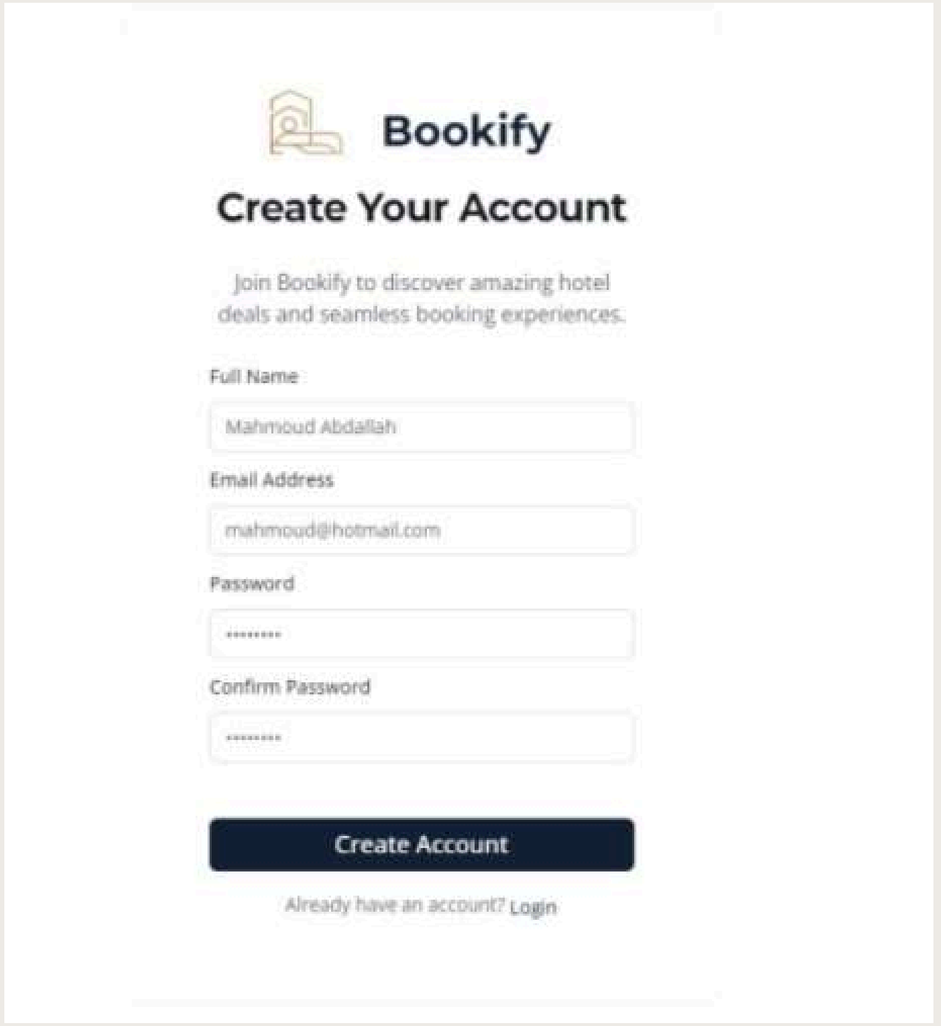

Bookify
Home
About
Rooms
Booking
Contact
Cart
Login
Sign Up
Get in Touch
Name
John Doe
Email
john.doe@example.com
Message
Your message to HotelEase...
Send Message
Our Location
123 Serenity Lane, Dreamland City, 54321, Fantasia
+1 (555) 123-4567
contact@hotelease.com
Quick Links
Legal
Support

UI/UX design


Bookify
Home
About
Rooms
Booking
Contact
Cart
Login
Sign Up
Your Booking Cart
Deluxe King Room
Check-in: 2024-07-15 | Check-out: 2024-07-18
Guests: 2
$250.00 / night
Remove
Executive Suite
Check-in: 2024-08-01 | Check-out: 2024-08-05
Guests: 3
$400.00 / night
Remove
Order Summary
Subtotal
$2350.00
Tax (10%)
$235.00
Total
$2585.00
Confirm Booking
Quick Links
Legal
Support

UI/UX design


Welcome Back
Sign in to access your bookings and personalized offers.
Email
Password
Remember me
Forgot Password?
Login
Don't have an account? Sign Up


Bookify
Create Your Account
Join Bookify to discover amazing hotel deals and seamless booking experiences.
Full Name
Mahmoud Abdallah
Email Address
mahmoud@hotmail.com
Password
Confirm Password
Create Account
Already have an account? Login

THANK YOU

PRESENTED BY MOHAMED ASHRAF