**Design Doc – BookMySport**

Design Document: User Registration and Authentication System

**1. Introduction**

This document outlines the design for the **User Registration and Authentication** system within the sports management platform. The system handles user registration, authentication, and role-based access control (RBAC) to ensure that users (admins, court owners, coaches, and regular users) can interact with the system securely and according to their roles.

**2. Objectives**

* Provide a secure method for user registration and login.
* Implement role-based access control (RBAC) to restrict certain operations based on user roles.
* Ensure secure password storage and management.
* Facilitate password recovery via email-based password reset.
* Provide JWT-based authentication for stateless user sessions.

**3. System Components**

**3.1. API Endpoints**

The system will provide a set of RESTful APIs to handle registration, authentication, and role management.

**4. API Specifications**

**4.1. User Registration**

* **Description**: Allows a new user to register an account by providing basic details (name, email, password) and selecting a role.
* **Endpoint**: POST /auth/register
* **Method**: POST
* **Request Body**:

{

"name": "John Doe",

"email": "john.doe@example.com",

"password": "securepassword123",

"role": "user"

}

**Response**:

* 201 Created for success:

{

"message": "User registered successfully",

"userId": "123456"

}

400 Bad Request for errors (e.g., email already exists, invalid role):

{

"message": "Email is already registered"

}

**Validations**:

* Ensure email is unique and valid.
* Password should meet minimum security standards (e.g., length, complexity).
* Role must be one of: user, admin, coach, court\_owner.

**4.2. User Login**

* **Description**: Authenticates a user by verifying email and password. Returns a JWT token on successful login.
* **Endpoint**: POST /auth/login
* **Method**: POST
* **Request Body**:

{

"email": "john.doe@example.com",

"password": "securepassword123"

}

**Response**:

* 200 OK for success:

{

"message": "Login successful",

"token": "jwt\_token\_here"

}

401 Unauthorized for invalid credentials:

{

"message": "Invalid email or password"

}

**JWT Details**:

* A stateless JWT token will be returned on successful login, allowing users to authenticate further requests without re-login.