**Name: Batool Hammoud**

**Project Title:** ESS Task

**Project Overview:** This project involved developing a web application using Visual Studio and .NET 6 that incorporates N-Layer architecture, multi-tenancy support with multiple databases, Entity Framework (EF) for data access, ASP.NET Identity for user management, and Blazor for the user interface. The application includes functionalities for tenant and user registration, tenant login, and a landing page displaying tenant-specific information.

**Key Components and Features:**

1. **N-Layer Architecture:**
   * Implemented a structured architecture comprising multiple layers: Core, Repos, Data, Services, ModelViews, and Pages.
2. **Multi-Tenancy with Dynamic Databases:**
   * Implemented multi-tenancy support where each tenant has its own separate database.
3. **Entity Framework Core and ASP.NET Core Identity:**
   * Utilized EF Code First approach for database schema generation.
   * Integrated ASP.NET Identity for user registration and login functionalities.
   * Designed database schema to include tables for user identity and tenant information only.
4. **Blazor for User Interface:**
   * Utilized Blazor framework to create interactive and responsive user interfaces based on the built-in Blazor style.
   * Implemented Blazor components for user registration and login.

**Technologies Used:**

* **Language:** C#
* **Framework:** .NET 6
* **Database Provider:** MySQL
* **Frontend:** Blazor for UI components
* **Backend:** ASP.NET Core with Entity Framework Core
* **Authentication and Authorization:** ASP.NET Identity
* **Development Environment:** Visual Studio

**Project Structure:**

1. **Core Layer:**
   * **Tenant.cs**: Entity representing tenant information.
   * **User.cs**: Entity representing user information.
2. **Data Layer:**
   * **AppDbContext.cs**: DbContext for handling database operations, including DbSet for **Tenant** and **User**.
3. **Repos:**

* **ITenant.cs:** interface for tenant methods.
* **IUser.cs:** interface for user methods.

1. **Services Layer:**
   * **TenantServices.cs**: Service handling tenant-related CRUD operations (Add, Get, Get All, Update, Delete).
   * **UserServices.cs**: Service managing user-related CRUD operations (Add, Get, Get All, Update, Delete).
2. **ModelViews:**

* **AuthenticateResult.cs**: result authentication error or success.
* **LoginViewModel**: custom user login model.
* **RegisterViewModel**: custom tenant register model.
* **RegisterationResult**: registration result error or success.

1. **Blazor UI Layer:**
   * **Pages/Register.razor**: Blazor component for user registration.
   * **Pages/Login.razor**: Blazor component for user login.
   * **Pages/Index.razor**: Main page.

**Challenges Faced:**

* I had some struggles dealing with Blazor pages.
* I found some trouble to manage changes in .net 6 from .net core 3.1 which I used to deal with.