Hotel Reservation System — Documentation

1. Project Overview

The **Hotel Reservation System** is a web application that allows users to search for hotels, view available rooms, and make reservations.

- **Users**: Request bookings for rooms.
- **Admins**: Manage hotels, rooms, and confirm or reject reservations.

This project simulates real hotel booking systems while keeping it simple and practical.

2. Roles

👤 User

- Register and login.
- Search hotels by city and dates.
- View hotel details and room availability.
- Submit a reservation request.
- View booking status (Pending, Confirmed, Rejected, Cancelled, Completed).
- Cancel confirmed bookings.
- (Optional) Leave reviews after a completed stay.

🛠️ Admin

- Login with admin account.
- Manage hotels and rooms.
- Manage availability (number of rooms per date).
- View all reservations.
- **Confirm or reject reservations** made by users.
- Cancel reservations if needed.
- View simple analytics (optional).

3. Data Models

User

- id
- name
- email
- password_hash
- role (ADMIN | USER)
- created_at

Hotel

- id
- name
- city
- address
- description
- stars
- thumbnail_url
- created_at

RoomType

- id
- hotel_id
- name (e.g., Deluxe, Suite)
- capacity
- bed_type
- base_price
- description

RoomInventory

- id
- room_type_id
- date
- total_rooms
- sold_rooms

Booking

- id
- user_id
- hotel_id
- room_type_id
- check_in
- check_out
- nights
- total_price
- status (PENDING | CONFIRMED | REJECTED | CANCELLED | COMPLETED)
- created_at

Review (optional)

- id
- user_id
- hotel_id
- rating (1–5)
- comment
- created_at

4. Features

Core

- User authentication (register/login).
- Search hotels by city and dates.
- Hotel details with rooms and prices.
- Reservation requests created by users.
- Admin confirmation/rejection of reservations.
- Prevent double booking by tracking availability.
- User booking history with statuses.

Optional

- Reviews and ratings for hotels.
- Extra role: **Receptionist** (confirm bookings but not manage hotels).
- Simple analytics for Admin (revenue, occupancy).

---

5. Backend (APIs with .NET)

Auth

- POST /api/auth/register → Register new user.
- POST /api/auth/login → Login and get token.

Hotels

- GET /api/hotels → List hotels (searchable).
- GET /api/hotels/{id} → Hotel details.
- POST /api/admin/hotels → Add hotel (Admin).
- PUT /api/admin/hotels/{id} → Update hotel.
- DELETE /api/admin/hotels/{id} → Delete hotel.

Room Types

- POST /api/admin/hotels/{id}/room-types → Add room type.
- PUT /api/admin/room-types/{id} → Update room type.
- DELETE /api/admin/room-types/{id} → Delete room type.

Availability

- GET /api/room-types/{id}/availability?from=...&to=...
- PUT /api/admin/room-inventory → Update inventory for rooms.

Bookings

- POST /api/bookings → Create **PENDING** booking.
- GET /api/me/bookings → User's bookings.
- PATCH /api/bookings/{id}/cancel → Cancel a confirmed booking.

Admin only

- GET /api/admin/bookings?status=PENDING → List pending bookings.
- PATCH /api/admin/bookings/{id}/confirm → Confirm booking.
- PATCH /api/admin/bookings/{id}/reject → Reject booking.

---

6. Frontend (HTML, CSS, Bootstrap, JS)

User Interface

- **Home/Search page**: search by city + date.
- **Hotel details**: photos, description, room types, availability.
- **Booking page**: select room, choose dates, submit request.
- **My Bookings**: list of bookings with statuses and cancel option.

Admin Dashboard

- **Login page**.
- **Dashboard home**: pending reservations.
- **Hotels management**: add/edit/delete hotels.
- **Room management**: add/edit/delete room types.
- **Inventory management**: set room availability.
- **Bookings**: confirm/reject user reservations, cancel if needed.

Responsive design using **Bootstrap grid system**.

7. Workflow Description

1. User registers or logs in.
2. User searches hotels by city + dates.
3. User views hotel details and chooses a room.
4. User submits booking → booking is saved as **PENDING**.
5. Admin dashboard shows **Pending reservations**.
6. Admin **confirms or rejects** reservation.
    - If confirmed → status = CONFIRMED, rooms count updated.
    - If rejected → status = REJECTED, user notified.
7. User sees updated status in "My Bookings".
8. User can cancel a **Confirmed** booking (availability is restored).
9. After check-out, booking marked as **Completed**.

---

8. Optional Extensions

Reviews and Rating

- After a **Completed** booking, users can rate the hotel (1–5 stars) and leave comments.
- Display average rating on hotel page.

Receptionist Role

- Can view bookings for assigned hotel.
- Can confirm/reject bookings (like Admin).
- Cannot create or edit hotels/rooms.