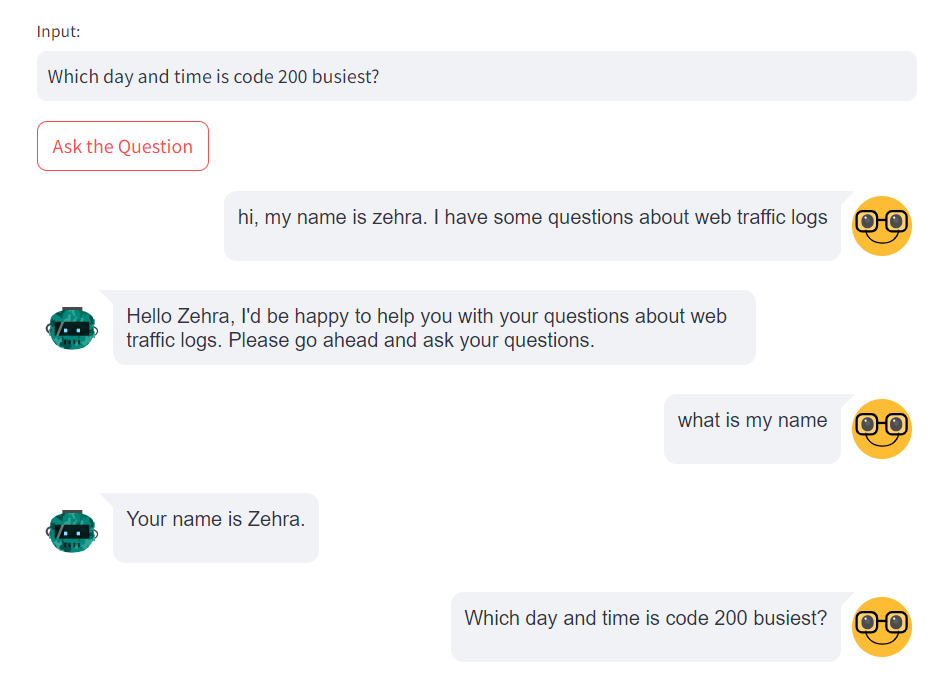
# **Proje Raporu: Sistem Entegrasyonu ve Performans Değerlendirmesi**

## **1. Giriş**

### **1.1 Proje Tanımı**

Bu proje Web Traffic Loglarını işleyip LLM modeli oluşturulmuştur.

Kullanıcıdan gelen sorulara yanıt verebilen bir sistem tasarlanmıştır.



### **1.2 Amaç ve Hedefler**

* Kullanıcı sorularını alarak vektör veri tabanında arama yapmak.
* En uygun log kayıtlarını bulmak.
* Bu kayıtları kullanarak dil modelinden bir yanıt oluşturmak.
* Sistemin doğruluğunu ve performansını değerlendirmek.

## **Sistem Entegrasyonu**

### **2.1 Veri İşleme ve Yükleme**

#### **2.1.1 Log Verilerinin Temizlenmesi**

Log verileri, ön işleme adımlarından geçirilerek temizlenmiştir. Verinin uygun bir biçime getirilmesi için gerekli adımlar atılmıştır.

#### **2.1.2 Vektör Veri Tabanı Oluşturulması**

Temizlenmiş log verileri kullanılarak vektör veri tabanı oluşturulmuştur. Bu aşamada Pinecone veri tabanı kullanılmıştır.

### **2.2 Model Entegrasyonu**

#### **2.2.1 RAG Modelinin Entegrasyonu**

Retrieval-Augmented Generation (RAG) modeli entegre edilmiştir. Bunun için Lanchain hub’tan prompt = hub.pull("rlm/rag-prompt") promptu kullanılmıştır.

**2.2.2 Memory yönetimi**

Hafıza yönetimi için streamlit’in chat history özelliğinden destek alındı ve buffer memory ile hafıza tutuldu.

**2.2.3 Kullanıcı Sorguları ile Yanıt Üretimi**

Kullanıcıdan gelen sorulara yanıt üretimi için sistem tasarlanmıştır. Bunun için get\_chatmodel\_response() adında bir fonksiyon oluşturulmuştur. Bu fonksiyonda:

* Kullanıcı sorusunun alınması.
* Vektör veri tabanında arama yapılarak en uygun log kayıtlarının bulunması.
* Bu kayıtlar kullanılarak dil modelinden yanıtın görüntülenmesi

**2.2.4 Kullanıcı Sorguları ile Yanıt Üretimi**

Deploy işlemleri için streamlit arayüzü oluşturulmuştur. Bir akış şeklinde robot ve human konuşması şeklinde bir konuşma akışı hazırlanmıştır.

### **2.3 Performans Testleri**

#### **2.3.1 Test Senaryoları ve Yöntemler**

Geliştirilen model sorular sorularak test edilmiştir, Hem hafızası hem de dokümanı ne kadar baz aldığı gözlemlenmiştir.

## **3. Performans Değerlendirmesi**

### **3.1 Sistem Performansı**

#### **3.1.1 Doğruluk ve Kalite**

Sistemin verdiği yanıtların doğruluğu ve kalitesi değerlendirilmesi yapılmıştır. Temperature değeri 0’a yaklaştığında cevaplarının doğruluğu daha’da artmıştır ve bunda gpt-4 kullanmanın etkisi de oldukça çoktur.

**3.1.2 Performans Metriği**

Yanıt süresini ölçmek için measure\_response\_time() şeklinde bir fonksiyon vardır bu da model cevap verdikçe altına ne kadar sürdüğünü gösterir:şu şekilde.

### **3.2 İyileştirme Önerileri**

#### **3.2.1 Kalite Artırma Yöntemleri**

Sistemin cevaplarının kalitesini artırmak için verilen dokumantasyon daha yalın hale getirilebilir. Ne kadar özgün bir dosya verirsek o kadar anlaşılır olur. Bunun yanında dokümantasyonu bölme işlemini doğru yerlerden yapmanında etkisi bi o denli fazladır.

#### **3.2.2 Performans İyileştirme**

Rag modelini doğru seçip, prompt kısmını daha da düzenlersek performansını arttırıp daha doğru yanıtlar vermesini sağlayabiliriz. Bunun yanı sıra doğru model, temperature seçimi de cevapların doğruluğunda etkilidir.

## **4. Sonuç**

### **4.1 Proje Sonuçları**

Projemiz istenilen şekilde çalışmaktadır.

Sorulara cevap verir, hafızada tutar sorulan soruları.

Modelin performansını arttırmanın yolları bulunmuştur, eksikleri anlaşılmıştır.