

KÜTÜPHANE UYGULAMASI

Kadir Yüksel, Eren Şapçı

170202059, 170202003

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

sapci.eren@gmail.com, 170202059@kocaeli.edu.tr

Özet

Projemizin amacı kullanıcılar ve yöneticiler olmak üzere iki tip kullanıcıya online kütüphane hizmeti sağlayan bir internet sitesi oluşturmaktır. Yönetici sisteme kitap adı ve kitap girdisi yapabilmekte ayrıca sistemde gün ötelemesi ve kullanıcıları listeleyebildiği bir tabloya erişebilmektedir. Öte yandan kullanıcılar sisteme giriş yaptığında kitap adı veya kitabın ISBN numarasını girerek kitap arama, kitap alma ve kitap verme gibi işlemleri gerçekleştirmektedir..

1.Giriş

Kullanıcılar veya admin siteye giriş yaptığında karşısına bir login ekranı çıkmaktadır. Bu ekranda kullanıcı girişi yaptığımızda karşımıza kullanıcı sayfası çıkmaktadır. Kullanıcı sayfasında kullanıcıya üç farklı seçenek sunulmaktadır. Kitapları veritabanında aramak istenirse kitap ara kısmından ISBN numarası ve Kitap adı aratılarak kullanıcıya pop-up lar ile kitabın var olup olmadığı bilgisi verilir. Kitap alma kısmında eğer kitap sistemdeyse veya başka bir kullanıcı tarafından alınmadıysa kullanıcıya kitap verilir. Kitap verme kısmında ise kullanıcı ISBN numarası gözükecek şekilde sisteme kitabın resmini yüklerse elindeki kitabı sisteme geri iade edebilir.Admin girişi yaptığımızda adminin karşısına sisteme kitap yükleyeceği bir ekran çıkar bu kısımda admin istediği bir kitabı ISBN numarası ve kitap adı ile beraber sisteme yükleyebilmektedir. Sistemimizden alınan bir kitabın bir hafta süre ile geri iade edilmesi gerekmektedir Sistemde başlangıç gününü öteleme işlemi yapılarak 1 haftalık süreyi görüntülemek mümkündür. Admin sayfamızda ayrıca kullanıcıları, aldığı kitapları, veriliş tarihlerini ve iade edilmesi gereken tarihi görüntülememiz mümkündür.

2.Temel Bilgiler

Projemiz Javascript programlama dili ile çeşitli html ve css kütüphaneleri kullanılarak tasarlanmıştır. Backend kısmında NodeJs kullanılarak proje gerçekleştirilmiştir. Veritabanı kısmında MongoDB kullanılarak veritabanı işlemleri gerçekleştirilmiştir. Proje Visual Studio Code kullanılarak oluşturulmuştur.

3.Tasarım

Kütüphane uygulamasının programlama aşamaları aşağıdaki başlıklar altında açıklanmıştır.

3.1 Algoritma

Sitemize ilk giriş yaptığımızda karşımıza çıkan login ekranından kullanan kişi giriş yaptığı zaman giriş yapan kişinin kullanıcı mı yoksa admin olup olmadığı kontrol edilip ona uygun sayfa ekrana getirilir.

Yönetici panelinde kitap ekleme işleminde kitap adı ve ISBN numarası istenir. Sisteme yüklenen resimde tesseract yardımı ile ISBN numarası çekilir. ISBN numarası ve kitap adı girildikten sonra sistemde böyle bir kitabın olup olmadığı kontrol edilir. Kitap yok ise kitap veritabanına eklenir. Kullanıcı listeleme işleminde ise sisteme kayıtlı tüm kullanıcılar aldıkları kitaplar ile birlikte listelenir. Ayrıca yine yönetici panelinden sistem tarihi yöneticinin istediği kadar ileri alınabilir.

Kullanıcı sayfasında kitap arama kısmında kullanıcı ISBN numarası ile ya da kitap adı ile veritabanından arama işlemini gerçekleştirir. Eğer kitap sistemde var veya yok ise pop-up ile kullanıcıya bunun bilgisi verilir. Kitap alma kısmında kullanıcı ISBN numarasını ya da kitap ismini yazar. Almak istediği kitabın veri tabanında olup olmadığı veya başka bir kullanıcı tarafından alınıp alınmadığı kontrol edilir. Alınmamış ise kullanıcı kitabı 7 gün iade koşulu ile alabilmektedir. Kitap iade etme kısmında ise yönetici panelindeki kitap ekleme kısmında olduğu gibi kullanıcı kitap resmini sisteme yükler. Sistem böyle bir kitabın olup olmadığını ve bu kullanıcının böyle bir kitaba sahip olup olmadığını kontrol eder ve ardından iade işlemini gerçekleştirir ya da pop-up yardımıyla işlemin neden gerçekleştirmediğini kullanıcıya bildirir.

4. Kazanımlar

- NodeJs ile kodlama konusunda yetkinlik kazanıldı.
- Veritabanı konusunda MongoDB ile ilk kez çalışıldı ve öğrenildi.
- Web tasarım konusunda yetkinlik kazanıldı.
- Local serverda çalışıldı.
- Admin ve kullanıcı kontrolleri konusunda gelişim sağlandı.

5. Kullanılan Paketler

```
"author": "no one",
"license": "MIT",
"dependencies": {
  "add": "^2.0.6",
  "express": "^4.17.1",
  "express-fileupload": "^1.1.6",
  "mongoose": "^5.9.4",
  "tesseract.js": "^2.1.1",
  "yarn": "^1.22.4"
},
"devDependencies": {
  "nodemon": "^2.0.2"
}
```

6. Kullanılan Bazı Fonksiyonlar

```
let bookName = req.body.bookname.toLowerCase();
let imageFile = req.files.bookImage;
let imageAddress = './isbnPictures/' + bookName + ".jpeg";
console.log("Resim dosyasının adı: " + imageFile.name);
//console.log("Image object: " + imageFile);
console.log("Kitabın Adı: " + bookName);

imageFile.mv(imageAddress, async function (error) {
  if (error) {
    console.log("Resim dosyası upload edilemedi.");
    console.log(error);
    errorCode = 0;
    res.json(errorCode);
  } else {
    console.log("Resim dosyası başarıyla upload edildi!");
    await readImageAndUploadBookInfo(imageAddress, bookName);
    console.log("ERRORCODE: " + errorCode);
    res.json(errorCode);
  }
})
```

- Veri Tabanı kontrol işlemleri

```
if (found) {
  query = query.toLowerCase();
  //console.log(query);
  const doc = await bookAdmin.find({ fileName: query }, async function (err, data) {
    if (err) {
      console.log("KITAP ADINI ARARKEN BİR ŞEYLER TERS GİTTİ: ", err);
      res.json("error");
    } else if (data.length == 0) {
      console.log("KITAP BULUNMADI!");
      res.json(obj);
    } else {
      await searchBookOwners(data[0].isbnNumber);
      //await getBook(data[0].isbnNumber);
      console.log("ARADIGINIZ: " + query);
      console.log("Kitap Adı: " + data[0].fileName + " ID: " + data[0]._id + " ISBN: " + data[0].isbnNumber);
      obj = { hasMoreThanThreeBooks: loggedUser.hasMoreThanThreeBooks, hasOutOfDateBook: loggedUser.hasOutOfDateBook, someOne };
      //result = JSON.stringify(obj);
      res.json(obj);
    }
  });
}
```

- Kullanıcı tarafında kitap arama işlemleri

```
let bookName = req.body.bookname.toLowerCase();
let imageFile = req.files.bookImage;
let imageAddress = './isbnPictures/' + bookName + ".jpeg";
console.log("Resim dosyasının adı: " + imageFile.name);
//console.log("Image object: " + imageFile);
console.log("Kitabın Adı: " + bookName);

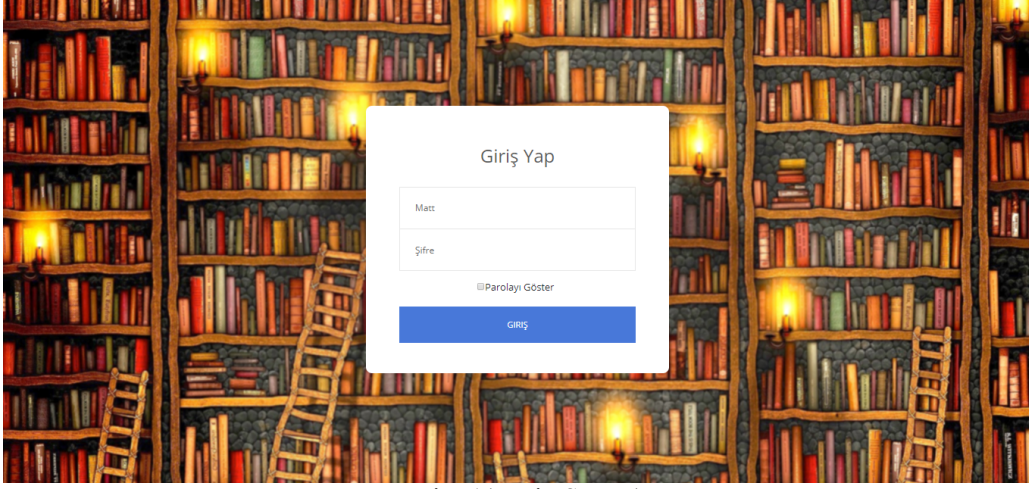
imageFile.mv(imageAddress, async function (error) {
  if (error) {
    console.log("Resim dosyası upload edilemedi.");
    console.log(error);
    errorCode = 0;
    res.json(errorCode);
  } else {
    console.log("Resim dosyası başarıyla upload edildi!");
    await readImageAndUploadBookInfo(imageAddress, bookName);
    console.log("ERRORCODE: " + errorCode);
    res.json(errorCode);
  }
})
```

- Sisteme resim yükleme işlemleri

7. Kaynakça

- [1] <https://www.mongodb.com/download-center/community>
- [2] <https://nodejs.org/en/knowledge/HTTP/servers/how-to-create-a-HTTP-server/>
- [3] <https://getbootstrap.com/docs/4.4/content/tables/>
- [4] <https://getbootstrap.com/docs/4.0/components/modal/>
- [5] <https://www.youtube.com/watch?v=waN5-o1quks>
- [6] <https://codeburst.io/writing-a-crud-app-with-node-js-and-mongodb-e0827cbbdafb>
- [7] <https://docs.mongodb.com/manual/tutorial/model-referenced-one-to-many-relationships-between-documents>
- [8] <https://www.npmjs.com/package/node-tesseract-ocr>
- [9] <https://tesseract.projectnaptha.com/>
- [10] <https://www.youtube.com/watch?v=1DtyAOHEHJY>
- [11] <https://dev.to/yvonnickfrin/ocr-in-javascript-with-tesseract-hn1>

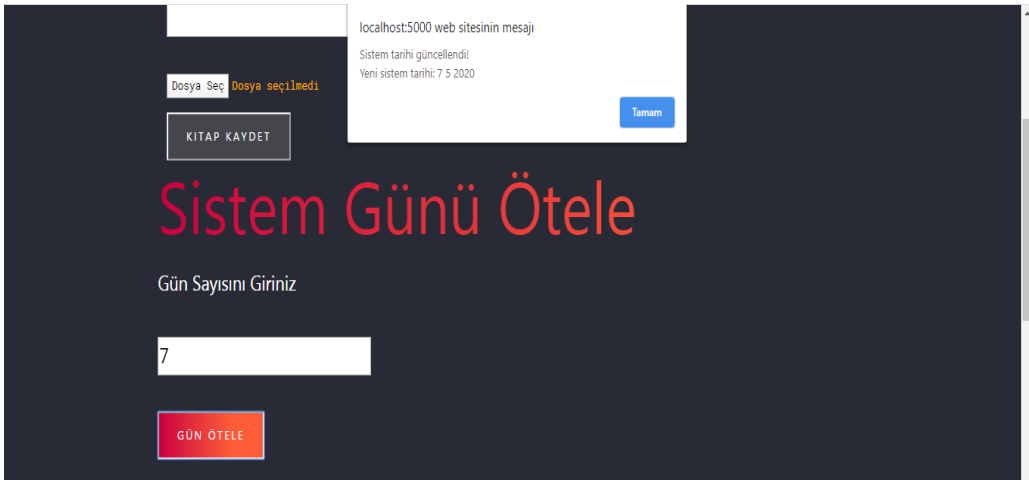
8. Deneysel sonuçlar



Resim-1(Login Screen)



Resim-2(User Interface)



Resim-3(Gün Öteleme)

9.Akış Şeması

