

Mini CRM Projesi -Geliştirme Sürecim ve Teknoloji Seçimlerim (ELIF CETİN)

Neden Bu Teknolojileri Seçtim?

1. Next.js 14 - Ana Framework

Neden Next.js?

- **Tek proje, iki işlev:** Hem web sitesi (frontend) hem de sunucu (backend) aynı projede
- **Hızlı geliştirme:** Hazır sayfa yapısı ve API sistemi
- **Kolay deploy:** Vercel gibi platformlarda tek tıkla yayınlama
- **SEO dostu:** Arama motorları için optimize edilmiş

Alternatifler nelerdi?

- React + Express: İki ayrı proje yönetmek gerekiyordu
- Vue.js: Daha az popüler, iş bulma açısından dezavantaj
- Vanilla JavaScript: Çok fazla kod yazmak gerekiyordu

2. TypeScript - Programlama Dili

Neden TypeScript?

- **Hata önleme:** Kod yazarken hataları yakalama
- **Daha iyi kod yazma:** Hangi veri tipini kullandığını bilme
- **İş piyasası:** Şirketler TypeScript tercih ediyor
- **Geliştirici deneyimi:** Daha iyi kod tamamlama

Alternatif: JavaScript

- Daha esnek ama hata yapma riski yüksek
- Büyük projelerde karmaşıklık yaratır

3. SQLite - Veritabanı

Neden SQLite?

- **Kurulum gerektirmez:** Dosya tabanlı, her yerde çalışır
- **Basit:** Karmaşık veritabanı sunucusu yok
- **Ücretsiz:** Ek maliyet yok
- **Taşınabilir:** Dosyayı kopyalayıp başka yere taşıyabilirsin

Alternatifler:

- PostgreSQL: Güçlü ama kurulum ve yönetim gerektirir
- MongoDB: NoSQL, farklı veri yapısı
- MySQL: Kurulum ve sunucu gerektirir

4. JWT - Kimlik Doğrulama

Neden JWT?

- **Güvenli:** Şifreli token sistemi
- **Stateless:** Sunucuda oturum bilgisi saklamaya gerek yok
- **Hızlı:** Her istekte kullanıcıyı tanıma
- **Yaygın:** Çoğu şirket kullanıyor

Alternatif: Session

- Sunucuda oturum bilgisi saklama gerektirir
- Daha karmaşık yönetim

5. TailwindCSS - Tasarım

Neden TailwindCSS?

- **Hızlı tasarım:** Hazır CSS sınıfları
- **Tutarlılık:** Tüm sayfalarda aynı görünüm
- **Responsive:** Mobil ve masaüstü uyumlu
- **Küçük boyut:** Sadece kullandığın CSS'ler dahil edilir

Alternatifler:

- Bootstrap: Daha büyük, daha az esnek
- CSS Modules: Daha fazla kod yazma
- Styled Components: React'e bağımlı

Proje Yapısı Neden Bu Şekilde?

Dosya Organizasyonum:

src/

```
|—— app/      # Sayfalar ve API'ler
|  |—— api/    # Backend kodları
|  |—— customers/ # Müşteri sayfaları
|  |—— login/   # Giriş sayfası
|  |—— register/ # Kayıt sayfası
|—— lib/      # Yardımcı fonksiyonlar
|—— types/    # Veri tipleri
|—— public/   # Resimler ve dosyalar
```

Neden bu yapı?

- Next.js 14 App Router: Yeni ve önerilen yapı
- **Kolay bulma:** Her şey kendi yerinde
- **Büyüme:** Yeni özellikler eklemek kolay

Güvenlik Seçimlerim

Şifre Hashleme

- **bcryptjs:** Şifreleri güvenli şekilde saklama
- **Salt:** Her şifreye özel ek güvenlik

JWT Token

- **Expiration:** Token'ların süresi doluyor
- **Secret Key:** Güvenli anahtar kullanımı

Input Validation

- **Sunucu tarafı:** Tüm veriler sunucuda kontrol ediliyor
- **TypeScript:** Veri tiplerini kontrol etme

Kullanıcı Deneyimi Seçimlerim

Responsive Tasarım

- **Mobil öncelikli:** Telefonlarda da çalışır
- **Touch-friendly:** Dokunmatik ekranlar için optimize

Loading States

- **Kullanıcı geri bildirimi:** İşlemler sırasında bilgi verme
- **Error handling:** Hataları güzel şekilde gösterme

Navigation

- **Kolay gezinme:** Menüler ve butonlar net
- **Breadcrumbs:** Nerede olduğunu bilme

Deploy Seçimlerim

Vercel

Neden Vercel?

- **Next.js uyumlu:** En iyi performans
- **Ücretsiz:** Kişisel projeler için yeterli
- **Otomatik deploy:** GitHub'a push ettiğinde otomatik yayınlama
- **Hızlı:** CDN ile dünya çapında hızlı erişim

Environment Variables

- **Güvenlik:** Hassas bilgileri kodda saklamama
- **Esneklik:** Farklı ortamlar için farklı ayarlar

Test Stratejim

Postman Koleksiyonu

Neden Postman?

- **API testi:** Backend'i test etme
- **Dokümantasyon:** Her endpoint'in nasıl kullanılacağı
- **Takım çalışması:** Başkaları da test edebilir

Alternatif: Jest

- Daha karmaşık kurulum
- Daha fazla kod yazma gerektirir

Gelecek Planlarım

Eklenebilecek Özellikler

- E-posta bildirimleri: Müşteri takibi
- **Raporlama:** Satış ve müşteri analizi
- Takvim entegrasyonu: Toplantı planlama
- **Dosya yükleme:** Müşteri belgeleri

Ölçeklendirme

- **PostgreSQL:** Daha büyük veritabanı
- **Redis:** Hızlı önbellek
- Microservices: Modüler yapı

Öğrendiğim Dersler

Teknoloji Seçimi

- **İhtiyaca göre seçim:** En popüler değil, en uygun
- **Basitlik:** Karmaşık çözümler yerine basit ve etkili
- **Gelecek odaklı:** Güncel teknolojiler kullanma

Proje Yönetimi

- **Adım adım geliştirme:** Her özelliği ayrı ayrı ekleme
- **Test etme:** Her aşamada kontrol etme
- **Dokümantasyon:** Kod açıklamaları ve README

Kullanıcı Odaklılık

- **Basit arayüz:** Karmaşık olmayan tasarım
- **Hızlı yükleme:** Performans önemli
- **Mobil uyumlu:** Her cihazda çalışma

Sonuç

Bu proje, modern web geliştirme teknolojilerini kullanarak, gerçek ihtiyaçları karşılayan bir uygulama oluşturmanın mükemmel bir örneğidir.

Her teknoloji seçimi, projenin ihtiyaçlarına ve gelecek planlarına göre yapılmıştır.

Basitlik, performans ve kullanıcı deneyimi ön planda tutulmuştur.