

YAZ OKULU Proje Raporu

NAGIP TAYLAN GÖZALAN - 4004

 <https://yazokulu.vercel.app>

 [voidnulljs/yazokulu](https://github.com/voidnulljs/yazokulu)

“Bu proje Vercel platformunda demo sürümü olarak yayınlanmaktadır.”

Proje Tanıtımı

Responsive Tasarım

Güncel Teknolojiler

Kolay Arayüz

Bu proje, yaz okulu süreçlerini yönetmek için geliştirilmiş kapsamlı bir web uygulamasıdır. Öğrencilerin ders seçimi, bakiye yükleme ve not görüntüleme işlemlerini; öğretmenlerin ise ders açma, öğrenci takibi ve not girişi işlemlerini gerçekleştirmesini sağlar.

Kullanılan Teknolojiler

Frontend

Next.js (App Router), React, Tailwind CSS, Lucide React (ikonlar)

Backend

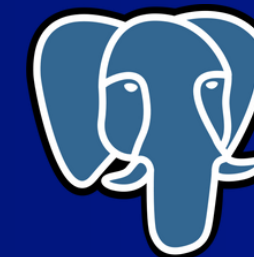
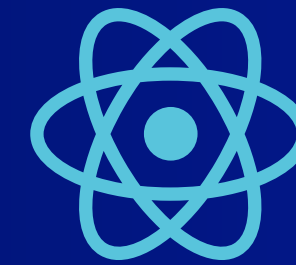
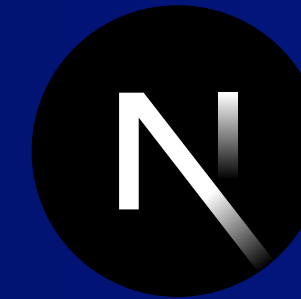
Supabase (PostgreSQL)

Programlama Dili

TypeScript

UI Bileşenleri

Shadcn/ui (Radix UI tabanlı)



Veritabanı Yapısı

- **ogrenciler:** Öğrenci bilgilerini tutar (ad soyad, öğrenci no).
- **ogretmenler:** Öğretmen bilgilerini tutar (ad soyad, unvan).
- **dersler:** Açılan derslerin bilgilerini içerir (kod, ad, kontenjan, fiyat, durum).
- **ders_talepleri:** Öğrencilerin açılmasını istediği ders taleplerini tutar.
- **ders_kayitlari:** Kesinleşen ders kayıtlarını ve not bilgilerini tutar.
- **bakiyeler:** Öğrencilerin güncel bakiye bilgisini tutar.
- **islemler:** Bakiye yükleme ve harcama geçmişini tutar.
- **yedek_listesi:** Kontenjanı dolan dersler için yedek kayıtları tutar.

Proje Yapısı ve Katmanlar

Proje, modern Next.js App Router mimarisine uygun olarak katmanlı bir yapıda geliştirilmiştir:

- **App Layer (`app/`)**: Sayfa yönlendirmeleri (routing), layout'lar ve sayfa bileşenlerini içerir.
 - **`app/ogrenci`**: Öğrenci paneli sayfaları (Ders seçimi, bakiye, vb.).
 - **`app/ogretmen`**: Öğretmen paneli sayfaları (Ders yönetimi, not girişi, vb.).
 - **`app/auth`**: Kimlik doğrulama sayfaları (Giriş, kayıt, çıkış).
- **Component Layer (`components/`)**: Uygulama genelinde tekrar kullanılan UI bileşenleri (Button, Card, Dialog vb.) burada bulunur.
- **Infrastructure Layer (`lib/`)**: Veritabanı bağlantıları ve yardımcı fonksiyonlar yer alır.
 - **`lib/supabase`**: Supabase istemci ve sunucu bağlantı konfigürasyonları.
- **Database Layer (`scripts/`)**: Veritabanı şemasını, saklı yordamları (stored procedures) ve tetikleyicileri (triggers) içeren SQL dosyaları.
- **Middleware**: Sayfa geçişlerinde ve oturum yönetiminde kullanılan ara katman yazılımı.

Veritabanı Bağlantısı ve Kurulum

Proje Supabase ile entegre çalışır. Kurulum için:

1. Bir Supabase projesi oluşturun.
2. `scripts/script.sql` dosyasındaki SQL komutlarını Supabase SQL Editöründe çalıştırın. Bu işlem tüm tabloları, fonksiyonları ve güvenlik politikalarını (RLS) oluşturacaktır.
3. Projenizi Supabase URL ve Anon Key ile bağlayın.

Sonuç ve Kazanımlar

Bu proje ile aşağıdaki kazanımlar elde edilmiştir:

- **Otomatik Kontenjan ve Yedek Yönetimi:** Ders kayıtlarında kontenjan dolduğunda öğrenciler otomatik olarak yedek listesine alınır.
- **Bakiye Kontrollü Kayıt:** Öğrenciler sadece bakiyeleri yeterliyse ders kaydı yapabilirler.
- **Dinamik Ders Yönetimi:** Öğretmenler dersleri taslak olarak oluşturup, talebe göre açabilir veya iptal edebilir.
- **Güvenli Veri Erişimi:** Row Level Security (RLS) politikaları ile veri güvenliği sağlanmıştır (Not: Geliştirme sürecinde kolaylık sağlaması için politikalar esnek tutulmuştur).
- **Kullanıcı Dostu Arayüz:** Modern ve responsive bir arayüz ile kolay kullanım hedeflenmiştir.



Teşekkürler