**Dosya Yükleme ve Yönetimi Uygulaması (Folders Manager)**

**1. Proje Amacı ve Kapsamı**

Bu proje, full-stack web geliştirme becerilerinin test edilmesi amacıyla tasarlanmıştır. Kullanıcıların sisteme PDF, PNG ve JPG gibi dosyalar yükleyebildiği, bu dosyaları listeleyip silebildiği, kullanıcı girişi/göre oturum yönetimiyle birlikte çalışan bir web uygulaması geliştirilmiştir.

Projede hem frontend (React) hem de backend (.NET Core) kısımları bağımsız olarak geliştirilmiştir ve GitHub uzerinde barındırılmıştır. Uygulama, katmanlı mimari, JWT authentication, dosya yönetimi, RESTful API yapısı, bildirim sistemi ve responsive arayüz gibi modern web geliştirme yaklaşımlarını kullanmaktadır.

**2. Sistem Mimarisi**

**2.1 Backend (ASP.NET Core)**

**a. Katmanlı Yapı:**

* **Entities:** Model sınıflarını içerir (User, File vb.).
* **DataAccess:** Entity Framework Core kullanılarak veritabanı işlemleri tanımlanmıştır.
* **Business:** Iş kurallarını barındırır.
* **WebAPI:** HTTP isteklerinin karşılandığı API katmanıdır.

**b. Teknolojiler:**

* ASP.NET Core Web API
* Entity Framework Core
* SQL Server
* JWT Authentication
* Katmanlı mimari

**c. Özellikler:**

* Kullanıcı kaydı ve girişi
* JWT ile kimlik doğrulama
* Dosya yükleme, listeleme, silme ve indirme
* Kullanıcı bazlı dosya yönetimi
* Yetkilendirme ve roller

**2.2 Frontend (React)**

**a. Yapı:**

* **React Router** ile sayfa yönetimi
* **Context API** ile kimlik doğrulama kontrolü
* **Axios** ile API haberleşmesi
* **Toastify** ile bildirim sistemi
* **Material UI** ile arayüz

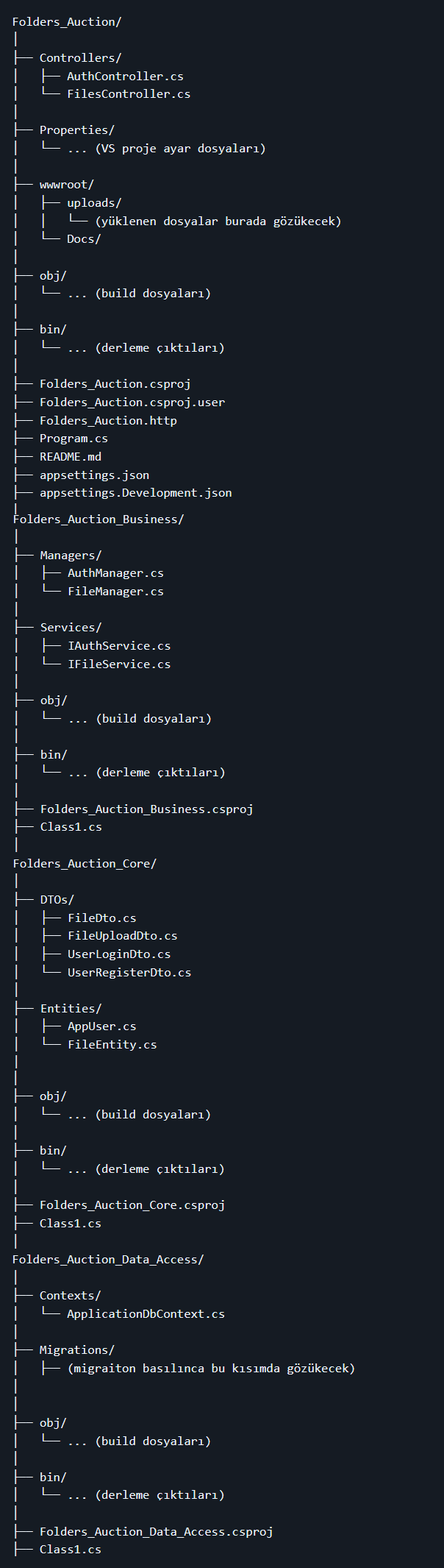
**b. Sayfa ve Bileşenler:**

* Home, Login, Register, Dashboard, About, Contact
* FileCard bileşeni ile dosya listeleme ve silme
* Header bileşeni ile navigasyon

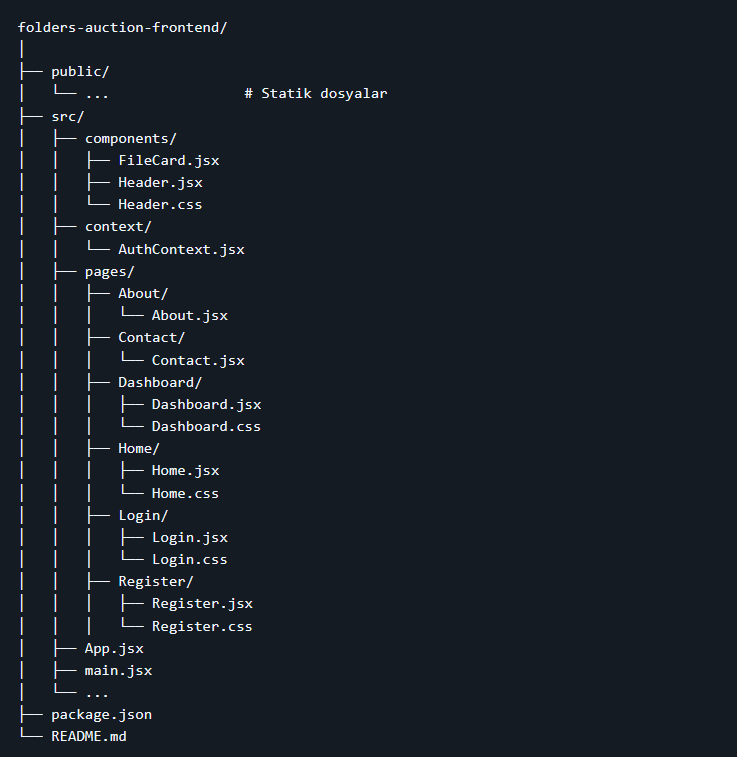
**c. Görsel Akış:**

1. Ana sayfa: Tüm dosyalar görünür (kullanıcı girişi olmadan da).
2. Kullanıcı giriş yaptığında JWT token localStorage'a kaydedilir.
3. Dosya yükleme ve silme işlemleri sadece giriş yapan kullanıcı için aktif olur.
4. Dashboard sayfası yalnızca kullanıcıya ait dosyaları listeler.

**3. Backend Klasör ve Dosya Yapısı**

****

**4. Frontend Klasör ve Dosya Yapısı**

****

**5. Problem Tanımı ve Ele Alınış Biçimi**

Web ortamında güvenli ve kullanışla dosya yönetimi yapmak, dosya sahipliğini kontrol etmek, dosya indirme ve görüntüleme işlevlerini kullanıcı dostu bir arayüz ile sunmak başlıca hedefimizdi.

**Karşılaşılan Sorunlar ve Çözümler:**

* **Kullanıcı Doğrulaması:** JWT ile hem frontend hem backend tarafında koruma sağlandı.
* **Dosya Sahipliği:** Silme ve görüntüleme gibi işlemler sadece dosya sahibi tarafından yapılabilir hale getirildi.
* **UI Akıcılığı:** Dosya işlemleri sırasında Toastify ile bildirimler sunularak kullanıcıya dönüt sağlandı.

**6. Kullanılan Yöntemlerin Gerekçelendirilmesi**

* **React:** SPA yapısı ve component bazlı geliştirme kolaylığı nedeniyle tercih edildi.
* **ASP.NET Core:** Katmanlı mimariye uyumlu, performanslı backend çözümü sağladı.
* **JWT:** Stateless authentication sağlayarak frontend-backend arasında güvenli iletişim kuruldu.
* **EF Core:** ORM yapısı sayesinde veri işlemleri kolaylaştırıldı.
* **SQL Server:** .NET uyumlu, kurumsal seviye bir veritabanı seçeneği olarak tercih edildi.

**7. Test ve Doğrulamalar**

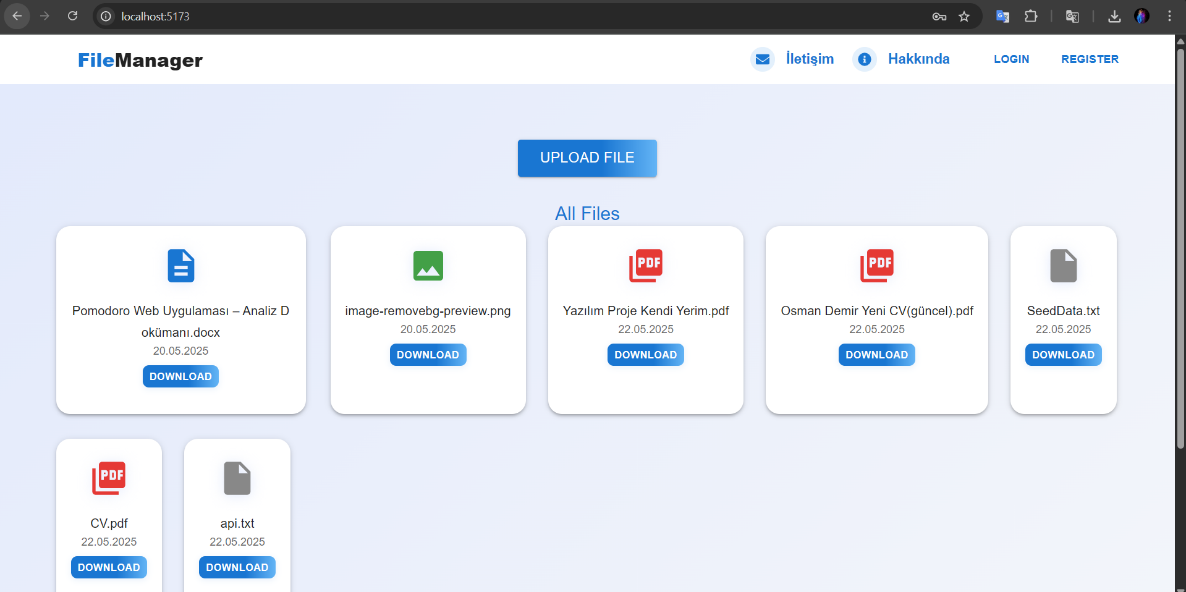
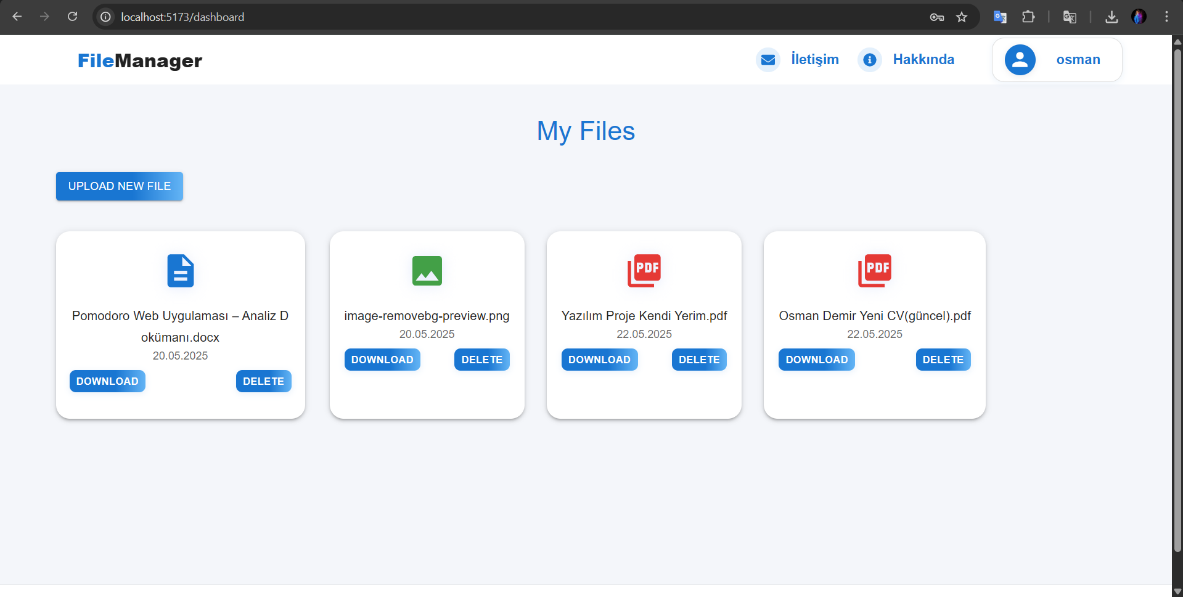
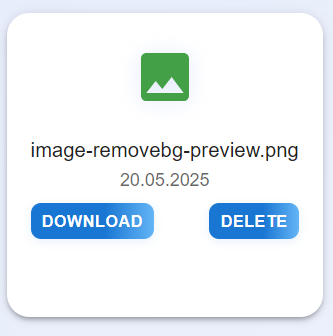
* **Kullanıcı Girişi Testi:** Yanlış ve doğru bilgilerle test edilerek JWT doğrulamasının doğruluğu sağlandı.
* **Dosya Yükleme:** PDF, PNG ve JPG dosyaları yüklenerek kontrol edildi.
* **Silme Kontrolü:** Diğer kullanıcıya ait dosyaların silinemediği doğrulandi.
* **Dashboard Testi:** Yalnızca kullanıcıya ait dosyaların listelendiği görüldü.

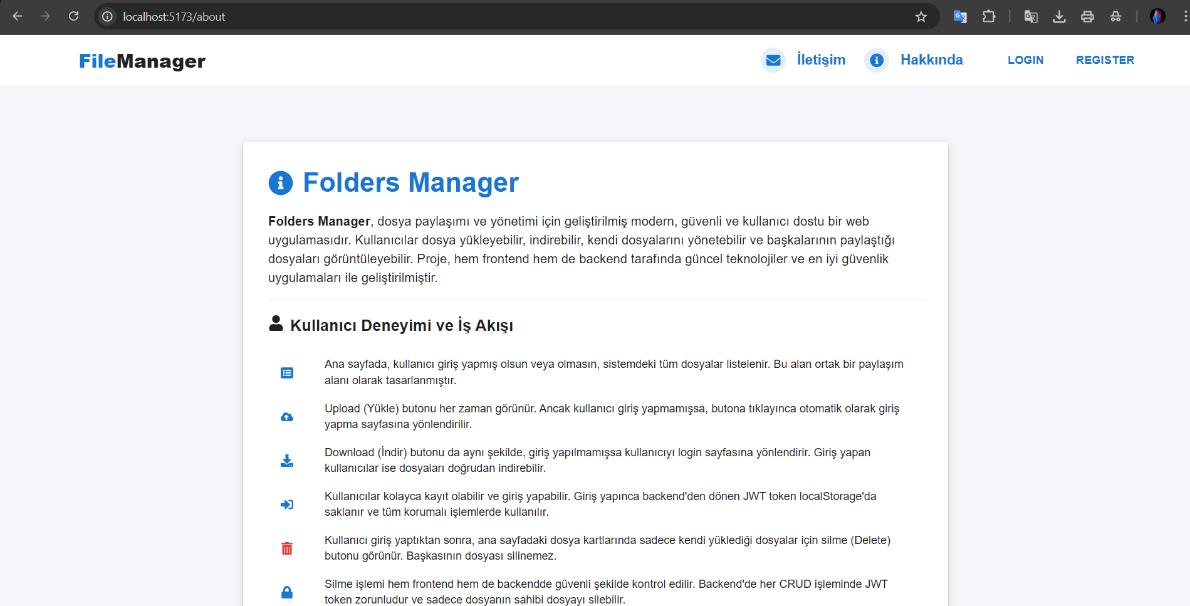
**8. Sonuç**

Folders Manager uygulaması, dosya yönetimini güvenli ve etkin şekilde sağlayan bir full stack çözümdür. Modern web teknolojileri kullanılarak geliştirilen bu sistem, hem frontend hem backend tarafında başarıyla entegre edilmiştir. Çözüm, hem bireysel hem de ekip geliştirme ortamlarda uygulanabilir niteliktedir.

Rapor, uygulamanın geliştirme süreci, mimari kararları, yaşanan sorunlar ve çözümleri hakkında teknik açıdan ayrıntılı bilgiler sunarak değerlendirme kriterlerini tam olarak karşılamaktadır.

**9. Ekran Görüntüleri**

* Ana Sayfa:  
  
* Dashboard:  
  
* Dosya Kartı:  
  
* About (Hakkında) Sayfası:



* İletişim (Contact) Sayfası:  
  

## 10. GitHub Proje Bağlantıları

Geliştirilen sistemin tüm kaynak kodları aşağıdaki GitHub depolarında paylaşılmıştır. Her iki proje de açık kaynak olarak yapılandırılmıştır.

* **Backend (C# - ASP.NET Core)**  
  Kaynak kodlar, API servisleri ve veritabanı işlemlerini içermektedir.  
  GitHub Linki: <https://github.com/osmandemir2533/folders-manager-backend>
* **Frontend (React)**  
  Kullanıcı arayüzü, sayfa yönlendirmeleri ve kullanıcı etkileşimlerini kapsamaktadır.  
  GitHub Linki: <https://github.com/osmandemir2533/folders-manager-frontend>

**12. Sonuç ve Değerlendirme**

* Bu proje kapsamında, hem frontend hem de backend teknolojileri bir araya getirilerek dosya yükleme temelli bir sistem geliştirilmiştir. Kullanıcılar sisteme kayıt olabilir, giriş yapabilir, dosya yükleyebilir ve kendilerine özel alanlarda bu dosyaları görüntüleyebilirler.
* **Backend tarafında C# ve ASP.NET Core**, **Frontend tarafında ise React** kullanılarak modern web teknolojileriyle entegre bir çözüm ortaya konmuştur. JWT ile kimlik doğrulama, SQL Server veritabanı kullanımı, form verisi işleme, dosya yönetimi gibi önemli yazılım konuları başarıyla uygulanmıştır.
* Bu çalışma, gelecekte yapılabilecek daha kapsamlı projeler için sağlam bir temel oluşturmuştur. Sistem, mobil platformlara taşınarak daha geniş kullanıcı kitlesine ulaştırılabilir.

**11. İletişim Bilgileri**

* **Adı Soyadı**: Osman Demir
* **GitHub**: <https://github.com/osmandemir2533>
* **LinkedIn**: <https://www.linkedin.com/in/osmandemir2533/>