Real-Life Brief: Hotel Reservation System

Project Overview:

You are tasked with building a web application for a hotel. The hotel needs a reservation system that allows users to search for available rooms, view details about the hotel and its rooms, make reservations, and manage their bookings. The staff should be able to manage room availability and reservation details. This system will include user authentication to ensure only authorized users can make or manage reservations.

Part 1: Web Application Features

1. Database Connection:

- **Objective:** Connect to a local MySQL database.
- **Details:**
  - You will create a MySQL database named `` `hotel_reservation` ``.
  - The database will have tables such as `` `users` ``, `` `rooms` ``, `` `reservations` ``, and `` `availability` ``.
  - Establish a connection to this database using PHP's `` `mysqli` `` or `` `PDO` `` extension.

2. Dynamic Search and Display:

- **Objective:** Allow users to search for available rooms based on certain criteria (e.g., check-in date, check-out date, number of guests) and dynamically display the results.
- **Details:**
  - Users will be able to search for rooms by providing check-in and check-out dates, and the number of guests.
  - The system will query the `` `availability` `` and `` `rooms` `` tables to fetch available rooms matching the search criteria.
  - The results will be displayed dynamically on the web page, with each room's details, price, and an option to reserve the room.

1. **Set Up the Database**:
    - Create the `` `hotel_reservation` `` database and the tables defined in the schema above.
2. **Connect to the Database**:
    - Write a PHP script to establish a connection to the MySQL database.
3. **Search Functionality**:
    - Create a search form that takes input for check-in date, check-out date, and number of guests.
    - Write a PHP script to handle the search query and fetch available rooms from the database.
    - Display the search results dynamically on the web page.
4. **Reservation Management**:
    - Develop a reservation form that captures user details and the selected room.
    - Write a PHP script to store reservation data and update room availability in the database.
    - Create a management interface for staff to view and update room availability.
5. **User Authentication**:
    - Implement user registration, login, and session management.
    - Protect pages with session checks, ensuring only logged-in users can make reservations.
    - Implement role-based access control for staff and regular users.

Further Enhancements:

- **Email Notifications**: Send confirmation emails to users after they make a reservation.
- **AJAX for Real-Time Updates**: Use AJAX to update search results dynamically without reloading the page.
- **Admin Panel**: Develop a more advanced admin panel with analytics and reporting features.

6. Authorization for Staff and Users:

- **Objective:** Implement role-based access control.
- **Details:**
  - Users should only be able to view and manage their reservations.
  - Staff members should have additional privileges, such as managing room availability and viewing all reservations.
  - The `users` table will include a `role` column that distinguishes between regular users and staff members.

Database Schema:

- users: `id`, `username`, `password`, `email`, `role`
- rooms: `id`, `room_number`, `type`, `price_per_night`, `description`
- reservations: `id`, `user_id`, `room_id`, `check_in`, `check_out`, `status`
- availability: `room_id`, `date`, `is_available`

Implementation Strategy:

1. **Set Up the Database**:
   - Create the `hotel_reservation` database and the tables defined in the schema above.
2. **Connect to the Database**:
   - Write a PHP script to establish a connection to the MySQL database.
3. **Search Functionality**:
   - Create a search form that takes input for check-in date, check-out date, and number of guests.
   - Write a PHP script to handle the search query and fetch available rooms from the database.
   - Display the search results dynamically on the web page.

3. Editing Database:

- **Objective:** Allow users (once authenticated) to make reservations, and allow staff to manage room availability.
- **Details:**
  - Once a user selects a room, they will be able to reserve it by providing their personal details (name, email, etc.).
  - The reservation details will be stored in the `` `reservations` `` table, and the `` `availability` `` table will be updated accordingly.
  - Hotel staff will have a separate interface to manage room availability and view all reservations. They can add or remove rooms from availability.

Part 2: User Authentication and Session Management

4. User Registration and Login:

- **Objective:** Implement user registration and login functionality.
- **Details:**
  - Users must register an account to make a reservation.
  - The `` `users` `` table will store user credentials (e.g., username, password, email).
  - Passwords should be securely hashed before being stored in the database.
  - Create a login system that validates user credentials and starts a session upon successful login.

5. Session Management:

- **Objective:** Use PHP sessions to manage user authentication and ensure only logged-in users can make reservations or access the staff interface.
- **Details:**
  - Once a user logs in, start a session and store the user’s information in session variables.
  - Use session data to restrict access to certain pages (e.g., reservation page, staff management page).
  - Include a logout feature that destroys the session.