

Main functions:

1. User Management

- **CRUD Operations :**
 - **Create:** Handle user registration with email check (can be simulated).
 - **Read:** View user profiles and account information.
 - **Update:** Manage profiles and account information; update roles (Admin).
 - **Delete:** Remove user accounts (implied for Admin role management).
- **Login Page Requirement:**
 - Single login page with email/password for all users.
 - Include password reset functionality.
 - Restrict access based on roles (Guest, Staff, Admin).
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Interface to assign/manage roles (Guest, Staff, Admin) and view user activity.
 - **Staff Dashboard:** Customer Support (Ms. Nadeesha) interface to assist with account issues (e.g., password reset help).
 - **User Dashboard:** Guest interface for registration, login, password reset, and profile updates.
- **Difficulty:** Medium

2. Event Management

- **CRUD Operations :**
 - **Create:** Create events with title, date, venue, category, description.
 - **Read:** View event details and status (draft, published, cancelled).
 - **Update:** Update event details; manage status.
 - **Delete:** Cancel events (implied by status management).

- **Login Page Requirement :**
 - Single login page restricts creation/updates to Staff/Admin.
 - Guests access browsing features post-login.
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Interface for approving event publication and assigning staff.
 - **Staff Dashboard:** Event Planner (Mr. Chamika) interface to create/update events; Customer Support views event details.
 - **User Dashboard:** Guest interface to browse events (title, date, category).
- **Difficulty:** Medium

3. Booking Management

- **CRUD Operations :**
 - **Create:** Allow users to book events with instant confirmation.
 - **Read:** Show real-time event availability and booking details.
 - **Update:** Edit bookings; update RSVP status.
 - **Delete:** Cancel bookings.
- **Login Page Requirement :**
 - Single login page for Guests to book, Staff to assist, Admin to monitor.
 - Role-based access ensures proper permissions.
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Interface to monitor all bookings.
 - **Staff Dashboard:** Event Planner views bookings; Customer Support assists with booking changes.
 - **User Dashboard:** Guest interface for booking, editing, cancelling, or RSVPing via forms.
- **Difficulty:** Medium

4. Payment Management

- **CRUD Operations:**
 - **Create:** Process transactions via mock payment gateway.
 - **Read:** Track payment status (pending, completed, failed, refunded); view receipts.
 - **Update:** Handle refunds.
 - **Delete:** Cancel transactions (implied by refund process).
- **Login Page Requirement:**
 - Single login page for Guests to pay, Staff to assist, Admin to monitor.
 - Role-based access restricts financial controls to Admin/Finance Manager.
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Finance Manager (Mrs. Ishara) interface to monitor transactions and process refunds.
 - **Staff Dashboard:** Customer Support interface to assist with payment issues.
 - **User Dashboard:** Guest interface for payment forms and receipt viewing.
- **Difficulty:** Medium

5. Venue Management

- **CRUD Operations:**
 - **Create:** Store venue details (location, capacity, features).
 - **Read:** View venue availability and usage reports.
 - **Update:** Update venue details; assign venues to events.
 - **Delete:** Remove venues (implied for management).
- **Login Page Requirement:**
 - Single login page restricts management to Staff/Admin.

- Guests view venue details during booking post-login.
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Interface to manage venues and view usage reports.
 - **Staff Dashboard:** Event Planner/Technical Coordinator interface to assign venues and check availability.
 - **User Dashboard:** Guest interface to view venue details during booking.
- **Difficulty:** Medium

6. Insights & Support

- **CRUD Operations:**
 - **Create:** Collect customer feedback and support tickets.
 - **Read:** View feedback, star ratings, and simple reports (booking counts, popular events).
 - **Update:** Respond to support requests.
 - **Delete:** Remove resolved tickets (implied for management).
- **Login Page Requirement:**
 - Single login page for Guests to submit feedback, Staff to respond, Admin to view reports.
 - Role-based access controls ticket and report access.
- **Role-Specific UI Dashboard Requirements:**
 - **Admin Dashboard:** Interface to view reports and feedback.
 - **Staff Dashboard:** Customer Support interface to respond to tickets; staff view for reports.
 - **User Dashboard:** Guest interface for feedback/rating forms and support requests.
- **Difficulty:** Easy

*** If full integration isn't possible, each function's CRUD and UI can be developed as standalone modules, linked via a single login page directing to role-specific dashboards, aligning with the PDF's modular structure.

Process Example:

- A user enters their email and password on the login page.
- The system verifies their credentials and identifies their role as "user."
- The user is redirected to the User Dashboard, which includes the Insights & Support component with forms for submitting feedback, star ratings, and support requests.
- Other roles (e.g., Staff or Admin) would be directed to their respective dashboards with different functionalities (e.g., Customer Support responds to tickets, Admins view reports).

** If time permits within the 12-week timeline, the project leader can merge the six core functions into a unified system by eliminating duplicate UI elements, such as **multiple login pages or redundant dashboards**. The project requires a single login page with role-based access control (for user, Staff, and Admin roles), meaning only one login interface should exist, redirecting users to their role-specific dashboard. Each function's dashboard can be consolidated into a single, unified dashboard with role-based views (e.g., tabs or sections) to remove duplication while maintaining function-specific features (e.g., Guests book events, Admins approve events, Staff manage tasks).

Process for Merging and Removing Duplicate UI:

- **Single Login Page:**
 - Implement one login page with email/password authentication and password reset functionality, as required for all roles (Guest, Staff, Admin).
 - The login page checks the user's role (stored in the database, e.g., MySQL) and redirects them to a unified dashboard tailored to their role.
 - Example: A Guest logs in and is directed to a dashboard with booking and feedback options; an Admin is directed to a dashboard with role management and approval options.

- Avoid creating separate login pages for each function to prevent duplication, aligning with the PDF's requirement for a centralized platform.
- **Unified Dashboard with Role-Based Views:**
 - Create a single dashboard interface that dynamically displays function-specific features based on the user's role, using conditional rendering (e.g., JavaScript to show/hide sections).
 - For each function, include role-specific UI components:
 - **User Management:** Admin sees role assignment and user activity; Staff (Customer Support) sees account assistance options; Guests see profile and login forms.
 - **Event Management:** Admin sees event approval and staff assignment; Staff (Event Planner) sees event creation/editing; Guests see event browsing.
 - **Booking Management:** Admin sees booking overview; Staff sees booking assistance; Guests see booking, editing, and RSVP forms.
 - **Payment Management:** Admin (Finance Manager) sees transaction monitoring and refunds; Staff sees payment issue assistance; Guests see payment forms and receipts.
 - **Venue Management:** Admin sees venue management and reports; Staff sees venue assignment and availability; Guests see venue details during booking.
 - **Insights & Support:** Admin sees reports and feedback; Staff sees ticket response options; Guests see feedback and support request forms.
 - Use tabs, menus, or sections to organize functions within the dashboard, ensuring role-based access control restricts unauthorized access (e.g., Guests can't see Admin's role management).
 - Eliminate redundant dashboards by consolidating all function-specific UIs into this single interface, avoiding separate dashboards for each function.
- **Removing Duplicate UI Elements:**
 - Ensure only one login page exists across all functions, redirecting to the unified dashboard.

- Avoid creating separate dashboards for each function (e.g., no standalone User Management dashboard or Booking Management dashboard). Instead, integrate all functions into one dashboard with role-based views.