

AĞ TABANLI TEKNOLOJİLER VE UYGULAMALARI



DÖNEM PROJE DOKÜMANI

notFacebook

Yılmaz Talha Kılıç

22290194

Github: <https://github.com/ylmztalhaklc/web-blog>

VideoLink:

<https://drive.google.com/file/d/1kKnhhy8yKbSi8ovrNWxUc6ylVjNoxce/view?usp=sharing>

1. Proje Tanımı ve Kapsamı

NotFacebook, temel sosyal medya işlevlerini içeren bir web uygulamasıdır. Proje kapsamında kullanıcıların sisteme kayıt olabildiği, giriş yapabildiği, gönderi paylaşabildiği, diğer kullanıcılarla etkileşime geçebildiği ve birebir mesajlaşabildiği bir yapı geliştirilmiştir.

Uygulama aşağıdaki temel işlevleri kapsamaktadır:

- Kullanıcı kayıt ve giriş işlemleri
- Gönderi oluşturma, listeleme ve silme
- Gönderilere beğeni ve yorum yapılması
- Kullanıcılar arası takip ilişkilerinin kurulması
- Birebir mesajlaşma
- Kullanıcı profillerinin görüntülenmesi

Proje, frontend ve backend bileşenlerinin birlikte çalıştığı, tam fonksiyonel bir web uygulaması geliştirmeyi hedeflemektedir.

2. Proje Mimarisi ve Dosya Yapısı

Uygulama istemci–sunucu (client–server) mimarisi ile geliştirilmiştir. Frontend ve backend katmanları ayrıdır ve REST tabanlı API’ler aracılığıyla haberleşmektedir.

Backend Yapısı

Backend tarafında katmanlı mimari kullanılmıştır:

- **NotFacebook.Core**
 - User.cs
 - Post.cs
 - PostComment.cs
 - PostLike.cs
 - Follow.cs
 - Message.cs

Bu sınıflar uygulamanın temel veri modellerini temsil etmektedir. Kullanıcılar, gönderiler, yorumlar, beğeniler, takip ilişkileri ve mesajlar bu katmanda tanımlanmıştır.

- **NotFacebook.Infrastructure**
 - AppDbContext.cs
 - Entity Framework Core yapılandırmaları

- Migration dosyaları

Bu katman veritabanı bağlantısını ve tablolar arası ilişkileri yönetmektedir. Veritabanı işlemleri ORM yaklaşımı ile gerçekleştirılmıştır.

- **NotFacebook.Api**

- AccountController.cs (Register, Login)
- PostsController.cs (Post oluşturma, silme, listeleme, beğeni, yorum)
- FollowController.cs (Takip ve takipten çıkışma işlemleri)
- MessagesController.cs (Mesaj gönderme ve listeleme)

API katmanı, frontend tarafından yapılan tüm isteklerin karşılandığı katmandır.

3. Frontend Yapısı ve Sayfa Bileşenleri

Frontend tarafında sayfa ve bileşen tabanlı bir yapı kullanılmıştır. Uygulamanın temel sayfaları ve sorumlulukları aşağıdaki gibidir:

- **AuthPage.jsx**

- Kullanıcı kayıt ve giriş işlemlerini yönetir.
- Register ve Login formları bu sayfa üzerinden kontrol edilmektedir.

- **HomePage.jsx**

- Uygulamanın ana sayfasıdır.
- Gönderi paylaşımı, gönderi akışı, beğeni ve yorum işlemleri bu sayfa üzerinden gerçekleştirilir.
- Takip edilen kullanıcıların gönderileri bu sayfada listelenir.
- Mesajlaşma paneli bu sayfa içerisinde açılmaktadır.

- **UserProfilePage.jsx**

- Kullanıcıya ait profil bilgilerini ve paylaşılan gönderileri gösterir.
- Takip ve takipçi bilgileri bu sayfada görüntülenmektedir.

- **Bileşenler (Components)**

- Gönderi kartları, yorum alanları ve mesajlaşma bileşenleri tekrar kullanılabilir component'ler olarak tasarlanmıştır.
- API iletişimini api.js dosyası üzerinden sağlanmaktadır.

4. Uygulama İşleyişi ve Etkileşim Mekanizmaları

Kullanıcı Yönetimi

- Kullanıcılar AccountController üzerinden sisteme kayıt olur ve giriş yapar.
- Giriş işlemi sonrası kullanıcı ana sayfaya yönlendirilir.
- Aynı e-posta adresi ile birden fazla hesap oluşturulması engellenmiştir.
- Kullanıcının 18 yaşından küçük olması durumunda kayıt işlemi reddedilmektedir.

Gönderi, Beğeni ve Yorum Sistemi

- Gönderiler PostsController üzerinden oluşturulmakta ve listelenmektedir.
- Beğeni sistemi tekil çalışır; bir kullanıcı aynı gönderiyi birden fazla kez beğenemez.
- Yorumlar PostComment yapısı üzerinden ilgili gönderi ile ilişkilendirilmiştir.
- Yorumlar yalnızca ilgili gönderi açıldığında görüntülenmektedir.

Takip Sistemi

- Takip işlemleri FollowController üzerinden gerçekleştirilmektedir.
- Kullanıcı kendisini takip edemez.
- Takip edilen kullanıcıların gönderileri ana akışta gösterilmektedir.

Mesajlaşma Sistemi

- Mesajlaşma işlemleri MessagesController üzerinden yürütülmektedir.
- Kullanıcılar metin ve görsel içeren mesajlar gönderebilmektedir.
- Mesajlarda okundu bilgisi veya teslim durumu gibi ek özellikler bulunmamaktadır.
- Mesajlaşma arayüzü frontend tarafında HomePage.jsx içinde açılan yan panel ile yönetilmektedir.

5. Profil Yapısı, Demo ve Sonuç

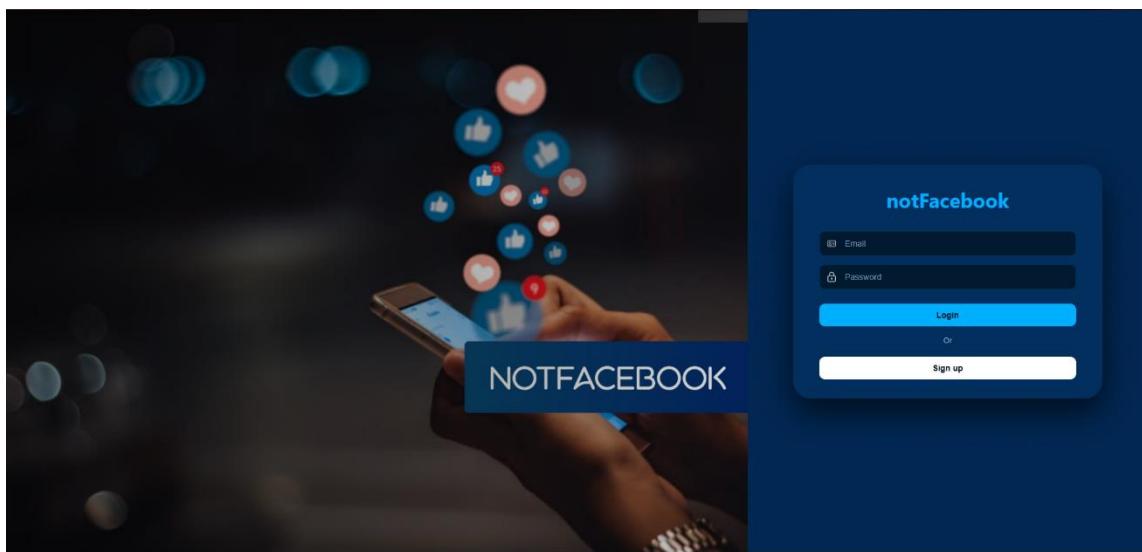
Profil işlemleri backend tarafında kullanıcı ve gönderi API'leri, frontend tarafında ise UserProfilePage.jsx üzerinden yönetilmektedir. Profil sayfasında:

- Kullanıcıya ait temel bilgiler
- Kullanıcının paylaştığı gönderiler
- Takip ve takipçi sayıları

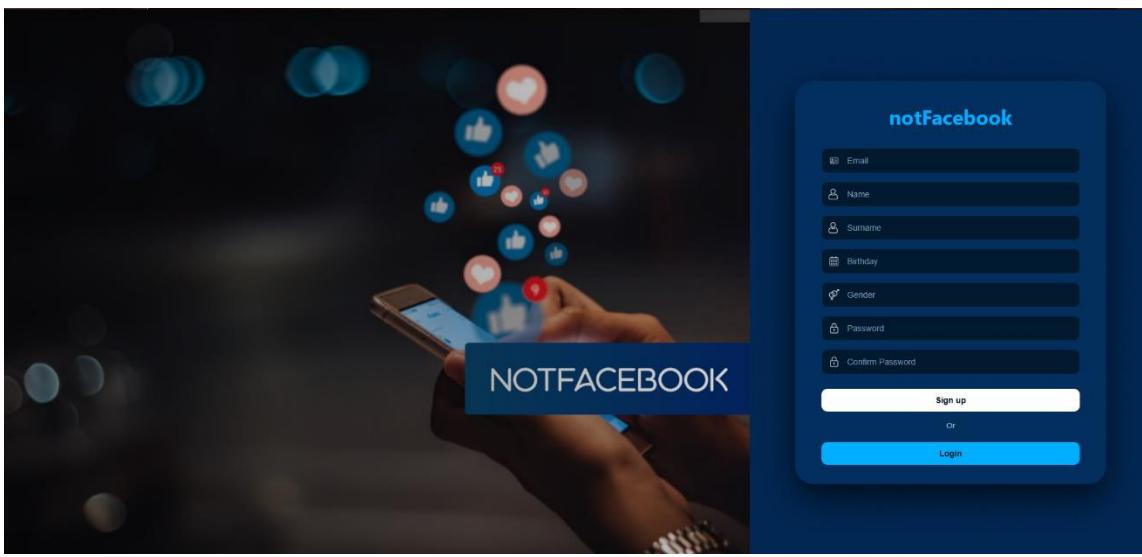
gösterilmektedir.

Uygulamanın çalışmasını göstermek amacıyla raporda aşağıdaki ekranlara ait demo görseller yer almaktadır:

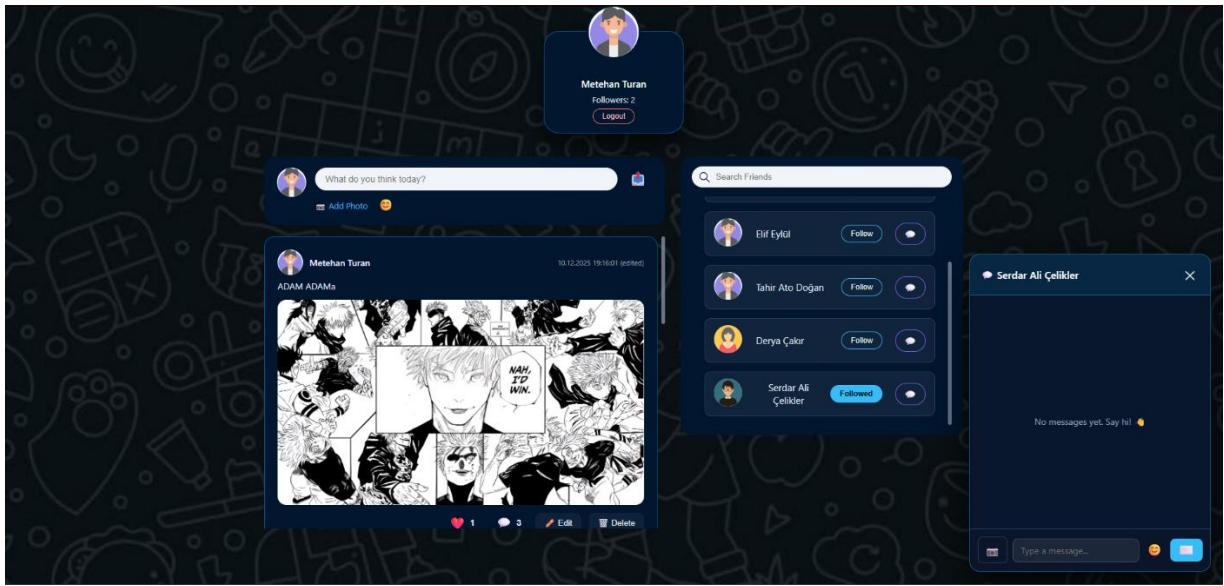
- AuthPage.jsx – Kayıt ve giriş ekranları
- HomePage.jsx – Gönderi paylaşımı ve akış
- Kullanıcı arama ve takip ekranları
- Mesajlaşma paneli
- UserProfilePage.jsx – Profil sayfası



1-Login Page



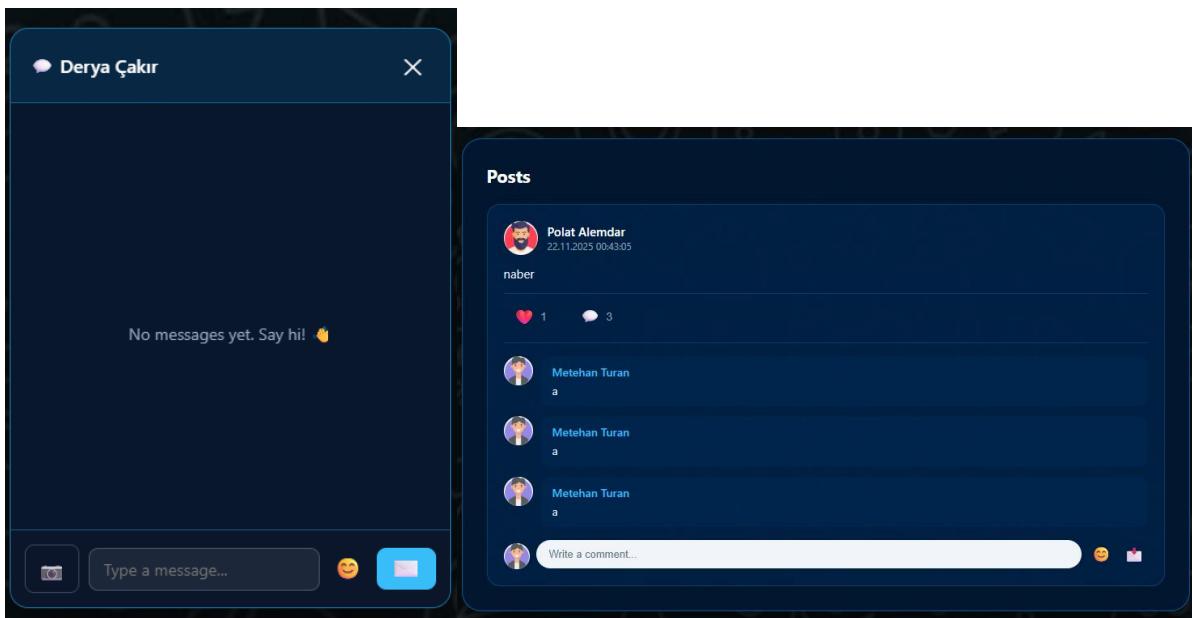
2-Register Page



3-Landing Page

A screenshot of a mobile application's profile page for "Polat Alemdar". At the top left is a "Back to Feed" button. The profile section features a circular profile picture of a man with a beard, the name "Polat Alemdar", and the email "polat@example.com". Below this are statistics: "1 POSTS" and "1 FOLLOWERS". There are two buttons: "Followed" (with a blue background) and "Message". The "Posts" section shows one post by "Polat Alemdar" from 22.11.2025 00:43:05 with the caption "naber". This post has 1 like and 3 comments. The entire profile card has a dark blue background.

4-Profile Page



5- Chat & Comments