**Web Text Combiner**

Berk Sunduri

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

berksunduri@gmail.com

**1-)Projenin Tanımı**

Bu kısım sadece projenin açıklamasını okuyup edindiğim ön bilgiye göre yazılmıştır.

Bize verilen pdf dosyasında projenin amacını öğrendim. Verilen isterleri dikkatlice okudum

Projede bizden istenen şeyin tekst birleştirmeye yarayan bir web applikasyonu olduğunu anladım.

**1.1-)Problem Tanımı**

Bu kısımda bizden yapmamız istenilenler içermektedir.

Birinci adımda bizden en az iki tane metin giriş butonları bulundurmamız istendi.

İkinci ister olarak girilen metinleri birleştirme butonu bulundurmamız istendi.

Üçüncü aşamada ise bizden girilen metinleri kaydetmemiz için bir buton istendi.

Dördüncü ister olarak bizden veriden elde edilen en uzun dizi listelerinin tümünü bir liste halinde tutmamız istendi

Son isterde ise bu metinleri birleştirmek için ne kadar süre geçtiğini hesaplanmamız istendi.

**2-)Yapılan Araştırmalar ve Karşılaşılan Sıkıntılar**

Bu kısım proje öncesi ve sonrası araştırmaları ve de projenin yapım aşamasındaki sıkıntıları ve çözümlerini içermektedir.

İlk karşılaştığım sorun hangi programlama dilini kullanacağım olduğuydu. Web Framework olarak Python Flask seçtim. NoSQL olarak MongoDB ve en sonda algoritmayı çalıştırmak için Java kodlama dilini seçtim.

Daha sonrasında back-end’i oluşturmak beni zorladı. İstenilen kadar input eklemek beni en çok zorlayan kısım oldu. Stackoverflow’dan yardım alarak bu zorluğu aştım

**2.1-Proje Sırasında Yararlanılan Teknolojiler**

Projeyi Python dili kullanarak PyCharm IDE’sinde Flask kullanarak yazdım. NoSQL olarak MongoDB kullandım. Java’yı yazmak içinde IntelliJ IDEA kullandım.

Programı yaparken Python’un ve Java’nın birkaç kütüphanesinden yararlandım.

**3-)Tasarım**

**3.1-Akış Diyagramı**

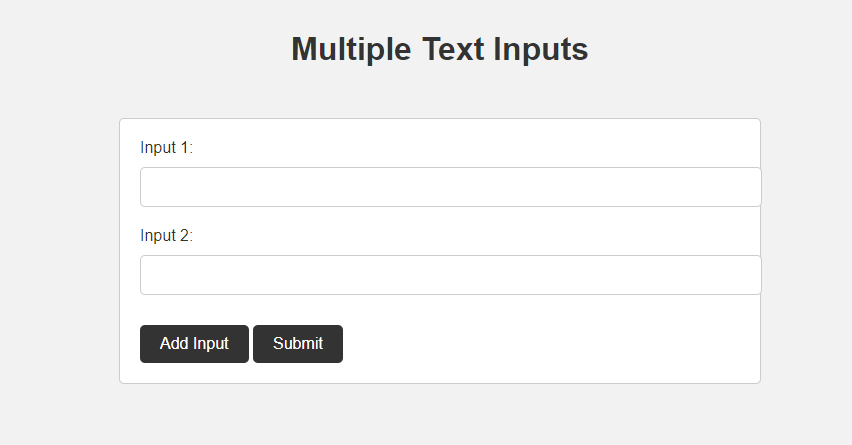
Kısım ektedir.(1)

**4-)Genel Yapı**

**4.1-Kullanıcı Kısmı**

Program çalıştığında karşımıza kullanıcının işlem yapmasına yarayan bir arayüz çıkmaktadır. Kullanıcı verilen inputları ve butonları kullanarak işlemler yapabilir.

**4.2-Kod Kısmı**

Kod kısmına baktığımızda ise kod tamamıyla fonksiyonlar yazılarak geliştirilmiştir.

Kod kısmında karşımıza birkaç flask fonksiyonu çıkacaktır.

Bunlar arayüzü oluşturmak için ve arkaplanda çalışacak işlemleri çalıştırmakta işimize yarar.

Algoritma şu şekilde çalışıyor:

Önce inputların boş olup olmadığını veya girilen String’in uzunluğunun sıfır olup olmadığını kontrol ediyor. Eğer öyleyse, algoritma bize boş bir dize döndürüyor.

Eğer inputta en az bir öge varsa ilk öge sonuç adlı bir değişkene atanıyor.

Ardından, algoritma, birinci indexten başlayarak geri kalan öğelerin hepsini test eder.

Yazılar dizisinde ki her öge için algoritma inputlar arasında ki en uzun örtüşmeyi arıyor.

Eğer bir örtüşme bulunursa algoritma bu geçerli öğenin örtüşmeyen kısmını sonuç adlı dizeye ekler.

Eğer örtüşme bulunmazsa geçerli öğenin tamamı sonuç dizesine eklenir.

Tüm elemanlar işlendikten sonra yöntemin çıktısı olarak nihai sonuç dizesi döndürülür.

Bu algoritma, bitişik dizeler arasındaki ortak örtüşmeleri bularak ve bunları birleştirerek texts dizisindeki dizeleri birleştirir, böylece gereksiz metin olmadan tüm orijinal dizeleri içeren tek bir birleştirilmiş dize elde edilir. Algoritmanın time complexitysi O(n\*m^2)’dir. Burda n girdinin uzunluğu ve m girdi dizisinde ki kelimelerin averaj uzunluğudur. Büyüme quadratictir eğer girdi çok uzunsa bekleme süresi uzayabilir.

**5-)Kütüphaneler**

Bu kısımda projeye include ettiğim kütüphaneler bulunmakta:

1-)flask

2-)Py4J

3-)PyMongo

**7-)Referanslar**

1-)w3schools html,css,js

”https://www.w3schools.com/”

2-) PDFMiner Documentation

” https://www.py4j.org/contents.html “

3-) Stack Over Flow ”[https://www.](https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm/)stackoverflow.com”

4-)Flask Documentation

“flask.palletsprojects.com”

5-)MongoDB Documentation

”https://www.mongodb.com/docs/”