

Bookify Hotel Reservation System - Project Documentation

Project Overview

Bookify is a comprehensive hotel reservation web application built with ASP.NET Core MVC. It provides a complete booking management system for hotels, allowing customers to search for available rooms, view room details, make reservations, and process payments securely. The application also includes a powerful admin panel for hotel staff to manage rooms, bookings, and customer data.

Key Objectives

- Build a robust, scalable hotel booking platform
 - Implement clean architecture with separation of concerns
 - Ensure secure payment processing
 - Provide intuitive user experience for both customers and admins
 - Maintain data integrity through transactions
-

Architecture

The application follows **N-Tier Architecture** with clear separation of concerns:

Layers

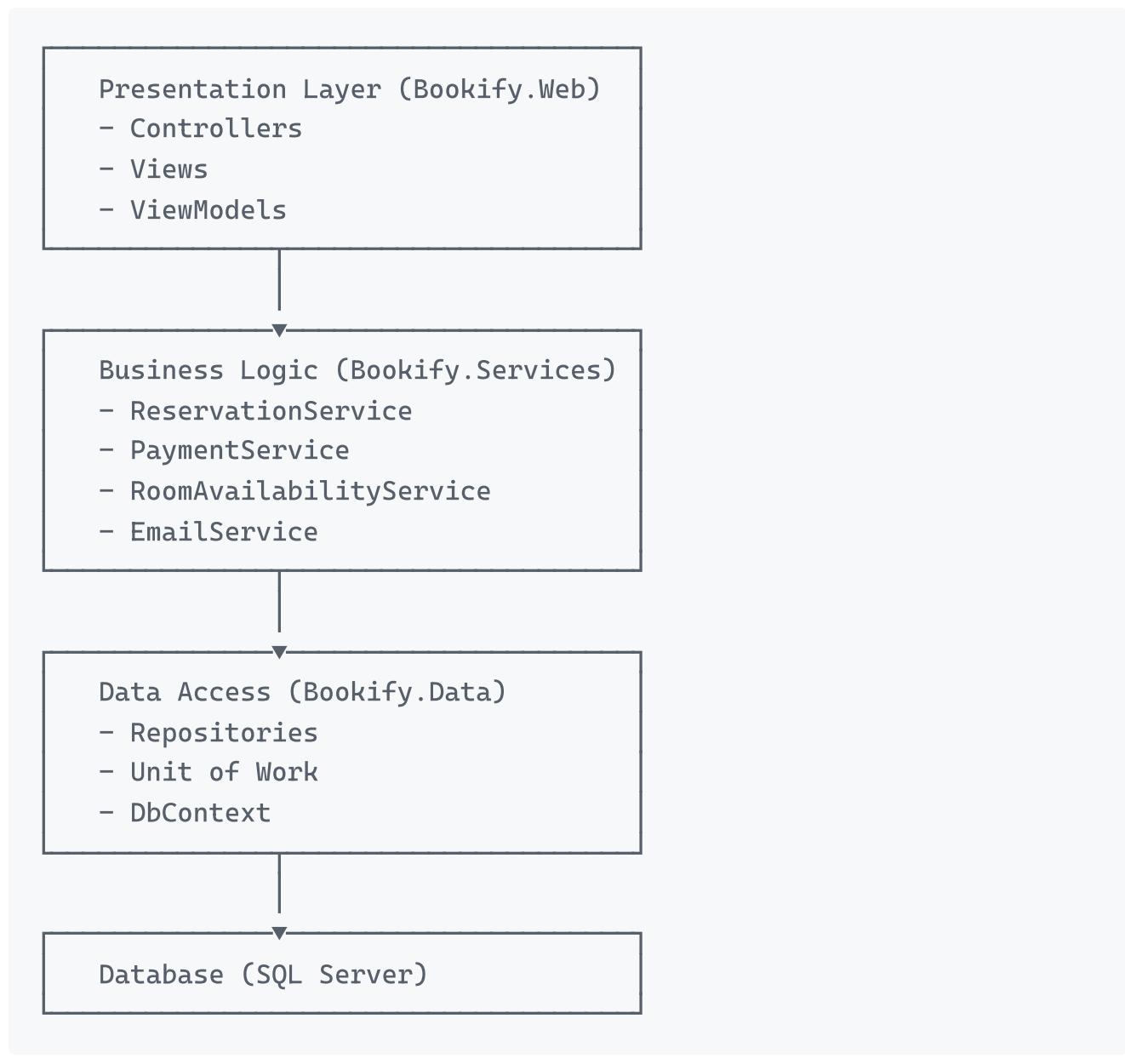
1. **Presentation Layer (Bookify.Web)**
 - ASP.NET Core MVC Controllers
 - Razor Views
 - ViewModels
 - Client-side assets (CSS, JavaScript)
 - Filters and Middleware
2. **Business Logic Layer (Bookify.Services)**
 - Service interfaces and implementations
 - Business rules and validations

- External service integrations (Stripe, SendGrid)

3. Data Access Layer (Bookify.Data)

- Entity Framework Core DbContext
- Repository Pattern implementation
- Unit of Work Pattern
- Entity configurations
- Database migrations

Architecture Diagram



Technology Stack

Backend

- **.NET 9.0** - Latest .NET framework
- **ASP.NET Core MVC** - Web framework
- **Entity Framework Core 9.0.11** - ORM
- **ASP.NET Core Identity** - Authentication & Authorization
- **SQL Server** - Database

Frontend

- **Bootstrap 5.3.0** - CSS framework
- **jQuery 3.7.1** - JavaScript library
- **DataTables 1.13.7** - Table plugin
- **Toastr.js** - Notification library
- **Font Awesome** - Icons

External Services

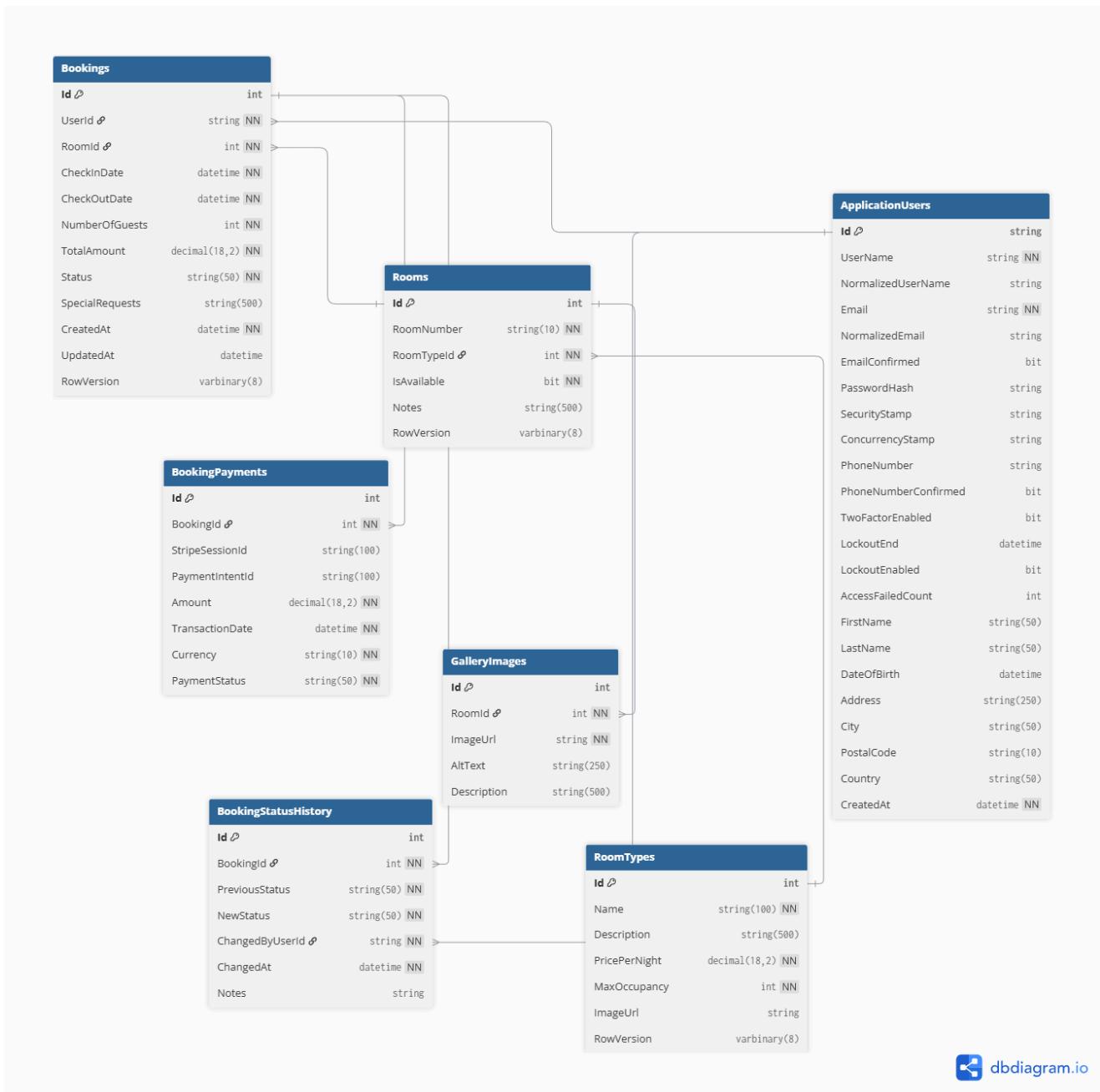
- **Stripe** - Payment gateway
- **SendGrid** - Email service

Development Tools

- **Serilog** - Structured logging
 - **Health Checks UI** - Application monitoring
 - **Entity Framework Migrations** - Database versioning
-

Database Schema

Entities



1. ApplicationUser (extends IdentityUser)

- Inherits from ASP.NET Identity
- Additional properties: FirstName, LastName, DateOfBirth, Address, City, PostalCode, Country

2. RoomType

- Id** (int, PK)
- Name** (string) - e.g., "Single", "Double", "Suite"
- Description** (string, nullable)
- PricePerNight** (decimal)
- MaxOccupancy** (int)
- ImageUrl** (string, nullable)

- `RowVersion` (byte[], for concurrency)

3. Room

- `Id` (int, PK)
- `RoomNumber` (string) - Unique room identifier
- `RoomTypeId` (int, FK → RoomType)
- `IsAvailable` (bool)
- `Notes` (string)
- `RowVersion` (byte[], for concurrency)

4. Booking

- `Id` (int, PK)
- `UserId` (string, FK → ApplicationUser)
- `RoomId` (int, FK → Room)
- `CheckInDate` (DateTime)
- `CheckOutDate` (DateTime)
- `NumberOfGuests` (int)
- `TotalAmount` (decimal)
- `Status` (BookingStatus enum)
- `SpecialRequests` (string, nullable)
- `CreatedAt` (DateTime)
- `UpdatedAt` (DateTime, nullable)
- `RowVersion` (byte[], for concurrency)

5. BookingPayment

- `Id` (int, PK)
- `BookingId` (int, FK → Booking)
- `Amount` (decimal)
- `PaymentStatus` (PaymentStatus enum)
- `PaymentIntentId` (string) - Stripe Payment Intent ID
- `TransactionId` (string, nullable)
- `PaymentDate` (DateTime)
- `RefundAmount` (decimal, nullable)
- `RefundDate` (DateTime, nullable)

6. BookingStatusHistory

- `Id` (int, PK)
- `BookingId` (int, FK → Booking)
- `PreviousStatus` (BookingStatus, nullable)
- `NewStatus` (BookingStatus)
- `ChangedBy` (string) - User ID
- `ChangedAt` (DateTime)
- `Notes` (string, nullable)

7. GalleryImage

- `Id` (int, PK)
- `RoomId` (int, FK → Room)
- `ImageUrl` (string)
- `AltText` (string, nullable)
- `DisplayOrder` (int)

Enums

BookingStatus

- `Pending` (0)
- `Paid` (1)
- `Cancelled` (2)
- `Completed` (3)

PaymentStatus

- `Pending` (0)
- `Completed` (1)
- `Failed` (2)
- `Refunded` (3)

Relationships

- RoomType → Rooms (One-to-Many)
- Room → Bookings (One-to-Many)
- Room → GalleryImages (One-to-Many)

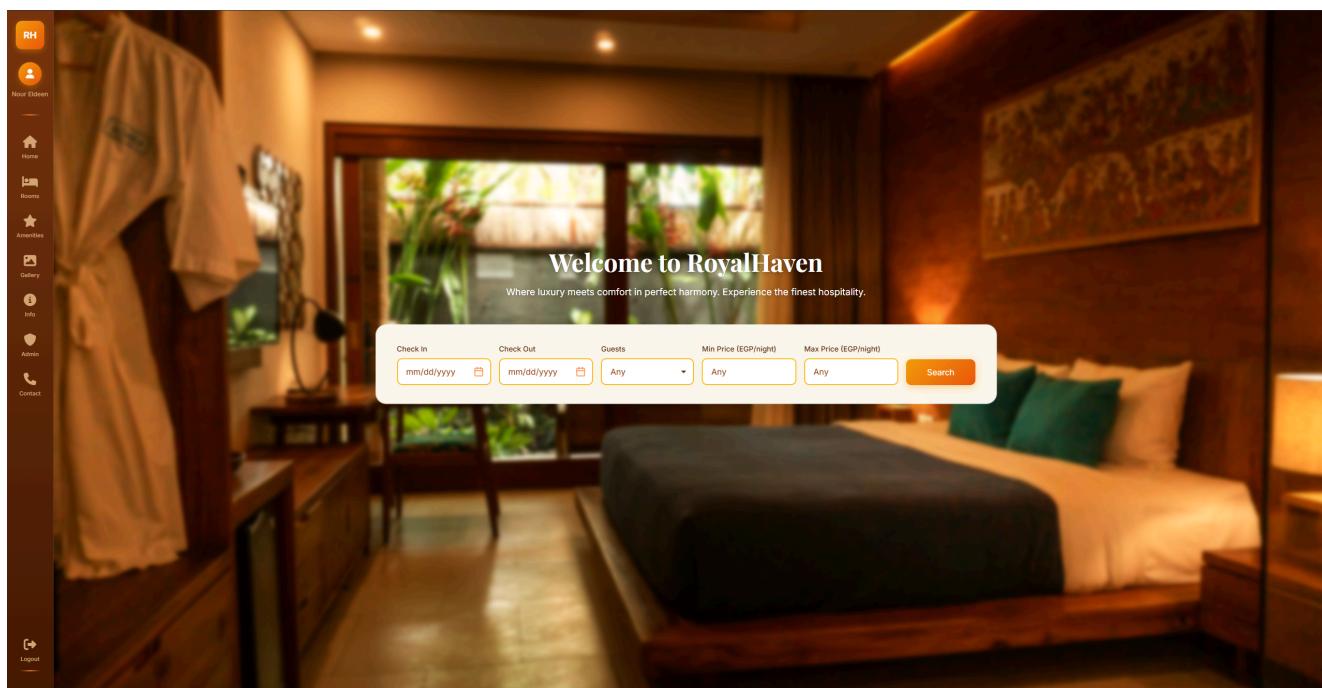
- ApplicationUser → Bookings (One-to-Many)
 - Booking → BookingPayments (One-to-Many)
 - Booking → BookingStatusHistory (One-to-Many)
-

Features

Customer Features

1. Room Search & Browsing

- Search available rooms by date range
- Filter by room type, price, occupancy
- View room details with images
- Real-time availability checking
- Pagination for large result sets



Our Rooms And Suites

Discover elegantly designed accommodations that blend comfort with sophistication

Room Type	Description	Price	Guest Capacity	Area	Services	Action
Single Room	Comfortable single room with one bed, perfect for solo travelers.	3,000 EGP per night	1 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Single Room	Comfortable single room with one bed, perfect for solo travelers.	3,000 EGP per night	1 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Single Room	Comfortable single room with one bed, perfect for solo travelers.	3,000 EGP per night	1 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Double Room	Spacious double room featuring two single beds.	3,500 EGP per night	2 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Double Room	Spacious double room featuring two single beds.	3,500 EGP per night	2 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Deluxe Suite	Luxurious suite with a king-size bed and private balcony.	10,000 EGP per night	4 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Deluxe Suite	Luxurious suite with a king-size bed and private balcony.	10,000 EGP per night	4 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details

1 2 Next

Our Rooms And Suites

Discover elegantly designed accommodations that blend comfort with sophistication

Single Room – Room Details
Room #202 | 1 Guests

Single Room
3,000 EGP / night
Max 1 guests

Included Services
✓ WiFi ✓ Breakfast

Amenities
Free WiFi | Flat-screen TV | Air Conditioning | Mini Bar | Room Service

[Book Now](#)

Room Type	Description	Price	Guest Capacity	Area	Services	Action
Single Room	Comfortable single room with one bed, perfect for solo travelers.	3,000 EGP per night	1 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Double Room	Spacious double room featuring two single beds.	3,500 EGP per night	2 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Double Room	Spacious double room featuring two single beds.	3,500 EGP per night	2 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Deluxe Suite	Luxurious suite with a king-size bed and private balcony.	10,000 EGP per night	4 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details
Deluxe Suite	Luxurious suite with a king-size bed and private balcony.	10,000 EGP per night	4 Guests	35 m ²	WiFi, Breakfast, TV, AC	View Details

2. User Authentication

- User registration with validation
- Secure login with password requirements
- Password reset functionality
- Account lockout after failed attempts

The image shows a mobile application's login screen. On the left is a dark-themed background image of a hotel room interior with a bed and greenery visible through a window. Overlaid on the top left is a white rectangular box containing the text "Welcome Back!" and "Sign in to continue your journey with RoyalHaven". To the right of this is a white login form with a light orange header that says "Login" and "Enter your credentials to continue". The form contains two text input fields: one for email with "Nour@nour.com" and one for password with "*****". Below these are "Remember me" and "Forgot password?" checkboxes. A large orange "Sign In" button is at the bottom. At the very bottom of the screen, there is small text: "Don't have an account? Create an Account".

The image shows a mobile application's account creation screen. On the left is a dark-themed background image of a hotel room interior with a bed and greenery visible through a window. Overlaid on the top left is a white rectangular box containing the text "Welcome, Sir!" and "Create your account to join RoyalHaven and start your journey.". To the right of this is a white registration form with a light orange header that says "Create Account" and "Fill in the form". The form has two text input fields: one for first name with "Hazem" and one for last name with "Hany". It also has two email input fields: one for "Hazem" with "Hazem@hazem.com" and one for "Hany" with "*****". Below these are two password input fields with "*****" and "*****". A large orange "Create Account" button is at the bottom. At the very bottom of the screen, there is small text: "Already have an account? Back to Login".

3. Booking Management

- View booking summary before checkout
- Complete booking with payment
- View booking history

Booking Summary

Single Room
Room #202

Booking Details

Check-in Date: 12/01/2025

Check-out Date: 12/02/2025

Number of Guests: 1
Men 1 guests

Nights: 1

Description: Comfortable single room with one bed, perfect for solo travelers.

Included Services
WiFi, Breakfast, Air Conditioning

Amenities
Free WiFi, Flat-screen TV, Air Conditioning, Mini Bar, Room Service, Private Bathroom, Safe

Price Breakdown

3,000 EGP × 1 nights	3,000 EGP
Tax (14%)	420 EGP
Total	3,420 EGP

Complete Your Booking

Special Requests (Optional)
Private massage

Payment Information

Enter your card details to complete the booking

Card number MM / YY CVC

Confirm & Pay 3,420 EGP

4. Payment Processing

- Secure payment via Stripe Payment Intent
- Payment confirmation emails

Booking Confirmed!

Thank you for your booking. Your reservation has been confirmed and a confirmation email has been sent to your email address.

Booking ID #4	Room Double Room	Room Number 305
Check-in Date Nov 30, 2025	Check-out Date Dec 01, 2025	Number of Guests 2
Number of Nights 1	Status Paid	
Special Requests None		

Payment Summary

3,500 EGP × 1 nights	3,500 EGP
Tax (14%)	490 EGP
Total Paid	3,990 EGP

View My Bookings **Back to Home**

5. User Profile

- View and update personal information
- View booking history
- Change password

The screenshot shows the Admin Dashboard interface. On the left, a vertical sidebar menu includes Home, Bookings, Analytics, Info, Admin, and Contact, with a Logout button at the bottom. The main content area has a header for 'Nour Eldeen Mahmoud' with a profile picture, email (nor.abdelmaksoud2004@gmail.com), and member status (Member since November 2025, 20 years old, mansoura, Egypt). The 'Profile Information' section contains fields for First Name (Nour Eldeen), Last Name (Mahmoud), Date of Birth (12/20/2004), Email Address (nor.abdelmaksoud2004@gmail.com), Address (mansoura), City (mansoura), Postal Code (1111), and Country (Egypt). A 'Save Changes' button is present. To the right, a summary box shows 'Total Bookings' (3), 'Total Spent' (416670 EGP), and 'Account Information' with status (Active), verification (Email Verified), and membership since (Nov 2025). Below this is a 'Quick Actions' section with links for Search Rooms, Change Password, and Logout. The 'My Bookings' section displays an upcoming booking for a Double Room from Nov 30, 2025, to Dec 01, 2025, for 2 guests, totaling 3,990 EGP. Buttons for Cancel Booking and View Details are shown.

Admin Features

1. Dashboard

- Overview statistics:
 - Total bookings
 - Pending/Paid bookings
 - Total rooms
 - Available rooms
 - Occupancy rate
 - Revenue (last 6 months)
- Recent bookings table
- Quick action cards

The Admin Dashboard displays various performance metrics and charts:

- Dashboard:** Total Bookings (3), Total Rooms (20), Available Rooms (15), Total Revenue (\$416,670.00), and Occupancy Rate (25%).
- Revenue Chart (Last 6 Months):** Shows a steady upward trend from June 2025 to November 2025.
- Occupancy Chart (Last 6 Months):** Shows occupancy rates fluctuating between 10% and 100% over the same period.
- Recent Bookings:** A table showing a recent booking for Nour Eldeen Mahmoud in room 305.

2. Room Management

- Create, Read, Update, Delete (CRUD) for rooms
- Manage room availability
- Add notes to rooms

The Rooms Management page lists 10 rooms:

Room Number	Room Type	Availability	Current Booking	Notes	Actions
101	Single Room	Unavailable	No active or upcoming booking	Ground floor; near elevator	
102	Single Room	Unavailable	Booked from 11/30/2025 to 12/01/2025	Ground floor; quiet area	
103	Single Room	Available	No active or upcoming booking	Under maintenance	
201	Single Room	Available	No active or upcoming booking	Second floor; city view	
202	Single Room	Available	No active or upcoming booking	Second floor; garden view	
301	Double Room	Available	No active or upcoming booking	Third floor; balcony	
302	Double Room	Unavailable	No active or upcoming booking	Third floor; sea view	
303	Double Room	Available	No active or upcoming booking	Currently occupied	
304	Double Room	Available	No active or upcoming booking	Third floor; corner room	
305	Double Room	Unavailable	Booked from 11/30/2025 to 12/01/2025	Third floor; premium view	

3. Room Type Management

- CRUD operations for room types
- Set pricing per night
- Configure max occupancy

Room Types

ID	Name	Description	Price Per Night	Max Occupancy	Image	Actions
1	Single Room	Comfortable single room with one bed, perfect for ...	\$3,000.00	1		
2	Double Room	Spacious double room with two beds, ideal for coup...	\$3,500.00	2		
3	Deluxe Suite	Luxurious suite with separate living area and prem...	\$10,000.00	4		
4	Family Room	Large family room with multiple beds, perfect for ...	\$50,000.00	5		
5	Presidential Suite	Ultra-luxurious presidential suite with all premiu...	\$20,000.00	6		
7	test room	test room	\$2,000.00	1	No Image	

Show 10 entries Search:

[CSV](#) [Excel](#) [PDF](#) [Print](#) Previous 1 Next

4. Booking Management

- View all bookings
- Filter by status, date, customer
- Update booking status
- Process refunds
- View booking details and payment history

Bookings Management

Booking ID	Customer Name	Room Number	Check-in Date	Check-out Date	Number of Guests	Total Amount	Status	Created Date	Actions
2	Nour Eideen Mahmoud	102	2025-11-30	2025-12-01	1	\$2,280.00	Completed	2025-11-29	
3	Nour Eideen Mahmoud	702	2025-11-30	2025-12-18	6	\$410,400.00	Cancelled	2025-11-29	
4	Nour Eideen Mahmoud	305	2025-11-30	2025-12-01	2	\$3,990.00	Paid	2025-11-29	

Show 10 entries Search:

[CSV](#) [Excel](#) [PDF](#) [Print](#) Previous 1 Next

5. User Management

- View all users
- Lock/unlock user accounts

RoyalHaven
Luxury Redefined

Quick Links

- Home
- Rooms
- Amenities
- Gallery

Connect With Us

reservations@royalhaven.com
+1 (555) 123-4567

[Facebook](#) [Instagram](#) [Twitter](#) [LinkedIn](#)

- View user booking history
- Manage user roles

User ID	Full Name	Email	Phone Number	Role	Total Bookings	Created Date	Status	Actions
2223d1e8...	Admin User	admin@bookify.com	N/A	Admin	0	2025-11-29	Active	<button>Edit</button>
d4b0f813...	Nour Eldeen Mahmoud	nor.abdelmaksoud2004@gmail.com	N/A	Admin	3	2025-11-29	Active	<button>Edit</button>
e10efbf2...	Nour Abdelmaksoud	noureideendev@gmail.com	N/A	Customer	0	2025-11-29	Active	<button>Edit</button>

Security

Authentication & Authorization

- **ASP.NET Core Identity** for user management
- **Role-Based Access Control (RBAC)**
 - Admin role for admin panel access
 - Customer role for regular users
- **Password Requirements:**
 - Minimum 8 characters
 - Requires digit, lowercase, uppercase, non-alphanumeric
 - At least 1 unique character

Account Security

- **Account Lockout:** 5 failed attempts → 10-minute lockout
- **Automatic vs Manual Lockout:**
 - Automatic: Failed login attempts → redirects to `/Account/Lockout`
 - Manual: Admin-imposed → redirects to `/Account/AccessDenied`
- **Cookie Security:**
 - HttpOnly cookies

- SameSite=Strict
- Secure policy
- 30-day expiration with sliding expiration

CSRF Protection

- **AutoValidateAntiforgeryTokenAttribute** on all POST requests
- Custom filter to exclude API routes
- Anti-forgery tokens in forms

Data Protection

- **Concurrency Control:** RowVersion (byte[]) for optimistic concurrency
- **Input Validation:** Data Annotations on ViewModels
- **SQL Injection Prevention:** Parameterized queries via EF Core
- **XSS Prevention:** Razor encoding by default

File Upload Security

- Maximum file size: 10MB
 - File type validation
 - Secure file storage
-

External Integrations

1. Stripe Payment Gateway

- **Integration Type:** Payment Intent API (inline checkout)
- **Features:**
 - Create payment intents
 - Process payments
 - Handle payment confirmations
 - Process refunds
 - Idempotency support
- **Configuration:** Secret Key, Publishable Key in `appsettings.json`

2. SendGrid Email Service

- **Purpose:** Transactional emails
- **Features:**
 - Booking confirmation emails
 - Payment confirmation emails
 - Password reset emails
- **Configuration:** API Key, FromEmail, FromName in `appsettings.json`

3. Health Checks

- **Database Health Check:** Verifies SQL Server connectivity
- **Email Health Check:** Validates SendGrid API key and configuration
- **Payment Health Check:** Validates Stripe API key and connectivity
- **UI:** Available at `/health-ui`

Project Structure

```

Bookify-Dev/
  └── Bookify.Data/
    ├── Data/
    │   ├── ApplicationDbContext.cs
    │   ├── Configurations/          # EF Core configurations
    │   ├── Enums/
    │   └── Seeding/                # BookingStatus, PaymentStatus
    ├── Models/
    └── Repositories/
      ├── Interfaces/
      ├── Implementations/
      ├── IUnitOfWork.cs
      └── UnitOfWork.cs            # Database seeders
    └── Migrations/               # Entity models

  └── Bookify.Services/
    ├── Interfaces/
    └── Services/                 # Business Logic Layer
      # Service interfaces
      # Service implementations

  └── Bookify.Web/
    ├── Controllers/             # Presentation Layer
    ├── Views/                   # MVC Controllers
    └── ViewModels/              # Razor views
      # View models

```

```
Filters/                                # Action filters
HealthChecks/                            # Health check implementations
wwwroot/
  css/
  js/
  images/
Program.cs                               # Application entry point
```

Design Patterns

1. Repository Pattern

- **Purpose:** Abstract data access logic
- **Implementation:**
 - `IRepository<T>` - Generic repository interface
 - `IRoomRepository` - Specific repository for rooms
 - `IBookingRepository` - Specific repository for bookings
- **Benefits:** Testability, maintainability, flexibility

2. Unit of Work Pattern

- **Purpose:** Manage transactions and coordinate repositories
- **Implementation:** `IUnitOfWork` and `UnitOfWork`
- **Benefits:**
 - Ensures atomic operations
 - Single database context per request
 - Transaction management

3. Dependency Injection

- **Purpose:** Loose coupling, testability
- **Implementation:** Built-in ASP.NET Core DI container
- **Lifetime:** Scoped (per request)

4. ViewModel Pattern

- **Purpose:** Separate presentation from domain models

- **Examples:** `RoomViewModel`, `RoomTypeViewModel`, `CheckoutViewModel`
- **Benefits:** Prevents over-posting, better validation

5. Service Layer Pattern

- **Purpose:** Encapsulate business logic
 - **Implementation:** Services in `Bookify.Services`
 - **Benefits:** Reusability, testability, separation of concerns
-

Controllers & Actions

HomeController

- `Index()` - Home page with room listings
- `GetRoomsPartial()` - AJAX endpoint for room listings
- `SearchRooms()` - Search and filter rooms

AccountController

- `Register()` - User registration
- `Login()` - User authentication
- `Logout()` - User sign out
- `ForgotPassword()` - Password reset request
- `ResetPassword()` - Password reset confirmation
- `Lockout()` - Account lockout page
- `AccessDenied()` - Access denied page

RoomsController

- `Details(int id)` - Room details page

BookingsController

- `Index()` - User's booking history
- `Details(int id)` - Booking details

PaymentController

- `CreatePaymentIntent()` - Create Stripe payment intent

- `ConfirmPayment()` - Confirm payment and create booking
- `Refund(int bookingId)` - Process refund

ProfileController

- `Index()` - User profile page
- `UpdateProfile()` - Update user information
- `ChangePassword()` - Change user password

AdminController

- `Dashboard()` - Admin dashboard
- `RoomTypes()` - List room types
- `CreateRoomType()` - Create new room type
- `EditRoomType(int id)` - Edit room type
- `DeleteRoomType(int id)` - Delete room type
- `Rooms()` - List rooms
- `CreateRoom()` - Create new room
- `EditRoom(int id)` - Edit room
- `DeleteRoom(int id)` - Delete room
- `Bookings()` - List all bookings
- `BookingDetails(int id)` - Booking details
- `UpdateBookingStatus()` - Update booking status
- `Users()` - List all users
- `LockUser(string id)` - Lock user account
- `UnlockUser(string id)` - Unlock user account
- `GetRecentBookings()` - AJAX endpoint for recent bookings

Setup & Installation

Prerequisites

- .NET 9.0 SDK
- SQL Server (LocalDB or full instance)
- Visual Studio 2022 or VS Code
- Stripe account (for payments)

- SendGrid account (for emails)

Steps

1. Clone the repository

```
git clone <repository-url>
cd Bookify-Dev
```

2. Configure Database

- Update connection string in `appsettings.json`:

```
"ConnectionStrings": {
    "DefaultConnection": {

        "Server=localhost;Database=BookifyDB;Trusted_Connection=True;
        TrustServerCertificate=True;"}
}
```

3. Run Migrations

```
cd Bookify.Web
dotnet ef database update --project ../Bookify.Data
```

4. Configure External Services

- Update `appsettings.json` with Stripe keys:

```
"Stripe": {
    "PublishableKey": "pk_test_...",
    "SecretKey": "sk_test_..."
}
```

- Update SendGrid configuration:

```
"SendGrid": {
    "ApiKey": "SG....",
    "FromEmail": "your-email@example.com",
    "FromName": "Bookify"
}
```

5. Run the Application

```
dotnet run --project Bookify.Web
```

6. Access the Application

- Customer site: <https://localhost:7293>
- Admin panel: <https://localhost:7293/Admin/Dashboard>

- Health checks UI: <https://localhost:7293/health-ui>

Default Admin Account

- Created via database seeding in Development environment
 - Check `IdentitySeeder.cs` for credentials
-

Configuration

appsettings.json Structure

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "Microsoft.AspNetCore": "Warning"  
    }  
  },  
  "SendGrid": {  
    "ApiKey": "SG...",  
    "FromEmail": "email@example.com",  
    "FromName": "Bookify"  
  },  
  "Stripe": {  
    "PublishableKey": "pk_test_...",  
    "SecretKey": "sk_test_..."  
  },  
  "ConnectionStrings": {  
    "DefaultConnection": "Server=...;Database=...;..."  
  },  
  "AllowedHosts": "*"  
}
```

Serilog Configuration

- Logs to console and file
 - File location: `logs/bookify-YYYYMMDD.log`
 - Retention: 30 days
 - Rolling interval: Daily
-

Future Enhancements

Potential Features

1. **Email Confirmation:** Require email verification
2. **Two-Factor Authentication:** Add 2FA support
3. **Reviews & Ratings:** Customer reviews for rooms
4. **Loyalty Program:** Points and rewards system
5. **Multi-language Support:** i18n implementation
6. **Mobile App:** Native mobile application
7. **Real-time Notifications:** SignalR integration
8. **Advanced Analytics:** Detailed reporting and analytics
9. **Room Recommendations:** AI-based suggestions
10. **Calendar View:** Visual booking calendar

Technical Improvements

1. **Unit Tests:** Add comprehensive unit tests
 2. **Integration Tests:** End-to-end testing
 3. **API Versioning:** RESTful API with versioning
 4. **Caching:** Redis caching for performance
 5. **Background Jobs:** Hangfire for scheduled tasks
 6. **Microservices:** Split into microservices if needed
-

Support & Contact

For questions or issues:

- Check the codebase documentation
 - Review the `req.txt` file for original requirements
 - Contact the development team
-

Acknowledgments

- ASP.NET Core team
- Stripe for payment processing

- SendGrid for email services
 - All open-source contributors
-

Document Version: 1.0

Last Updated: 30/11/2025

- **Maintained By:**

- Nour Eldeen Mahmoud (Team Leader / Back End | [GitHub](#))
- Hazem Hany Samy (Front End | [GitHub](#))
- Omar Ashraf (Front End | [GitHub](#))
- Shahd Ayman (Back End | [GitHub](#))
- Alia Harb (Back End | [GitHub](#))
- Mohamed Soultan (Back End | [LinkedIn](#))