**BSC Room Booking Web Application - Requirements Document**

**1. Login Page (All Users)**

* **Modal:** Provide a modal for users to input their username and password to log in.
* **Authentication:** Ensure secure login functionality with proper validation for username and password.

**2. Home Page (All Users)**

* **Booking Form:**
  + **Room Selection:** Dropdown for selecting a room.
  + **Default Date:** The default start date should be set to the current date.
  + **Multiple Days Selection:** A checkbox for multiple days selection. When checked:
    - An **End Date** field will be displayed.
    - The system should allow consecutive days bookings.
  + **Start Time and End Time:**
    - Timeslot intervals should be in 30-minute increments, from 6:00 AM to 12:00 AM.
    - Only available timeslots should be selectable based on the room’s availability.
  + **Event:** A text input field for the event name.
  + **Frequency:** Default set to “single-use,” but allow options for recurring events (daily, weekly, monthly).
  + **Status:** Default set to “pending.”
* **Calendar View:**
  + Each room should have a dedicated monthly calendar tab.
  + Only **approved** bookings should appear, showing:
    - Event Name
    - Start time - End time
    - User who booked
  + For bookings spanning consecutive days, the event should span across necessary cells.
  + Clicking on a booking cell should display a modal with:
    - Start time - End time
    - User
    - Event details
    - Approval status

**3. My Bookings Page (All Users)**

* **Filter Form:** Allow filtering by:
  + Room
  + Date
* **Booking Table:**
  + Show a paginated list of the user’s bookings with the following columns:
    - Created Date
    - Room
    - Start and End times
    - Event Name
    - Status (pending, approved, rejected)
    - Approved on (if applicable)
  + **Edit Pending Bookings:** Users can edit their "pending" bookings.
  + **Delete Bookings:** Users can delete their own bookings.

**4. Admin Page (Admin Role Only)**

* **Filter Form:** Provide filters for:
  + User
  + Room
  + Date
  + Status
* **Booking Table:**
  + Admins can view all bookings in a paginated table.
  + Admins can sort bookings by headers (alphabetical, datetime).
* **Actions:**
  + Admins can manage bookings:
    - **Approve/Reject Pending Bookings**
    - **Edit Bookings**
    - **Delete Bookings**
    - **Manage Recurring Bookings:**
      * Ability to delete or edit individual or all future bookings in a recurring series.
* **Enhanced Booking Form:**
  + Admins can book on behalf of any user.
  + Admins should be able to create single-use or recurring bookings.
  + **Recurring Bookings:** Set frequency (weekly, biweekly, monthly, nth week).
  + **Create New User/Admin Accounts:** Admins can create new users and assign roles (user/admin).
  + **Logging:** All changes made by admins should be logged with the timestamp for traceability.

**5. User and Booking Schema**

* **User Schema:**
  + id (unique string)
  + username (unique string)
  + name (string)
  + password (string)
  + role (string) [user, admin]
* **Booking Schema:**
  + id VARCHAR(36) PRIMARY KEY,
  + user\_id VARCHAR(36) NOT NULL,
  + user\_name VARCHAR(255) NOT NULL,
  + room VARCHAR(255) NOT NULL,
  + event\_name VARCHAR(255) NOT NULL,
  + start\_date DATE NOT NULL,
  + start\_time VARCHAR(5) NOT NULL,
  + end\_date DATE NOT NULL,
  + end\_time VARCHAR(5) NOT NULL,
  + series\_id VARCHAR(36) DEFAULT NULL,
  + frequency VARCHAR(50) DEFAULT 'single', -- optional for UI clarity
  + frequency\_start DATETIME DEFAULT NULL,
  + frequency\_end DATETIME DEFAULT NULL,
  + created\_at DATE NOT NULL,
  + status VARCHAR(20) DEFAULT 'pending', -- pending, approved, rejected
  + approved\_at DATETIME DEFAULT NULL,
  + approved\_by VARCHAR(36) DEFAULT NULL,
  + FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,
  + FOREIGN KEY (series\_id) REFERENCES repeat\_series(id) ON DELETE SET NULL
* **Room Schema:**
  + id (unique string)
  + name (unique string)
  + floor (int) [2, 3, 4, 5]
  + pax (integer, number of people the room can accommodate)
* **Logs Schema:**
  + id (unique string)
  + timestamp (datetime)
  + action (string) [admin action logs such as approval, update, delete]
* **Repeat Series Schema:**
* repeat\_series (
* id VARCHAR(36) PRIMARY KEY,
* created\_by VARCHAR(36) NOT NULL,
* repeat\_type ENUM('daily', 'weekly', 'monthly', 'yearly') NOT NULL,
* repeat\_interval INT NOT NULL DEFAULT 1, -- Every X days/weeks/etc.
* repeat\_on JSON DEFAULT NULL, -- e.g., ["Monday", "Wednesday"] or ["15"] or ["February", "April"]
* ends\_after INT DEFAULT NULL, -- Repeat N times
* ends\_on DATE DEFAULT NULL, -- Or until specific date (max 2 years from now)
* created\_at DATETIME NOT NULL,
* FOREIGN KEY (created\_by) REFERENCES users(id) ON DELETE CASCADE

**6. User Interface Requirements**

* **Frontend: React + Tailwind CSS**
* **Backend:** Node.js
* **Database:** MySQL (localhost:3306 root:P@55w0rd)

**7. Testing Accounts**

* **user1:**
  + Username: user
  + Name: user
  + Password: user
  + Role: user
* **admin1:**
  + Username: admin
  + Name: admin
  + Password: admin
  + Role: admin

**8. Security and Authentication**

* **Role-based Access Control:** Ensure that users can only access appropriate pages (admin pages for admin role only).
* **Session Management:** Use secure sessions or JWT for user authentication.

**9. Other Notes**

* Ensure **pagination** for booking tables on both user and admin pages.
* Implement **validation** for form inputs to prevent errors.
* Provide detailed **error messages** when necessary (e.g., for invalid input, failed actions).