# 🚀 WhatsApp Messaging SaaS Platform

WhatsApp SaaS React TypeScript Supabase

Modern ve kapsamlı bir WhatsApp mesajlaşma mikroservisi platformu. React frontend, Supabase backend ve WhatsApp Web entegrasyonu ile tam özellikli bir SaaS çözümü.

## ✨ Özellikler

### 🔐 Kimlik Doğrulama

* Güvenli kullanıcı kaydı ve giriş sistemi
* Email doğrulama
* Rol tabanlı erişim kontrolü (Admin/User)
* Session yönetimi

### 📱 WhatsApp Entegrasyonu

* **WhatsApp Web QR Kod Bağlantısı**: Gerçek WhatsApp hesabınızı platform üzerinden yönetin
* **Toplu Mesaj Gönderimi**: Birden fazla kişiye aynı anda mesaj gönderin
* **Media Desteği**: Resim, video ve dosya paylaşımı
* **Şablon Mesajları**: Önceden hazırlanmış mesaj şablonları

### 👥 Kişi Yönetimi

* Kişi ekleme, düzenleme ve silme
* Kişi grupları oluşturma
* CSV dosyası ile toplu kişi içe aktarma
* Kişi filtreleme ve arama

### 📝 Mesaj Şablonları

* Özelleştirilebilir mesaj şablonları
* Değişken kullanımı (isim, tarih, vs.)
* Şablon kategorileri
* Drag & drop editör

### 📊 Analitik ve Raporlama

* Gönderilen mesaj istatistikleri
* Başarı oranları
* Detaylı analitik dashboard
* Export özellikleri

### 🎛️ Admin Paneli

* Kullanıcı yönetimi
* Sistem ayarları
* Uygulama konfigürasyonu
* Güvenlik ayarları

## 🛠️ Teknoloji Stack

### Frontend

* **React 18.x** - Modern UI framework
* **TypeScript** - Type-safe JavaScript
* **Vite** - Lightning fast build tool
* **TailwindCSS** - Utility-first CSS framework
* **React Router** - Client-side routing
* **React Hook Form** - Form management
* **Lucide React** - Beautiful icons

### Backend

* **Supabase** - Backend-as-a-Service
* **PostgreSQL** - Robust database
* **Row Level Security (RLS)** - Database security
* **Edge Functions** - Serverless API
* **Real-time subscriptions** - Live updates

### DevOps & Deployment

* **Vercel** - Frontend deployment
* **Railway** - Microservice deployment
* **GitHub Actions** - CI/CD pipeline
* **ESLint & Prettier** - Code quality

## 🚀 Hızlı Başlangıç

### Ön Gereksinimler

* Node.js 18.x veya üzeri
* npm veya pnpm
* Supabase hesabı
* WhatsApp telefon numarası

### Kurulum

1. **Repository’yi klonlayın**

git clone https://github.com/hiktan44/whatsappmini.git  
cd whatsappmini

1. **Bağımlılıkları yükleyin**

# Frontend  
cd pywhatapp-platform  
pnpm install  
  
# Microservice  
cd ../whatsapp-railway-microservice  
npm install

1. **Environment dosyalarını ayarlayın**

# Frontend .env dosyası  
VITE\_SUPABASE\_URL=your\_supabase\_url  
VITE\_SUPABASE\_ANON\_KEY=your\_supabase\_anon\_key  
  
# Microservice .env dosyası  
SUPABASE\_URL=your\_supabase\_url  
SUPABASE\_SERVICE\_ROLE\_KEY=your\_service\_role\_key

1. **Supabase database’i kurun**

# Tabloları oluştur  
psql -f supabase/tables/\*.sql  
  
# Migrations'ları çalıştır  
cd supabase  
supabase db push

1. **Uygulamayı başlatın**

# Frontend (Development)  
cd pywhatapp-platform  
pnpm dev  
  
# Microservice  
cd whatsapp-railway-microservice  
npm start

## 📁 Proje Yapısı

whatsappmini/  
├── pywhatapp-platform/ # React frontend  
│ ├── src/  
│ │ ├── components/ # Reusable components  
│ │ ├── pages/ # Page components  
│ │ ├── hooks/ # Custom hooks  
│ │ ├── contexts/ # React contexts  
│ │ └── lib/ # Utilities  
├── whatsapp-railway-microservice/ # WhatsApp microservice  
├── supabase/ # Database & Edge Functions  
│ ├── functions/ # Edge Functions  
│ ├── migrations/ # Database migrations  
│ └── tables/ # Table schemas  
├── docs/ # Documentation  
└── tests/ # Test files

## 🔧 Konfigürasyon

### Supabase Kurulumu

1. [Supabase Console](https://supabase.io)’dan yeni proje oluşturun
2. Database şemalarını supabase/tables/ klasöründen içe aktarın
3. Edge Functions’ları deploy edin
4. RLS politikalarını etkinleştirin

### WhatsApp Entegrasyonu

1. Uygulamaya giriş yapın
2. “WhatsApp Web” sayfasına gidin
3. QR kodu telefon uygulaması ile tarayın
4. Bağlantı kurulduktan sonra mesaj göndermeye başlayın

## 🧪 Test Edilmiş Özellikler

* ✅ Kullanıcı kimlik doğrulama sistemi
* ✅ WhatsApp Web QR kod bağlantısı
* ✅ Kişi yönetimi (CRUD işlemleri)
* ✅ Mesaj şablonları
* ✅ Media upload sistemi
* ✅ Admin panel erişim kontrolü
* ✅ Database security (RLS)
* ✅ Real-time güncellemeler

## 🔒 Güvenlik

* **Row Level Security (RLS)** - Database seviyesinde güvenlik
* **JWT Token Authentication** - Güvenli kimlik doğrulama
* **HTTPS Enforced** - Şifreli veri iletimi
* **Input Validation** - XSS ve injection koruması
* **Rate Limiting** - API abuse koruması

## 🚀 Deployment

### Frontend (Vercel)

1. Vercel hesabınızı GitHub’a bağlayın
2. Repository’yi import edin
3. Environment variables’ları ayarlayın
4. Deploy edin

### Microservice (Railway)

1. Railway’e proje yükleyin
2. Environment variables’ları configure edin
3. Automatic deployment ayarlayın

## 📊 Performance

* **Frontend**: 95+ Lighthouse Score
* **API Response Time**: <200ms average
* **Database Queries**: Optimized with indexes
* **Real-time Updates**: <100ms latency

## 🤝 Katkıda Bulunma

1. Fork yapın
2. Feature branch oluşturun (git checkout -b feature/amazing-feature)
3. Commit edin (git commit -m 'Add amazing feature')
4. Branch’i push edin (git push origin feature/amazing-feature)
5. Pull Request açın

## 📄 Lisans

Bu proje MIT lisansı altında lisanslanmıştır. Detaylar için <LICENSE> dosyasına bakın.

## 👨‍💻 Geliştirici

**hiktan44** - [GitHub Profile](https://github.com/hiktan44)

## 🆘 Destek

Herhangi bir sorun yaşarsanız: - [Issues](https://github.com/hiktan44/whatsappmini/issues) sayfasında yeni bir konu açın - [Discussions](https://github.com/hiktan44/whatsappmini/discussions) sayfasında soru sorun

## 🙏 Teşekkürler

* [Supabase](https://supabase.io) - Harika BaaS hizmeti için
* [Vercel](https://vercel.com) - Ücretsiz hosting için
* [Railway](https://railway.app) - Kolay deployment için
* [TailwindCSS](https://tailwindcss.com) - Modern styling için

⭐ **Bu projeyi beğendiyseniz yıldız vermeyi unutmayın!**