

## Ağ Tabanlı Teknolojiler ve Uygulamaları Proje Sunumu

19290214  
Elif Ayçiçek

Projem 3 kısımdan oluşmaktadır.(Frontend,Backend ve sql data server)  
Github linkimde bulunan “e” dosyasının içindeki kısım backend ve frontend kısmını içermektedir.  
Frontend ile ilgili kodlarım e dosyasında,backend ile ilgili kodlarım a dosyası içinde bulunmaktadır.  
Github linkimin içinde aynı zamanda sql database’inin projede kullandığım tüm tablalarının scripti bulunmaktadır.(script.sql olarak)

### -BACKEND

Backend ile ilgili kodlarımın hepsi “a” dosyasının içinde yer almaktadır.Dosyayı açtığımızda belgeler şekil 1.1deki gibi gözükmetedir.



..	
Controllers	push to github
Models	push to github
Properties	push to github
bin/Debug/net6.0	push to github
obj	push to github
Program.cs	push to github
a.csproj	push to github
a.csproj.user	push to github
appsettings.Development.json	push to github
appsettings.json	push to github

Şekil 1.1

Şekil 1.1’e baktığımızda gördüğümüz Models dosyasının içerisinde sqlden gelecek verileri tanımladığım c# modelleri ve onların database contextleriyle(Models dosyası içinde) beraber http get/post/delete/put metodlarını(CRUD) gerçekleştirecek controllerların(Controllers dosyası içerisinde) kodları yer almaktadır.

Örnek verecek olursam sql veritabanından kitap bilgilerini getirmek için ilk olarak “books.cs” isimli bir model tanımlaması yaptım.(Şekil 1.2)

```

using Microsoft.VisualBasic;
using System.ComponentModel.DataAnnotations;

namespace a.Models
{
    public class Books
    {
        [Key]
        public int? BookId { get; set; }

        public string? BookName { get; set; }
        public string? Category { get; set; }

        public string? BookAuthor { get; set; }
        public string? BookCover { get; set; }

        public string? BookSummary { get; set; }
    }
}

```

Şekil 1.2

ardından Booksdbcontext.cs dosyasının içine modelin database contextini tanımladım.(Şekil 1.3)

```

using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System;

namespace a.Models
{
    public class Booksdbcontext : DbContext
    {
        public Booksdbcontext(DbContextOptions<Booksdbcontext> context) : base(context)
        {
        }

        public DbSet<Books> Books { get; set; }
    }
}

```

Şekil 1.3

Daha sonra ise Controllers dosyası içinde bulunan BookController.cs dosyasında sql database'inden kitapların adını,yazarını,özetini,türünü getirmek için get metodunu kullandım.(Şekil 1.4)

```

namespace a.Controllers;

[Route("api/[controller]")]
[ApiController]
public class BookController : ControllerBase
{
    private readonly Booksdbcontext _reactcontext;

    public BookController(Booksdbcontext reactcontext)
    {
        _reactcontext = reactcontext;
    }

    [HttpGet]
    public async Task<ActionResult> GET()
    {
        var books = await _reactcontext.Books.ToListAsync();
        return Ok(books);
    }

    [HttpGet]
    [Route("{bookId:int}")]
    public async Task<IActionResult> Get(int bookId)
    {
        var bookById = await _reactcontext
            .Books.Where(_ => _.BookId == bookId)
            .FirstOrDefaultAsync();
        return Ok(bookById);
    }
}

```

Şekil 1.4

Diğer sql table verilerim için de aynı yöntemleri uyguladım.

-Events.cs ve Eventdbcontext.cs (Kitaplar için etkinlik oluşturma).Controllerı => EventController.cs

-Books.cs ve Booksdbcontext.cs (Kitapların bilgilerini getirmek için).Controllerı => BookController.cs

- Register.cs ve Registerdbcontext.cs(Kayıtlı kullanıcıları tutmak için).Controllerı => RegistrationController.cs

-BookComments.cs ve BookCommentdbcontext.cs(Kitap yorumlarını göstermek için).Controllerı=>BookCommentController.cs

## FRONTEND

Frontend kısmı için projemde react kullanmayı seçtim.Burda yazmış olduğum tüm kodlar “e” dosyasında “src” kısmının içindedir.

App.js dosyası sayfaların yönlendirilmesinin yapıldığı ana dosyamızdır. Diğer tüm dosyalar “components” içindedir.

1-Projemizi açtığımızda ekrana gelen ilk sayfa “Login.js” sayfasıdır.Bu sayfada kullanıcı eğer siteye kayıtlı değilse,kullanıcı adı,şifresi,emaili ve favori kitabını yazarak kayıt olabiliyor ve bilgileri sql veritabanına kaydediliyor.

The image shows a web form titled "BookWorm" with a "Create Account" section. The form is set against a dark purple background with a light blue rounded rectangle containing the input fields. The fields are: Username, Email Address, Password, and Favourite Book. Below these fields is a dark purple button labeled "CREATE ACCOUNT". At the bottom of the form, there is a link that says "HAVE AN ACCOUNT? SIGN IN".

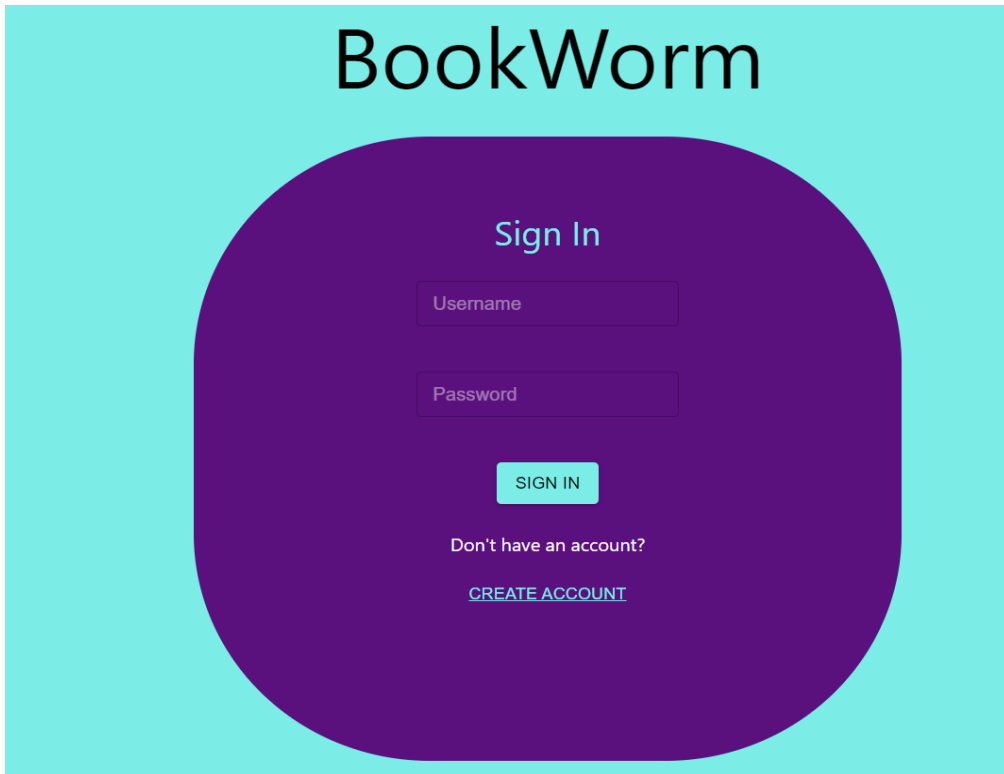
## BookWorm

### Create Account

CREATE ACCOUNT

[HAVE AN ACCOUNT? SIGN IN](#)

2-Eğer zaten kayıtlı ise “have and account? Sign in” butonuna tıklayarak giriş sayfasına yönlendiriliyor.Giriş sayfasının kodları “Sign.js” dosyasının içinde yer almaktadır.Bu sayfada kullanıcı bilgilerini girerek kendi kullanıcı sayfasına yönlendiriliyor.Eğer hatalı girdiyse karşısına bir error mesajı çıkıyor.

The image shows a web form titled "BookWorm" with a "Sign In" section. The form is set against a light blue background with a dark purple rounded rectangle containing the input fields. The fields are: Username and Password. Below these fields is a light blue button labeled "SIGN IN". Below the button, there is a link that says "Don't have an account?" and a link that says "CREATE ACCOUNT".

## BookWorm

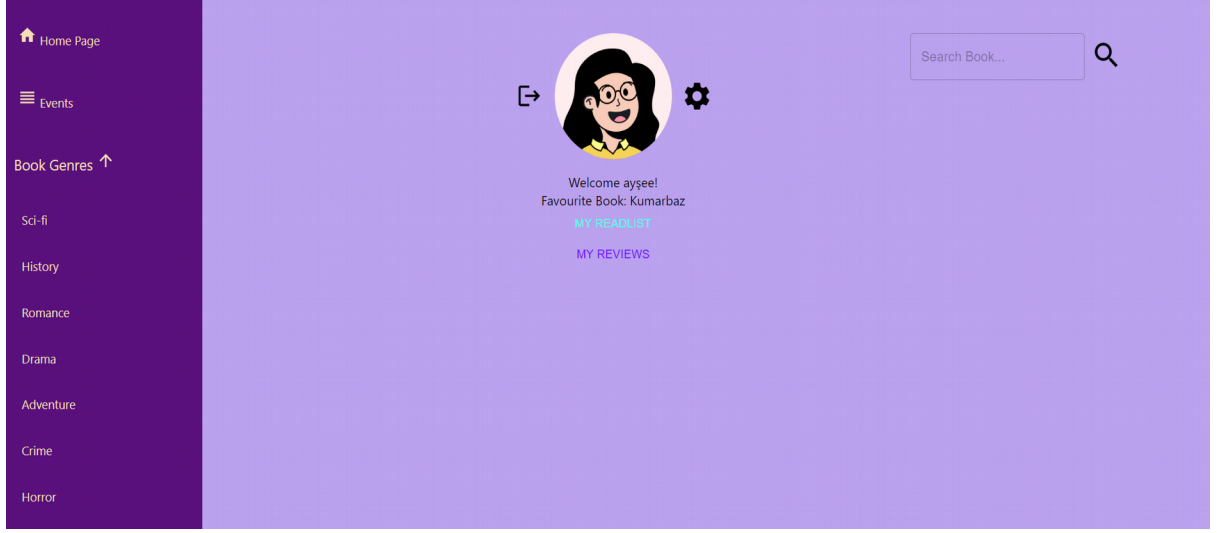
### Sign In

SIGN IN

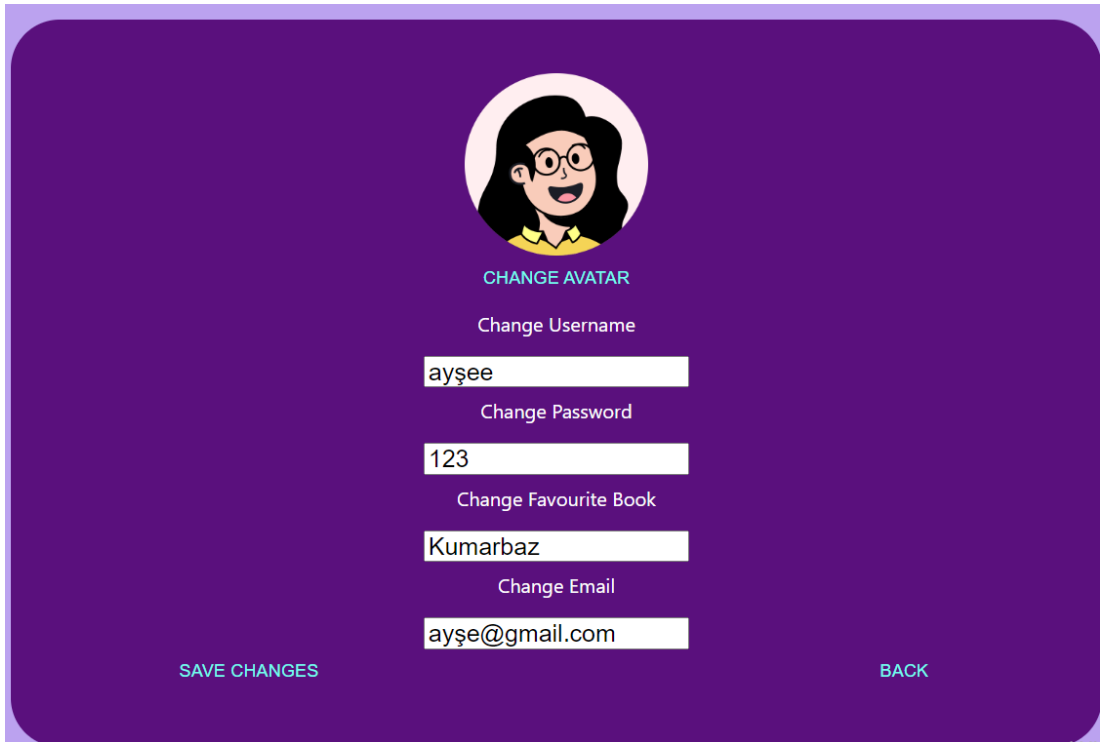
[Don't have an account?](#)

[CREATE ACCOUNT](#)

3-Kullanıcının giriş yaptıktan sonra önüne gelen sayfa “Userpage.js” dosyası içinde bulunmaktadır. Bu sayfada kullanıcının avatarı,kullanıcının adı ve favori kitap bilgileri getirilmektedir.Bu sayfadan kullanıcı çıkış yapabilir ve ayarlar butonuna tıklayarak ayarlar sayfasına yönlendirilir.Kullanıcının kitaplar ve eventler arasında gezinmesi için bir de sidebar navigation bulunmaktadır.



4-Kullanıcının ayarlar sayfasına ait kodlar “Settings.js” içinde bulunmaktadır.Bu sayfada kullanıcı kullanıcı adını,favori kitabını,şifresini ve emaili değiştirip güncelleyebilir.

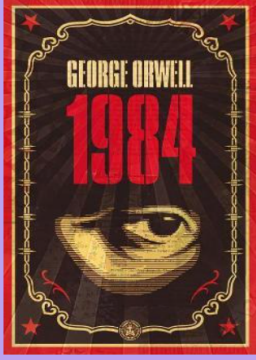


5-Tüm kitaplar sidebar navigation içinde türlerine göre yerleştirilmiştir.macara türü “Adventure.js” içinde, suç türü “Crime.js” içinde, dram türü “Drama.js” içinde, korku türü “Horror.js” içinde, bilimkurgu türü “Scifi.js” içinde,tarihi türü “History.js” içinde,romantik türü “Romance.js” içinde yazılmıştır.Tüm bu js dosyaları içinde türlerine göre kitaplar isim,yazar ve kapak fotoğraflarıyla sıralanmaktadır.

Örn Scifi sayfası:



6-Kitaba özel özetini,yazarı,kitap fotoğrafını ve adını gösteren,kullanıcıların yorum yapabildiği ve kitaba dair tüm yorumları gösteren kodların sayfası “BookComments.js” dosyası içindedir.



## 1984-George Orwell

Winston Smith works for the Ministry of Truth in London, chief city of Airstrip One. Big Brother stares out from every poster, the Thought Police uncover every act of betrayal. When Winston finds love with Julia, he discovers that life does not have to be dull and deadening, and awakens to new possibilities. Despite the police helicopters that hover and circle overhead, Winston and Julia begin to question the Party; they are drawn towards conspiracy. Yet Big Brother will not tolerate dissent - even in the mind. For those with original thoughts they invented Room 101. . .

#Sci-fi

ADD COMMENT

Recommended by ayşee

11.01.2023

bayıldım

7-Tüm eventlerin gösterildiği,yeni eventlerin oluşturulabildiği ve event türüne göre eventlerin filtrelenerek gösterildiği kısma ait kodlar “Events.js” içindedir.

Events

Event Type:

Book Reading

CREATE EVENT

Sefiller

Book Reading

Marslı

Book Reading

Create event kısmı:

Select Event Type

Select Total Participants:

3

Book Name

Add Comment about Event...

Select Event Date:

2023/01/18



Select Event Time:

hh:mm



CANCEL

ADD EVENT

8-Tıklanılan evente ait detaylı bilgilerin gösterildiği kodlar “ShowEvents.js” içerisinde.

Bookname:	Suç ve Ceza	
Event Type:	Book Discussing	
Event Date:	05.01.2023	
Event Time:	16:30:00	
Host's Comment:	kitaptaki karakterlerin incelemesini ve analizini yapacağız.	
Participants:	1/4	ENROLL
GO BACK		

## VERİTABANI

Veritabanı olarak sql kullandım.Veritabanı için 4 farklı table kullandım.

- 1-dbo.Books:Kitap bilgilerini tutan table(kitap adı,yazarı,türü,özeti ve kitap kapak fotoğrafı)
- 2-dbo.UserRegistration:Kullanıcı bilgilerini tutan table(kullanıcı adı,şifresi,emaili ve favori kitabı)
- 3-dbo.BookComments:Kitap yorumlarını tutan table(kullanıcı adı,yorum atılan tarih,kitap yorumu,öneriliyor mu/önerilmiyor mu)
- 4-dbo.CreateEvent:Eventleri tutan table(kitap adı,event türü,saati,günü,event açıklaması ve katılımcı sayısı)