



SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

T.C.
**SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ**



SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Selçuk AI Akademik Asistan

203311008 Ali Yıldırım

203311066 Doğukan Balaman

Bilgisayar Mühendisliği Uygulamaları

Prof. Dr. NURETTİN DOĞAN

Dr. Öğr. Üyesi ONUR İNAN

Konya

İÇİNDEKİLER

2/9

- 1. Proje Özeti
- 2. Problem Tanımı
- 3. Amaç ve Hedefler
- 4. Literatür/Mevcut Çözümler
- 5. Metodoloji
- 6. Sistem Mimarisi
- 7. Kullanılan Teknolojiler
- 8. Uygulama Detayları
- 9. Ekran Görüntüleri/Demo
- 10. Test ve Değerlendirme
- 11. Sonuçlar
- 12. Gelecek Çalışmalar
- 13. Kaynaklar

PROJE ÖZETİ

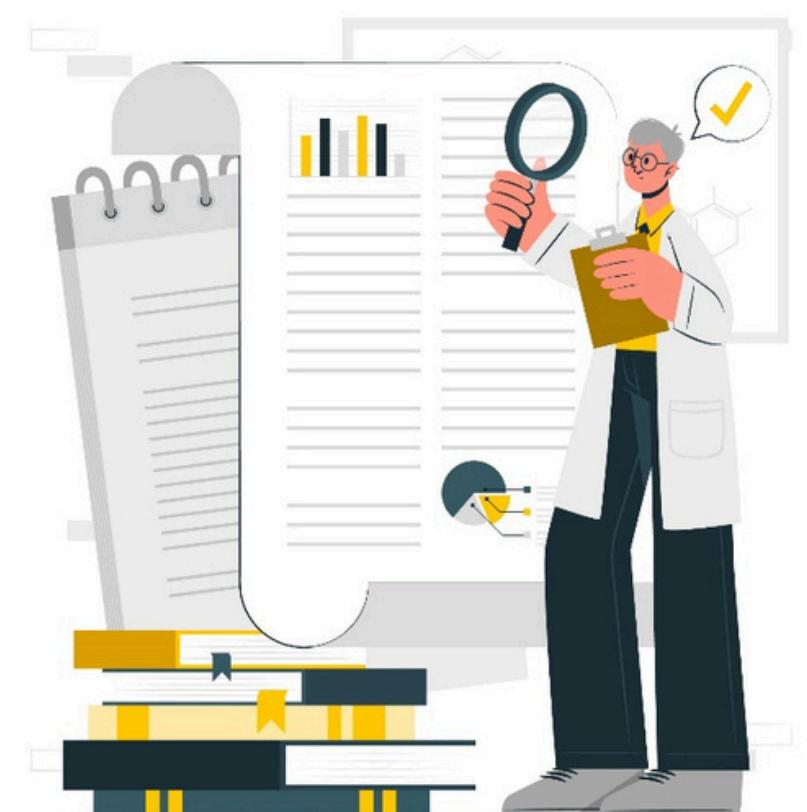
3/9

Selçuk AI Akademik Asistan, Selçuk Üniversitesi için gizliliğe odaklı, yerel LLM tabanlı bir bilgi asistanıdır. Flutter istemci ve FastAPI backend ile çoklu platformda çalışır; /chat ve /chat/stream üzerinden yanıt üretir. RAG (FAISS + SentenceTransformer) ile kaynaklı yanıtlar ve strict mod sunar.

Ollama ana sağlayıcıdır; HuggingFace opsiyonel olarak desteklenir.

Görsel Önerisi: Mimariyi özetleyen tek satırlık akış diyagramı.

Konuşmacı Notu (özet): Yerel çalışma, RAG ve çoklu sağlayıcı temaları öne çıkarılır.



PROBLEM TANIMI

4/9

Mevcut Durum:

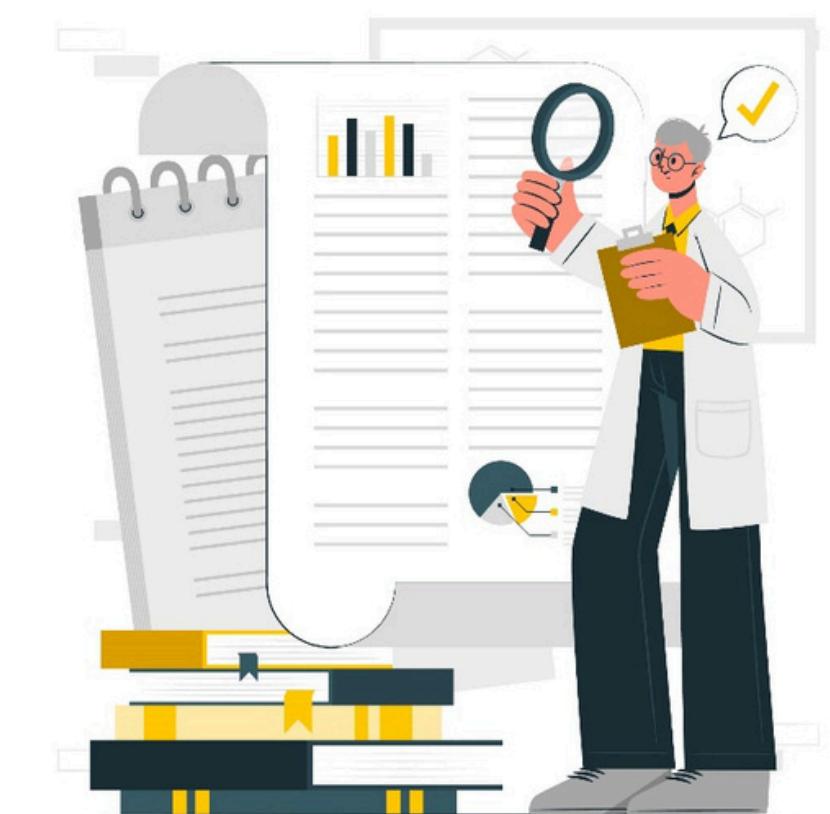
- Üniversite bilgileri dağınık; erişim ve doğrulama gücü
- 7/24 destek ve hızlı yönlendirme eksikliği
- Tekrarlayan sorular için insan kaynağı yükü
- Bulut tabanlı çözümlerde gizlilik riski

Etkilenen Kitle:

- 100.000+ Selçuk Üniversitesi öğrencisi (KB verisi)
- Akademik ve idari personel
- Aday öğrenciler ve ziyaretçiler

Görsel Önerisi: “Sorun–Etkilenen Kitle” ikonlu infografik.

Konuşmacı Notu (özet): Gizlilik ve doğruluk ihtiyacı vurgulanır.



AMAÇ VE HEDEFLER

5/9

Ana Amaç:

Selçuk Üniversitesi'ne özel, doğru ve kaynaklı bilgi veren yerel AI asistanı geliştirmek.

Hedefler:

- Yerel LLM ile gizlilik ve offline çalışma
- RAG ile kaynaklı yanıt ve strict mod
- TR/EN dil desteği ve tutarlı sistem istemi
- Çoklu sağlayıcı/model seçimi (Ollama + opsiyonel HF)
- Kullanıcı dostu, çoklu platform arayüz ve SSE akışı

Görsel Önerisi: Hedefleri simgeleyen beş madde kartı.

Konuşmacı Notu (özet): Hedefler doğrudan repo dokümantasyonundan türetilmiştir.



LİTERATÜR / MEVCUT ÇÖZÜMLER

6/9

Benzer Çözümler:

- Üniversite chatbot'ları ve SSS portalları
- Bulut tabanlı LLM destekli yardım masaları
- RAG tabanlı bilgi asistanları (Lewis et al., 2020)

Bu Projenin Farkı:

- Selçuk Üniversitesi'ne özel doğrulanmış bilgi tabanı
- Yerel LLM (Ollama) ile gizlilik ve bağımsızlık
- RAG + citations + strict mod ile hatalı bilgi riski azaltma
- Provider Pattern ile model esnekliği

Görsel Önerisi: Karşılaştırma tablosu (klasik chatbot vs RAG/yerel LLM).

Konuşmacı Notu (özet): Farklılaştırıcılar net ve kısa şekilde aktarılır.

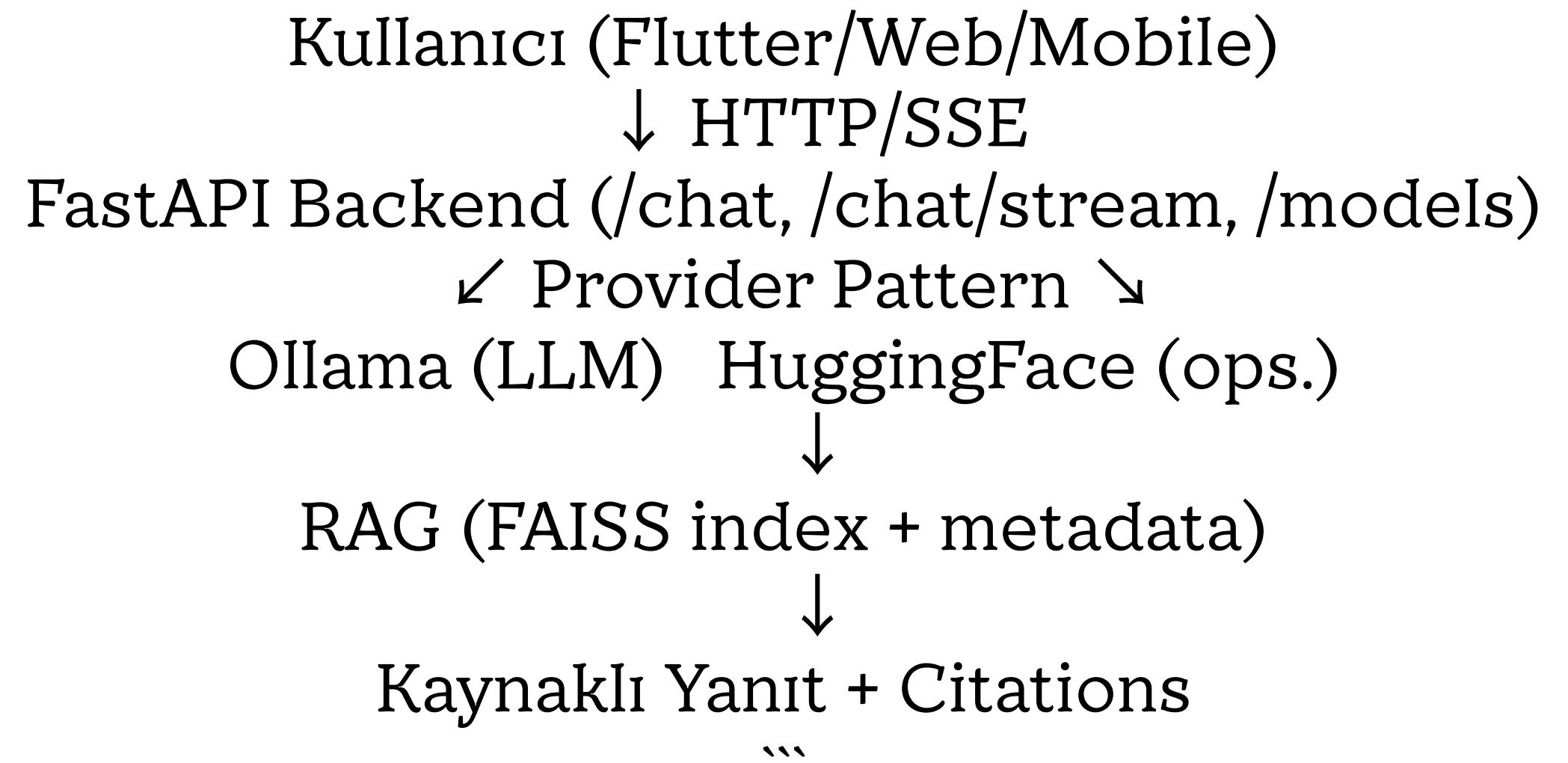


- Geliştirme Adımları (repo akışı):
- 1. İhtiyaç ve gizlilik analizi
- 2. Veri toplama: resmi kaynaklar (tools/collect_sources.py)
- 3. Bilgi tabanı oluşturma: selcuk_knowledge_base.json
- 4. QA veri seti: selcuk_qa_dataset.jsonl + doğrulama (validate_knowledge.py)
- 5. RAG indeksleme: rag_ingest.py (FAISS)
- 6. Backend API: FastAPI + Provider Pattern
- 7. Frontend: Flutter + GetX + Hive
- 8. Test/Benchmark: pytest, ruff, mypy, benchmark/run.py
- 9. Dağıtım: Local, Docker, Docker Compose, Nginx



SİSTEM MİMARİSİ

8/9



Görsel Önerisi: Katmanlı mimari diyagramı.

Konuşmacı Notu (özet): SSE streaming ve RAG strict akışı anlatılır.

KULLANILAN TEKNOLOJİLER

Backend:

- Python, FastAPI, Pydantic, Uvicorn
- requests/httpx, python-dotenv

AI/ML:

- Ollama (yerel LLM, varsayılan model seçilebilir)
- HuggingFace Transformers (opsiyonel)
- RAG: FAISS + SentenceTransformer embeddings

Frontend:

- Flutter (Material 3) + GetX
- Hive (local storage), flutter_secure_storage
- SSE istemcisi, markdown desteği

Veri/Depolama:

- JSON/JSONL bilgi tabanı
- FAISS index + metadata.json
- Opsiyonel: Appwrite (auth/log)

Deployment:

- Local run, Docker, Docker Compose, Nginx
- Görsel Önerisi: Teknoloji logoları ile katmanlı kolaj.

Konuşmacı Notu (özet): Stack, requirements ve pubspec üzerinden doğrulanmıştır.

VERİ TABANI YAPISI

8/9

Knowledge Base İçeriği:

- Üniversite genel bilgileri (Konya, 1975, kampüsler)
- Fakülte/bölüm bilgileri
- Akademik takvim özetleri
- İletişim bilgileri ve SSS

Veri Formatı:

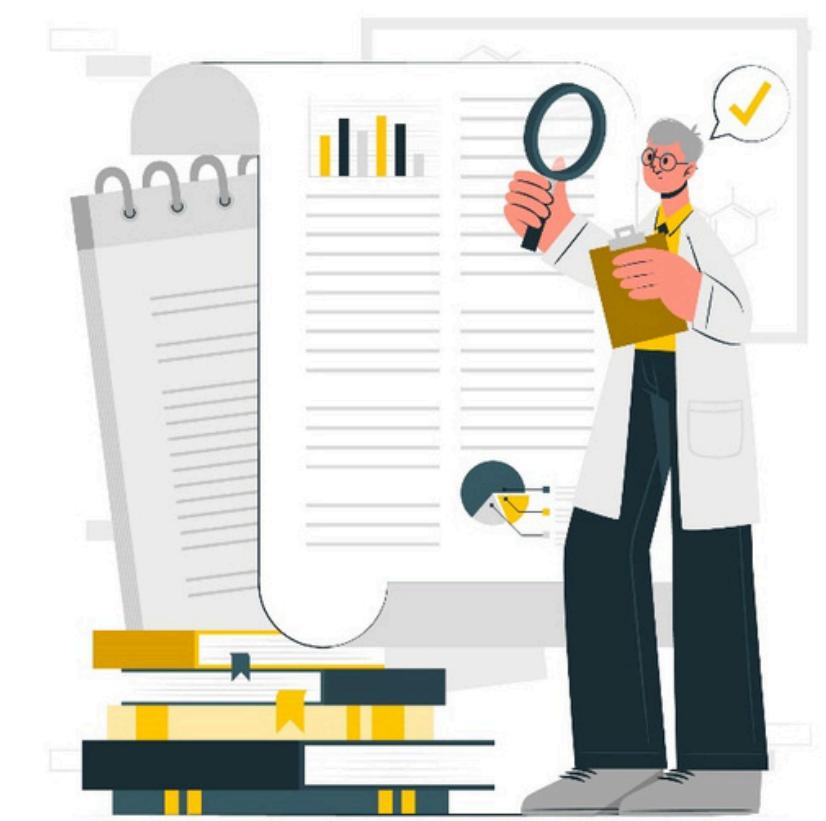
- `backend/data/selcuk_knowledge_base.json`
- `backend/data/selcuk_qa_dataset.jsonl`
- RAG: `backend/data/rag/index.faiss` + `metadata.json`

Uygulama Verisi:

- Flutter: Hive ile konuşma geçmişi
- Opsiyonel: Appwrite ile oturum ve sohbet logları

Görsel Önerisi: Dosya ağacı görseli (data/ dizini).

Konuşmacı Notu (özet): Dosya isimleri doğrudan repo veri dizinlerinden alınmıştır.



UYGULAMA DETAYLARI - Backend

8/9

API Endpoints:

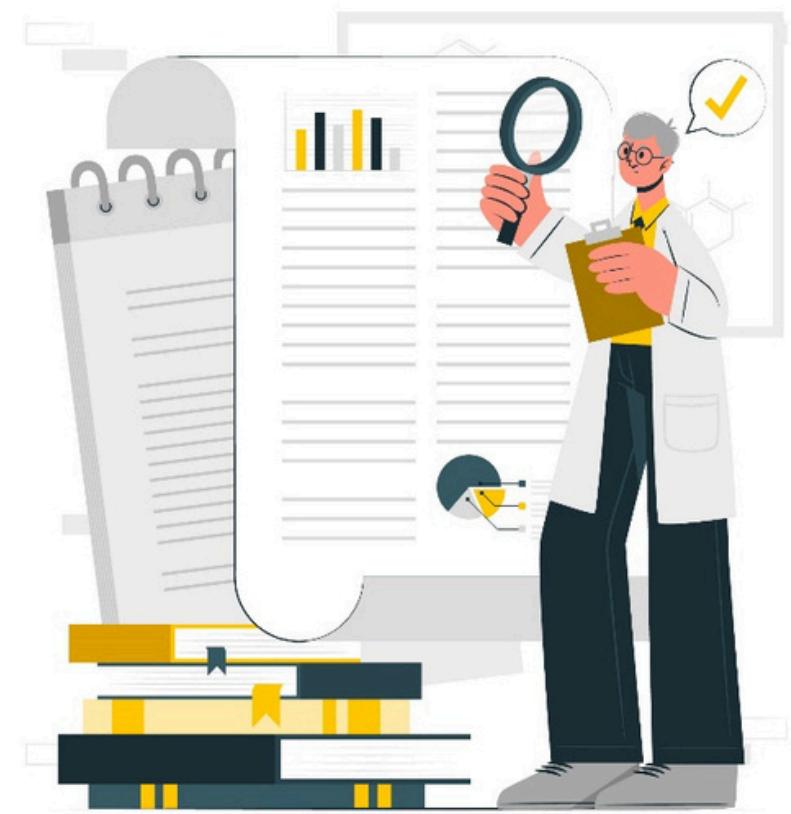
- GET /health, /health/ollama, /health/hf
- GET /models
- POST /chat
- POST /chat/stream (SSE)

Core Functions:

- Model routing: `providers/registry.py`
- RAG context: `rag_service.py`
- Prompt ve strict mode: `prompts.py`
- Streaming response temizleme
- Opsiyonel Appwrite loglama

Görsel Önerisi: API endpoint listesi + küçük akış şeması.

Konuşmacı Notu (özet): Endpoint'lerin amacı ve SSE farkı belirtilir.



UYGULAMA DETAYLARI - AI ENTEGRASYON

8/9

LLM Konfigürasyonu (istemci varsayılanı):

- temperature: 0.2, top_p: 0.9, max_tokens: 256
- Model sağlayıcı: Ollama (varsayılan), HF opsiyonel

System Prompt (prompts.py):

“Selçuk Üniversitesi'nin resmi AI asistanısın... Konya (İzmir değil), 1975...”

RAG Pipeline:

1. Soru alınır
2. Embedding'e çevrilir (SentenceTransformer)
3. FAISS top_k (varsayılan 4) araması
4. Kaynaklar prompta eklenir (strict mod destekli)
5. Yanıt + citations üretilir

Görsel Önerisi: RAG akış diyagramı.

Konuşmacı Notu (özet): Konya vurgusu ve strict mod mesajı özellikle belirtilir.



UYGULAMA DETAYLARI - Frontend

8/9

Arayüz Özellikleri:

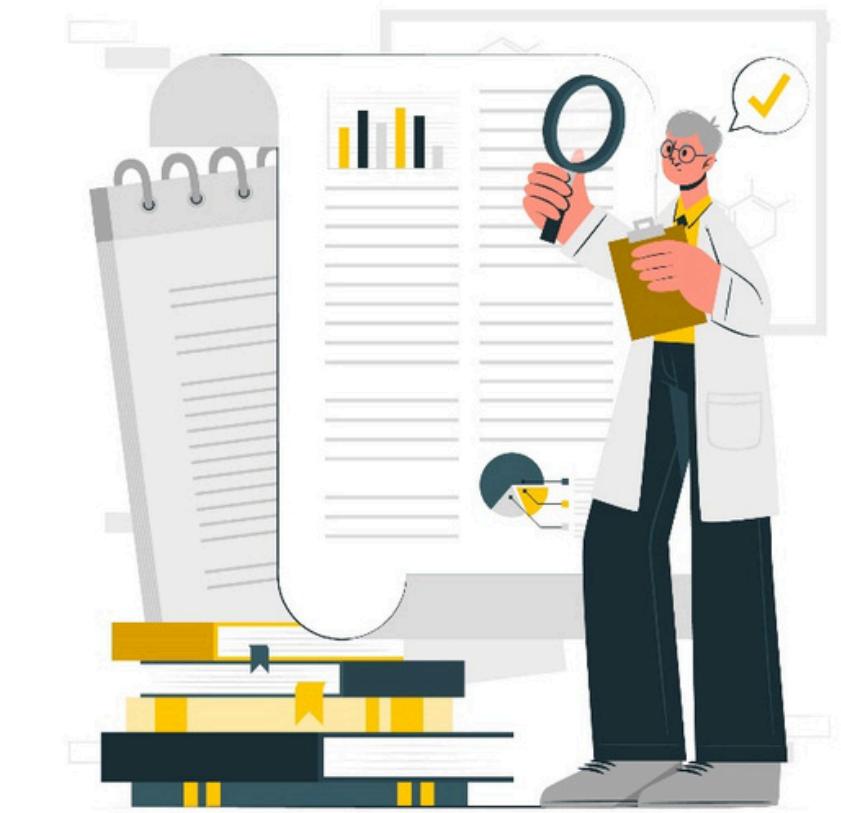
- ChatGPT benzeri sohbet arayüzü
- Streaming SSE ve Markdown destekli mesajlar
- Sesli giriş (speech_to_text)
- Model seçici ve tanılama ekranı
- RAG ve strict mod ayarları
- Sohbet geçmişi, düzenleme, yeniden üretme, dışa aktarma

Kullanıcı Deneyimi:

- TR/EN dil desteği
- Açık/Koyu tema
- Mobil, web ve desktop uyumluluğu

Görsel Önerisi: Chat ekranı + ayarlar ekranı yan yana.

Konuşmacı Notu (özet): Temel UI özellikleri FEATURES.md ve Flutter kodu ile uyumlu.



EKRAN GÖRÜNTÜLERİ - ANA SAYFA

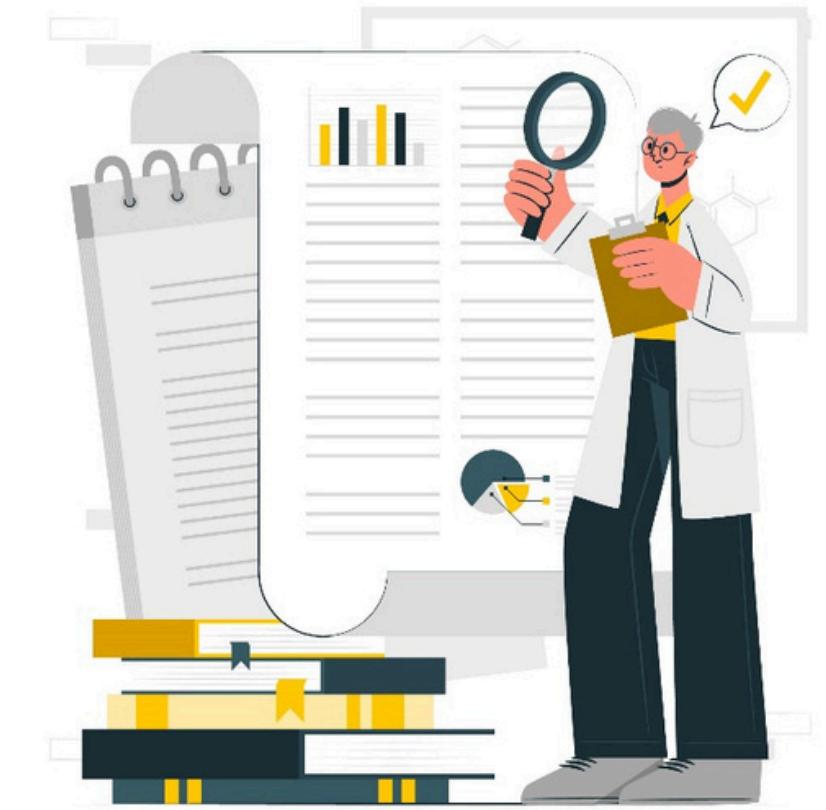
8/9

Önerilen Görseller:

- Yeni sohbet ekranı (NewChatScreen)
- Sol menü: konuşma listesi
- Üst bar: tema ve menü aksiyonları
- Alt bar: mikrofon + mesaj kutusu + gönder/durdur

Görsel Önerisi: Gerçek uygulamadan ekran görüntüsü.

Konuşmacı Notu (özet): Kullanıcı akışı ilk bakışta anlaşılır şekilde gösterilir.



EKRAN GÖRÜNTÜLERİ - SOHBET ÖRNEĞİ

8/9

Örnek Sohbet (KB verisi):

Kullanıcı: "Selçuk Üniversitesi nerede?"

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

Görsel Önerisi: Sohbet balonları + kaynaklar (citations) alanı.

Konuşmacı Notu (özet): Doğru şehir vurgusu (Konya, İzmir değil) öne çıkarılır.



EKRAN GÖRÜNTÜLERİ - FARKLI SORU TIPLERİ

8/9

Desteklenen Soru Kategorileri (KB örnekleri):

- Konum ve kampüs bilgileri
- Fakülte/bölüm bilgileri
- Akademik takvim özetleri
- İletişim ve öğrenci işleri
- Rektörlük ve yönetim bilgileri

Görsel Önerisi: 3-4 farklı soru örneği grid görünümü.

Konuşmacı Notu (özet): Soru çeşitliliği bilgi tabanı ile örtüşür.



TEST SENARYOLARI

8/9

Test Senaryoları (tanımlı script/rapor):

| Soru | Beklenen | Sonuç |

| --- | --- | --- |

| “Selçuk Üniversitesi nerede?” | Konya | Doğrulama kuralı mevcut
(validate_knowledge.py) |

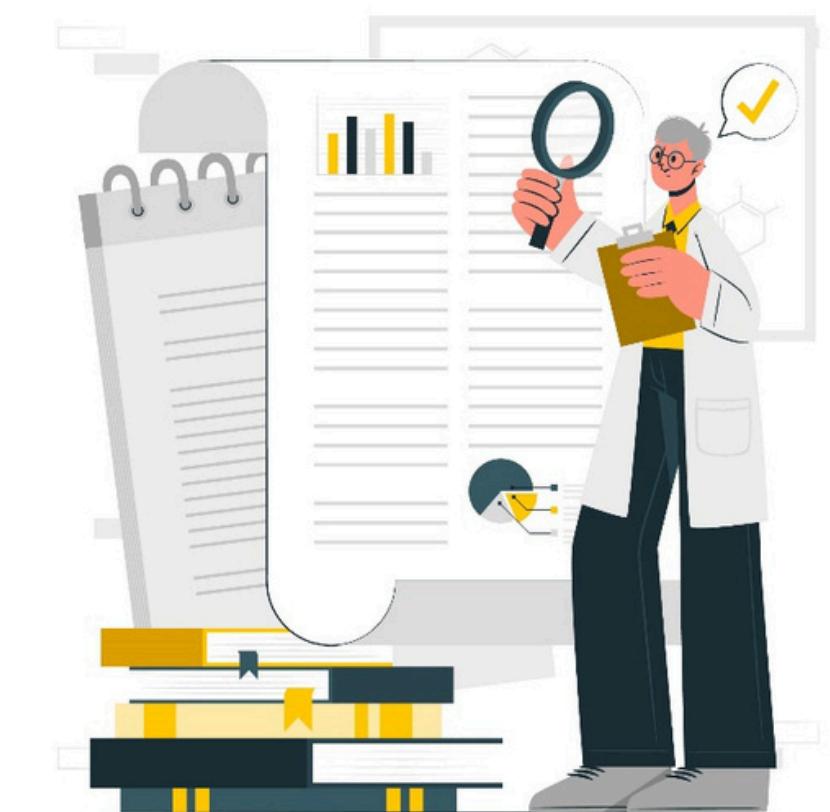
| “Ne zaman kuruldu?” | 1975 | Doğrulama kuralı mevcut
(validate_knowledge.py) |

| “Bilg. Müh. hangi fakültede?” | Teknoloji Fakültesi | Doğrulama
kuralı mevcut (validate_knowledge.py) |

| “Bilg. Müh. akredite mi?” | MÜDEK var | Doğrulama kuralı mevcut
(validate_knowledge.py) |

| “Kaç fakülte var?” | 23 | KB verisi (selcuk_knowledge_base.json) |
Görsel Önerisi: Test tablosu + kısa log alıntısı.

Konuşmacı Notu (özet): Test raporu ve doğrulama scriptleri
referans gösterilir.



PERFORMANS METRİKLERİ

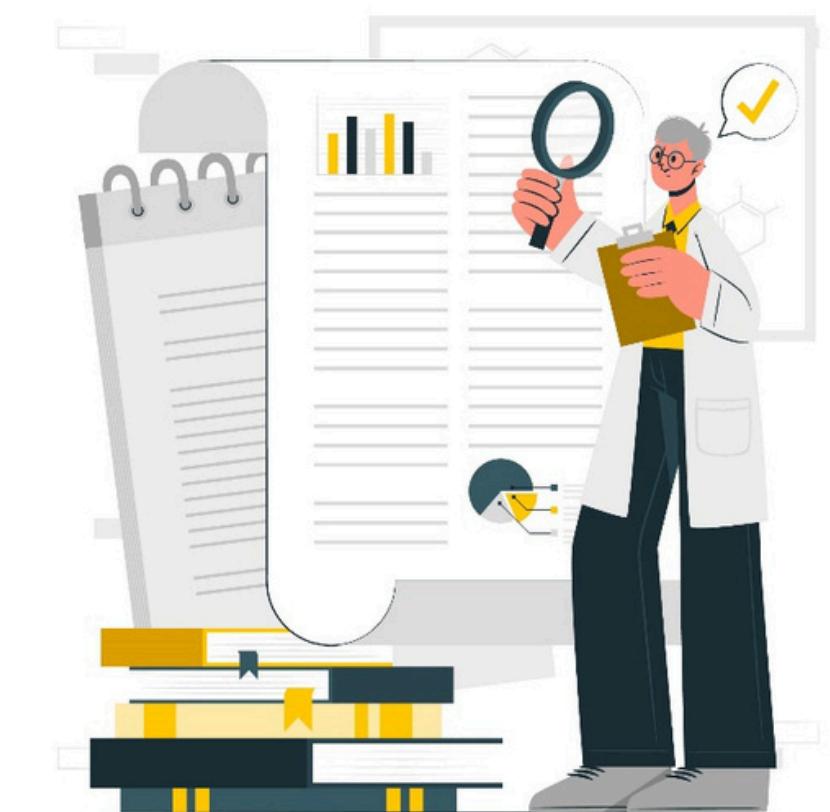
8/9

Benchmark Özeti (docs/BENCHMARK_RAPORU.md):

- Ilama3.2:3b (12 örnek): Avg TTFT 5180 ms, 5.41 tok/s, Avg total 8.643 s
- turkcell-Ilm-7b (6 örnek): Avg TTFT 10126 ms, 4.10 tok/s
- selcuk_ai_assistant (6 örnek): Avg TTFT 10186 ms, 3.49 tok/s
- SSE streaming ile algılanan gecikme düşürülür

Görsel Önerisi: Bar chart (TTFT ve tok/s).

Konuşmacı Notu (özet): Performans verileri benchmark raporundan alınmıştır.



KARŞILAŞILAN ZORLUKLAR

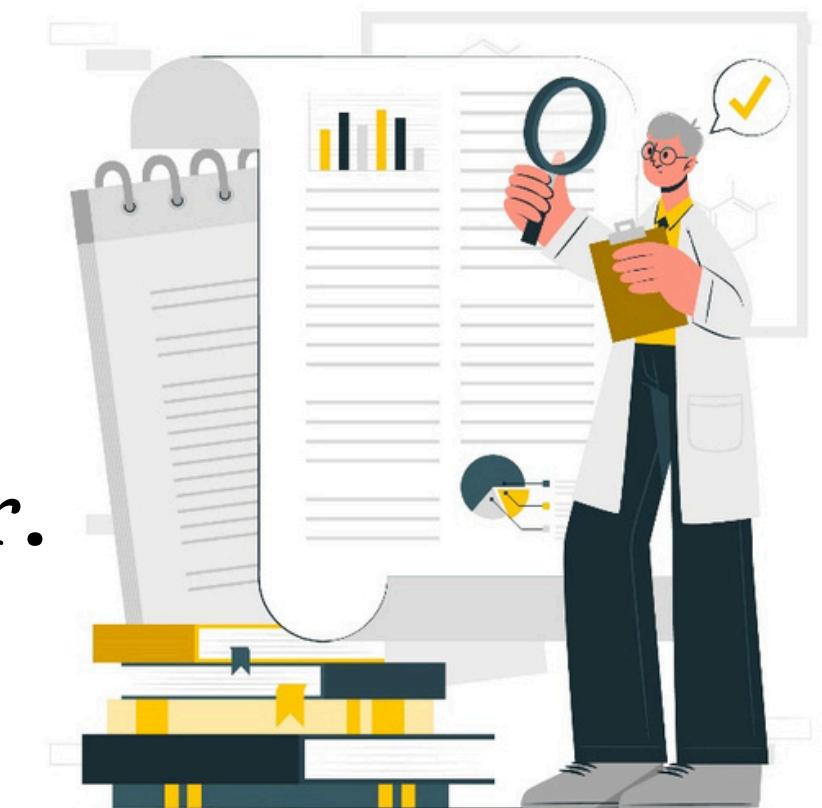
8/9

Zorluklar:

- Hallucination riski ve yanlış bilgi üretimi
- Türkçe dil yapısı ve resmi üslup ihtiyacı
- Yerel LLM performans/latency yönetimi
- Bilgi tabanının güncel tutulması

Görsel Önerisi: “Risk–Çözüm” eşleştirme görseli.

Konuşmacı Notu (özet): Her zorluğun repo'daki karşılığı belirtilir.



ÇÖZÜMLER VE İYİLEŞTİRMELER

8/9

Uygulanan Çözümler:

- RAG + strict mod: kaynak yoksa “Bu bilgi kaynaklarda yok.”
- Doğrulama scripti: validate_knowledge.py
- SSE streaming ile kullanıcı deneyimi
- Provider Pattern ile model esnekliği
- Türkçe hata mesajları ve input doğrulama

Görsel Önerisi: Çözüm ikonları ve kısa örnek yanıt.

Konuşmacı Notu (özet): Çözümler doğrudan kod ve dokümantasyona dayanır.



SONUÇLAR

8/9

Başarılar:

- Çalışan yerel LLM + RAG prototipi
- Kaynaklı ve doğrulanabilir yanıtlar
- Cross-platform Flutter arayüzü
- CI/test kalite kapıları (pytest, ruff, mypy, flutter)

Görsel Önerisi: Başarı checklist'i.

Konuşmacı Notu (özet): Test/CI vurgusu akademik güvenilirliği artırır.



SWOT ANALİZİ

8/9

Güçlü Yönler:

- Gizlilik odaklı yerel LLM
- RAG ile kaynaklı yanıtlar
- Çoklu sağlayıcı/modele açık mimari

Zayıf Yönler:

- Yerel donanım gereksinimi
- Veri güncelleme süreçlerinin maliyeti

Fırsatlar:

- Resmi üniversite entegrasyonları
- LoRA ile alan özelleştirme

Tehditler:

- Model sürüm değişimleri
- Güncel veri ihtiyacı

Görsel Önerisi: 2x2 SWOT matrisi.

Konuşmacı Notu (özet): Güçlü/zayıf taraflar repo gerçeklerine dayanır.

GELECEK ÇALIŞMALAR

8/9

Kısa Vadeli:

- RAG kaynaklarının genişletilmesi
- KB doğrulama ve güncelleme otomasyonu

Orta Vadeli:

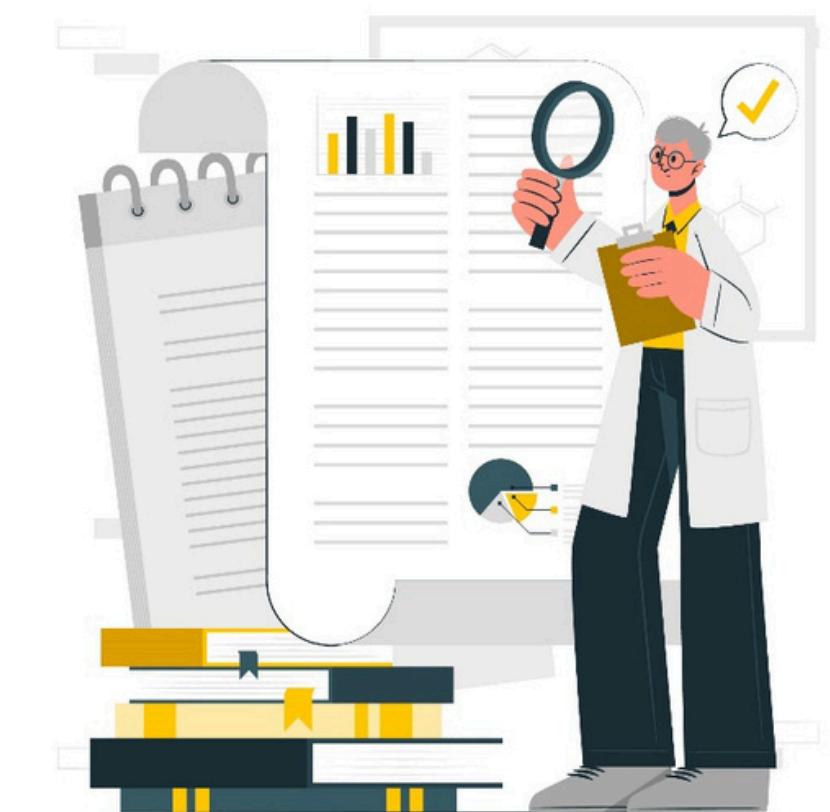
- LoRA/QLoRA ince ayar (docs/LORA_PLANI.md)
- Appwrite entegrasyonunun genişletilmesi

Uzun Vadeli:

- Resmi üniversite sistemleriyle entegrasyon
- Çoklu dil ve kişiselleştirme

Görsel Önerisi: Yol haritası (P1–P2–P3).

Konuşmacı Notu (özet): Yol haritası ve LoRA planı referans gösterilir.



DEMO

8/9

Canlı Demo:

- Yerel ortam (backend: <http://localhost:8000>, Flutter app)

Test Edilecek Sorular:

1. "Selçuk Üniversitesi nerede?"
2. "Kampüsler hangileri?"
3. "Kaç fakülte var?"
4. "Rektör kim?"
5. "Bilgisayar Mühendisliği hangi fakültede?"

Görsel Önerisi: Demo akış adımları + ekran görüntüsü.

Konuşmacı Notu (özet): Demo script adımları DEMO_SCRIPT.md'de detaylıdır.





T.C.
SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Teşekkürler...

9/9

KAYNAKLAR (Slayt sınırı dışında)

8/9

- [1] Selçuk AI Asistanı README ve dokümantasyonları (README.md, docs/*.md)
- [2] Selçuk Üniversitesi Resmi Web Sitesi - <https://www.selcuk.edu.tr>
- [3] FastAPI Documentation - <https://fastapi.tiangolo.com>
- [4] Flutter Documentation - <https://docs.flutter.dev>
- [5] Ollama Documentation - <https://ollama.com/docs>
- [6] FAISS Documentation - <https://github.com/facebookresearch/faiss>
- [7] Sentence-Transformers - <https://www.sbert.net>
- [8] RAG: Lewis et al., 2020, Retrieval-Augmented Generation

