Yapay Zeka Destekli Sohbet Sistemi (AI Chat Bot)

Final Gerçekleştirim Raporu ve Ortak Çalışma Kılavuzu

1. Giriş

Bu rapor, modern web teknolojileri ve büyük dil modelleri (LLM) entegrasyonu ile geliştirilen **"AI Chat Bot"** projesinin gerçekleştirim sürecini, teknik mimarisini ve sonuçlarını kapsamaktadır. Proje, Yazılım İnşaası dersi kapsamında hazırlanan analiz ve tasarım raporları temel alınarak; nesne yönelimli programlama (OOP) prensipleri, SOLID kuralları ve modüler mimari esaslarına uygun olarak hayata geçirilmiştir.

Raporun amacı; sistemin kodlama sürecini belgelemek, analiz-tasarım uyumunu değerlendirmek, geliştirilen özellikleri teknik detaylarıyla sunmak, kullanıcı kılavuzunu sağlamak ve proje ekibinin kullandığı ortak çalışma metodolojilerini (Git/GitHub) sunmaktır.

2. Gerçekleştirim Süreci ve Kullanılan Araçlar

Sistemin geliştirilmesinde, yüksek performans, tip güvenliği ve ölçeklenebilirlik sağlayan modern bir teknoloji yığını (Tech Stack) tercih edilmiştir. Uygulama, analiz raporunda belirtilen sistem kısıtlarına tam uyumlu şekilde web tabanlı olarak kodlanmıştır.

Gerçekleştirim süreci şu adımlar izlenerek yürütülmüştür:

2.1. Proje Ekibi ve Görev Dağılımı

Proje geliştirme süreci, Agile prensiplerine uygun olarak yürütülmüş ve görevler uzmanlık alanlarına göre aşağıdaki gibi dağıtılmıştır:

Adı Soyadı	Rolü	Sorumluluk Alanları
Abdulmouen kanatri	**Proje Lideri & Backend Dev.**	Proje mimarisinin kurulması (Init), Veritabanı tasarımı (Prisma/PostgreSQL), API Route geliştirimi, Git Repository yönetimi ve Merge işlemleri.
MOHAMED ABIYA	**Frontend Geliştirici**	Kullanıcı arayüzü (UI) tasarımı, React bileşenleri (Components), Responsive tasarım ve İstemci tarafı (Client-Side) mantığı.

1. **Proje Kurulumu ve Versiyon Kontrolü :**Proje Lideri tarafından GitHub üzerinde ortak depo (Repository) oluşturulmuş, feature/project-initialization dalı üzerinden Next.js mimarisi ve Backend altyapısı kurulmuştur. Takım üyeleri, belirlenen kurallar çerçevesinde özellik dalları (feature branches) açarak **geliştirmelere katkı sağlamıştır."**
2. **Katmanlı Kodlama:**
    a. **Sunum Katmanı (Presentation):** `app/` dizini altında React bileşenleri ve Tailwind CSS ile kullanıcı arayüzü tasarlanmıştır.
    b. **İş Mantığı Katmanı (Business Logic):** Vercel AI SDK kullanılarak yapay zeka iletişim mantığı `api/chat` rotasında merkezileştirilmiştir.
    c. **Veri Katmanı (Data):** Prisma ORM kullanılarak veritabanı şemaları oluşturulmuş ve PostgreSQL bağlantısı Singleton deseni ile sağlanmıştır.

3. **Test ve İyileştirme:** Sohbet akışı, geçmiş kaydı ve oturum açma işlemleri test edilmiş, "Streaming" (akışkan yanıt) optimizasyonları yapılmıştır.

Kullanılan Temel Teknolojiler:

- **Next.js (App Router):** Server ve Client Components ayrımı ile performans maksimizasyonu.
- **TypeScript:** Statik tipleme ile kod güvenliği ve hata minimizasyonu.
- **Vercel AI SDK:** LLM modelleri ile streaming (akışkan) iletişim.
- **Prisma & PostgreSQL:** Güvenli veri saklama ve ORM yapısı.
- **Tailwind CSS:** Modern ve responsive arayüz tasarımı.

3. Analiz ve Tasarım Raporları ile Uyum

Sistem, analiz ve tasarım aşamalarında belirlenen gereksinimleri eksiksiz karşılayacak şekilde geliştirilmiştir.

3.1. Fonksiyonel Gereksinimlerin Karşılanması

Gereksinim	Karşılanma Durumu	Açıklama
FR1 – Sohbet Yönetimi	Tamamen Karşılandı	AI modelleriyle gerçek zamanlı veri akışı ve yanıt üretimi.
FR2 – Geçmiş Yönetimi	Tamamen Karşılandı	Eski sohbetler veritabanında saklanır, listelenir ve yönetilir.
FR3 – Kimlik Doğrulama	Tamamen Karşılandı	NextAuth ile güvenli oturum açma ve veri izolasyonu sağlanmıştır.
FR4 – Arayüz İşlevleri	Tamamen Karşılandı	Kod blokları renklendirilir (Syntax Highlighting) ve Markdown desteklenir.

3.2. Tasarım Uyumu

Tasarım raporunda öngörülen **Katmanlı Mimari (Layered Architecture)** yapıya sadık kalınmıştır:

- **Sunum Katmanı:** Kullanıcı etkileşimi `app/` dizinindeki React bileşenleri ile yönetilir.
- **İş Mantığı Katmanı:** AI servis mantığı API rotalarında izole edilmiştir.
- **Veri Katmanı:** Veri erişimi Prisma Client (Singleton Pattern) üzerinden güvenle yönetilir.

4. Gerçekleştirilen Sistem Özellikleri

4.1. Akıllı Sohbet Arayüzü

Kullanıcı mesajı yazdığında sayfa yenilenmeden yanıt ekrana kelime kelime yazılır (Streaming). Bu özellik, bekleme süresi algısını yok eder.

4.2. Sohbet Geçmişi ve Kenar Çubuğu

Kullanıcıların eski sohbetlerine erişebilmesi için sol tarafta bir kenar çubuğu (Sidebar) tasarlanmıştır. Başlıklar otomatik olarak sohbetin içeriğine göre üretilir.

5. Kullanıcı Kılavuzu

Bu bölüm, uygulamanın nasıl çalıştırılacağını ve kullanılacağını açıklar.

5.1. Programın Başlatılması

1. Projeyi GitHub'dan bilgisayarınıza indirin (Clone).
2. Terminali açın ve proje klasörüne gidin.
3. Bağımlılıkları yükleyin: `npm install`

4. Veritabanını hazırlayın: `npx prisma db push`
5. Uygulamayı başlatın: `npm run dev`
6. Tarayıcınızda http://localhost:3000 adresine gidin.

5.2. Kullanım Senaryoları

- **Giriş Yapma:** Ana sayfada "Giriş Yap" butonuna tıklayın ve kimlik bilgilerinizi girin.
- **Sohbet Başlatma:** Alt kısımdaki metin kutusuna sorunuzu yazın ve "Gönder" ikonuna tıklayın.
- **Geçmişe Erişim:** Sol menüden eski bir sohbet başlığına tıklayarak konuşmaya kaldığınız yerden devam edin.
- **Yeni Sohbet:** Menüdeki "+" butonuna basarak ekranı temizleyin ve yeni bir konu başlatın.

6. Tasarımdan Sapmalar ve Alınan Kararlar

Proje kapsamında, performans ve güncel teknolojilere uyum sağlamak amacıyla bazı tasarım kararları güncellenmiştir:

- **Veritabanı Tercihi:** Tasarım raporunda dosya sistemi (TXT) önerilmiş olsa da, verilerin ilişkisel yapısı ve sorgu performansı nedeniyle **PostgreSQL** kullanılmıştır.
- **Framework Seçimi:** Python yerine, gerçek zamanlı web uygulamaları (SPA) için endüstri standardı olan **Next.js** tercih edilmiştir.

7. Karşılaşılan Zorluklar ve Çözümler

Geliştirme sürecinde karşılaşılan başlıca zorluklar ve çözümleri şunlardır:

1. **Yanıt Gecikmesi (Latency):** LLM modellerinin yanıt üretmesi zaman almaktadır.

    a. *Çözüm:* "Streaming UI" tekniği ile yanıt tamamlanmadan ekrana yansıtılarak kullanıcı deneyimi iyileştirildi.
2. **Veritabanı Bağlantı Sınırları:** Serverless ortamda çok fazla bağlantı açılması sorunu yaşandı.
    a. *Çözüm:* Prisma Client için "Singleton Pattern" uygulanarak tek bir bağlantı havuzu oluşturuldu.

- **3. Git Çakışmaları (Merge Conflicts):** Takım çalışması sırasında aynı dosyalar üzerinde yapılan değişiklikler çakışmalara yol açtı.
- **Çözüm:** Proje lideri yönetiminde Pull Request (PR) incelemeleri yapılarak ve git merge stratejileri kullanılarak kod bütünlüğü korundu.

8. Eksikler ve Gelecek Geliştirme Önerileri

Sistem temel gereksinimleri karşılamasına rağmen şu geliştirmeler planlanmaktadır:

- **RAG (Retrieval-Augmented Generation):** Kullanıcıların kendi PDF dokümanlarını yükleyip bunlar üzerinden soru sorabilmesi.
- **Çoklu Model Desteği:** Kullanıcının GPT-4, Claude veya Gemini modelleri arasında seçim yapabilmesi.
- **Mobil Uygulama:** Sistemin React Native ile mobil platformlara taşınması.

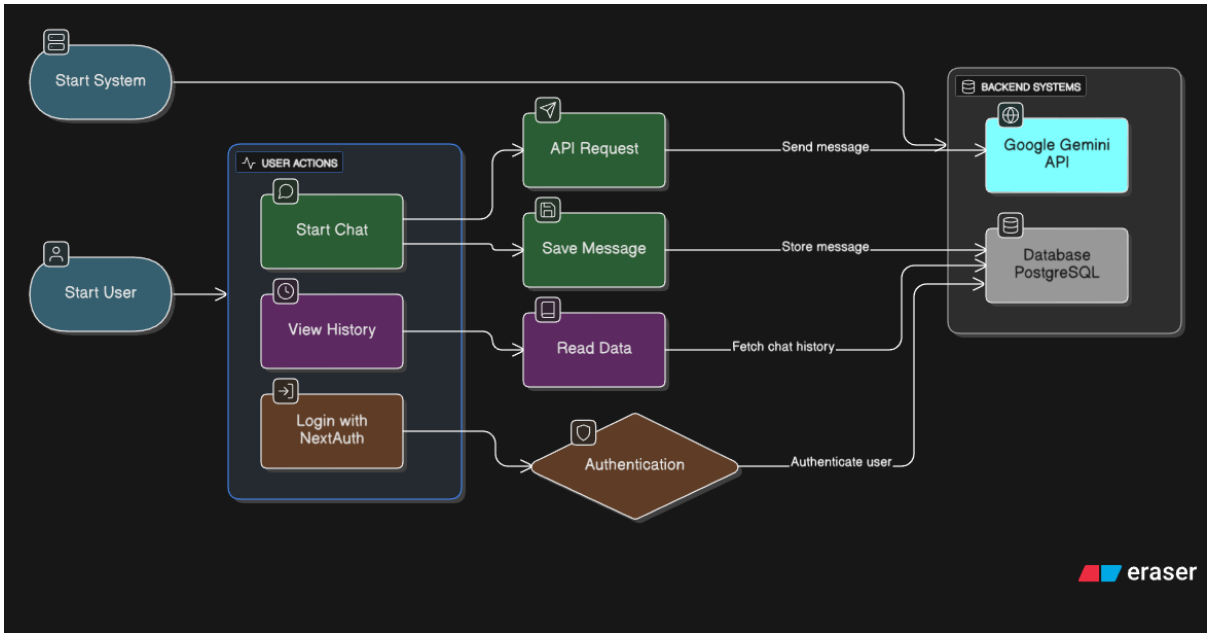
9. Sonuç

Bu çalışma kapsamında geliştirilen AI Chat Bot, yazılım mühendisliği yaşam döngüsünün analiz, tasarım ve gerçekleştirim aşamalarını eksiksiz ve tutarlı biçimde içeren bütüncül bir projedir. Sistem; nesne yönelimli prensiplere uygun, katmanlı, modüler ve genişletilebilir bir mimari

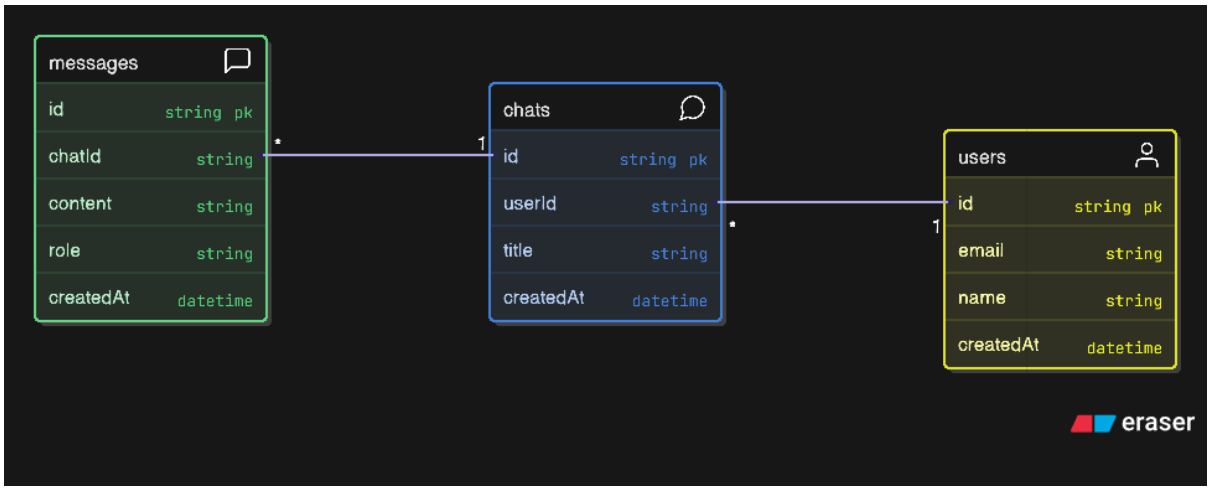
sunmaktadır. Analiz raporunda tanımlanan kullanım senaryoları başarıyla hayata geçirilmiş ve grup çalışması (Git/GitHub) ile profesyonel bir geliştirme süreci deneyimlenmiştir.

EK 2: UML Diyagramları

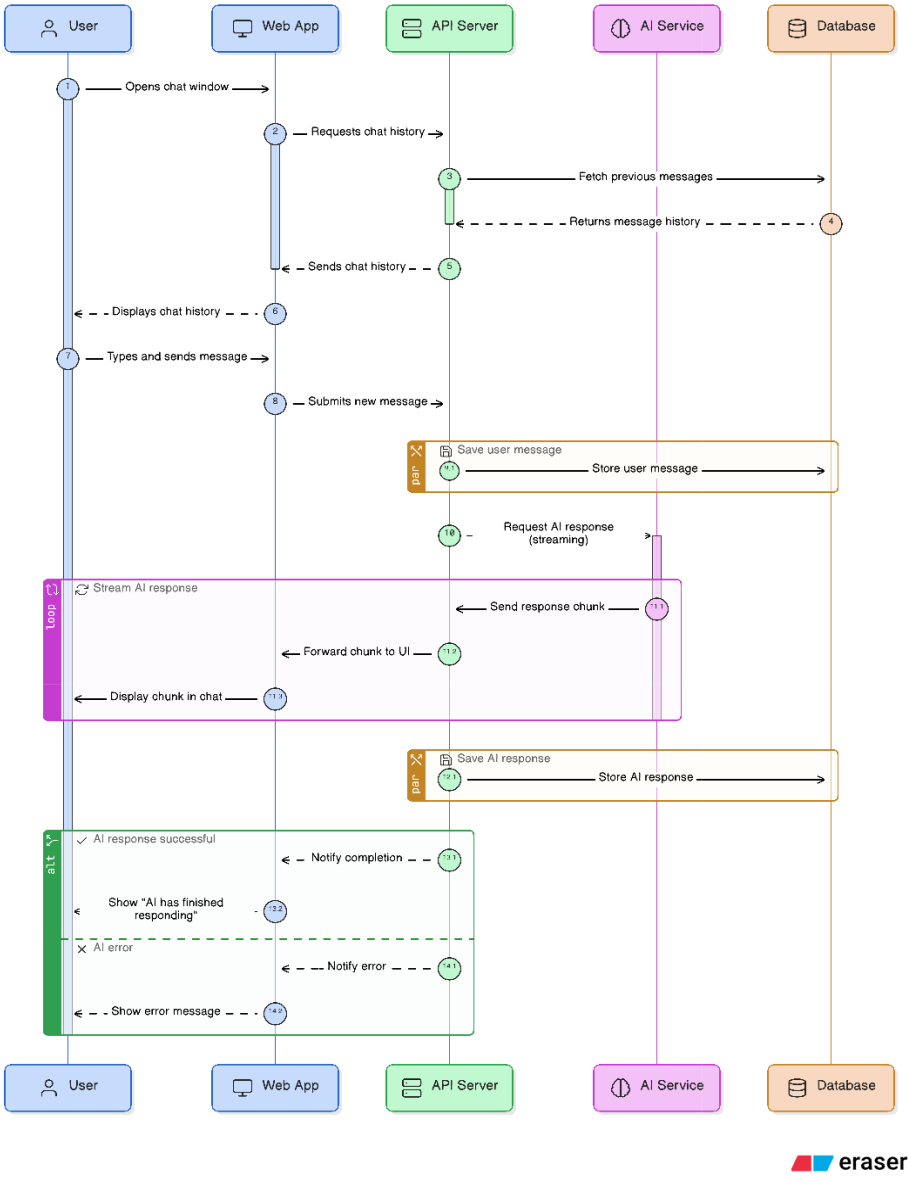
1. Use Case Diagram (diyagram 1)



2. Class Diagram (diyagram 2)



3. Sequence Diagram (diyagram 3)



EK: GİT VE GITHUB ORTAK ÇALIŞMA KILAVUZU

(Bu bölüm, proje ekibinin geliştirme sürecinde uyduğu kuralları ve teknik standartları içerir.)

1. ÇALIŞMANIN AMACI

Bu kılavuzun amacı; **"AI Chat Bot"** projesini geliştirirken Git ve GitHub teknolojilerini etkin kullanmak, kod çakışmalarını önlemek ve profesyonel bir takım çalışması deneyimi kazanmaktır.

2. PROJE KLASÖR YAPISI

Directory	Description
app/	Sayfalar (Frontend & Backend API)
components/	UI Bileşenleri (ChatBox, Sidebar vb.)
lib/	Yardımcı fonksiyonlar (Prisma, Utils)
prisma/	Veritabanı modelleri
public/	Görseller
docs	Raporlar ve UML diyagramları
.gitignore	Git dışı bırakılacak dosyalar
package.json	Bağımlılıkla