



İZMİR BAKIRÇAY ÜNİVERSİTESİ
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL440 YZ DESTEKLİ YAZILIM GELİŞTİRME
2025-26 GÜZ, FINAL PROJESİ

AI-AUGMENTED SOFTWARE
LIFECYCLE PROJECT

HAZIRLAYANLAR:

Sudenaz ONARAN 220601060

Ece KARAALP 220601046

İÇİNDEKİLER

1. ANALİZ AŞAMASI	4
1.1. Problem Tanımı	4
1.2. Proje Hedefleri	4
1.3. Kullanıcı Gereksinimleri	4
1.4. Sistem Gereksinimleri	5
1.5. Akıllı Özellikler ve Yapay Zekâ Gereksinimleri	5
1.6. Kısıtlar ve Varsayımlar	5
1.7. Kullanıcı Hikayeleri	6
2. TASARIM AŞAMASI – AI Prompt Kullanımı	6
2.1. UML Component Diagram (Bileşen Diyagramı)	8
3. GELİŞTİRME	9
3.1. Kullanılan Teknolojiler	9
3.2. Geliştirilen Ana Modüller	9
3.2.1. Doküman Yükleme ve Yönetimi	9
3.2.2. Öztleme Sistemi	9
3.2.3. Akıllı Asistan (Soru-Cevap)	9
3.2.4. Sohbet Geçmişsi Kaydı	9
3.2.5. Geri Bildirim ve Hata Raporlama	10
3.3. Yapay Zekâ Destekli Geliştirme Süreci	10
3.4. Versiyon Kontrolü ve Commit Yapısı	10
3.5. Karşılaşılan Teknik Zorluklar	10
4. TEST VE HATA AYIKLAMA AŞAMASI	10
4.1. Test Yaklaşımı	10
4.2. Fonksiyonel Testler (AI Destekli)	11
4.3. Edge-Case (Sınır Durum) Senaryoları	11
4.4. Yanlış / Eksik AI Çıktısı (Bilinçli Raporlama)	12
4.5. Hata Ayıklama Süreci	12
4.6. Test Aşamasının Sonuçları	13
5. ETİK, GÜVENLİK VE LİSANS DEĞERLENDİRMESİ	13
5.1. Kod Lisansı Riski	13

5.2.	Veri Gizliliği	14
5.3.	Güvenlik Açıkları	14
5.4.	AI Hallucination ve Eksik Bağlamsal Çıktılar	15
5.4.1.	Etik Kullanım.....	15
5.5.	Genel Değerlendirme.....	15
6.	AI DECISION LOG.....	16

1. ANALİZ AŞAMASI

1.1. Problem Tanımı

Kullanıcılar, PDF ve metin tabanlı dokümanlar başta olmak üzere birçok farklı belge içerisinde aradıkları bilgiye hızlı bir şekilde ulaşmakta zorlanmaktadır. Bilgiye erişilse dahi, içeriklerin uzun ve karmaşık olması sebebiyle kullanıcılar bu bilgileri anlamlandırmakta ve sonuca ulaşmakta güçlük çekmektedir.

Bu projenin amacı; kullanıcıların yüklediği dokümanları sistemli bir şekilde saklayan, doküman içeriği üzerinde arama yapabilen ve yapay zekâ destekli özetleme ile bilgiye erişimi kolaylaştıran web tabanlı bir sistem geliştirmektir.

1.2. Proje Hedefleri

Projenin temel hedefi; kullanıcıların yüklediği dokümanları saklayabilen, içerik üzerinde arama yapabilen ve yapay zekâ destekli özetleme ile bilgiye hızlı erişim sağlayan bir sistem geliştirmektir.

1.3. Kullanıcı Gereksinimleri

Bu proje, kullanıcıların dokümanlar üzerinde yapay zeka destekli analiz yapabilmesini ve içerikleri daha verimli inceleyebilmesini amaçlayan bir web uygulamasıdır. Kullanıcı gereksinimleri, sistemin temel işlevlerini ve kullanıcı beklentilerini karşılaşacak şekilde belirlenmiştir.

Kullanıcıların temel ihtiyaçları şunlardır:

- Doküman yükleyebilmek
- Doküman içeriğini görüntüleyebilmek
- Anahtar kelime ile arama yapabilmek
- Kısa ve detaylı özet oluşturabilmek
- Özetleri kaydedip görüntüleyebilmek
- Kaydedilen özetleri silebilmek
- Doküman hakkında soru sorabilmek
- Önceki soru-cevap geçmişini görüntüleyebilmek
- Yapay zekanın hatalı cevaplarını raporlayabilmek

1.4.Sistem Gereksinimleri

Sistem, modern web teknolojileri ve yapay zekâ entegrasyonu kullanılarak geliştirilmiş olup güvenli ve performanslı çalışacak şekilde tasarlanmıştır.

1.4.1. Fonksiyonel Gereksinimler

- Kullanıcı doküman yükleyip görüntüleyebilmelidir.
- Sistem anahtar kelime araması yapabilmelidir.
- Kısa ve detaylı özet oluşturulabilmelidir.
- Özeler veritabanına kaydedilip silinebilmelidir.
- Kullanıcı doküman hakkında soru sorabilmelidir.
- Sistem yalnızca doküman içeriğine göre cevap vermelidir.
- Soru-cevap geçmişi görüntülenebilmelidir.
- Hatalı AI cevapları raporlanabilmelidir.

1.4.2. Fonksiyonel Olmayan Gereksinimler

- Özelleme ve soru-cevap işlemleri makul sürede tamamlanmalıdır.
- Arayüz sade ve kullanıcı dostu olmalıdır.
- Veritabanı işlemleri yalnızca sunucu tarafında yapılmalıdır.
- Sistem artan kullanıcı sayısını destekleyebilmelidir.
- Kullanıcı verileri güvenli şekilde saklanmalıdır.

1.5.Akıllı Özellikler ve Yapay Zekâ Gereksinimleri

Sistem, yapay zeka destekli aşağıdaki özellikleri sunmaktadır:

- Otomatik kısa özet oluşturma
- Otomatik detaylı özet oluşturma
- Doküman tabanlı soru-cevap sistemi
- AI hata raporlama mekanizması

Yapay zeka, yalnızca kullanıcı tarafından yüklenen doküman içeriğini kullanarak cevap üretmelidir. Dış kaynaklardan bilgi eklememelidir.

1.6.Kısıtlar ve Varsayımlar

Kısıtlar:

- Yapay zeka API kotası sınırlıdır.

- İnternet bağlantısı gereklidir.
- Büyük dokümanlar daha uzun sürede işlenebilir.

Varsayımlar:

- Kullanıcıların temel bilgisayar bilgisi vardır.
- Yüklenen dokümanlar metin tabanlıdır.
- Kullanıcılar doğru ve anlamlı sorular soracaktır.

1.7.Kullanıcı Hikayeleri

KH-1:

Bir kullanıcı olarak dokümanımı yükleyip içeriğini görüntülemek istiyorum, böylece dokümani sistem üzerinden inceleyebilirim.

KH-2:

Bir kullanıcı olarak kısa özet almak istiyorum, böylece dokümanın ana fikrini hızla öğrenebilirim.

KH-3:

Bir kullanıcı olarak detaylı özet almak istiyorum, böylece dokümanın tüm içeriğini daha iyi anlayabilirim.

KH-4:

Bir kullanıcı olarak yapay zekaya soru sormak istiyorum, böylece doküman hakkında hızlı cevap alabilirim.

KH-5:

Bir kullanıcı olarak önceki soru-cevapları görmek istiyorum, böylece geçmişte sorduklarımı tekrar inceleyebilirim.

KH-6:

Bir kullanıcı olarak hatalı cevapları raporlamak istiyorum, böylece sistemin gelişmesine katkı sağlayabilirim.

2. TASARIM AŞAMASI – AI Prompt Kullanımı

Bu projede tasarım süreci, yapay zekâ destekli bir başlangıç prompt'u ile şekillendirilmiştir. Sistem mimarisi, kullanıcı arayüzü bileşenleri, yapay zekâ entegrasyonu ve temel fonksiyonlar

ilk olarak aşağıdaki AI-Generated prompt üzerinden planlanmıştır. Ancak nihai tasarım kararları insan değerlendirmesiyle verilmiştir.

Prompt (AI-Generated):

Amaç: Akıllı Doküman Arama ve Öztleme Sistemi için web tabanlı bir uygulama tasarla ve geliştir.

Uygulama Tanımı: Kullanıcının PDF veya metin formatındaki dokümanları yükleyebildiği, bu dokümanlar üzerinde hem anahtar kelimeye dayalı hem de anlamsal (doğal dil) arama yapabildiği bir web uygulaması oluştur.

Teknoloji ve Platform:

Frontend: Modern ve sade bir web arayüzü (React veya Next.js tercih edilebilir)

Backend: Firebase (Authentication, Firestore, Storage, Functions)

Yapay Zekâ: Doküman içeriğine dayalı soru-cevap ve öztleme için büyük dil modeli entegrasyonu

Temel Özellikler:

Kullanıcı Arayüzü

- Giriş / kayıt ekranı (Firebase Authentication)
- Doküman yükleme alanı (PDF veya TXT)
- Yüklenen dokümanların listelendiği panel
- Arama çubuğu (anahtar kelime ile arama)
- Doğal dilde soru sorma alanı

Doküman Yönetimi

- Yüklenen dokümanları Firebase Storage'da sakla
- Doküman metinlerini Firestore'da indekslenebilir şekilde kaydet

Arama Özellikleri

- Anahtar kelimeye dayalı klasik metin araması
- Kullanıcının doğal dilde sorduğu soruya, doküman içeriğine dayanarak cevap üret

- Üretilen cevabın hangi doküman(lar)a dayandığını açıkça belirt

Özetleme

- Her doküman için otomatik kısa özet üret

- Kullanıcı isterse detaylı özet oluştur

Akıllı Davranış ve Raporlama

- Yapay zekâdan bilinçli olarak en az bir yanlış veya eksik cevap üretebilecek bir senaryo oluştur

- Bu hatalı çıktıyı kullanıcıya raporlayabileceği veya işaretleyebileceği bir alan ekle

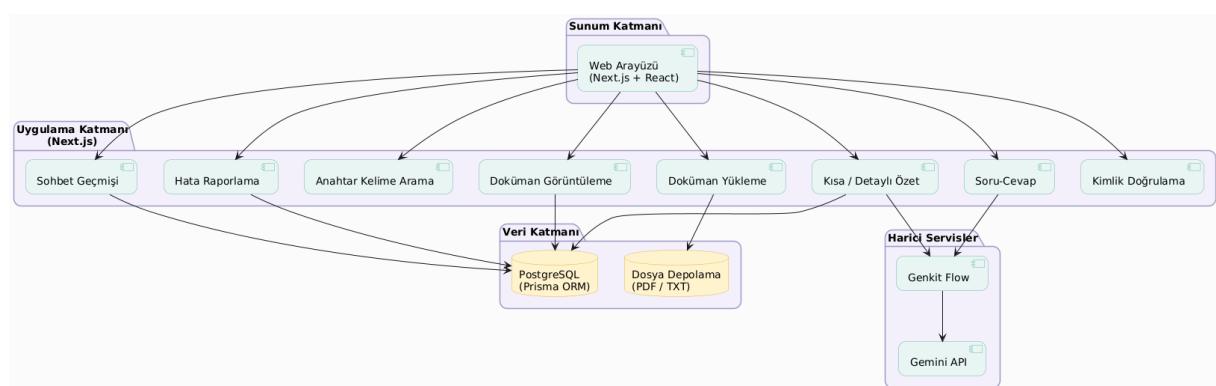
Tasarım Beklentileri:

- Minimal, modern ve öğrenci projesi sunumuna uygun tasarım
- Kart tabanlı doküman listesi
- Okunabilirlik odaklı tipografi

Çıktı:

- Sayfa yapısı ve bileşenler
- Firebase koleksiyon yapısı
- Örnek UI akışı
- Yapay zekâ entegrasyon mimarisi

2.1. UML Component Diagram (Bileşen Diyagramı)



3. GELİŞTİRME

Bu aşamada, tasarım sürecinde belirlenen mimari ve teknolojiler kullanılarak sistemin yazılım geliştirme süreci gerçekleştirılmıştır.

3.1. Kullanılan Teknolojiler

Projede aşağıdaki teknolojiler kullanılmıştır:

- **Next.js:** Sunucu işlemleri ve sayfa yönetimi
- **TypeScript:** Tip güvenliği
- **Tailwind + ShadCN UI:** Arayüz geliştirme
- **PostgreSQL:** Veri saklama
- **Prisma:** Veritabanı işlemleri
- **Firebase:** Hosting
- **Genkit + Gemini:** Yapay zekâ özellikleri

3.2. Geliştirilen Ana Modüller

3.2.1. Doküman Yükleme ve Yönetimi

Kullanıcılar PDF veya metin tabanlı dokümanları sisteme yükleyebilmekte ve bu dokümanlar veritabanında saklanmaktadır.

3.2.2. Öztleme Sistemi

Dokümanlar için kısa ve detaylı özetler, Genkit üzerinden Gemini API kullanılarak oluşturulmaktadır.

3.2.3. Akıllı Asistan (Soru-Cevap)

Kullanıcılar doküman içeriğine dayalı sorular sorabilmekte, yapay zekâ yalnızca verilen metne göre cevap üretmektedir.

3.2.4. Sohbet Geçmişi Kaydı

Soru-cevap geçmişi veritabanına kaydedilmekte ve kullanıcı arayüzünde görüntülenebilmektedir.

3.2.5. Geri Bildirim ve Hata Raporlama

Kullanıcılar, yapay zekâ cevapları için geri bildirim verebilmekte ve hatalı cevapları raporlayabilmektedir.

3.3. Yapay Zekâ Destekli Geliştirme Süreci

Geliştirme sürecinde ChatGPT, GitHub Copilot, Firebase Studio ve Gemini API araçlarından faydalanyılmıştır. Ancak tüm AI çıktıları insan kontrolünden geçirilerek düzenlenmiştir.

3.4. Versiyon Kontrolü ve Commit Yapısı

Proje GitHub üzerinden yönetilmiş ve commit mesajları şu şekilde etiketlenmiştir:

- [AI-generated]
- [AI-assisted]
- [Human-written]

Bu yapı, kodların nasıl üretildiğinin şeffaf biçimde izlenmesini sağlamıştır.

3.5. Karşılaşılan Teknik Zorluklar

Geliştirme sürecinde Prisma ilişki hataları, API kota sınırları, veri senkronizasyonu ve AI çıktılarının bağlama uygunluğu gibi sorunlar yaşanmış, bu sorunlar manuel müdahalelerle çözülmüştür.

4. TEST VE HATA AYIKLAMA AŞAMASI

Bu aşamada, geliştirilen sistemin doğruluğu, güvenilirliği ve yapay zekâ çıktılarının tutarlılığı test edilmiştir. Test sürecinde hem geleneksel yazılım test yaklaşımı hem de yapay zekâ destekli değerlendirmeler kullanılmıştır.

Test sürecinin temel amaçları:

- Sistem fonksiyonlarının doğru çalıştığını doğrulamak
- Yapay zekâ çıktılarının bağlama uygunluğunu kontrol etmek
- Hatalı veya eksik AI cevaplarını tespit etmek
- Edge-case (sınır durum) senaryolarını değerlendirmek

4.1. Test Yaklaşımı

Test süreci üç ana başlıkta yürütülmüştür:

1. Fonksiyonel Testler
2. Edge-Case Senaryoları
3. Yapay Zekâ Çıktı Analizi

Testlerin bir kısmı **ChatGPT** ve **Gemini** gibi yapay zekâ araçlarının önerileriyle oluşturulmuş, ancak nihai değerlendirme insan tarafından yapılmıştır.

4.2.Fonksiyonel Testler (AI Destekli)

Aşağıdaki test senaryoları ChatGPT tarafından önerilmiş ve geliştirici tarafından uygulanmıştır:

Test ID	Test Senaryosu	Beklenen Sonuç	Sonuç
T-1	Doküman yükleme	Dosya başarıyla yüklenir	Başarılı
T-2	Kısa özet oluşturma	Kısa özet üretilir	Başarılı
T-3	Detaylı özet oluşturma	Detaylı özet üretilir	Başarılı
T-4	Soru-cevap	Dökümana uygun cevap	Başarılı
T-5	Özet silme	Özet DB'den silinir	Başarılı
T-6	Sohbet geçmişi	Geçmiş görüntülenir	Başarılı

Bu test senaryoları, sistemin temel fonksiyonlarının çalıştığını göstermektedir.

4.3.Edge-Case (Sınır Durum) Senaryoları

Aşağıdaki edge-case senaryoları Gemini tarafından önerilmiş ve manuel olarak test edilmiştir:

Senaryo	Beklenen Davranış	Sonuç
Boş doküman yükleme	Sistem hata vermelii	Doğru
Çok büyük doküman	Yavaş ama çalışmalii	Doğru
Alakasız soru	“Bilgi yok” cevabı	Doğru
Dökümanda olmayan bilgi	Cevap verilmemeli	Doğru
API kotası dolu	Uyarı mesajı	Doğru

Bu testler, sistemin olağan dışı durumlara karşı dayanıklı olduğunu göstermektedir.

4.4.Yanlış / Eksik AI Çıktısı (Bilinçli Raporlama)

Ders gereksinimleri doğrultusunda, yapay zekâ destekli akıllı asistanın verdiği eksik bağlamsal bir cevap bilinçli olarak analiz edilmiştir.

Test sürecinde, yapay zekânın bazı durumlarda eksik bağlamsal açıklamalar yaptığı gözlemlenmiştir.

Örnek 1:

Dokümanda tarih bilgisi bulunmayan bir soruya doğru ancak yönlendirici olmayan bir cevap verilmiştir. Bu durum, kullanıcı deneyimini sınırladığı için sistemde hata raporlama mekanizması geliştirilmiştir.

Örnek 2:

Test sürecinde, Türkçe yüklenen dokümanlar için özetlerin İngilizce üretildiği tespit edilmiştir. Yapılan incelemede, bu durumun özetleme prompt'unun İngilizce olmasından kaynaklandığı anlaşılmıştır. Prompt Türkçeye çevrilmiş ve sistem yeniden test edilmiştir. Sonuç olarak özetlerin doğru dilde üretildiği doğrulanmıştır.

4.5.Hata Ayıklama Süreci

Test sırasında karşılaşılan başlıca sorunlar:

Sorun	Neden	Cözüm
Prisma ilişki hatası	Eksik model ilişkisi Manuel düzenleme	
Gemini API kotası	Günlük limit doldu	Test ertelendi
UI buton konumu	Yanlış yerleşim	Manuel düzeltildi
AI bağlam hatası	Prompt yetersiz	Prompt revize edildi

Bu sorunlar, hem AI önerileri hem de insan müdahalesiyle giderilmiştir.

4.6.Test Aşamasının Sonuçları

Bu aşamada:

- Sistem fonksiyonları doğrulanmıştır
- AI çıktıları analiz edilmiştir
- En az bir eksik bağlamaşal AI cevabı raporlanmıştır
- İnsan müdahalesinin önemi açıkça görülmüştür

Bu sonuçlar, yapay zekânın doğru bilgi verebilmesine rağmen kullanıcı deneyimi ve bağlam açısından yetersiz kalabileceğini göstermektedir.

5. ETİK, GÜVENLİK VE LİSANS DEĞERLENDİRMESİ

Bu bölümde, geliştirilen yapay zekâ destekli doküman analiz sisteminin etik, güvenlik, lisans ve veri gizliliği açısından oluşturabileceği riskler değerlendirilmiştir. Ayrıca, yapay zekâ kaynaklı yanlış veya yanıldıcı çıktılar (hallucination) örnekleri tartışılmıştır.

5.1.Kod Lisansı Riski

Projede kullanılan temel teknolojiler:

- Next.js
- Firebase Studio
- Prisma ORM
- PostgreSQL
- Gemini API / ChatGPT

Bu araçların büyük çoğunluğu **açık kaynak lisansları** ile sunulmaktadır. Ancak:

- Yapay zekâ tarafından üretilen kodların lisans durumu belirsiz olabilir.
- AI tarafından önerilen kod parçalarının doğrudan kopyalanması, telif riski doğurabilir.

Bu nedenle:

- AI çıktıları doğrudan kopyalanmamış
- Kodlar insan tarafından yeniden yazılmış ve düzenlenmiştir
- Commit mesajlarında **[AI-assisted]** etiketi kullanılmıştır

Bu yaklaşım, lisans riskini azaltmayı amaçlamaktadır.

5.2.Veri Gizliliği

Sistemde kullanıcı tarafından yüklenen dokümanlar ve sohbet geçmişleri veritabanında saklanmaktadır.

Olası riskler:

- Kullanıcı dokümanlarında kişisel veri bulunabilir
- AI servislerine gönderilen içerikler üçüncü taraf sistemlere ilettilir

Alınan önlemler:

- Hassas veri yüklenmemesi kullanıcıya bildirilmiştir
- Sadece gerekli metin parçaları AI API'ye gönderilmiştir
- Sohbet geçmişi yalnızca kullanıcıya özeldir

Bu sayede veri gizliliği riskleri minimize edilmiştir.

5.3.Güvenlik Açıkları

Sistemde oluşabilecek güvenlik riskleri:

Risk	Açıklama
Yetkisiz erişim	Kullanıcı verilerine izinsiz erişim
API anahtarlarının sızması	Gemini / Firebase anahtarları
SQL Injection	Veritabanı sorguları

Alınan önlemler:

- Ortam değişkenleri (.env) kullanıldı
- Prisma ORM ile güvenli sorgular yazıldı
- API anahtarları GitHub'a eklenmedi
- Firebase Authentication kullanıldı

Bu önlemler, sistemin temel güvenlik seviyesini sağlamaktadır.

5.4.AI Hallucination ve Eksik Bağlamsal Çıktılar

Yapay zekâ sistemleri her zaman yanlış bilgi üretmese bile, bazı durumlarda eksik, yönlendirmesiz veya bağlamı dar cevaplar verebilmektedir. Bu durum, kullanıcıyı yeterince bilgilendirmeme riskini doğurur.

5.4.1. Etik Kullanım

Proje kapsamında yapay zekâ:

- Yardımcı araç olarak kullanılmış
- Nihai kararlar insan tarafından verilmiştir
- AI çıktıları eleştirel süzgeçten geçirilmiştir
- Yanlış veya eksik çıktılar raporlanmıştır

Bu yaklaşım:

- Akademik dürüstlüğü
- Mühendislik etik ilkelerini
- Sorumlu AI kullanımını

desteklemektedir.

5.5.Genel Değerlendirme

Bu projede:

- Yapay zekâ güçlü bir yardımcı araç olarak kullanılmış
- Ancak körük körüğe güvenilmemiştir
- Eksik ve bağımsız olarak zayıf çıktılar tespit edilmiştir
- İnsan kontrolü her aşamada korunmuştur

Bu durum, projenin etik, güvenli ve bilinçli AI kullanımı açısından uygun olduğunu göstermektedir.

6. AI DECISION LOG

Aşama	Kullanılan YZ	YZ Önerisi	Nihai Karar	Gerekçe
Analiz	ChatGPT	Proje iskeleti, kullanıcı gereksinimleri ve kullanıcı hikayelerinin oluşturulması	Kabul edildi	Standartlara uygun, kapsamlı ve anlaşılır bir yapı sundu
Analiz	Gemini	Risk analizine token limiti ve maliyet kısıtlarının eklenmesi	Kabul edildi	Teknik gerçekçiliği artırdığı için
Analiz	Gemini	OCR entegrasyonu eklenmesi	Reddedildi	MVP kapsamını aştığı için
Tasarım	ChatGPT	ShadCN UI kullanımı	Kabul edildi	Modern ve erişilebilir arayüz sağladığı için
Tasarım	Gemini	AI Flow mimarisi (Genkit)	Kabul edildi	Modüler ve yönetilebilir yapı sunduğu için
Tasarım	Firebase Studio	Firebase Hosting kullanımı	Kabul edildi	Kolay kurulum ve dağıtım sağladığı için
Geliştirme	ChatGPT	Prisma ChatHistory modeli eklenmesi	Kabul edildi	Veri yapısı doğrudydu, küçük düzenleme yapıldı
Geliştirme	GitHub Copilot	UI buton ve modal yapıları	Düzenlenerek kullanıldı	Erişilebilirlik için değişiklik yapıldı
Geliştirme	Gemini	İngilizce prompt ile özetleme	Reddedildi	Türkçe dil uyumu bozulduğu için
Test	ChatGPT	Fonksiyonel test senaryoları	Kabul edildi	Mantıklı ve yeterliydi
Test	Gemini	Edge-case (sınır durum) senaryoları	Kabul edildi	Gerçekçi senaryolar sundu
Test	ChatGPT	“Cevap teknik olarak doğru” yorumu	Kısmen reddedildi	Kullanıcı deneyimi açısından yetersizdi
Test	Gemini	İngilizce özet üretimi normaldir	Reddedildi	Kullanıcı dili Türkçe olduğu için
Test	ChatGPT	Prompt Türkçeye çevrilmeli	Kabul edildi	Dil uyumu sağlandı