

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 447 MOBİL UYGULAMA GELİŞTİRME**

**E-LABORATUVAR SİSTEMİ**

**B221210350 – Abdullah BİLGİ**  
**B201210554 - Eray SARI**

**Bölüm** : **BİLGİSAYAR MÜHENDİSLİĞİ**  
**Danışman** : **Öğr. Üyesi Ahmet ŞANSLI**

**2024-2025 Güz Dönemi**

## **1. PROJENİN AMACI**

Verilen kılavuzlara göre hastanın kan tahlili değerlerini değerlendirip yorumlamak.

## **2. BACKEND**

Projenin backend tarafı Springboot ile yapıldı.Database tarafında ise PostgreSql kullanıldı.

### **2.1. Bootstrap Package**

Bu adımda verilen kılavuz değerleri database'e kaydedildi.Ek olarak admin de eklendi.

### **2.2. Config Package**

Bu adımda config sınıfında Authentication ve Authorization işlemleri yapıldı.Password şifrelemesi için ByCryiptPasswordEncoder kullanıldı..React native için de CORS konfigürasyonları yapıldı.

### **2.3. Controller Package**

Bu adımda User,BloodAnalysis ve ValueRanges için endpointler oluşturuldu.Gerekli istekler yapılarak veri tabanından veriler gönderildi.

### **2.4.Enums Package**

Bu adımda projedeki sabitler yazıldı.Tekrar tekrar kullanmamak için.

### **2.5. Handler Package**

Bu adımda oluşabilecek hatalar yakalandı ve hata oluşunca gerekli hata mesajları gönderildi.

## **2.6.Model Package**

Bu adımda projede kullanılmak üzere kurgulanan User,BloodAnalysis ve ValueRanges sınıfları oluşturuldu ve içleri gerekli şekilde dolduruldu.

## **2.7. Repository Package**

Bu adımda modeller için veri tabanı işlemleri yapıldı.

## **2.8. Service Package**

Bu adımda repository den gelen verilere ek işlemler uygulandı ve isteğe bağlı proje logicleri kullanıldı.

## **2.9. Application.properties Dosyası**

Bu adımda veri tabanı bağlantısı yapıldı. Cors için react native port adresi verildi.

# **3. REACT-NATIVE**

Bu bölümde, hastanın kan tahlili değerlerini kılavuzlara göre yorumlayan bir mobil uygulama geliştirildi. Proje, kullanıcı dostu bir arayüz sunmak ve rehberlere dayalı doğru analizler yapmak amacıyla oluşturuldu.

## **3.1. Kullanılan Teknolojiler ve Araçlar**

- React Native: Mobil uygulamanın geliştirilmesi için kullanıldı.
- React Query: API isteklerini yönetmek ve veri önbelleklemesi sağlamak için kullanıldı.

- React Navigation: Uygulamada sayfa geçişleri ve yönlendirme işlemleri için kullanıldı.
- Formik ve Yup: Form doğrulama işlemleri için kullanıldı.
- Tailwind CSS: Arayüzde hızlı ve temiz bir stil uygulamak için kullanıldı.
- React Context: Kullanıcı oturum yönetimi ve global state yönetimi için kullanıldı.

### 3.2. API İstekleri ve Veri Yönetimi

Projenin veri yönetimi React Query ile sağlandı. useQuery ve useMutation hook'ları kullanılarak aşağıdaki işlemler gerçekleştirildi:

- Kullanıcı bilgilerini almak için useQuery kullanıldı.
- Yeni kan tahlili analizleri eklemek ve rehber değerlerini sorgulamak için useMutation kullanıldı.
- Veriler React Query ile önbelleğe alındı, bu da performansı artırdı ve kullanıcı deneyimini geliştirdi.

### 3.3. Uygulama Yapısı

- Context (AuthContext): Kullanıcı oturum bilgilerini yönetmek için bir React Context kullanıldı. Kullanıcı giriş yapıldığında, getUser fonksiyonu çağrılarak kullanıcı bilgileri çekildi ve oturum bilgileri global state'e aktarıldı.
- Formlar: Hasta bilgileri ve kan tahlili değerlerinin girilmesi için dinamik formlar oluşturuldu. Formik ve Yup kullanılarak bu formlar valide edildi.
- Tablolar: Kan tahlili değerleri, kılavuzdaki aralıklara göre bir tabloda gösterildi. Tablo bileşeni tamamen özelleştirildi ve her bir rehber değeri için karşılaştırma sonucu açıkça belirtildi.