Specification: User Authentication System

To implement a secure and extendable user authentication system for BadmintonApp that supports different user roles (Player, Trainer, Admin).

**Scope**

- Registration

- Login

- JWT-based Authentication

- Role-based access control

**Technologies**

- ASP.NET Core Web API

- EF Core

- Identity

- JWT Token

- SQL Server

**Entities**

User:

- guid, unique

- Email

- FirstName

- LastName

- PasswordHash

- Role (enum: Player, Trainer, Admin)

- CreatedAt

- IsActive

- DoB

- Rank

- Level

- Club (guid, referrer to club entity)

**Endpoints**

- POST /api/auth/register — user registration

- POST /api/auth/login — login and receive JWT

- GET /api/users/{id} — get current user info (auth required)

**Validation Rules**

- Email must be unique

- Password must meet security requirements - later

- Role must be valid enum value

**Security**

- Passwords stored as hashes using ASP.NET Identity

- JWT token generated with secret key and expiry

- Role claims embedded into JWT

**Architecture and Integration**

- API project should reference both Application and Infrastructure layers

- Application contains business logic and DTOs

- Infrastructure implements database access and user repositories

- Swagger should be configured in the API for documentation and testing endpoints

**Extensibility**

- Add refresh tokens later

**Notes**

- No CQRS used; services directly called from controllers or Razor pages

- DTOs are used for input/output in Application layer

- Enums, entities and Repository interfaces should be stored in Domain project

- Create mapping class which should contain from dto and to dto :   
A screenshot of a computer program

AI-generated content may be incorrect.

- Entity Framework Core should be configured in the API with proper DbContext and connection string.

- Dependency Injection should be set up in Program.cs to register services from Application and Infrastructure layers.

- For example AuthService in Application project should have IUserRepository object injected by DI

A screenshot of a computer program

AI-generated content may be incorrect.