

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4531 Ağ Tabanlı Teknolojiler ve Uygulamaları
Moto Rota (Motosiklet tutkunları için rota planlama, tur organizasyonu ve sosyal etkileşim platformu.)
Ağ Tabanlı Uygulama Dokümantasyonu

Yavuz Selim Demirhan

22290565

Github:

<https://github.com/yavuzdemirhan/BLM4531>

Video Linki:

<https://drive.google.com/file/d/1XYhiJv5lcrgrBvtalkOIN-vHZBxuNsMa/view?usp=sharing>

1. Proje Özeti ve Konsept

Proje Adı: Moto Rota

Konsept: Motosiklet tutkunlarını bir araya getiren, yeni rotalar keşfetmelerini ve toplu sürüş etkinlikleri (tur) düzenlemelerini sağlayan bir sosyal platformdur.

Amaç: Motosiklet kullanıcılarının sürüş deneyimlerini paylaşması, güvenli sürüş etkinlikleri organize etmesi ve bir topluluk oluşturmaları hedeflenmiştir. Kullanıcılar kendi rotalarını oluşturabilir, var olan turlara katılabilir ve diğer sürücülerle etkileşime geçebilir.

2. Kullanılan Teknolojiler

Proje, modern web teknolojileri kullanılarak Full-Stack bir mimaride geliştirilmiştir.

Backend (Sunucu Taraflı):

- Dil & Framework:** C# / ASP.NET Core Web API
- Veritabanı Erişimi:** Entity Framework Core (ORM)
- Mimari:** MVC (Model-View-Controller) tabanlı API yapısı
- Temel Özellikler:** RESTful API servisleri, Dependency Injection.

Frontend (İstemci Taraflı):

- Dil & Kütüphane:** JavaScript (ES6+), React.js
- Build Aracı:** Vite (Hızlı geliştirme ve derleme için)
- Yönlendirme (Routing):** React Router Dom
- Stil:** CSS (Modüler CSS dosyaları)

Veritabanı:

- Veritabanı:** MSSQL (Microsoft SQL Server)
- Yapı:** İlişkisel Veritabanı Modeli (Relational Database)

3. Proje Özellikleri ve Fonksiyonlar

Uygulama aşağıdaki temel modüllerden oluşmaktadır:

A. Kullanıcı İşlemleri (Account Management)

- Kayıt Ol & Giriş Yap:** Kullanıcılar sisteme güvenli bir şekilde üye olabilir ve giriş yapabilir.
- Profil Yönetimi (Profil.jsx):** Kullanıcılar profil bilgilerini güncelleyebilir.
- Garajım (Garage):** Kullanıcılar sahip oldukları motosikletleri sistemdeki "Garaj" bölümüne ekleyebilir (Motorcycle modeli).
- Takip Sistemi (Follows):** Kullanıcılar birbirlerini takip edebilir.

B. Tur ve Rota İşlemleri (Tour Management)

- Tur Oluşturma (TurEkle.jsx):** Kullanıcılar rota duraklarını, tur tarihini ve açıklamasını belirleyerek yeni bir tur oluşturabilir.
- Tur Detayları (TurDetay.jsx):** Turların detayları, rota bilgisi ve katılımcı listesi görüntülenebilir.
- Rota Durakları (RouteStop):** Her tur belirli durak noktalarından oluşur.

C. Etkileşim ve Sosyal Özellikler

- Tura Katılım (Participations):** Kullanıcılar "Katıl" butonu ile turlara katılım isteği gönderebilir (Etkinliklerim.jsx).
- Yorum ve Değerlendirme:** Turlara yorum yapılabılır (Comments) ve puan verilebilir (Ratings).
- Favoriler:** Kullanıcılar beğenikleri turları favorilere ekleyebilir.

D. Diğer Özellikler

- Acil Durum Bilgileri (Emergency):** Sürücüler için acil durum iletişim bilgileri saklanabilir.
- İpuçları ve Bilgi (Ipuclari.jsx):** Motosiklet sürüşü ile ilgili faydalı bilgiler veya blog içerikleri sunulur.
- Admin Paneli:** Yönetici işlemleri için özel panel.

4. Veritabanı Tabloları ve Görevleri

Projede kullanılan veritabanı tabloları ve her birinin kullanım amacı aşağıda açıklanmıştır:

Tablo Adı	Açıklama ve Kullanım Amacı
AspNetUsers	Kullanıcıların kimlik bilgilerini (e-posta, şifre, kullanıcı adı), profil fotoğrafını ve kişisel bilgilerini saklar.
Tours	Oluşturulan turların temel bilgilerini (Başlık, Tarih, Açıklama, Başlangıç Lokasyonu vb.) tutar.
RouteStops	Bir tura ait rota duraklarını tutar. Her turun birden fazla durağı olabilir (One-to-Many ilişkisi).
Motorcycles	Kullanıcıların garajlarına ekledikleri motosikletlerin marka, model ve yıl bilgilerini saklar.
Participations	Hangi kullanıcının hangi tura katıldığını takip eder. Kullanıcı ve Tur arasındaki çoka-çok ilişkiyi yönetir.
Comments	Kullanıcıların turlar hakkında yaptığı yorumları içerişir.
TourRatings	Kullanıcıların turlara verdikleri puanları (1-5 arası) saklar.
Favorites	Kullanıcıların favorilerine eklediği turları listeler.
UserFollows	Kullanıcıların birbirlerini takip etme durumlarını (Takip eden - Takip edilen) saklar.
EmergencyInfos	Kullanıcıların acil durumlarda aranacak kişi bilgilerini ve kan grubu gibi detayları tutar.

5. Proje Geliştirme Takvimi (2 Aylık Süreç)

Projenin geliştirme süreci toplamda 8 haftalık bir plana yayılmıştır:

Hafta	Aşama	Yapılan Çalışmalar
1. Hafta	Planlama ve Analiz	Proje konusunun belirlenmesi, gereksinim analizi, veritabanı şemasının tasarllanması.
2. Hafta	Kurulum ve Altyapı	Frontend (React+Vite) ve Backend (ASP.NET Core) projelerinin oluşturulması, Git reposunun açılması.
3. Hafta - I	Backend Geliştirme	Veritabanı bağlantısı (EF Core), Kimlik doğrulama (Auth) yapısının kurulması, User controller'ların yazılması.
4. Hafta - II	Backend Geliştirme	Tur işlemleri, Rota durakları ve İletişim API'lerinin geliştirilmesi. Swagger ile testler.
5. Hafta - I	Frontend Geliştirme	Temel tasarımın oluşturulması (CSS), Giriş/Kayıt sayfalarının ve Ana Sayfanın kodlanması.
6. Hafta - II	Frontend Geliştirme	Tur Ekleme, Profil, Detay sayfalarının yapılması. API entegrasyonlarının sağlanması.
7. Hafta	Entegrasyon ve Test	Sosyal özelliklerin (Yorum, Puan, Takip) eklenmesi. Hataların giderilmesi ve test süreçleri.
8. Hafta	Dokümantasyon ve Final	Proje dokümanının hazırlanması, kod temizliği ve projenin teslime hazır hale getirilmesi.