

# Assignment Project: Simple Hotel Booking System (Procedural PHP)

---

## 1. Project Brief

---

Build a **Simple Hotel Booking System** using:

- PHP (Procedural style only, no OOP frameworks)
- HTML
- CSS
- JavaScript
- MySQL (or MariaDB)

This project should simulate a small hotel website where users can view rooms, check availability, and submit a booking request.

## 2. Goal

---

The main goal is to practice:

- Writing clean procedural PHP code
- Connecting PHP with MySQL
- Building forms with validation
- Performing CRUD operations
- Creating a full mini project from frontend to backend

## 3. Core Features (Required)

---

### A. Public Side (Customer)

1. **Home Page**
  - Hotel name and short description
  - Navigation links
2. **Rooms Page**
  - Display all available rooms from database
  - Show: room name/type, price per night, capacity, description, and image
3. **Booking Form Page**
  - Customer inputs:
    - Full name
    - Email
    - Phone number
    - Room selection
    - Check-in date
    - Check-out date
    - Number of guests
  - Validate all fields before saving
  - Show success/error messages
4. **Booking Confirmation Page**
  - After successful booking, show a summary of submitted booking details

### B. Admin Side (Simple Panel)

1. **Admin Login (Basic)**
  - Hardcoded username/password is acceptable for this assignment
2. **Manage Rooms (CRUD)**
  - Add new room
  - View room list
  - Edit room details
  - Delete room
3. **Manage Bookings**
  - View all bookings
  - Update booking status ( Pending, Confirmed, Cancelled )
  - Delete booking

## 4. Technical Requirements

---

- Use **procedural PHP only** (`mysqli_*` or PDO in procedural usage).
- Organize code into reusable files:
  - `config.php` (database connection)

- o header.php , footer.php
  - o separate files for each page
- Use SQL database with proper tables (example below).
  - Use server-side validation in PHP.
  - Use basic JavaScript validation for better UX.
  - Protect against SQL injection (prepared statements recommended).
  - Keep UI simple but clean and responsive.

## 5. Suggested Database Tables

---

### rooms

- id (INT, PK, AI)
- room\_name (VARCHAR)
- room\_type (VARCHAR)
- price\_per\_night (DECIMAL)
- capacity (INT)
- description (TEXT)
- image (VARCHAR)
- is\_available (TINYINT)

### bookings

- id (INT, PK, AI)
- full\_name (VARCHAR)
- email (VARCHAR)
- phone (VARCHAR)
- room\_id (INT, FK -> rooms.id)
- check\_in (DATE)
- check\_out (DATE)
- guests (INT)
- status (ENUM: Pending, Confirmed, Cancelled)
- created\_at (TIMESTAMP)

## 6. Business Rules

---

- Check-out date must be later than check-in date.
- Guests count must not exceed room capacity.
- A room cannot be booked by multiple customers for overlapping dates.
- Bookings are Pending by default.

## 7. Folder Structure (Example)

---

```
hotel-booking/
|-- assets/
|   |-- css/
|   |-- js/
|   |-- images/
|-- admin/
|   |-- login.php
|   |-- dashboard.php
|   |-- rooms/
|   |-- bookings/
|-- config.php
|-- header.php
|-- footer.php
|-- index.php
|-- rooms.php
|-- book.php
|-- booking-confirmation.php
|-- database.sql
```

## 8. Deliverables

---

Submit:

1. Complete project folder
2. database.sql file
3. README.md with:

- project setup steps
  - database import steps
  - admin login credentials
4. Screenshots (minimum 5):
- Home page
  - Rooms page
  - Booking form
  - Admin rooms management
  - Admin bookings list

## 9. Evaluation Criteria (100 Marks)

---

1. Functionality completed as required - **40**
2. Code structure and procedural PHP quality - **20**
3. Database design and query correctness - **15**
4. Validation and error handling - **10**
5. UI clarity and responsiveness - **10**
6. README and submission quality - **5**

## 10. Bonus (Optional)

---

- Search/filter rooms by date and capacity
- Email notification on booking (use mail mock or simulation)
- Pagination in admin bookings list

## 11. Submission Instructions

---

- Zip the project as: `yourname_hotel_booking.zip`
  - Ensure project runs without errors on localhost
  - Include all files and SQL dump
  - Late submissions will receive a penalty as per course policy
- 

## Student Checklist

---

- Project runs on localhost
- Database imports successfully
- All required pages are completed
- CRUD works for rooms and bookings
- Date and booking overlap checks work
- README is complete