



MEF Kütüphane - Sanal Asistanı: ChatGPT Tabanlı Bir Chatbot Geliştirme Süreci

The MEF Library - Virtual Assistant: The Development Process of a ChatGPT-based Chatbot

Ramazan ÇELİK* , Neslihan ERDOĞAN** , Gaye KOCATÜRK***

ÖZET

Bu makale, MEF Üniversitesi Kütüphanesi tarafından geliştirilen yapay zekâ tabanlı bir sanal asistan olan MEF – Kütüphane Chatbot projesinin geliştirme sürecini kapsamlı biçimde ele almaktadır. Proje, kullanıcıların bilgiye hızlı ve etkili erişimini sağlamak, kütüphane hizmetlerini dijitalleştirmek ve kullanıcı memnuniyetini artırmak amacıyla başlatılmıştır. OpenAI’nin ChatGPT API’si kullanılarak geliştirilen bu Chatbot, sıkça sorulan soruları yanıtlayabilmekte, kütüphane etkinlikleri hakkında bilgi verebilmekte ve çeşitli birimlerden gelen soruları tek bir noktadan yönetebilmektedir. Geliştirme sürecinde React tabanlı bir ön yüz mimarisi, GitHub ve Vercel gibi modern yazılım araçları kullanılmıştır. API anahtarı yönetimi, dosya yükleme, model ayarları ve kullanıcıya özel yanıt üretimi gibi teknik detaylar titizlikle planlanmıştır. Chatbot’un yapılandırılması sırasında kullanıcı profilleri, yanıt üslubu, hata yönetimi ve bilgi kaynakları gibi unsurlar detaylı biçimde tanımlanmıştır. Proje, yalnızca teknik bir uygulama değil, aynı zamanda MEF Kütüphane’nin dijital dönüşüm vizyonunun bir parçası olarak konumlandırılmıştır. Makale, kütüphane hizmetlerinde yapay zekâ tabanlı sanal asistan uygulamalarının tasarım, entegrasyon ve kullanıcı deneyimi boyutlarını ortaya koyarak akademik kütüphanelerde dijital dönüşüm süreçlerine uygulanabilir bir model sunması bakımından literatüre özgün bir katkı sağlamaktadır.

Anahtar Kelimeler: MEF Kütüphane - Sanal Asistan, Chatbot, ChatGPT, Kütüphane, API

* Direktör Yardımcısı, MEF Üniversitesi, Kütüphane Direktörlüğü, celikr@mef.edu.tr,

<https://orcid.org/0000-0002-6509-3071>

** Uzman Yardımcısı, MEF Üniversitesi, Kütüphane Direktörlüğü, erdoganne@mef.edu.tr,

<https://orcid.org/0009-0002-5629-5517>

*** Uzman Yardımcısı, MEF Üniversitesi, Kütüphane Direktörlüğü, yilmazgaye@mef.edu.tr,

<https://orcid.org/0009-0003-0698-6296>

ABSTRACT

This article provides a comprehensive overview of the development process of the MEF Library Chatbot project, which is an artificial intelligence-based virtual assistant developed by the MEF University Library. The project was initiated to provide users with fast and effective access to information, to digitize library services and to increase user satisfaction. Built using OpenAI's ChatGPT API, the chatbot can answer frequently asked questions, provide information about library events and handle enquiries from various departments via a single interface. The development process involved the use of a React-based front-end architecture and modern software tools such as GitHub and Vercel. Technical details such as API key management, file uploading, model settings and user-specific response generation were meticulously planned. During chatbot configuration, elements such as user profiles, response style, error management and information sources were defined in detail. The project is not only positioned as a technical application, but also as part of the MEF Library's vision for digital transformation. This article makes a unique contribution to the literature by presenting a model applicable to the digital transformation of academic libraries. It highlights the design, integration and user experience of artificial intelligence-based virtual assistant applications in library services.

Keywords: MEF Library - Virtual Assistant, Chatbot, ChatGPT, Library, API

GİRİŞ

Son yıllarda, yapay zekâ (YZ) birçok alanda büyük bir ilgi odağı haline gelmiştir. Çeşitli sektörlerde entegrasyonu, hizmetlerin yeniden tasarlanıp sunulmasında önemli ölçüde dönüşüm sağlanmıştır. Bilgi yayılımının temel merkezleri olan kütüphaneler, teknolojik yenilik dalgasından etkilenen alanlar arasında yer almaktadır. Kullanıcıların daha hızlı, kişiselleştirilmiş ve kolay erişilebilir bilgilere yönelik artan beklentileri doğrultusunda, geleneksel kütüphane hizmetleri akıllı dijital araçlarla dönüşüm geçirmektedir. Ortaya çıkan ihtiyaçları ve YZ'nin kullanıcı deneyimini geliştirme potansiyelini fark eden MEF Kütüphane, YZ destekli bir sohbet robotu sistemi geliştirme ve uygulama konusunda öncü bir girişim başlatmıştır.

MEF Kütüphane Chatbot Projesi, hızlı, etkili ve etkileşimli bir iletişim yolu sunarak kullanıcılar (Akademisyen, Öğrenci, İdari Personel) ile kütüphane kaynakları arasındaki bağı kuvvetlendirmeyi amaçlayan bir sanal asistandır. YZ tabanlı bu sanal asistan, kullanıcıların kütüphane hizmetlerine kolay bir erişimini sağlamak üzere geliştirilmiştir. Sıkça sorulan sorulara yanıt verebilen, kullanıcıları uygun kaynaklara yönlendirebilen, kütüphane hizmetleri ve etkinlikleri hakkında gerçek zamanlı bilgi sağlayabilen sanal bir referans asistanı olarak işlev görmektedir. Sohbet robotu, rutin soruları otomatikleştirerek yalnızca kütüphanenin kullanıcı hizmetlerinin verimliliğini artırmanın yanında kütüphane personelinin diğer görevlere daha fazla zaman ayırmasını sağlamaktadır.

Sanal asistan, doğal dil işleme ve üretiminde yüksek başarı gösteren OpenAI'nin konuşma tabanlı YZ modeli olan ChatGPT dil modelini kullanmaktadır (Chaves vd., 2020). ChatGPT'nin lisanslı API¹ sürümü, güvenilir, doğru yanıtlar sağlamak için sanal asistan sistemine entegre edilmiştir. Geliştirme sürecinde, kod yönetimi ve projenin web uygulama kısmının barındırılması, ilgili platforma kurulum (deploy) işlemlerinin yapılması için Vercel² platformu tercih edilmiştir.

¹ API (Application Programming Interface), bir yazılımın başka bir yazılım ile konuşmasını sağlayan arayüzdür. Chatbot süreçlerinde API'ler, botun dış sistemlerden veri almasını, veri göndermesini veya belirli işlemleri otomatik yapmasını sağlar.

² Vercel, modern web uygulamalarının geliştirilmesi, test edilmesi ve yayınlanması için kullanılan bulut tabanlı bir platformdur.

Altyapı, hızlı geliştirme, sürekli dağıtım ve ölçeklenebilirlik sağlayarak sanal asistanın gelecekteki geliştirmelere ve kullanıcı geri bildirimlerine uyum sağlaması bakımından önemli görülmektedir.

Teknik işlevselliğin ötesinde, sohbet robotu MEF Kütüphane'nin yükseköğrenim ortamlarında inovasyonu ve dijital dönüşümü benimsemeye yönelik daha geniş bir stratejik taahhüdünü yansıtmaktadır. Kütüphane, temel hizmetlerine akıllı teknolojileri yerleştirerek yalnızca kullanıcılarının mevcut taleplerini karşılamayı değil, aynı zamanda daha ilgi çekici, duyarlı ve dijital olarak akıcı bir öğrenme ortamı oluşturmayı da amaçlamaktadır.

Bu çalışmada, MEF Kütüphane Chatbot Projesinin geliştirilme süreci kavramsal temelleri ve teknik uygulama boyutlarıyla bütüncül bir çerçevede incelenmektedir. İlk olarak YZ chatbot mimarisi ve API anahtarı kullanımı gibi temel kavramlara ilişkin açıklamalar sunulmakta ve ardından projenin ortaya çıkış gerekçesini oluşturan kullanıcı ihtiyaçları ve çözülmesi hedeflenen yapısal sorunlar ele alınmaktadır. Devam eden bölümlerde, (GPT) tabanlı sanal asistanın tasarım ilkeleri, model konfigürasyonları, dosya yükleme süreçleri ve talimat setlerinin oluşturulmasına yönelik yöntemler ayrıntılı şekilde açıklanmaktadır. Bunu takiben, kodlama sürecine ilişkin teknik aşamalar ile GitHub³ ve Vercel tabanlı entegrasyon süreçleri tanıtılmaktadır. Son aşamada ise sistem testleri, web sayfası entegrasyonu ve geliştirilen sanal asistanın kullanıcı deneyimine sağladığı katkılar değerlendirilmektedir. Böylece çalışma, hem yazılım geliştirme sürecine yönelik teknik bir yol haritası hem de kütüphane hizmetlerinde YZ uygulamalarının işlevselliğine odaklanmaktadır.

Tanım ve Temel Kavramlar

YZ, makinelerin ve bilgisayar sistemlerinin deneyimlerden öğrenme, dili anlama, kalıpları tanımlama ve seçim yapma gibi insani bilişsel süreçlerini taklit etme yeteneği olarak tanımlanabilir (Çelik, Çuhadar, 2025).

- API⁴ anahtarı, bir uygulamanın belirli bir servise güvenli biçimde erişmesini sağlayan, rastgele oluşturulmuş özel bir kimlik doğrulama kodudur.
- Chatbot, kullanıcılarla doğal dil aracılığıyla etkileşim kurabilen, önceden tanımlanmış kurallar veya YZ tabanlı algoritmalarla çalışan dijital asistanlardır.

Gelişim Süreci

- 2000'li yılların başından itibaren, doğal dil işleme (NLP- Natural Language Processing) ⁵alanındaki çalışmalar, Chatbot teknolojisinin geliştirilmesine ivme kazandırmıştır.

³ GitHub, 2008 yılında kullanıma sunulan ve günümüzde Microsoft bünyesinde hizmet veren çevrimiçi bir yazılım geliştirme platformudur.

⁴ API anahtarı (Application Programming Interface Key), bir yazılım uygulamasının belirli bir servise veya veri kaynağına API aracılığıyla güvenli ve kontrollü bir şekilde erişimini sağlayan, genellikle uzun, benzersiz ve rastgele üretilmiş karakter dizilerinden oluşan bir kimlik doğrulama aracıdır.

⁵ Doğal dil işleme (NLP), bilgisayarların insan dillerini anlamasını, yorumlamasını ve üretmesini sağlayan yapay zekâ alanıdır.

- Bulut bilişim altyapısının yaygınlaşması, Chatbot'ların daha erişilebilir ve ölçeklenebilir hale gelmesini sağlamıştır.
- Makine öğrenmesi algoritmalarındaki gelişmeler, Chatbot'ların daha doğru ve bağlamsal yanıtlar üretmesine olanak tanımıştır.
- Büyük veri kaynaklarına erişimin artması, Chatbotların kullanıcı davranışlarını analiz ederek daha etkili öğrenme süreçleri geliştirmesini mümkün kılmıştır.
- Modern Chatbot sistemleri, gelişmiş YZ modelleriyle desteklenmiştir. Bu sayede Chatbotlar istemi anlayabilmekte, etkileşimlerden öğrenebilmekte ve insan benzeri yanıtlar üretebilmektedir.
- Web siteleri ve mesajlaşma platformlarına entegrasyon, bu sistemlerin kullanıcı deneyimini artırmak ve hizmet süreçlerini optimize etmek amacıyla yaygın biçimde kullanılmasını sağlamıştır.

Problem

MEF Kütüphane, hafta içi, hafta sonu ve final dönemlerinde farklı çalışma saatlerine sahiptir. Ancak kullanıcıların bilgiye erişim ihtiyaçları bu zaman dilimlerinin ötesine geçmektedir. Özellikle mesai saatleri dışında da kütüphane hizmetlerine ulaşmak isteyen araştırmacılar için zaman ve mekân sınırlamaları önemli bir engel oluşturmaktadır.

Çözüm

Bu doğrultuda, MEF Kütüphane olarak kullanıcıların ihtiyaç duydukları bilgilere her zaman ve her yerden erişebilmelerini sağlamak amacıyla bir sanal asistan geliştirilmesi planlanmıştır. Sanal asistanın, kütüphane hizmetleri hakkında sıkça sorulan sorulara yanıt vermesi, kullanıcıları doğru kaynaklara yönlendirmesi ve kütüphane ile ilgili temel bilgilere anında ulaşmalarını sağlaması hedeflenmiştir. Hedeflenen yaklaşım, kullanıcıların bilgiye erişimini kolaylaştırmak, hizmet sürekliliğini sağlamak ve kütüphane deneyimini daha zengin hale getirmektir.

Proje kapsamında hedeflenen temel kazanımlar şunlardır:

Proje kapsamında, zaman ve mekân sınırlaması olmaksızın bilgiye erişimin sağlanması, kütüphane hizmetleri hakkında anlık yanıtlar sunularak kullanıcıların bilgiye hızlı biçimde erişebilmesi, kullanıcı memnuniyeti ile hizmet sürekliliğinin artırılması ve kütüphane personelinin iş yükünün azaltılması temel kazanımlar olarak öngörülmüştür. Sanal asistanın, MEF Kütüphane'nin dijital dönüşüm sürecine katkı sağlaması, kullanıcı deneyimini iyileştirmesi ve bilgiye erişimin daha hızlı, kolay ve etkili hale gelmesini sağlayacaktır.

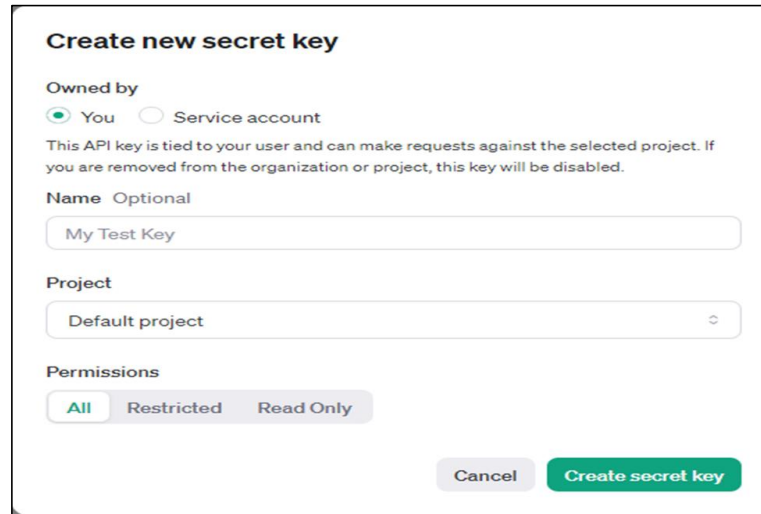
API Anahtarı Kullanım Alanları

API anahtarları, servis sağlayıcı tarafından istemcinin kimliğinin doğrulanması, erişim haklarının tanımlanması ve kullanıcıya özel veri aktarımının sağlanması amacıyla kullanılmaktadır (OpenAI, 2024).

API anahtarları, bilgi sistemlerinin güvenliğini sağlama, ağ trafiğini izleme, erişim yetkilendirmelerini yönetme ve hizmetlerin farklı dijital platformlarla yüksek hızda entegre edilmesini mümkün kılan temel bileşenler arasında yer almaktadır. Bu anahtarlar, sistemin işlevselliğini artırmakta, aynı zamanda veri bütünlüğü ve erişim kontrolü açısından kritik rol üstlenmektedir. MEF Kütüphane tarafından geliştirilen YZ tabanlı sanal asistan ile kullanılan API anahtarı, teknik entegrasyonu sağlamakta, sistemin sürdürülebilirliğini, esnekliğini ve ölçeklenebilirliğini desteklemektedir. Bu nedenle, özellikle dış kaynaklı hizmetlerle çalışan uygulamalarda API anahtarlarının kullanımı kritik bir öneme sahiptir.

Sanal asistan projesinde, OpenAI tarafından sunulan lisanslı ChatGPT dil modeli kullanılmaktadır. Bu modelin etkinleştirilmesi ve dış sistemle entegrasyonu için bir API'ye ihtiyaç duyulmuştur. Gerekli değerlendirmelerin ardından kullanım kapsamı ve gereksinimlere uygunluğu doğrultusunda OpenAI'nın sağladığı API tercih edilmiştir.

Bu doğrultuda, OpenAI üyeliği üzerinden MEF Kütüphane adına bir API anahtarı sağlanmıştır. İlgili anahtar, sistem içerisinde “MEF Chatbot” adıyla tanımlanmıştır. Anahtar bir metin dosyasına kaydedilerek kütüphane arşivinde saklanmıştır. API anahtarının harici bir dosyada muhafaza edilmesinin temel gerekçesi, ilerleyen aşamalarda aynı anahtarın tekrar kullanımının sağlanması ve yetkisiz erişim riskinin en aza indirilmesidir. Yürütülen süreç, YZ tabanlı uygulamaların geliştirilmesinde güvenlik, sürdürülebilirlik ve verimlilik gibi temel ilkeler açısından API anahtarlarının kritik bir rol üstlendiğini ortaya koymaktadır. API Anahtarı satın alma sürecine ilişkin ekran görselleri aşağıda verilmektedir.



Create new secret key

Owned by
☒ You ☐ Service account

This API key is tied to your user and can make requests against the selected project. If you are removed from the organization or project, this key will be disabled.

Name Optional
My Test Key

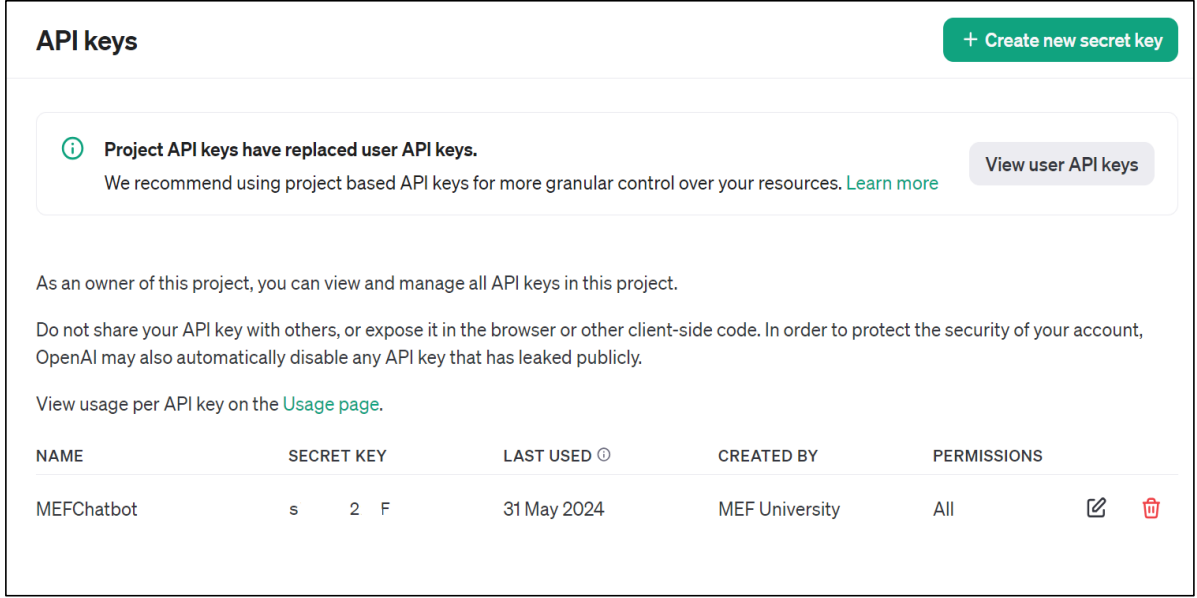
Project
Default project

Permissions
☒ All ☐ Restricted ☐ Read Only

Cancel Create secret key

Görsel 1. API Key Satın Alım Süreci Görseli (Adım -1)

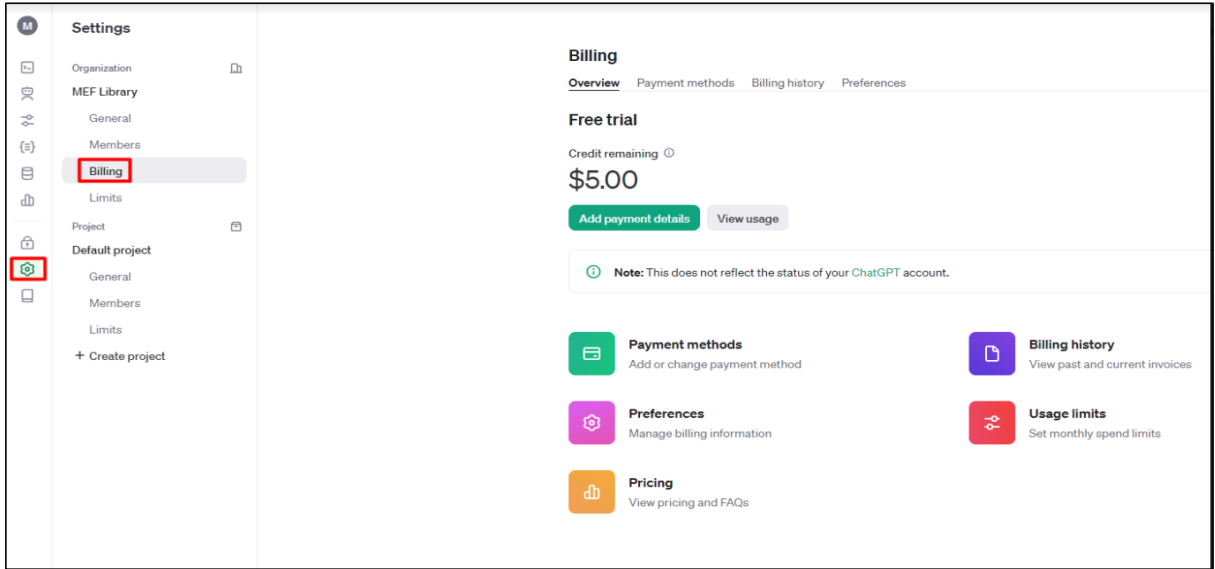
Kaynak: (<https://platform.openai.com/api-keys>)



Görsel 2. API Key Satın Alım Süreci Görseli (Adım -2)

Kaynak: (<https://platform.openai.com/api-keys>)

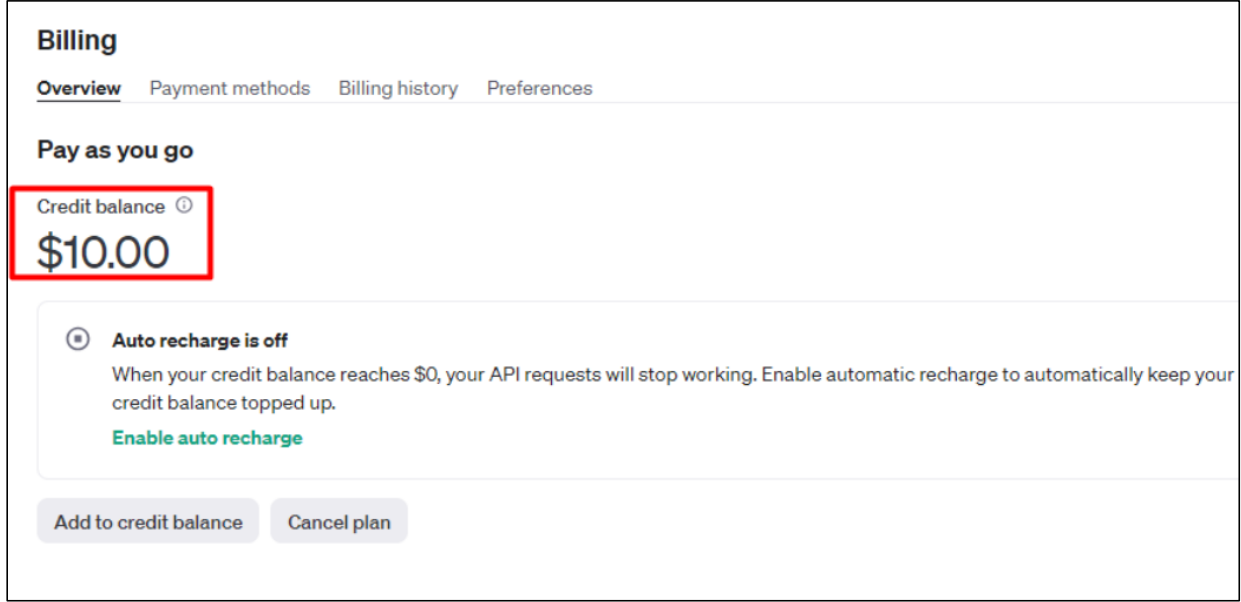
API Key satın alındıktan sonra, sanal asistanda kullanılmak üzere API hesabına artı 5 ABD doları tutarında bakiye yüklenmiştir (Görsel 3).



Görsel 3. API Key Bakiye Yükleme Adımları Görseli

Kaynak: (<https://platform.openai.com/settings/organization/billing/overview>)

Kullanıcı hesabına ilk başlangıçta sistem tarafından 5\$ hediye bakiye tanımlanmış olup, kredi kartı ile yüklenen artı 5\$ ile sistemde toplamda 10\$ kullanım kredisi sağlanmıştır (Görsel 4). Söz konusu kredi, her bir soru iletimi ve asistanın yanıt üretimi aşamasında önceden belirlenmiş oranlarda kademeli olarak kredi (token) düşmektedir.



Görsel 4. OpenAI Platformu Bakiye Yükleme Görseli

Kaynak: (<https://platform.openai.com/settings/organization/billing/overview>)

Sanal Asistan'ın Geliştirilme Süreci

MEF Kütüphane hizmetlerinin dijital ortamda sürdürülebilirliğini artırmak amacıyla, OpenAI tarafından sağlanan API anahtarı temin edildikten sonra sanal asistan geliştirme sürecine geçilmiştir. İlk aşamada, sistem yönetici paneline erişim sağlanarak ayarlar (Settings) menüsünden yönetim paneli (Dashboard) ekranına geçilmiş ve asistanlar (Assistants) sekmesi altında yeni bir asistan profili tanımlanmıştır.

Asistanın yapılandırılmasında önce kurum kimliğine uygun olarak “MEF Kütüphane Sanal Asistanı” ismi belirlenmiştir. Ayrıca kullanılacak YZ modelinin sürümü ve yanıt üretme ayarları detaylı şekilde belirlenmiştir. Metin tabanlı ve ekli doküman kaynaklı soruların işlenebilmesi amacıyla sisteme ilgili dosyalar (kütüphane katalog verileri, sık sorulan sorular derlemesi, yönerge vb.) yüklenmiştir. Bu dosyalar üzerinden yapılacak bilgi sorgulamalarında, asistanın güncel ve doğru veriyi kullanıcıya sunabilmesi için dosya seçme ve önceliklendirme kuralları tanımlanmıştır. Ayrıca, asistanın yanıt formatı belirlenerek sorulara cevap verirken davranış biçimleri tanımlanmıştır.

Asistan ile araştırmacıların etkileşim kalitesini yükseltmek üzere aşağıdaki seçenekler için kapsamlı bir talimat seti hazırlanmıştır:

- Amaç Tanımı: Asistanın kütüphane hizmetleri, ödünç kitap sorgulama, makale erişimi, referans yönetimi ve kütüphane politikaları hakkında bilgi verme rolü açıkça tanımlanmıştır.
- Hedef Kitle ve Üslup: Kullanıcı (lisans, yüksek lisans, doktora öğrencileri, akademik ve idari personel) profilleri için uygun resmi/tam cümleli, kavram odaklı yanıt biçimi belirlenmiştir.

- Yanıt Kuralları: Kaynak gösterme gerekliliği, teknik terimlerin açıklanması, gerektiğinde örnek vaka sunulması ve adım adım yol gösterme prensipleri detaylı bir şekilde yazılmıştır.
- Hata Yönetimi: Bilgi eksikliği durumunda nasıl geri bildirimde bulunulacağı, belirsizlik içeren sorularda hangi ek soruların sorulacağı ve kullanıcıyı yönlendirme yöntemleri tanımlanmıştır.

Talimatlar Aşaması ve İçerik Örnekleri

Asistanın bilgi kaynağı olarak kullanacağı yanıt seti, yöneltilen sorulara sistematik ve tutarlı biçimde yanıt üretebilme kapasitesine sahip olacak şekilde yapılandırılmıştır. Bu yapı, bilgi bütünlüğünü koruyarak kullanıcı etkileşimlerinde yüksek doğruluk ve güvenilirlik sağlamaya odaklanmıştır. Aşağıda talimat setine dâhil edilen bazı temel örnekler sunulmuştur:

Üyelik ve Ödünç Yayın Kullanımı

- MEF Üniversitesi mensupları kütüphanenin doğal üyesidir. MEF Üniversitesi kimlik kartı bulunan kullanıcılar kütüphaneden ödünç yayın alabilirler (MEF Üniversitesi Kütüphanesi, 2024a).

MEF Kütüphane Çalışma Saatleri

- Pazartesi - Cuma: 08.30 – 22:00
- Cumartesi - Pazar: 10.00 - 18.00
- Final Proje dönemlerinde haftanın dört günü 7/24

Mezun Yararlanma Koşulları

- Telif hakları kanunu gereğince alınan Üniversite Senato kararı ile öğrencinin talebi üzerine mezuniyet tarihi itibarıyla (90) doksan gün MEF Kütüphane Online - Kaynaklarından Kampüs Dışı Erişim ile yararlanmaya devam etmeleri olanaklıdır.
- MEF Kütüphane, Kampüste C Blok 3. katta yer almaktadır.

Bu talimat örnekleri, asistanın kullanıcı taleplerine anında ve eksiksiz yanıt sunabilmesi için temel bilgi altyapısını oluşturmaktadır.

Model Ayarları, Dosya Yükleme ve Yanıt Parametreleri

Talimat metnlerinin tanımlanmasının ardından, sanal asistanın YZ modeline ilişkin konfigürasyon süreci tamamlanmıştır. Model sürümü ve işlem kapasitesi, yanıt kalitesi ile işlem hızı arasında optimal dengeyi sağlayacak biçimde belirlenmiştir. Devamında “Dosya Yükle” özelliği aktifleştirilmiş ve kütüphane katalog listesi sisteme yüklenmiştir. Bu sayede kitap tarama ve kaynak arama işlemleri, otomatik indeksleme yoluyla

hızlandırılmıştır. Son olarak değerler (temperature) parametresi optimize edilmiştir; Chatbot'larda (özellikle dil modellerinde) değerler parametresi, model yanıtlarının olasılıklarını (veya yaratıcılığını) kontrol eden bir ayardır. Bu optimizasyon ile daha doğru, tutarlı ve tekrarlanabilir sonuçlar üretmektedir.



Görsel 5. “Temperature” (Değerler) Ayar Görseli

Kaynak: (https://platform.openai.com/assistants/asst_8xq9u1N6Z6L5jPL0SgDRCRkZ)

Sanal Asistanın Kodlama Süreci

Sanal Asistanın kodlama sürecinde, ön yüz tarafında React tabanlı bir mimari kullanılarak OpenAI’nin API’sine entegrasyon sağlanmıştır. Bu mimari, kullanıcı ile doğal dil düzeyinde etkileşim kurabilen bir sohbet arayüzü oluşturmak üzere tasarlanmıştır. React’in bileşen tabanlı yapısı ve sanal DOM (Document Object Model) optimizasyonu sayesinde performanslı ve modüler bir yapı elde edilmiştir (React, 2024). API çağrılarının asenkron yönetimi için Axios gibi HTTP istemci kütüphaneleri kullanılmış, useState ve useEffect kancalarıyla hem kullanıcı girdisi hem de API yanıtları “state” durum içinde kontrollü biçimde saklanıp güncellenmiştir. Bu sayede, mesaj gönderme ve alma iş akışları kesintisiz ve tutarlı bir şekilde işlenmiştir (Naithani, 2023).

Kütüphane katalog verilerine yönelik arama-dönüş işlemlerinde ise RESTful servisler kullanılmıştır. GET istekleriyle envanter sorgulamaları gerçekleştirilirken, gelen JSON yanıtları React bileşenlerine uygun veri modellerine dönüştürülmüştür. Filtreleme ve sıralama algoritmaları, kullanıcı deneyimini iyileştirmek amacıyla ön yüzde optimize edilmiştir. POST istekleri kullanılarak kullanıcı etkileşimlerine dair kayıtlar oluşturulmuştur. Süreç içerisinde hata yakalama (error handling) ve tekrar deneme (retry) mekanizmaları entegre edilerek, API anahtarı yönetimi güvenlik odaklı bir yaklaşımla yapılandırılmıştır (Aleedy vd., 2019; OpenAI, 2024). Tüm bu adımlar, MEF Kütüphane’nin dijital hizmet vizyonu ile örtüşen, ölçeklenebilir ve sürdürülebilir bir sanal asistan altyapısının oluşturulmasına imkân tanımıştır.

GitHub ile Web Entegrasyonu

Dağıtık sürüm kontrol sistemi üzerine inşa edilen GitHub, kullanıcıların kaynak kodlarını depo (repository) adı verilen ortamlarda güvenli barındırmalarına, farklı sürümler arasında geçiş yapmalarına ve dallanma (branching) ile birleştirme (merge) mekanizmaları aracılığıyla eşzamanlı geliştirme süreçlerini yönetmelerine imkân tanır. 2018 yılında Microsoft tarafından satın alınmasının ardından, platformun benimsenme oranı hızla artmış ve Mayıs 2025 itibarıyla yaklaşık 150 milyon kayıtlı geliştirici ile 420 milyonu aşkın proje barındırır hâle gelmiştir (GitHub, 2024).

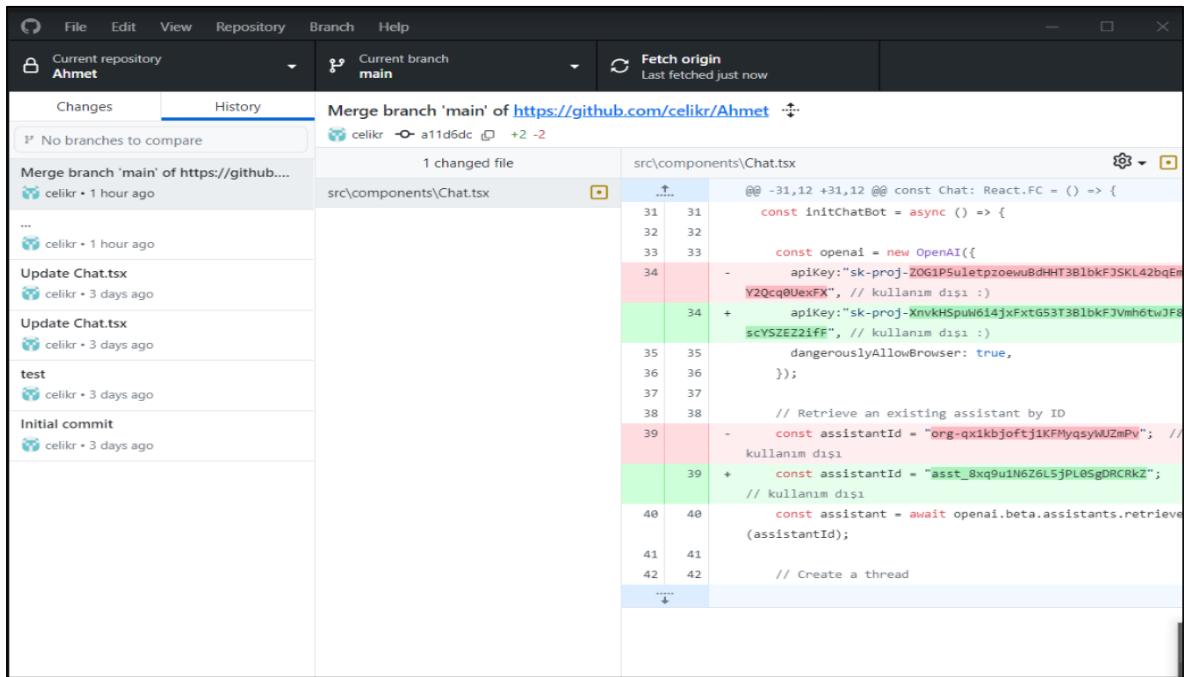
GitHub'ın sunduğu başlıca işlevler arasında hata ve özellik takibi (issue tracking), kod inceleme (pull request) süreçleri, kod parçacığı paylaşımı (Gist) ve statik web sitesi barındırma (GitHub Pages) yer alır (GitHub, 2024). Ayrıca, GitHub Actions ile sürekli entegrasyon ve sürekli dağıtım (CI/CD) iş akışları otomatikleştirilerek derleme, test ve dağıtım adımlarını kolaylaştırılmaktadır. Platform, herkese açık (public) olmakla birlikte özel (private) depo seçeneği sunmaktadır. GitHub sürüm kontrolünün yanı sıra hata takibi, kod incelemesi ve proje yönetimi araçlarıyla da yazılım geliştirme yaşam döngüsünü bütüncül biçimde desteklemektedir (GitHub, 2024).

GitHub Kullanım Adımları

Sanal asistanın geliştirme sürecinde, sürüm kontrolü, kodlama altyapısının oluşturulması, gizli verilerin güvenli yönetimi ve web ortamına yayımlama işlemlerini kapsayan GitHub tabanlı bütünleşik bir yazılım geliştirme ve dağıtım mimarisi yapılandırılmıştır. Bu süreçteki adımlar aşağıda verilmiştir:

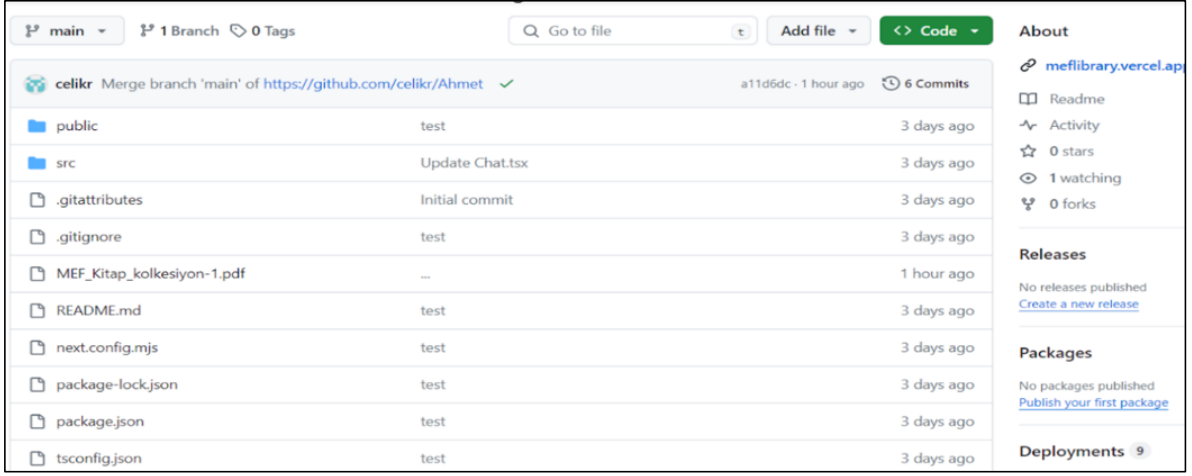
- Hesap Oluşturma: GitHub web sayfası (<https://github.com/>) üzerinden kurum e-posta adresiyle kullanıcı kaydı tamamlanmıştır.
- GitHub Desktop Kurulumu: Yerel geliştirme ortamının sürüm kontrolü ile entegre çalışabilmesi amacıyla GitHub Desktop uygulaması indirilmiş ve bilgisayara kurulumu gerçekleştirilmiştir.
- Kodlama ve Şablon Hazırlığı: DZone'da yayımlanan örnek uygulama şablonlarından yararlanılarak MEF Kütüphane'ye özgü kodlama altyapısı oluşturulmuş; proje dizini içinde temel kaynak dosyaları tanımlanmıştır.
- Otomatik Yapılandırma Dosyalarının Oluşturulması: Platform, yapılandırma dosyalarını otomatik olarak üretmiş; sürüm kontrolü için gerekli meta veriler tanımlanmıştır.
- Gizli Verilerin Eşleştirilmesi: API anahtarları, kurum kimlik numarası ve organizasyon ID'si gibi hassas bilgiler, yerel ortamda çalışan bir betik aracılığıyla ilgili değişken dosyasına aktarılmış ve genel depo erişiminden izole edilmiştir.

- İlk Sürüm ve Senkronizasyon: Oluşturulan tüm proje dosyaları yerel GitHub Desktop deposuna eklenip sürüm (commit) işlemi yapılarak gönder (push) komutuyla web ortamındaki merkezi depoya gönderilerek senkronizasyon sağlanmıştır.
- Depo Düzenlemesi ve Sınıflandırma: GitHub web arayüzünde, projenin dokümantasyonuna yönelik içerikler dokümanlar (documents) kategorisi altında gruplanmış; Yönetim panelinden dosya eşleştirme ve sürüm etiketleme işlemleri tamamlanmıştır.
- GitHub Pages ile 30 Günlük Deneme Süreci: Ayarlar (settings), sayfalar (pages) menüsü üzerinden statik web sitesi yayımlama özelliği etkinleştirilmiştir.



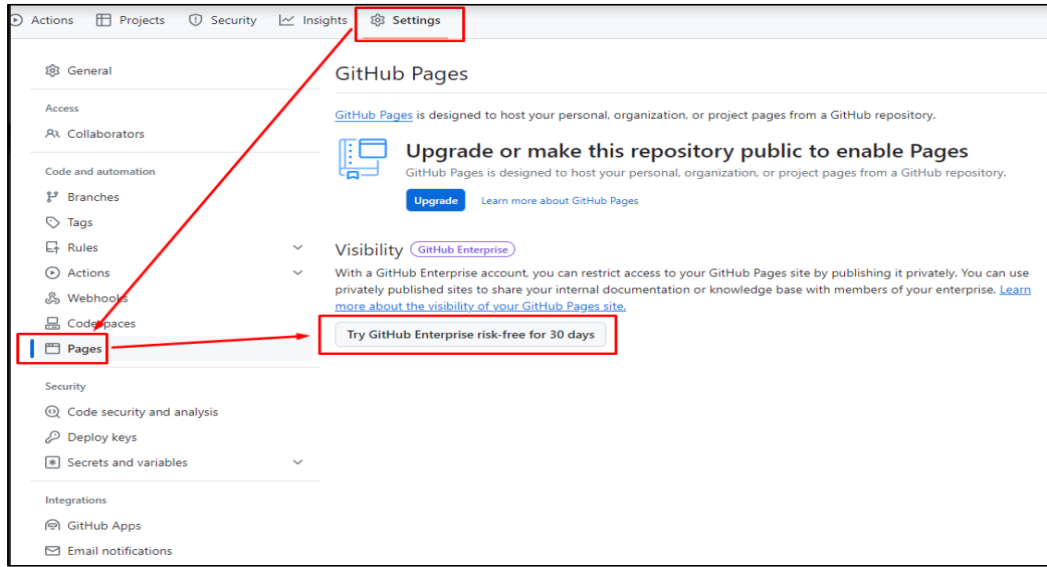
Görsel 6. GitHub Desktop Arayüz Görseli

Kaynak: (<https://github.com/dashboard>)



Görsel 7. GitHub- Dashboard Üzerinden Eşlenmiş Dosyalar Görseli

Kaynak: (https://github.com/dieistowake/chatbot_project/tree/main)



Görsel 8. Github Platformu 30 Günlük Deneme Süreci Görseli

Kaynak: (<https://github.com/dashboard/settings>)

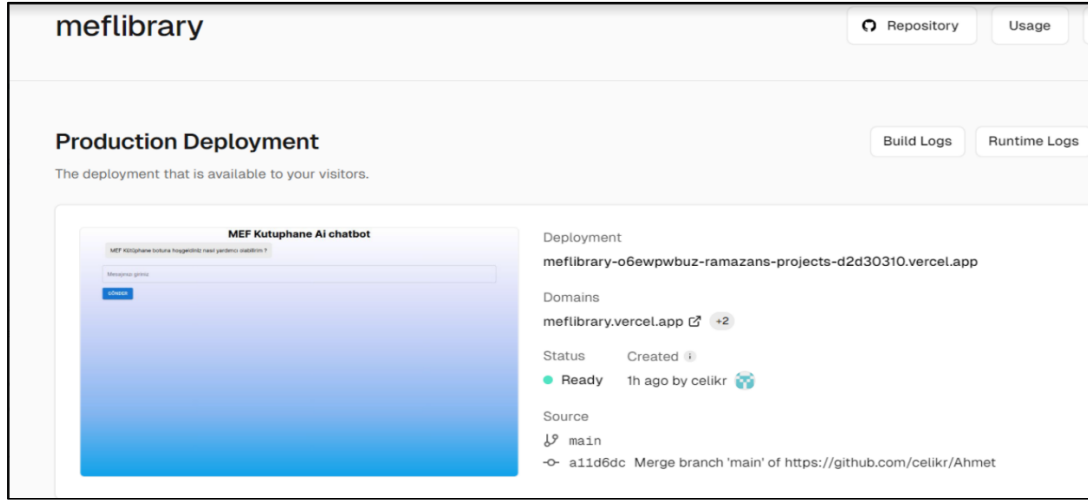
GitHub üzerinden yapılan kullanımında beklenen verim alınamamıştır. Kullanım sırasında proje dizininde yer alan API anahtarına ilişkin güvenlik uyarısı alınmıştır ve API anahtarı GitHub tarafından bloke edilmiştir. Bu nedenle GitHub ile uyumlu çalışan Vercel entegrasyonu kullanılarak proje çalışmalarına devam edilmiştir. Bu yöntemle, proje dosyaları doğrudan Vercel platformuna aktarılarak hem iş akışları korunmuş hem de hassas verilerin korunması sağlanmıştır.

Vercel Nedir ve Ne İşe Yarar?

Vercel, modern web uygulamalarının geliştirilmesi, test edilmesi ve yayınlanması için kullanılan bulut tabanlı bir platformdur. Özellikle Next.js framework'ü ile güçlü bir entegrasyona sahip olan Vercel, geliştiricilere hızlı, güvenli ve ölçeklenebilir bir dağıtım süreci sunmaktadır (Vercel, 2024).

Temel Özellikleri

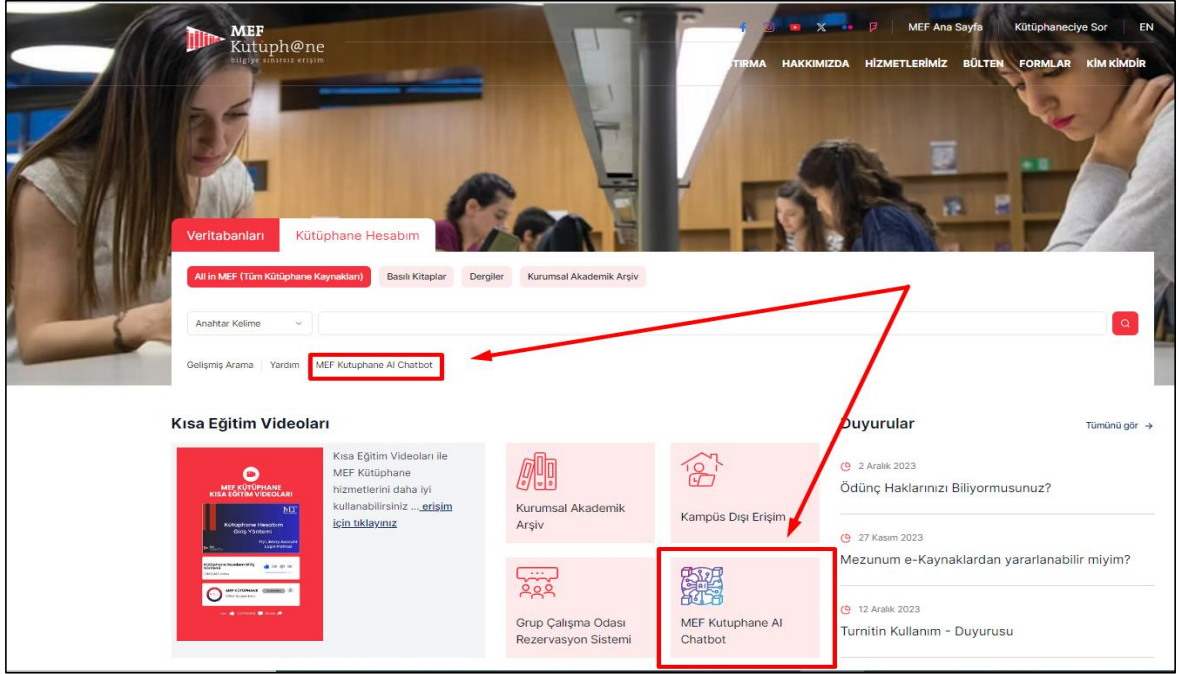
- Otomatik Yayınlama (Deploy): GitHub, GitLab veya Bitbucket gibi kaynak kontrol sistemleriyle entegre çalışarak her kod değişikliğinde otomatik olarak yeni bir sürüm yayınlar.
- Ön izleme Linkleri: Her geliştirme aşaması için ayrı bir ön izleme bağlantısı oluşturur, bu sayede ekip üyeleri yapılan değişiklikleri canlı olarak görebilir.
- Sunucusuz Fonksiyonlar: Arka planda çalışan (backend) işlemleri için sunucusuz fonksiyonlar tanımlanabilir. Bu, API oluşturma gibi işlemleri kolaylaştırır.
- Global CDN: İçeriği dünya çapında hızlı bir şekilde sunmak için içerik dağıtım ağı (CDN) kullanır.
- Performans ve Ölçeklenebilirlik: Trafik arttığında otomatik olarak ölçeklenir, böylece uygulamanın performansı korunur.



Görsel 9. GitHub platformu ve Vercel Entegrasyonu Sonucu MEF Kütüphane AI Chatbot Görseli

Kaynak: (<https://vercel.com/ramazans-projects-d2d30310/meflibrary>, <https://meflibrary.vercel.app/>)

Tüm bu işlem süreçleri sanal asistanın işlevsel ve kesintisiz biçimde hizmet verebilir hale gelmesini mümkündür. Sanal Asistan beş MEF Kütüphane çalışanı ve sekiz asistan öğrenci tarafından bir ay süre ile test edildikten sonra MEF Kütüphane web sayfası ile entegre edilmiş ve kullanıcıların hizmetine sunulmuştur (MEF Üniversitesi Kütüphanesi, 2024b).



Görsel 10. MEF Kütüphane Web Sayfası Sanal Asistan Görseli

Kaynak: (<https://library.mef.edu.tr>)



Görsel 11. MEF Kütüphane Sanal Asistan Uygulama Örneği Görseli

Kaynak: (<https://meflibrary.vercel.app>)

SONUÇ

MEF Üniversitesi Kütüphanesi tarafından hayata geçirilen YZ tabanlı sanal asistan projesi, kütüphanecilik hizmetlerinin dijital dönüşüm vizyonu çerçevesinde yeniden yapılandırılmasında öncü bir model sunmaktadır. Projenin temel motivasyonu, kütüphane hizmetlerinin YZ desteği ile 7/24 sunulmasıdır.

Sanal asistanın mimarisi ve geliştirme sürecinde doğal dil işleme yetkinliği yüksek olan OpenAI alt yapısı kullanılmıştır. Geliştirme sürecinde ise bileşen tabanlı yapısı, yeniden kullanılabilir arayüz elemanları sunması ve sanal DOM mimarisi sayesinde performanslı kullanıcı deneyimi sağlaması nedeni ile React tabanlı bir ön yüz mimarisi tercih edilmiştir. Kod yönetimi ve sürüm kontrolü için GitHub, hızlı ve ölçeklenebilir dağıtım için ise Vercel platformları tercih edilmiştir. Sistemin başarısı, teknolojik alt yapıyla birlikte, kullanıcı beklentileri çerçevesinden yapılandırılan talimat setlerine, kütüphane katalog verilerine ve sıkça sorulan soruların sisteme entegre edilmesine dayanmaktadır. Özellikle değer (temperature) ve olasılık seçeneklerinin optimize edilmesi, sanal asistanın daha tutarlı ve doğru yanıtlar üretmesini sağlamıştır.

Sanal asistan zaman ve mekân kısıtlarını ortadan kaldırarak hizmet sürekliliği sunmaktadır. Sanal asistan ile ilgili olarak kullanıcılardan öneriler ve geri bildirimler alınmıştır. Bu bildirimler projenin amacına uygun olarak iyileştirilmesi ve geliştirilmesinde motivasyon kaynağı olmuştur. Rutin ve tekrar eden soruların sistem tarafından yanıtlanması, kütüphane personelinin rutin işlerini azaltmış, onların daha nitelikli görev ve projelere odaklanmasına olanak tanımıştır. Ayrıca kullanıcıların danışma soruları, kampüs dışı erişim işlemleri gibi hizmetler için YZ desteği ile cevap alması süreçlerin hızlanmasına, doğal olarak kullanıcı memnuniyetinin artmasını sağlamıştır.

Sürdürülebilirlik ve gelecek vizyonunda proje başlangıçta ücretli API modelleri ile kurgulanmış olsa da sürdürülebilirlik ilkeleri doğrultusunda maliyet etkinliği sağlamak adına açık kaynaklı ve ücretsiz alternatiflere yönelme stratejisi benimsenmiştir. Gelecekte sanal asistanın derinleştirilmiş doğal dil işleme yetkinlikleri ve makine öğrenmesi tabanlı öneri mekanizmaları ile geliştirilmesi planlanmaktadır.

Sonuç olarak, MEF Kütüphane sanal asistanı, sadece teknik bir araç değil, aynı zamanda akademik kütüphaneler için uygulanabilir, ölçeklenebilir ve kullanıcı odaklı bir dijital hizmet modeli teşkil etmektedir. Bu çalışma, YZ'nın kütüphane ekosistemine entegrasyonunun kurumsal verimliliği ve eğitimde mükemmellik vizyonunu nasıl desteklediğini açıkça ortaya koymaktadır.

KAYNAKÇA

- Aleedy, M., Shaiba, H. ve Bezbradica, M. (2019). Generating and analyzing chatbot responses using natural language processing. *International Journal of Advanced Computer Science and Applications* 10(9). <http://dx.doi.org/10.14569/IJACSA.2019.0100910>
- Chaves, A. P., Gerosa, M. A., Chaves, A. C. ve Gerosa, M. A. (2020). How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction* 37(8). <https://doi.org/10.1080/10447318.2020.1841438>
- Çelik, S. ve Çuhadar, S. (2025). Yapay zekânın bilimsel araştırmalarda kullanımı ve etik. Ö. E. Akgün (Ed.), *Eğitimde Yapay Zeka Teknolojileri* içinde (ss. 397-435). İdealkültür Yayınları. <https://doi.org/10.36657/ihu.yz.2025.0>
- GitHub. (2024). *GitHub*. Wikipedia. <https://en.wikipedia.org/wiki/GitHub>
- MEF Üniversitesi Kütüphanesi. (2024a). *Sık sorulan sorular*. <https://library.mef.edu.tr/tr/sss>
- MEF Üniversitesi Kütüphanesi. (2024b). *Vercel platformu*. <https://meflibrary.vercel.app/>
- Naithani, A. (2023). *ReactJS with ChatGPT: Building conversational AI your web apps*. DZone. <https://dzone.com/articles/reactjs-with-chatgpt-building-conversational-ai-in>
- OpenAI. (2024). *ChatGPT API reference*. <https://platform.openai.com/docs/api-reference/chat>
- React. (2024). *React – A JavaScript library for building user interfaces*. <https://reactjs.org/docs/getting-started.html>