



Proje Raporu: Araba Uzmanı ChatBot Geliştirme Süreci

Konu: Yapay Zeka Destekli Araba Teknik Asistanı

Github: Araba Uzmanı ChatBot

Hazırlayan: Ahmet Yeşil

1. Proje Özeti ve Kapsamı

Bu proje, kullanıcıların araçlarıyla ilgili yaşadıkları teknik sorunlara (motor, fren, elektrik vb.) çözüm önerileri sunan, yapay zeka tabanlı bir chatbot geliştirme sürecini kapsamaktadır. Sistem, **LangChain** mimarisi ve **Google Gemini API** kullanılarak inşa edilmiştir. Projenin temel amacı, karmaşık teknik terimleri basitleştirmek ve kullanıcıya adım adım rehberlik etmek ve güvenli sürüş tavsiyeleri vermektedir.

2. Chatbot Akış Tasarımı ve Niyet (Intent) Yönetimi

Ödev gereksinimlerine uygun olarak chatbot, aşağıdaki niyet türlerini başarıyla yönetmektedir:

- Selamlama & Vedalaşma:** Kullanıcıyı dostane bir dille karşılar ve görüşmeyi sonlandırır.
- Reddetme (Filtreleme):** Araba dışı konular (sağlık, yemek, yazılım, siyaset vb.) sorulduğunda, sistem `BLOCKED_KEYWORDS` listesini kullanarak bu talepleri kibarca reddeder.
- Konuya Özgü Senaryolar:** Motor arızaları, fren sistemleri, elektrik/akü problemleri, klima ve şanzıman gibi spesifik teknik konularda uzmanlaşmıştır.

3. Veri Seti ve RAG (Retrieval-Augmented Generation) Yapısı

Projede sabit bir veri setinden ziyade, dinamik bir **RAG (Retrieval-Augmented Generation)** mimarisi tercih edilmiştir.

- Veri Formatları:** Sistem; `.pdf`, `.docx` ve `.xlsx` formatındaki teknik belgeleri okuyabilmektedir.
- İşleme Süreci:** `DocumentProcessor` sınıfı ile dokümanlar küçük parçalara (chunks) bölünür ve `SimpleDocumentStore` içinde saklanır.
- Bağlamsal Arama:** Kullanıcı bir soru sorduğunda, sistem dokümanlar içinde anahtar kelime araması yaparak ilgili metni bulur ve bunu LLM'e (Gemini) "bağlam" (context) olarak iletir.

4. LLM Model Seçimi ve Teknolojiler

Projede iki farklı modelin değerlendirilmesi hedeflenmiş, uygulama aşamasında güncel ve hızlı olan **Gemini 2.5 Flash** modeli temel alınmıştır.

- **Kullanılan Araçlar:** LangChain (LLM yönetimi), Streamlit (Arayüz), python-dotenv (Güvenlik).
- **Neden Gemini?:** Geniş bağlam penceresi ve Türkçe dil desteği kapasitesinin yüksek olması nedeniyle seçilmiştir.

6. Uygulama Arayüzü (UI)

Kullanıcı arayüzü **Streamlit** kütüphanesi ile modern bir "Koyu Tema" (Dark Mode) tasarımına sahip olacak şekilde geliştirilmiştir.

- **Sidebar (Yan Panel):** Geçmiş sohbetlerin yönetimi ve yeni doküman yükleme alanı.
- **Hızlı Erişim Butonları:** 6 farklı arıza kategorisi için hazır soru şablonları.
- **Sohbet Geçmişi:** Kullanıcıların önceki konuşmalarına dönebilmesi için `session_state` yönetimi.