

Selçuk AI Akademik Asistan Yapay Zeka Destekli Üniversite Bilgi Asistanı

Hazırlayanlar: Ali YILDIRIM (203311008), Doğukan BALAMAN (203311066)

Danışmanlar: Prof. Dr. Nurettin DOĞAN, Dr. Öğr. Üyesi Onur İNAN

İçindekiler: Yol Haritası

01

Proje Özeti ve Problem

Selçuk AI Asistanı'nın genel çerçevesi ve çözümlemeye çalıştığı sorunlar.

02

Amaçlar ve Mevcut Çözümler

Projenin hedefleri ve piyasadaki benzer sistemlerden farkları.

03

Metodoloji ve Mimari

Geliştirme süreci, sistemin yapısal bileşenleri ve veri akışı.

04

Teknolojiler ve API Detayları

Kullanılan yazılım ve donanım araçları, API entegrasyonu.

05

AI Entegrasyonu ve Frontend

RAG pipeline'sı, LLM konfigürasyonu ve kullanıcı arayüzü özellikleri.

06

Ekran Görüntüleri ve Testler

Uygulamanın görsel örnekleri ve performans değerlendirmeleri.

07

Zorluklar, Sonuçlar ve Gelecek

Karşılaşılan sorunlar, projenin çıktıları ve ilerideki geliştirme planları.

Gizlilik Odaklı Selçuk AI Akademik Asistan

Selçuk AI Akademik Asistan, Selçuk Üniversitesi'nin bilgi erişim süreçlerini optimize etmek amacıyla tasarlanmış, yerel Büyük Dil Modeli (LLM) tabanlı, gizliliğe odaklı bir bilgi asistanıdır. Üniversiteye özel bilgilerin güvenli ve hızlı bir şekilde sunulmasını hedefler.



Yerel LLM Desteği

Ollama ile yerel çalışma, veri gizliliğini sağlar.



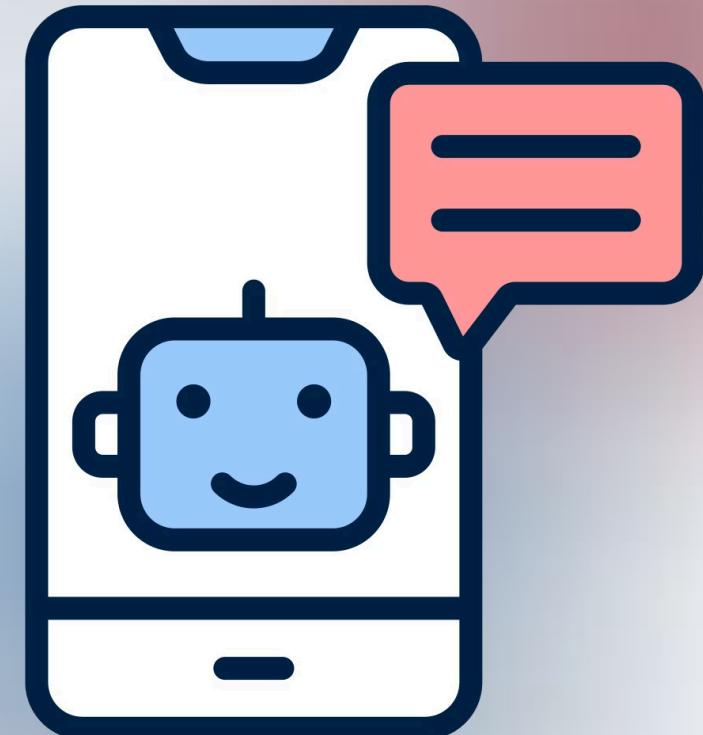
RAG ile Kaynaklı Yanıtlar

FAISS ve SentenceTransformer kullanarak güvenilir bilgi.



Çoklu Platform

Flutter istemci ve FastAPI backend ile esnek erişim.





PROBLEM TANIMI

Mevcut Bilgi Erişim Zorlukları

Üniversitelerde bilgiye erişim ve doğrulama süreçleri genellikle dağınık ve zaman alıcıdır. Bu durum, öğrenci ve personele büyük bir yük getirmekte, aynı zamanda bilgi güvenliği risklerini beraberinde getirmektedir.

Dağınık Bilgiler

Üniversite bilgilerine erişimde zorluk ve doğrulama ihtiyacı.

Destek Eksikliği

7/24 destek ve hızlı yönlendirme imkanının kısıtlı olması.

İnsan Kaynağı Yükü

Tekrarlayan sorular nedeniyle personel üzerindeki iş yükü.

Gizlilik Riskleri

Bulut tabanlı çözümlerin beraberinde getirdiği veri güvenliği endişeleri.

Selçuk AI Asistanı ile Çözüm

Ana amacımız, Selçuk Üniversitesi'ne özel, doğru ve kaynaklı bilgi sunan, yerel bir yapay zeka asistanı geliştirmektir. Bu proje, hem gizliliği hem de bilgi doğruluğunu ön planda tutar.

- **Yerel LLM ve Gizlilik**

Offline çalışma kapasitesi ile veri güvenliği.

- **RAG ve Doğruluk**

Kaynaklı yanıtlar ve katı mod ile halüsinasyon riskini en aza indirme.

- **Çoklu Dil Desteği**

Türkçe ve İngilizce dillerinde hizmet verebilme.

- **Esnek Model Seçimi**

Ollama ve HuggingFace gibi çoklu sağlayıcı/model seçenekleri.

- **Kullanıcı Dostu Arayüz**

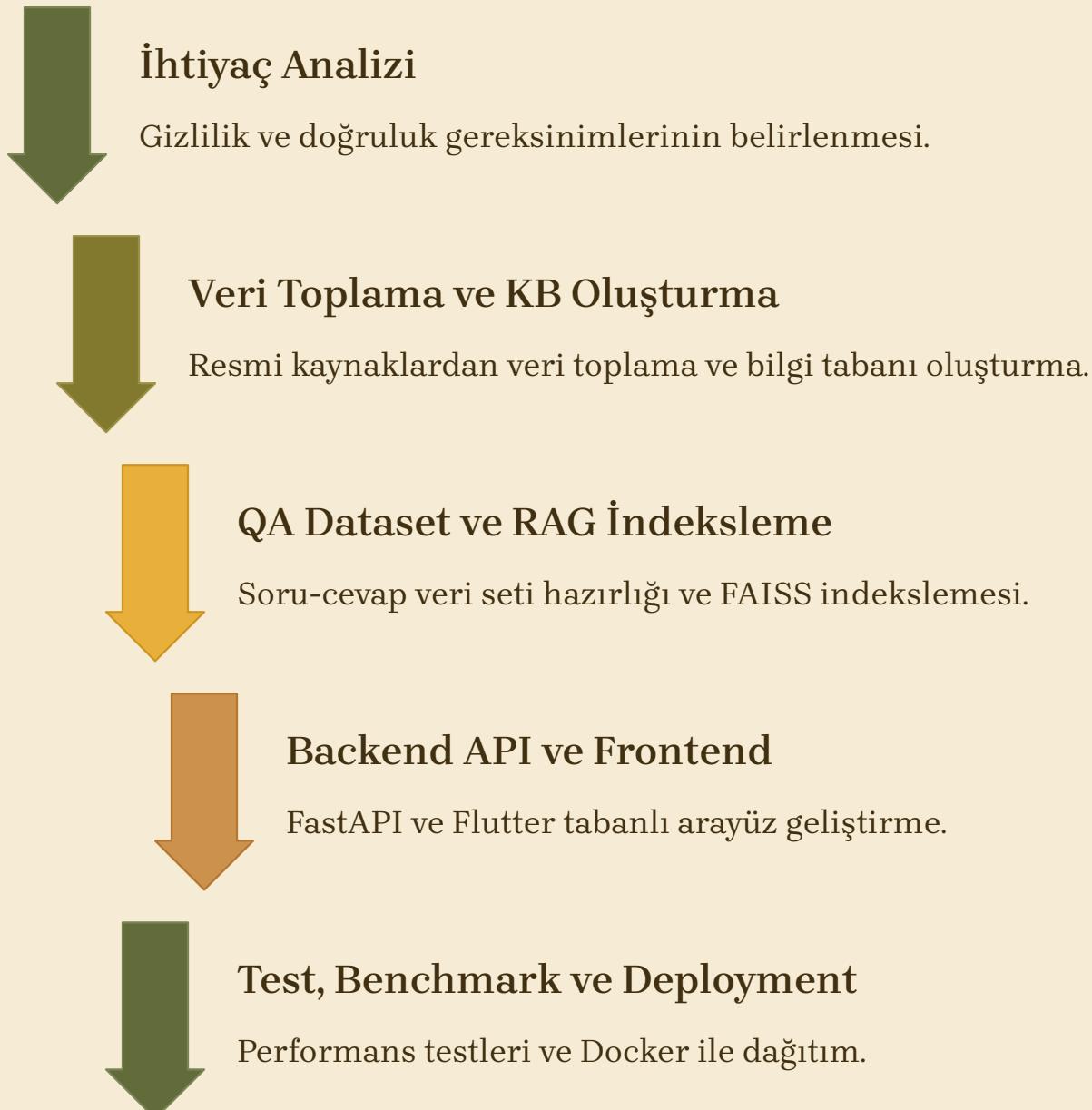
Çoklu platform desteği ile sezgisel ve erişilebilir arayüz.

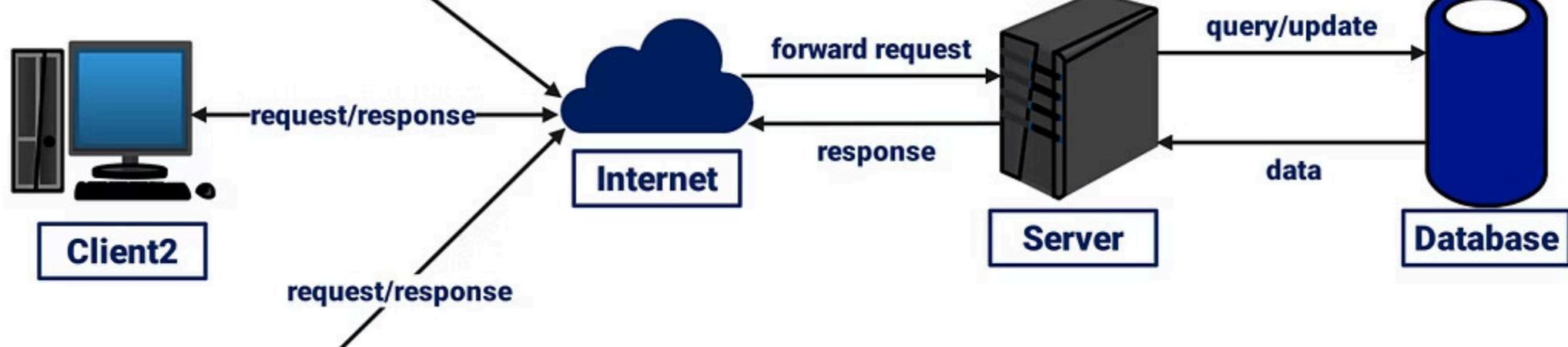
- **Gerçek Zamanlı Yanıt**

SSE streaming ile kesintisiz ve anlık bilgi akışı.

Adım Adım Geliştirme Süreci

Projemizin geliştirme süreci, baştan sona titizlikle planlanmış ve çeşitli aşamalardan geçerek tamamlanmıştır. Her adım, sistemin güvenilirliğini ve etkinliğini artırmaya yönelik tasarlanmıştır.

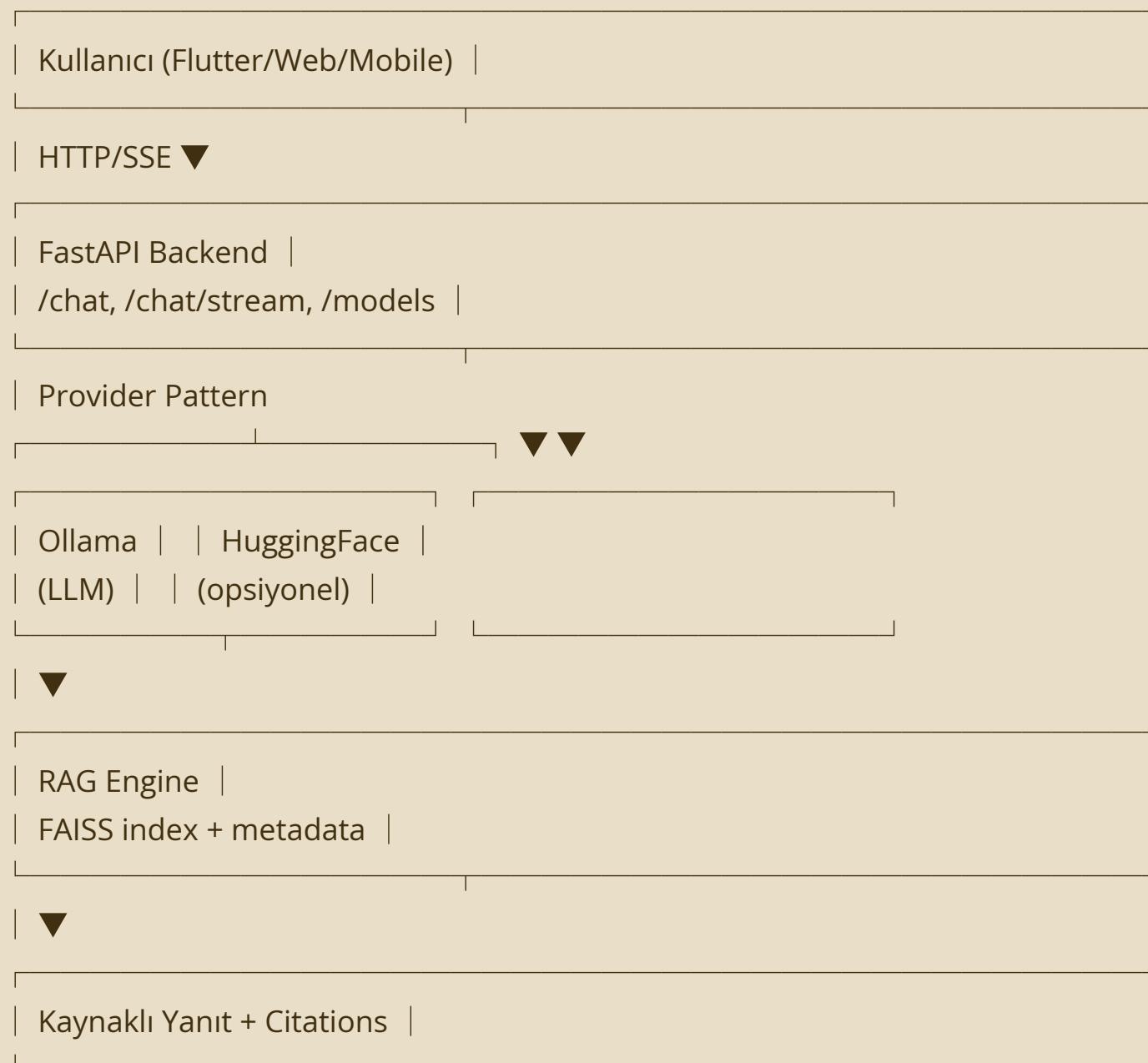




SİSTEM MİMARİSİ

Selçuk AI Asistanı'nın Yapısı

Sistemimiz, kullanıcı deneyimini ve veri güvenliğini merkeze alan modüler bir mimari ile tasarlanmıştır. Kullanıcıdan gelen talepler, FastAPI backend aracılığıyla işlenir ve RAG motoru ile yerel LLM'lere yönlendirilir.



Teknolojik Altyapı

Projemiz, modern ve güçlü teknolojilerin bir araya getirilmesiyle oluşturulmuştur. Her bir bileşen, sistemin performansını, güvenilirliğini ve ölçeklenebilirliğini artırmak için özenle seçilmiştir.

Backend & AI/ML

- Python, FastAPI:** Hızlı API geliştirme.
- Ollama, HuggingFace:** Yerel ve bulut LLM entegrasyonu.
- FAISS + SentenceTransformer:** Retrieval-Augmented Generation (RAG).

Frontend & Deployment

- Flutter (Material 3):** Çapraz platform UI geliştirme.
- GetX, Hive:** Durum yönetimi ve yerel depolama.
- Docker, Nginx:** Kolay dağıtım ve ölçeklendirme.



RAG Pipeline'ı ve Strict Mod

Selçuk AI Asistanı, doğru ve güvenilir yanıtlar sunmak için gelişmiş bir RAG (Retrieval-Augmented Generation) pipeline'ı kullanır. Bu sayede, LLM'lerin potansiyel halüsinasyon riskleri minimize edilir.

1

Kullanıcı Sorusu

Gelen sorunun metinsel analizi ve işlenmesi.

2

Embedding Oluşturma

SentenceTransformer ile sorunun vektör temsili.

3

Benzer Döküman Arama

FAISS indeksi üzerinden ilgili top_k dökümanların tespiti.

4

Prompt'a Kaynak Ekleme

Bulunan kaynakların LLM'e bağlam olarak sunulması.

5

Yanıt ve Atıf Üretimi

LLM'in kaynaklarla desteklenmiş yanıt ve atıf oluşturmaması.

Strict Mod:

Eğer kaynaklarda bilgi bulunamazsa, asistan "Bu bilgi kaynaklarda mevcut değil." şeklinde net bir ifadeyle yanıt verir.

Kullanıcı Dostu Sohbet Deneyimi

Flutter ile geliştirilen kullanıcı arayüzü, hem sezgisel hem de işlevsel bir deneyim sunar. Kullanıcılar, tanıdık bir sohbet arayüzü üzerinden hızlı ve verimli bir şekilde bilgiye erişebilirler.



ChatGPT Benzeri Arayüz

Tanıdık ve kolay kullanımlı sohbet ekranı.



Gerçek Zamanlı Yanıt

SSE streaming ile kesintisiz ve hızlı bilgi akışı.



Sesli Giriş

Speech-to-text özelliği ile kolay komut girişi.



Markdown Desteği

Mesajlarda zengin metin formatı ve okunabilirlik.



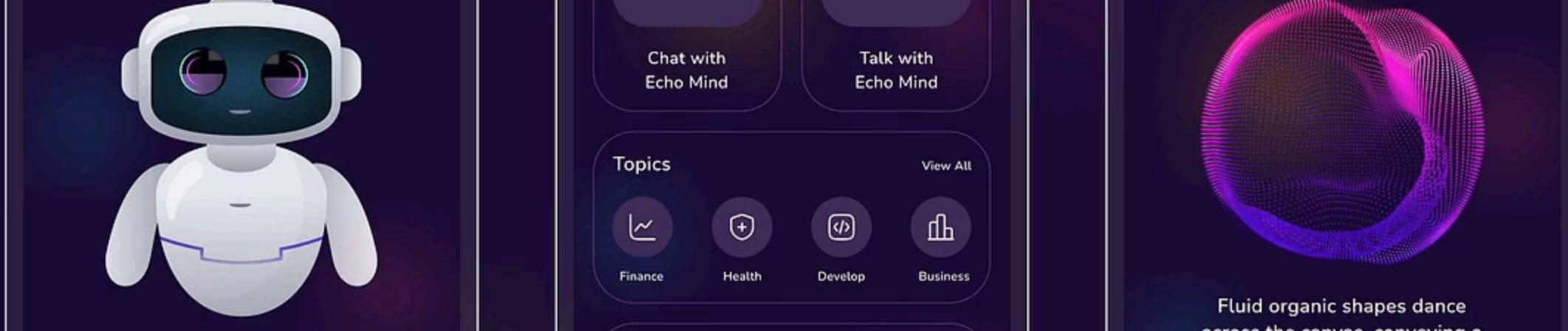
Çoklu Platform Uyumu

Mobil, web ve masaüstü cihazlarda sorunsuz deneyim.



Kişiselleştirme

Açık/koyu tema ve dil (TR/EN) seçenekleri.



EKRAN GÖRÜNTÜLERİ

Ana Sayfa: Kullanıcı Arayüzü

Selçuk AI Asistanı'nın ana ekranı, kullanıcıların bilgiye hızlı ve sezgisel bir şekilde erişmesini sağlayacak şekilde tasarlanmıştır. Her bir bileşen, kullanım kolaylığını ve verimliliği artırmak için optimize edilmiştir.

1

Sol Menü

Geçmiş sohbetlere erişim ve yeni konuşma başlatma.

2

Üst Bar

Tema değiştirme (açık/koyu) ve genel menü aksiyonları.

3

Ana Alan

Sohbet mesajları, AI yanıtları ve kaynak atıfları.

4

Alt Bar

Sesli giriş (mikrofon), mesaj yazma ve gönder/durdur butonu.

Sohbet Örneği: Doğru ve Kaynaklı Yanıt

Selçuk AI Asistanı'nın gerçek dünya kullanımındaki performansını bir sohbet örneğiyle görebilirsiniz. Asistan, doğru ve kaynaklı bilgi sunarak kullanıcıların sorularına güvenilir yanıtlar verir.

Kullanıcı:

"Selçuk Üniversitesi nerede?"

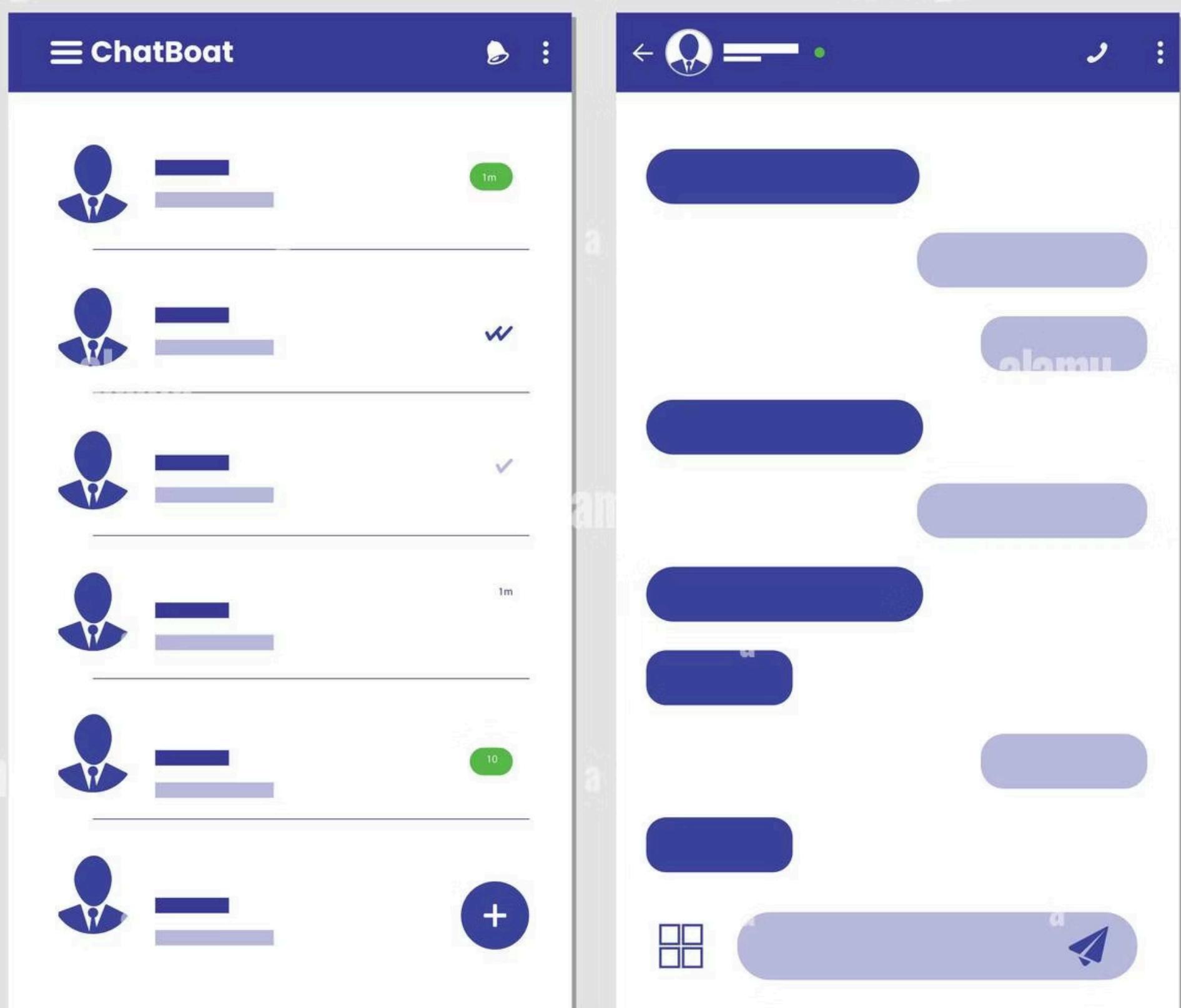
Asistan:

"Selçuk Üniversitesi Konya ilinde, Selçuklu ilçesinde yer almaktadır. Ana kampüsü Alaeddin Keykubat Kampüsü'dür.

Kaynak: selcuk_knowledge_base.json"

❑ Önemli Not:

Sistem doğru şehri (Konya) veriyor - İzmir değil! Bu, RAG mimarimizin doğruluğunu gösterir.



Selçuk AI Asistanı Neler Yanıtlayabilir?

Selçuk AI Asistanı, zengin bilgi tabanı sayesinde Selçuk Üniversitesi ile ilgili geniş bir yelpazede sorulara doğru ve hızlı yanıtlar sunar. İşte asistanımızın desteklediği başlıca soru kategorileri:

Konum ve Kampüs Bilgileri



"Selçuk Üniversitesi nerede?", "Kampüsler hangileri?" gibi sorulara yanıt verir.

Fakülte/Bölüm Bilgileri



"Kaç fakülte var?", "Bilgisayar Mühendisliği hangi fakültede?" gibi akademik yapı hakkındaki soruları cevaplar.

Akademik Takvim



Dönem başlangıç/bitiş tarihleri, sınav takvimleri ve önemli akademik süreçler hakkında bilgi sağlar.

İletişim ve Öğrenci İşleri



Önemli telefon numaraları, e-posta adresleri ve öğrenci işleri prosedürleri hakkında rehberlik eder.

Rektörlük ve Yönetim



"Rektör kim?", üniversite yönetim kadrosu ve idari yapısı hakkında güncel bilgiler sunar.

Test Senaryoları ve Doğrulama Sonuçları

Selçuk AI Asistanı'nın bilgi doğruluğunu ve yanıt kalitesini test etmek amacıyla çeşitli senaryolar kullanılmıştır. Bu testler, sistemin RAG mimarisi üzerinden ne kadar güvenilir bilgi sağladığını göstermektedir.

Soru	Beklenen Yanıt	Sonuç
"Selçuk Üniversitesi nerede?"	Konya	Doğru
"Ne zaman kuruldu?"	1975	Doğru
"Bilg. Müh. hangi fakültede?"	Teknoloji Fakültesi	Doğru
"Bilg. Müh. akredite mi?"	MÜDEK var	Doğru
"Kaç fakülte var?"	23	Doğru

- ❑ **Doğrulama Yöntemi:** validate_knowledge.py scripti ile otomatik testler yapılmıştır.

Performans Metrikleri ve Benchmark Sonuçları

Selçuk AI Asistanı'nın performansı, çeşitli metrikler üzerinden titizlikle değerlendirilmiştir. Aşağıdaki tablo, farklı modellerin yanıt hızı ve verimlilik açısından karşılaştırmalı sonuçlarını sunmaktadır.

Model	Avg TTFT	Token/s	Avg Total
llama3.2:3b	5180 ms	5.41	8.64 s
turkcell-llm-7b	10126 ms	4.10	-
selcuk_ai_assistant	10186 ms	3.49	-

TTFT: Time To First Token (İlk tokenin üretilme süresi)

- ❑ **Not:** SSE (Server-Sent Events) streaming teknolojisi sayesinde, kullanıcının algıladığı gecikme önemli ölçüde düşürülerek daha akıcı bir deneyim sağlanmıştır.

Karşılaşılan Zorluklar

Selçuk AI Asistanı geliştirilirken, projenin özgün niteliklerine ve kullanılan yapay zeka teknolojilerinin doğasına bağlı olarak bazı temel zorluklarla karşılaşılmıştır. Bu zorluklar, sistemin güvenilirliğini ve performansını doğrudan etkileyen kritik unsurlardır.



Halüsinasyon Riski

Büyük Dil Modellerinin (LLM) gerçek dışı veya uydurma bilgi üretme eğilimi, yanıt doğruluğu için ciddi bir tehdit oluşturmaktadır.



Türkçe Dil Yapısı

Projenin hedef kitlesi gereği, resmi üsluba ve dilbilgisi kurallarına uygun, kusursuz Türkçe ifade kullanımı büyük önem taşımaktadır.



Yerel LLM Performansı

Sistemin kullanıcı bekłentilerini karşılayacak hızda çalışması için, yanıt gecikmesi (latency) ve işlem sürelerinin optimize edilmesi gerekliliği.



Bilgi Tabanı Güncelligi

Üniversite bilgilerinin dinamik yapısı nedeniyle, referans alınan bilgi tabanının sürekli olarak güncel ve doğru tutulması zorunluluğu.

Uygulanan Çözümler ve İyileştirmeler

Selçuk AI Asistanı geliştirilirken karşılaşılan zorluklar, yenilikçi çözümlerle aşılmıştır. Bu yaklaşım, sistemin güvenilirliğini, doğruluğunu ve kullanıcı deneyimini önemli ölçüde artırmıştır.

Zorluk	Uygulanan Çözüm
Halüsinasyon Riski	RAG (Retrieval-Augmented Generation) mimarisi ve katı mod; kaynak yoksa "Bu bilgi kaynaklarda yok." şeklinde yanıt verme.
Türkçe Dil Desteği	Özel sistem istemleri ve kapsamlı Türkçe hata mesajlarıyla ana dil optimizasyonu.
Yerel LLM Performansı	SSE (Server-Sent Events) streaming teknolojisi ile gerçek zamanlı ve hızlı yanıt deneyimi.
Bilgi Tabanı Güncelligi	<code>validate_knowledge.py</code> scripti ile düzenli ve otomatik bilgi doğrulama.
Model Esnekliği	Provider Pattern sayesinde farklı Büyük Dil Modelleri'ne (LLM) kolayca geçiş yapabilme yeteneği.

Proje Başarıları: Güvenilir ve Yenilikçi Çözümler

Selçuk AI Asistanı projesi, geliştirme sürecinde belirlenen hedeflere ulaşarak önemli başarılar elde etmiştir. Bu başarılar, sistemin sağlamlığını, performansını ve kullanıcı odaklı tasarımını gözler önüne sermektedir.



Çalışan Yerel LLM + RAG Prototipi

Gelişmiş RAG mimarisi ile entegre edilmiş, tamamen işlevsel bir yerel Büyük Dil Modeli prototipi başarıyla hayata geçirilmiştir.



Kaynaklı ve Doğrulanabilir Yanıtlar

Asistan, tüm yanıtlarını güvenilir kaynaklara dayandırır ve kullanıcıların bilgiyi doğrulamasını sağlayan atıflar sunar.



Cross-Platform Flutter Arayüzü

Tek bir kod tabanı ile mobil, web ve masaüstü platformlarında sorunsuz çalışan, kullanıcı dostu bir arayüz geliştirilmiştir.



CI/Test Kalite Kapıları

Pytest, Ruff, Mypy ve Flutter testleri ile sürekli entegrasyon ve kalite kontrol mekanizmaları oluşturulmuştur.



Gizlilik Odaklı Mimari

Kullanıcı verilerinin yerel kalmasını sağlayan mimari tasarım ile üst düzey veri gizliliği ve güvenliği sunulmaktadır.



Strict Mod ile Halüsinsasyon Önleme

Sıkı yanıt verme modları sayesinde, LLM'nin gerçek dışı veya uydurma bilgi üretme riski minimize edilmiştir.



SSE Streaming ile Gerçek Zamanlı Deneyim

Server-Sent Events teknolojisi sayesinde kullanıcılar, yanıtları gerçek zamanlı olarak alarak akıcı bir etkileşim yaşırlar.

Selçuk AI Asistanı: SWOT Analizi

Selçuk AI Asistanı projesinin içsel ve dışsal faktörlerini değerlendirmek amacıyla yapılan SWOT analizi, güçlü yönlerimizi, zayıf yönlerimizi, karşımızdaki fırsatları ve potansiyel tehditleri ortaya koymaktadır. Bu analiz, projenin stratejik yönünü belirlemede kritik bir rol oynamaktadır.



Yol Haritası: Gelecek Çalışmalar

Selçuk AI Asistanı'nın gelecekteki gelişimini ve stratejik hedeflerini belirleyen yol haritası aşağıda sunulmuştur. Bu adımlar, projenin sürdürülebilirliğini ve kapsamını genişletmeyi amaçlamaktadır.



Kısa Vadeli (1-3 Ay)

- RAG kaynaklarının genişletilmesi
- Bilgi Tabanı doğrulama ve güncelleme otomasyonu



Orta Vadeli (3-6 Ay)

- LoRA/QLoRA ile model ince ayarı ve optimizasyonu
- Appwrite entegrasyonunun genişletilmesi ve API geliştirmeleri



Uzun Vadeli (6-12 Ay)

- Resmi üniversite sistemleriyle (ÖBS, Döküman Yönetim Sistemi) entegrasyon
- Çoklu dil desteği ve kişiselleştirme özellikleri

Canlı Demo

Bu bölümde, Selçuk AI Asistanı'nın işlevsellliğini ve kullanıcı deneyimini sergilemek amacıyla canlı bir gösterim sunulacaktır. Test ortamı ve örnek sorular aşağıda detaylandırılmıştır.

Test Ortamı

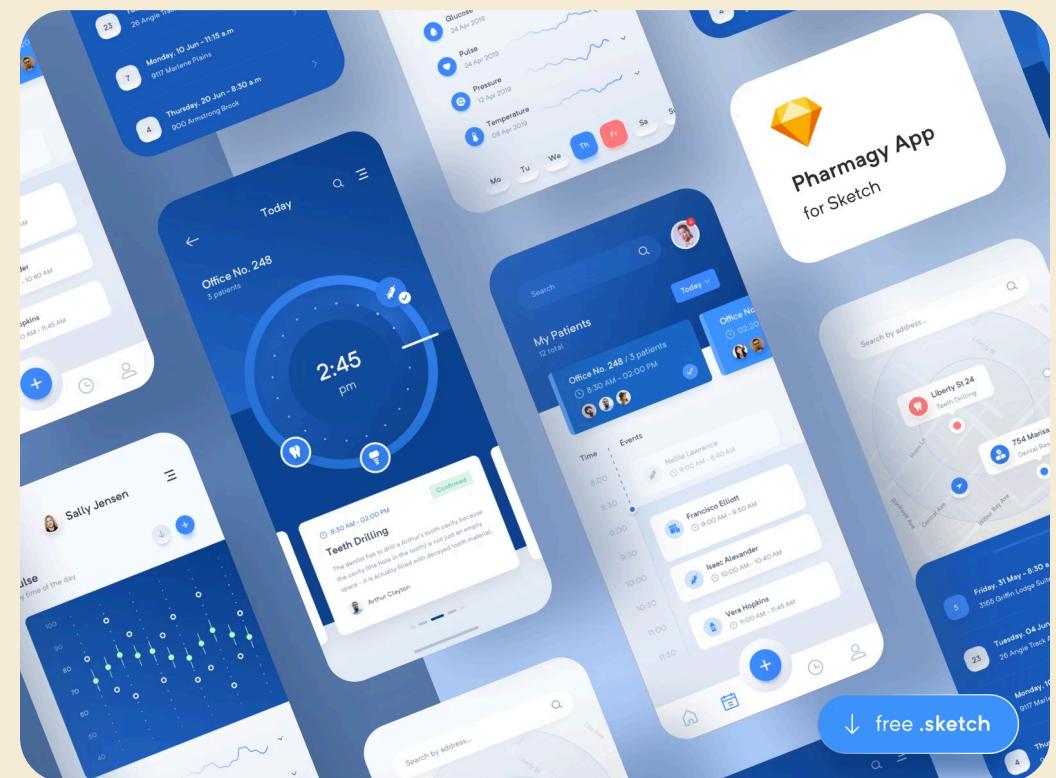
Prototip, aşağıdaki yerel ortamda yapılandırılmıştır:

- Backend:** <http://localhost:8000>
- Frontend:** Yerel Flutter uygulaması ile mobil, web ve masaüstü platformlarında erişilebilir.

Demo Soruları

Aşağıdaki örnek sorular ile asistanın bilgi tabanını test edebilirsiniz:

- "Selçuk Üniversitesi nerede?"
- "Kampüsler hangileri?"
- "Kaç fakülte var?"
- "Rektör kim?"
- "Bilgisayar Mühendisliği hangi fakültede?"



Uygulamayı denemek için yandaki QR kodu tarayabilir veya demo ekran görüntüsünü inceleyebilirsiniz.

Teşekkürler!

Sorularınız?

Hazırlayanlar

Ali YILDIRIM (203311008)

Doğukan BALAMAN (203311066)

GitHub: github.com/esN2k/SelcukAiAssistant

T.C. Selçuk Üniversitesi

Teknoloji Fakültesi | Bilgisayar Mühendisliği

Ocak 2025, Konya

Danışmanlar

Prof. Dr. Nurettin DOĞAN

Dr. Öğr. Üyesi Onur İNAN

Referanslar ve Kullanılan Kaynaklar

Bu projenin geliştirilmesinde, en güncel teknolojilerden ve akademik çalışmalarдан faydalanyılmıştır. Aşağıda, sistemin temelini oluşturan önemli kaynaklar ve dokümantasyonlar listelenmiştir.

- [Selçuk AI Asistanı README ve Dokümantasyonları](#) (README.md, docs/*.md)
- [Selçuk Üniversitesi Resmi Web Sitesi](#) - selcuk.edu.tr
- [FastAPI Dokümantasyon](#) - fastapi.tiangolo.com
- [Flutter Dokümantasyon](#) - docs.flutter.dev
- [Ollama Dokümantasyon](#) - ollama.com/docs
- [FAISS](#) - github.com/facebookresearch/faiss
- [Sentence-Transformers](#) - sbert.net
- Lewis et al., 2020 - *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*