

YAZILIM LABORATUVARI 1

WEB UYGULAMASI

190202106-190202058

İÇERİK

Bir bölümde yapılan projelerin sisteme yüklenebildiği ve bu dokümanlarının temel bilgilerinin çıkarılıp sorgulanabildiği bir web uygulaması gerçekleştirmeniz beklenmektedir.

Web sitesinde yönetim ve kullanıcı paneli bulunmalıdır.

Yönetim paneli aracılığıyla yeni kullanıcı ekleme, silme ve güncelleme işlemleri yapılabilir.

Kullanıcıların sisteme PDF yükleyebileceği bir alan bulunmalıdır. Bu alanda projeyi teslim

eden öğrencilerin PDF'leri yüklenecektir. Bu PDF'ler için kullanılacak algoritma ile aşağıdaki

bilgilerin otomatik olarak elde edilmesi gerekmektedir:

- Yazar bilgileri (ad, soyad, öğrenci numarası, öğretim türü)
- Ders adı (araştırma problemleri veya bitirme projesi)
- Proje özeti
- Projenin teslim edildiği dönem (Örn. 2020-2021 Güz)
- Proje başlığı
- Anahtar kelimeler
- Danışman bilgileri (ad, soyad, unvan)
- Jüri bilgileri (ad, soyad, unvan)

Elde edilen bilgiler ve PDF dokümanı ilişkisel bir veritabanına kaydedilerek sorgulanabilmelidir.

Yönetim paneli aracılığıyla kullanıcıların yüklediği PDF'lere ve bu dokümanlardan elde edilen

bilgilere, erişim sağlanabilmelidir. Yönetim kısmında kullanıcı bilgileri, dersin adı ve dönem

bilgilerine erişim sağlanabilmelidir.

Aşağıdaki sorgular gerçekleştirilmelidir:

- Sorgu 1: Elde edilen bilgilerden yazara, derse, proje adına, anahtar kelimelere ve döneme göre ayrı ayrı sorgular
- Sorgu 2: Belirli dönemde bir kullanıcının bir derste yüklediği projeler (Örn. 2018-2019 Bahar dönemi A. A. kullanıcısının Bitirme dersi için yüklediği projeler)

KAZANIMLAR

İnternet sitesi yapmayı öğrendik.

İlişkisel bir veritabanı tasarladık. Her kullanıcının sahip olduğu tezler vardır. Bu tezleri ayrı bir tablo oluşturarak orada tuttuk. Daha sonra tez tablosundaki kullanıcı ID'sini kullanıcı tablosundaki ID'ye bağladık. NodeJs kullanarak oluşturduğumuz uygulamada bu veritabanını bağladık. Verileri gösterdik. Kullanıcı ekleme, silme ve güncelleme işlemlerini yaptık.

Kullanıcı giriş sayfasını oluşturduk. Her kullanıcının kendi bilgilerini görmesi için JWT kullandık. Bu token her giriş yapan kullanıcının ID'sini içerir. Buna göre kimin giriş yaptığı anlaşılır. Daha sonra bu ID'den kullanıcının bilgilerine erişilir. Yazdığımız SQL sorgularıyla bunları yaptık.

Tez yükleme sayfası oluşturduk. Kullanıcı buradan PDF formatındaki dosyaları yükler. Daha sonra back-end tarafında regex kullanarak tezin içindeki istenen bilgileri çektik ve veri tabanına kaydettik. Bu bilgileri alırken satıra göre bilgileri bulduk. Öğretim türünü bulurken öğrenci numarasının içinde 202 varsa ikinci öğretim 201 varsa birinci öğretim şeklinde bir algoritma kullandık.

Öğrenci adını, soyadını, öğrenci numarasını bu bilgileri içeren satırdan bulduk.

Sorguları kullanıcı ID'ye göre yaptık. Bu kullanıcı ID'yi giriş yapan kullanıcının token'ından elde ettik ve ona göre veritabanından sorgualdık. Daha sonra bu kullanıcının yüklediği PDF'leri ve PDF'deki bilgileri tablo şeklinde gösterdik.

Axios kullanarak back-end'den verileri çektik ve önyüzde gösterdik. Kullanıcı giriş yaptığında kullanıcının profili ve yüklemiş olduğu PDF'ler ekranda gösterildi. Menüden de sorgular ve dosya yükleme sayfalarına erişim sağladık. Admin giriş yaptığında admin panel ekranı gösterilir. Burada da kullanıcı ekleme, silme ve güncelleme işlemleri yapılır. Ayrıca sorgularda yapılır.

Bu projede back-end'de SQL sorguları ile veriyi ekleme, silme ve güncelleme işlemlerini öğrendik. Back-end'de yaptığımız bu işlemleri önyüzde göstermeyi öğrendik.

Regex kullanımını öğrendik.

YORUMLAR

Yaptığımız Web uygulaması bize göre kullanıcı dostudur.

Bir hata bulunmamaktadır.

Tüm isterler yerine getirilmiştir.

Sorgular hatasız çalışmaktadır.

Kullanıcı giriş işlemlerinde hangi kullanıcının giriş yaptığını bulmada hatayla karşılaştık. Ve çözümünü bulduk.

GENEL YAPI

Projenin back-end'ini nodeJs kullanarak yaptık. Front-end kısmını react kullanarak yaptık. Veri tabanı olarak MySQL kullandık ve tabloları birbirine bağladık.

API'de ID'ye göre kullanıcı getirme ve PDF getirme yöntemini kullandık.

Multer kütüphanesinden yararlanarak PDF'leri bir dosyada depoladık. Bu dosya API'nin içinde localhost adresiyle bağlantılı PDF'lere bu adresten erişilmesini sağladık.

Front-end kısmında router kullanarak link işlemlerini sağladık.

PDF yükleme işlemini post metodu kullanarak yaptık. Post metoduyla yüklenen PDF içindeki veriler anında bulunur ve veri tabanına yazdırılır. Buradan da anasayfadaki tabloya aktarılır. Sorguların hepsi için ayrı bir sayfa yaptık. Bu sayfaların back-end'inde SQL sorgularını döndürdük.

KAYNAKÇA

<https://jasonwatmore.com/post/2021/11/22/nodejs-mysql-crud-api-example-and-tutorial>

<https://tugrulbayrak.medium.com/jwt-json-web-tokens-ile-express-rest-api-güvenliđi-2-517cc44e846c>

<https://www.freecodecamp.org/news/how-to-parse-pdfs-at-scale-in-nodejs-what-to-do-and-what-not-to-do-541df9d2eec1/>

<https://www.yusufsezer.com.tr/node-js-express-route/>

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

<https://medium.com/devopsturkiye/react-axios-ile-web-request-47c61815a494>

AKIŞ ŞEMASI



