



FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği Bölümü

YMH310 - WEB TASARIMI VE PROGRAMLAMA

Dersi Proje Ön Raporu

Eğitmenler İçin Proje Yönetimi Uygulaması

Geliştiren

225541034 Mehmet Said HÜSEYİNOĞLU

Proje Yürütücüleri
Öğr.Gör. Eyüp ERÖZ

MART – 2025

İÇİNDEKİLER

1. Proje Özeti.....	3
2. Problem Tanımı.....	3
3. Çözüm Önerisi.....	3
3.1 Temel Özellikler.....	3
3.2 Teknik Altyapı.....	4
4. Fonksiyonel Programlama Uygulamaları.....	4
5. Kullanıcı Senaryoları.....	4
Senaryo 1: Becayış Eşleşmesi.....	4
Senaryo 2: Mevzuat Bilgisi Alma.....	5
6. İş Planı Ve Zaman Çizelgesi.....	5
7. Teknik Gereksinimler.....	5
8. Beklenen Faydalar.....	5
9. Riskler Ve Zorluklar.....	6
10. Kullanılması Planlanan Kaynaklar.....	6

1. Proje Özeti

Bu proje, öğretmenlerin projelerini yönetmelerini ve projelerdeki belgeleri incelemelerini sağlayan bir web uygulaması geliştirmeyi amaçlamaktadır. Uygulama, yüklenen PDF belgelerindeki metinleri yerel bir LLM modeli veya API kullanarak analiz eder. Kullanıcılar, belgelerdeki belirli bilgilerin bulunup bulunmadığını kolayca kontrol edebilir ve sonuçları veritabanına kaydedebilir.

2. Problem Tanımı

Eğitmenler, öğrencilerin sunduğu projeleri değerlendirirken belgeleri manuel olarak incelemekte ve belirli bilgilerin varlığını doğrulamak için zaman kaybetmektedir. Bu süreç, özellikle birden fazla belgeyi incelemeyi gerektiren durumlarda verimsiz hale gelmektedir. Eğitmenlerin daha hızlı ve etkili bir şekilde projeleri değerlendirebilmeleri için belgelerdeki belirli bilgilerin otomatik olarak analiz edilip sunulmasına ihtiyaç vardır.

3. Çözüm Önerisi

Bu proje, öğretmenlerin PDF formatındaki belgeleri daha hızlı ve verimli bir şekilde inceleyebilmesi için Node.js tabanlı bir web uygulaması geliştirmeyi hedeflemektedir. Proje, metin bazlı arama ve analiz işlemlerini yerel bir dil modeli kullanarak gerçekleştirecek, verileri MongoDB üzerinde depolayacaktır:

3.1 Temel Özellikler

- **PDF Yükleme ve Metin Dönüştürme:**
 - Kullanıcıların PDF dosyalarını yükleyip metin formatına dönüştürmelerini sağlar.
- **Akıllı Metin Analizi:**
 - Belgelerdeki belirli bilgileri (örneğin, proje amacı, teknik altyapı, iş paketi gibi) tespit eder ve özet halinde sunar
- **Bildiri Arama ve Doğrulama:**
 - Kullanıcıların belirli anahtar kelimeleri veya başlıkları arayarak ilgili içeriklerin varlığını kontrol etmesini sağlar.
- **Veri Depolama:**
 - İncelenen belgeler ve analiz sonuçları MongoDB veritabanında saklanır ve gerektiğinde tekrar erişilebilir.
- **API Entegrasyonu (Opsiyonel):**
 - API kullanarak harici dil modelleri ile analiz yapılabilir (örneğin, Hugging Face API, Cohere API).

3.2 Teknik Altyapı

- **Web Backend:** Node.js (Express) Veritabanı: MongoDB
- **Dil Modeli:** Yerel LLM (örneğin, GPT-Neo, LLaMA) veya API tabanlı modeller (Hugging Face, Cohere).
- **Metin İşleme:** pdf-parse veya benzeri kütüphanelerle PDF'den metin çıkarımı.
- **Frontend (Opsiyonel):** React.js veya basit HTML/CSS ile kullanıcı arayüzü.

4. Fonksiyonel Programlama Uygulamaları

Bu proje kapsamında fonksiyonel programlama prensiplerinden yararlanarak dil modeliyle etkileşim ve metin işleme süreçleri optimize edilecektir. Fonksiyonel programlama uygulamaları şunları içerecektir:

1. İmmutability (Değişmezlik):

- Veriler üzerinde değişiklik yapılmak yerine her işlemde yeni veri oluşturulacak. Örneğin, PDF dosyalarından çıkarılan metinler üzerinde yapılan tüm işlemler saf fonksiyonlar aracılığıyla gerçekleştirilecek.

2. Saf Fonksiyonlar (Pure Functions):

- PDF dosyalarını metin formatına dönüştürme, metin analizi yapma ve özetleme gibi işlemler saf fonksiyonlar kullanılarak yapılacaktır. Bu fonksiyonlar yalnızca girdilerine bağlı olacak ve dışsal durumları değiştirmeyecek.

3. Yüksek-Düzy Fonksiyonlar (Higher-Order Functions):

- Metin analiz fonksiyonları, harici API çağrılarını kapsayarak dinamik olarak işlevler oluşturulabilecek ve zincirleme işlemler uygulanabilecek.

4. Fonksiyonel Kompozisyon:

- PDF metinlerinin işlenmesi ve analiz edilmesi gibi birbirine bağlı işlemler, bir dizi fonksiyonun ardışık olarak uygulanmasıyla gerçekleştirilecek.

5. Lazy Evaluation (Tembel Değerlendirme):

- Büyük boyutlu PDF dosyalarında sadece gerekli kısımlar işlenecek. Metin analizi ve arama işlemleri, yalnızca kullanıcı istekte bulunduğu anda tetiklenecek.

6. Map, Filter ve Reduce Kullanımı:

- Metin üzerinde çeşitli işlemler (örneğin, anahtar kelime araması, belirli bölümlerin çıkarılması) map, filter ve reduce gibi fonksiyonlar kullanılarak gerçekleştirilecek.

7. Fonksiyonel Hatalar ve İstisna Yönetimi:

- Hatalar fonksiyonel yapılar ile ele alınacak ve kodun güvenilirliği artırılacak.

5. Kullanıcı Senaryoları

Senaryo 1: Dosya Yükleyerek Bilgi Arama

- Kullanıcı, platforma bir veya birden fazla PDF dosyası yükler.
- Kullanıcı, yüklenen dosyalar üzerinde bir anahtar kelime araması yapmak ister.
- Sistem, ilgili anahtar kelimenin bulunduğu tüm bölümleri hızlıca listeleterek kullanıcıya sunar.
- Kullanıcı, ilgili sonuçlardan birine tıklayarak detaylı içeriği görüntüler.

Senaryo 2: Belge Özeti Oluşturma

- Kullanıcı, bir PDF dosyasını platforma yükler.
- Kullanıcı, tüm belgeyi veya belirli bir bölümü özetlemek ister.
- Sistem, dil modelini kullanarak metni analiz eder ve kısa, anlamlı bir özet oluşturur.
- Kullanıcı, özetlenen metni kopyalayabilir veya kaydedebilir.

Senaryo 3: Sorulara Yanıt Bulma

- Kullanıcı, platforma bir PDF dosyası yükler.
- Kullanıcı, dosyanın içeriği hakkında spesifik bir soru yazar (örneğin, “Bu belgede yapay zeka uygulamalarından bahsediliyor mu?”).
- Sistem, ilgili bölümleri analiz ederek soruya uygun yanıt döndürür.

Senaryo 4: Çoklu Dosya Analizi

- Kullanıcı, birden fazla PDF dosyasını aynı anda yükler.
- Kullanıcı, dosyalar arasında karşılaştırmalı bir analiz yapmak ister (örneğin, benzer konuların tespiti).
- Sistem, dosyaları işleyerek benzer veya ilgili bölümleri gruplar halinde kullanıcıya sunar.

Senaryo 5: Kullanıcı Özelleştirmeleri

- Kullanıcı, arama veya analiz ayarlarını özelleştirmek ister (örneğin, sadece belirli başlıkları analiz etmek).
- Sistem, kullanıcının tercihlerini alır ve bu doğrultuda analiz işlemlerini gerçekleştirir.

6. İş Planı ve Zaman Çizelgesi

Aşama	Görevler	Süre	Tarih Aralığı
1. Planlama	- Proje gereksinimlerinin belirlenmesi	2 Hafta	17 Mart - 26 Mart
	- Teknik altyapı ve kullanılacak model ve sağlayıcı'ların seçimi		
	- Proje tasarımı ve kullanıcı senaryolarının hazırlanması		
2. Altyapı Kurulumu	- Python miniconda env. ve MongoDB kurulumlarının yapılması	2 Hafta	27 Mart - 12 Nisan
	- Model entegrasyonlarının sağlanması (Hugging Face models , ollama ve lm studio)		
	- PDF dosyalarını işleyebilecek dosya yükleme sistemi oluşturma		
3. Backend Geliştirme	- Dosya yükleme ve işleme işlemlerinin geliştirilmesi	3 Hafta	13 Nisan - 2 Mayıs
	- Metin analizi ve bilgi çıkarım fonksiyonlarının oluşturulması		
	- Veritabanı entegrasyonu ve yapılandırması		
4. Frontend Geliştirme	- Kullanıcı arayüzü tasarımı ve geliştirilmesi	2 Hafta	3 Mayıs - 24 Mayıs
	- Kullanıcı senaryolarının uygulanması		
	- API entegrasyonu ile arayüz bağlantısı		
5. Test ve Hata Ayıklama	- Fonksiyonların test edilmesi	2 Hafta	25 Mayıs - 10 Haziran
	- Hataların düzeltilmesi ve performans iyileştirmeleri		
	- Güvenlik ve uyumluluk testlerinin yapılması		
6. Dağıtım ve Sonuçlandırma	- Projenin sunucuya dağıtımı veya sunum dosyalarının hazırlanması	1 Hafta	11 Haziran - 20 Haziran
	- Belgelerin hazırlanması ve proje raporunun yazılması		

Toplam Süre: Yaklaşık 3 ay (17 Mart - 20 Haziran).

7. Teknik Gereksinimler

- Geliştirme Ortamı: Visual Studio Code, miniconda
- Programlama Dilleri ve Teknolojiler: Python, MongoDB, HTML, CSS, JavaScript
- Bulut Veritabanı: MongoDB Atlas
- Kullanılacak modeller: Deepseek, Ilma3, mistral nemo vs.
- Kullanılacak API'ler: Gemini, Openai, Claude
- Araçlar ve Kütüphaneler: Ollama ve Lm studio (sağlayıcılar için), Mongoose (MongoDB bağlantısı için), Multer (Dosya yükleme işlemleri için)
- İşletim Sistemi: Windows, MacOS veya Linux
- Test Araçları: Jest veya Mocha (Birim ve entegrasyon testleri için)

8. Beklenen Faydalar

- Kullanıcıların PDF dosyalarını hızlı ve etkili bir şekilde analiz edebilmesi.
- Belgelerden anlamlı bilgiler çıkararak raporlama ve analiz süreçlerini hızlandırma.
- Yapay zeka modellerinin gücünden faydalanarak kullanıcıların verimliliğini artırma.
- Geniş kapsamlı doküman incelemelerinde zaman tasarrufu sağlama.
- Kullanıcıların projeleriyle ilgili bilgilere kolay erişim sağlanması ve sonuçları kaydedip inceleyebilmesi.

9. Riskler ve Zorluklar

- Localde model çalıştırma için gerekli donanımların pahalı olması veya ulaşımının zor olması.
- API kullanımının maliyetleri (Hugging Face veya Cohere gibi platformların fiyatlandırma modelleri).
- Kullanıcı verilerinin gizliliği ve güvenliğinin sağlanması.
- Yüksek boyutlu PDF dosyalarının işlenmesinde performans sorunları.
- Metin analizi sonuçlarının doğruluğunu sağlamak.
- Kullanıcı arayüzü ile backend arasındaki entegrasyonun düzgün çalışmaması.
- Projenin geliştirilme süresinin sınırlı olması.

10. Kullanılması Planlanan Kaynaklar

- Python ve miniconda dokümantasyonu.
- MongoDB ve Mongoose dokümantasyonu.
- Ollama ve Lm studio sdk ları
- Hugging Face API dokümantasyonu.
- PDF işleme kütüphaneleri ve dokümantasyonları.
- JavaScript, HTML, CSS gibi web teknolojileri üzerine kaynaklar.
- API istekleri ve entegrasyonları için Axios gibi yardımcı kütüphaneler.

Kaynaklar:

- Github: <https://github.com/SIYAKS-ARES/proje-yonetim-sistemi>
- LinkedIn: <https://www.linkedin.com/in/mehmet-said-huseyinoglu>