

Requirement Specification – Register & Login System

1. System Overview

The system provides a secure authentication process for four types of users:

- **Admin** – manages the whole system.
 - **User** – regular users who browse the website.
 - **Organizer** – event organizers who can create/manage Perahera events.
 - **Service Provider** – vendors who provide Perahera services (dancers, elephants, lighting, items, etc.).
-

2. Functional Requirements

2.1 Registration

1. Registration form must include:
 - Username
 - Email (unique)
 - Password + Confirm Password
 - User Type (`user`, `organizer`, `service_provider`)
2. If **Service Provider** is selected, system must display a **Service Type dropdown** with options:
 - `perahara_item`
 - `perahara_service`
 - `elephant_service`

3. System must validate inputs (email format, password match, unique email).
 4. Registered users are saved in the database with hashed passwords.
 5. **Admin users cannot register via UI** – they must be created manually in the database.
-

2.2 Login

1. Login form must include:
 - Email
 - Password
 2. System must validate credentials against the database.
 3. Based on `user_type`, the user is redirected:
 - **Admin** → Admin Dashboard
 - **User** → Website/User Dashboard
 - **Organizer** → Website/Organizer Dashboard
 - **Service Provider** → Website/Service Provider Dashboard
 4. If credentials are invalid → show error message.
 5. System must maintain **session management** (stay logged in until logout).
-

2.3 Service Provider – Add Services

1. After logging in, Service Providers must be able to add services.
2. Each service must include:
 - Service Name (e.g., Dancing Team, Lighting, Elephant Hire)
 - Description

- Price
 - Availability (optional)
3. Services are stored in a separate table linked to the service provider.
-

2.4 Logout

1. Every user must have the option to logout.
 2. Logout must destroy the session and redirect to the homepage/login page.
-

3. Non-Functional Requirements

- **Security:** Passwords stored using hashing (bcrypt/argon2).
- **Role-based Access:** Only Admin can access Admin Dashboard, etc.
- **Usability:** Responsive and user-friendly UI with clear error messages.
- **Performance:** Login/registration actions must respond within 1 second.
- **Scalability:** Database should support thousands of users and services.

1.user

Column	Type	Description
id	integer(PK)	Auto-increment primary key
username	string(100)	User's display/username
email	string(150) UNIQUE	User's email
password	string(255)	Hashed password
user_type	enum('admin','user','organizer','service_provider')	Role of the user
service_type_id	integer(FK, nullable)	If user is a service provider → link to service_types table
created_at	TIMESTAMP	Record created date
updated_at	TIMESTAMP	Record updated date

2. service_types

Column	Type	Description
id	integer(PK)	Auto-increment
name	string	Type of service
description	text	Optional description
created_at	TIMESTAMP	Record created date
updated_at	TIMESTAMP	Record updated date

3. services

Column	Type	Description
id	integer(PK)	Auto-increment
user_id	integer(FK → users.id)	The service provider who added it
service_type_id	integer(FK → service_types.id)	Type of service
name	string(150)	Service name (e.g., Kandyan Dancing, Tusker Elephant)
description	text	Details of the service
price	decimal(10,2)	Optional price/rate
created_at	TIMESTAMP	Record created date
updated_at	TIMESTAMP	Record updated date