**Teknik Rapor**

**Proje Adı: JWT Tabanlı Kimlik Doğrulama ile Dosya Yükleme Sistemi**

**1) Amaç ve Problem Tanımı**

Bu projede geliştirilen sistemin amacı, kullanıcıların dosya yüklemesine, listelemesine ve silmesine izin veren bir Web API oluşturmaktır.

Proje, herhangi bir veritabanı entegrasyonu olmaksızın, bellekte tutulan kullanıcı listesi ile JWT (JSON Web Token) kullanarak kimlik doğrulama yapmaktadır.

**2) Geliştirme fikri**

Kayıt olma ve Giriş yapma özellikleri güvenlik amacıyla kullanılabilir sadece giriş yapan kullanıcıların kendi dosyalarını görebilmesi gibi özellikler eklenebilir. Bu tarz özellikler güvenlik açısından çok önemlidir.

**3) Yöntem ve Kullanılan Teknolojiler**

|  |  |
| --- | --- |
| **Teknoloji** | **Amaç** |
| ASP.NET Core Web API | API altyapısını oluşturmak |
| JWT | Kullanıcı kimlik doğrulama |
| Swagger | API test arayüzü |
| HTML + JS | Basit frontend arayüz |

**4) Sistem Akışı ve Mimarisi**

* Kullanıcı /api/auth/register ile kayıt olur.
* /api/auth/login ile sistemden giriş yapar ve JWT token alır.
* Kullanıcı token almış olsa bile, dosya yükleme, silme ve listeleme işlemleri şu anda herhangi bir Bearer yetkilendirme kontrolü olmadan yapılmaktadır.
* Dosya işlemleri api/files uç noktalarından gerçekleştirilir.

**5. Klasör Yapısı**

|  |  |
| --- | --- |
| Klasör/Dosya | Açıklama |
| Controllers/ | API denetleyicileri |
| ├── AuthController.cs | Kayıt ve giriş işlemleri |
| └── FilesController.cs | Dosya yükleme/listeleme/silme |
| Models/ | Model sınıfları |
| ├── RegisterModel.cs | Kayıt modeli |
| └── UploadModel.cs | Dosya yükleme modeli |
| Services/ | Servis ve arayüz sınıfları |
| ├── IUserService.cs | Servis arayüzü |
| └── UserService.cs | Kullanıcı işlemleri |
| appsettings.json | Yapılandırma dosyası |
| Program.cs | Ana uygulama başlatıcısı |
| index.html | Basit kullanıcı arayüzü |
| README.md | Proje açıklamaları |

**6. Tercih Edilen Yöntem ve Gerekçeleri**

* **Dosya sistemine yazma**, veritabanı karmaşıklığı olmadan hızlı test için uygun.
* **Swagger** kullanılarak geliştiriciler için kullanışlı API testi sağlandı.
* **Basit HTML arayüz**, kullanıcının dosya yükleme ve listeleme işlemlerini rahat test etmesini sağlar.
* JWT kimlik doğrulama yalnızca login ve register için aktif edilmiştir. Dosya işlemlerinde token kontrolü uygulanmamıştır.

**7. Arayüz ve Kullanıcı Deneyimi**

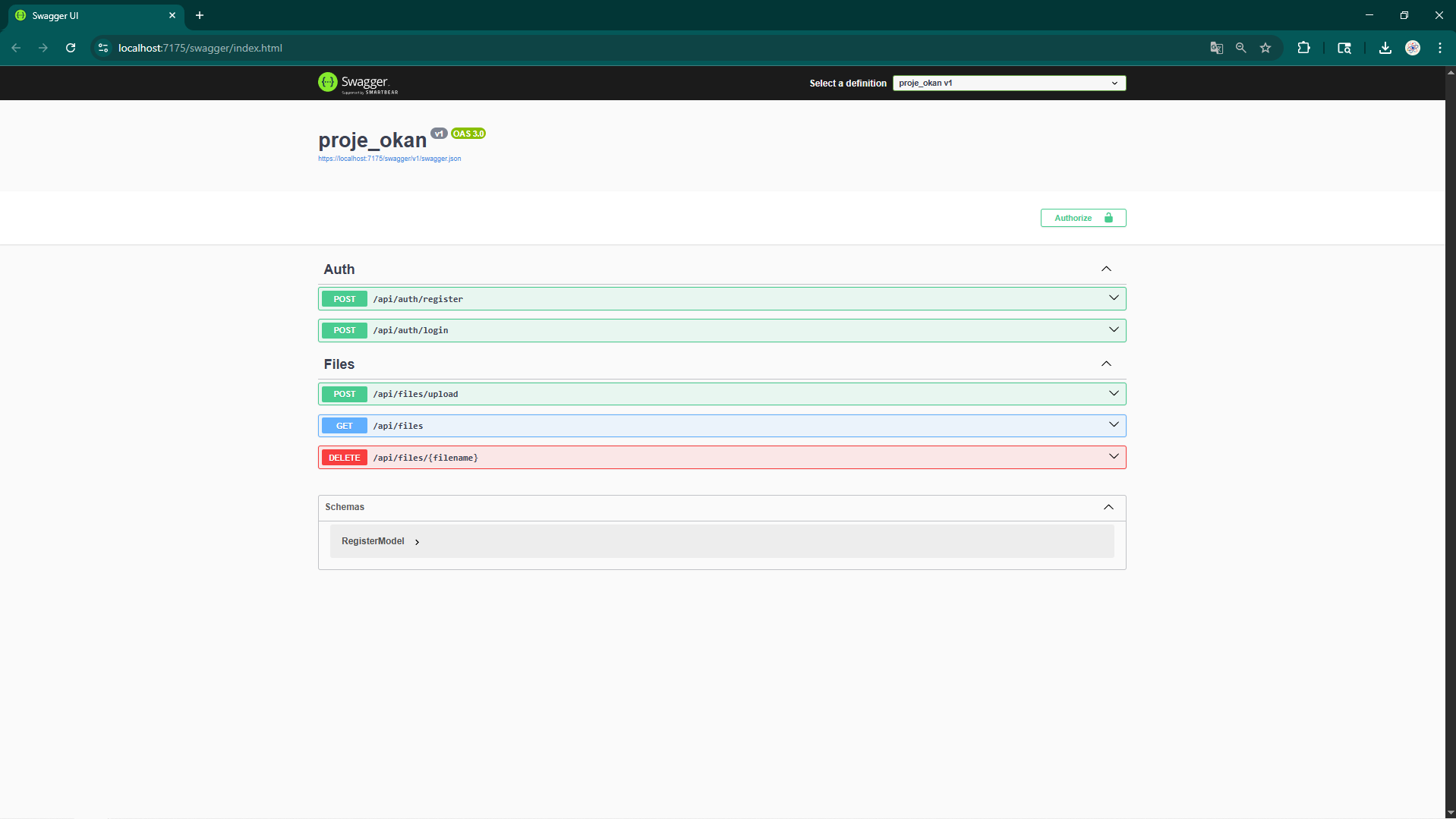
Frontend tek bir index.html sayfasından oluşur. Kullanıcı:

* Kullanıcı adı/parola ile giriş yapar
* Dosya yükler
* Listeleme yapar
* Tek tıkla dosyayı siler

Giriş yapıldıktan sonra token alınmakta fakat dosya işlemlerinde yetkilendirme kontrolü uygulanmamaktadır.

**8. Ekran Görüntüleri ile**

**1. Projenin Ana sayfası**



**2. Register Bölümü (/api/auth/register)**

**Kullanıcının sisteme yeni bir hesap kaydetmesini sağlar.**

* **“Try it out” butonuna tıklanarak giriş alanları aktif hale getirilir.**
* **Açılan metin kutularına bir username ve password girilir.**
* **“Execute” butonuna tıklanarak kayıt işlemi tamamlanır.**
* A black line on a white background

  AI-generated content may be incorrect.**Eğer aynı kullanıcı daha önce kayıt olduysa sistem hata döner, değilse başarılı şekilde kayıt gerçekleşir.**

**3. Login Bölümü (/api/auth/login)**

Bu bölümde kullanıcı adı ve şifre ile giriş yapılır.

* Önce “**Try it out**” butonuna basılır.
* Açılan kutucuklara kayıtlı username ve password girilir.
* Ardından “**Execute**” butonuna tıklanarak giriş yapılır.
* Eğer bilgiler doğruysa, cevap olarak bir JWT (token) döner.

A black line on a white background

AI-generated content may be incorrect.

**4. Dosya Yükleme Bölümü (/api/files/upload)**

Kullanıcı bu alanda sunucuya dosya yükleyebilir.

* “**Try it out**” butonuna tıklanır.
* Açılan alandan bir dosya seçilir.
* Format olarak .pdf, .png, .jpg, .jpeg desteklenir.
* “**Execute**” ile dosya yüklenir ve yüklenen dosyanın adı döner.

A screenshot of a phone

AI-generated content may be incorrect.!Dosyalar Uploads klasöründe saklanır.

**5. Dosya Listeleme Bölümü (/api/files)**

Bu GET işlemi, sunucuda yüklü olan dosyaların listesini getirir.

* “**Try it out**” butonuna tıklanır ve ardından “**Execute**” yapılır.
* Yüklenmiş tüm dosyalar liste şeklinde döner.

A screenshot of a phone

AI-generated content may be incorrect.Bu liste, frontend tarafında kullanıcıya gösterilebilir.

**6. Dosya Silme Bölümü (/api/files/{filename})**

Belirli bir dosyayı silmek için kullanılır.

* “**Try it out**” tıklanır.
* filename kısmına silmek istenilen dosyanın adı yazılır.
* “**Execute**” ile silme işlemi gerçekleştirilir.

A screenshot of a computer

AI-generated content may be incorrect.Dosya bulunamazsa Not Found hatası, başarılıysa OK mesajı döner.

**9) Sonuç**

Projede temel bir kimlik doğrulama sistemi ile dosya yükleme ve listeleme işlevleri sade ve etkili şekilde çalışmaktadır. Yetkilendirme kodları uygulanmış fakat yalnızca kullanıcı kimlik doğrulamasında (login/register) kullanılmıştır.

**10. Teslim Takibi**

* Proje GitHub'a yüklenmiştir.
* Kodlar, README dahil olacak şekilde yüklenmiştir.