Ders Programı Sitesi ve AR Uygulaması

Süleyman Burak Çınar

211307031

Ömer Baver Yıldız

211307064

Deniz Ilgın Koca

211307017

Bilişim Sistemleri Mühendisliği

Teknoloji Fakültesi

Kocaeli Üniversitesi

I. Giriş

Bu rapor HTML, CSS, Node.js ve SQL ile geliştirilmiş ders programı ve C# ile yazılmış AR projesi hakkındadır. Proje, kullanıcılara güncel ders programına erişip derslerin yerini değiştirme, dersleri güncelleme, dersleri silme özellikleri ve AR ile de fakültenin krokisini görüntüleyip sınıflara tıklayıp o sınıftaki ders programını gösterme özelliklerini sağlamaktadır.

II. Proje Amaçları ve Kapsamı

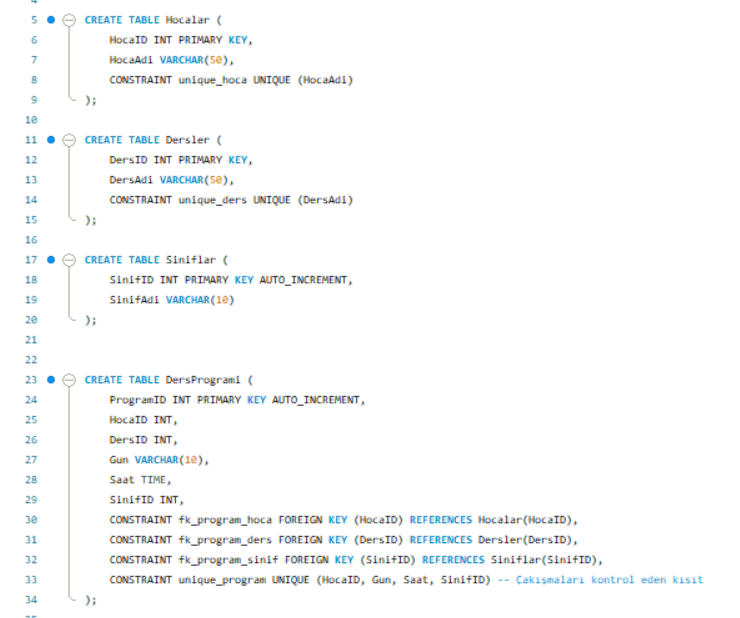
## Projenin temel amaçları şunlardır:

* Kullanıcıların ders programına erişip düzenleyebilmesini sağlamak.
* ARCore teknolojisi kullanılarak fakülte haritasına erişip sınıflardaki ders programını görebilmek.

III. Geliştirme Süreci

İlk olarak ders programı için verilen istencelerle birlikte bir veri tabanı oluşturuldu. Veri tabanının pseudo kodu:

* ders\_programi isimli bir veritabanı oluştur.
* Hocalar tablosu oluştur.
* Hoca adını kısıt olarak ekle.
* Dersler tablosu oluştur.
* Ders adını kısıt olarak ekle.
* Siniflar tablosu oluştur.
* DersProgrami tablosu oluştur.
* Hocalar, Dersler ve Siniflar tablosundaki ID'leri kullansın.
* Tabloya ek olarak gün ve saat elemanları ekle.
* Hocaları, dersleri, sınıfları ve ders programı içeriğinin çakışmaması için kısıt olarak ekle.
* Derslere eleman ata.
* Sınıflara eleman ata.
* Hocalara eleman ata.
* Ders programına elamanları ata.



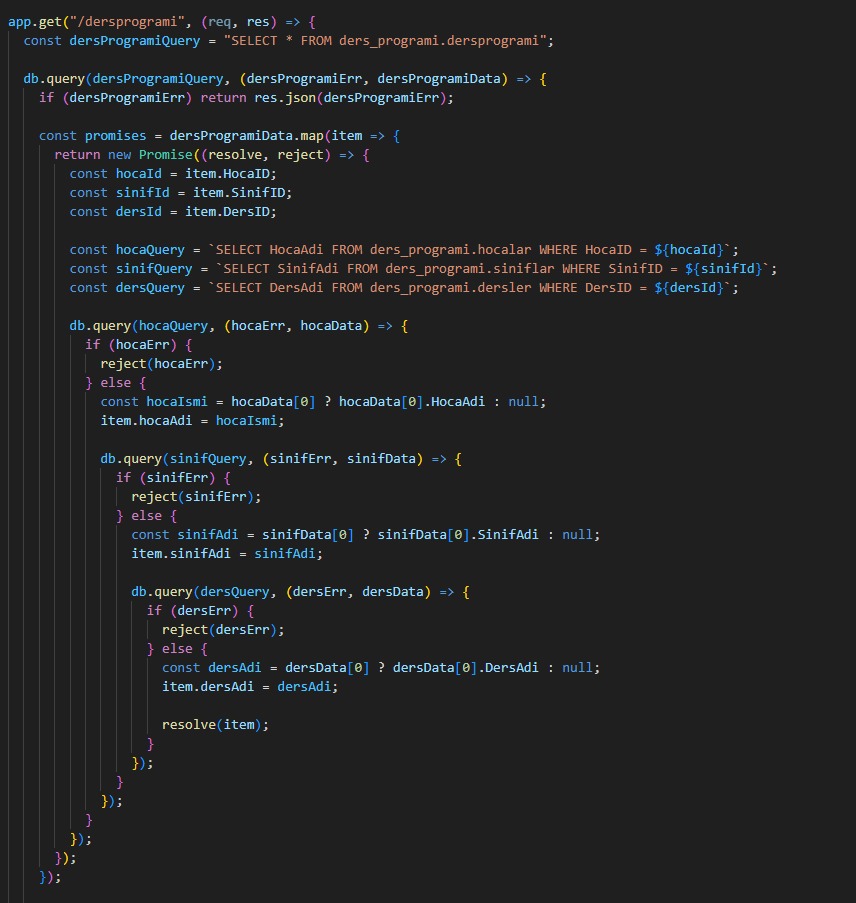
Veri tabanını gösterebilmek ve ders ekleme, çıkarma, silme işlemleri için localhost’u kullanan bir web sitesi oluşturuldu. Sitenin pseudo kodu:

Frontend:

* Gerekli modüller ve bileşenler import edilir.
* State değişkenleri ve state'i güncellemek için setter fonksiyonları tanımlanır.
* useEffect kancası kullanılarak bileşenin ilk render edildiğinde API'lerden veri çekme işlemi gerçekleştirilir.
* Ders programı tablosunu oluşturan ve her hücreye uygun bilgileri yerleştiren bir fonksiyon (handleTable) tanımlanır.
* Ders eklemek için kullanılan modal penceresi açma fonksiyonu (handleOpenModal) tanımlanır.
* Ders eklemek için kullanılan formun submit işlemini gerçekleştiren fonksiyon (handleSubmit) tanımlanır.
* Ders güncellemek ve silmek için kullanılan modal penceresi açma fonksiyonu (handleOpenUpdateModal) tanımlanır.
* Ders güncellemek için kullanılan formun submit işlemini gerçekleştiren fonksiyon (handleUpdateSubmit) tanımlanır.
* Dersi silmek için kullanılan fonksiyon (handleDelete) tanımlanır.
* JSX içinde, tablo ve modaller render edilir. State değişkenleri ve fonksiyonlar uygun yerlere bağlanır.
* Modal bileşenleri, kullanıcı bir modalı kapattığında state değişkenlerini sıfırlayan onCloseModal ve onCloseUpdateModal fonksiyonlarına bağlanır.
* Dersler, hocalar, sınıflar, günler ve saatler için select (seçim) inputları oluşturulur ve uygun API verileriyle doldurulur.
* Axios kütüphanesi kullanılarak API istekleri gerçekleştirilir (POST, DELETE, PUT).
* Bileşen export edilir.

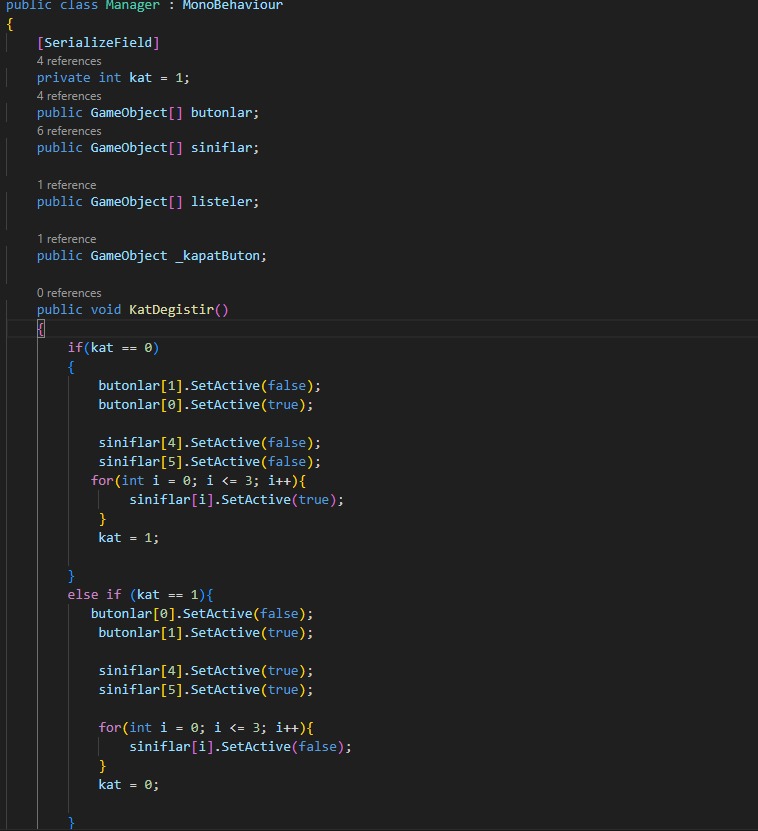
Backend:

* Gerekli modülleri içe aktar
* import express
* import mysql
* import cors
* Bir Express uygulaması oluştur
* app = express()
* app.use(express.json())
* app.use(cors())
* MySQL veritabanı bağlantısını oluştur
* db = mysql.createConnection({
* host: "localhost",
* user: "root",
* password: "619123",
* database: "ders\_programi",
* })
* Rotaları ve ilgili veritabanı sorgularını tanımla
* Rota: Ana sayfa
* app.get("/", (req, res) => {
* res.json("merhaba, bu arka uçtır")
* })
* Rota: Tüm dersleri getir
* app.get("/dersler", (req, res) => {
* 'dersler' tablosunda bir SELECT sorgusu çalıştır
* Sonucu veya bir hatayı JSON formatında döndür
* })
* Rota: Tüm hocaları getir
* app.get("/hocalar", (req, res) => {
* 'hocalar' tablosunda bir SELECT sorgusu çalıştır
* Sonucu veya bir hatayı JSON formatında döndür
* })
* Rota: Tüm sınıfları getir
* app.get("/siniflar", (req, res) => {
* 'siniflar' tablosunda bir SELECT sorgusu çalıştır
* Sonucu veya bir hatayı JSON formatında döndür
* })
* Rota: Tüm ders programlarını getir
* app.get("/dersprogrami", (req, res) => {
* 'dersprogrami' tablosunda bir SELECT sorgusu çalıştır
* Her bir program için ilgili veriyi getir (hoca adı, sınıf adı, ders adı)
* Sonucu veya bir hatayı JSON formatında döndür
* })
* Rota: Yeni bir ders programı oluştur
* app.post('/dersprogrami/create', (req, res) => {
* İsteğin gövdesinden veriyi çıkar
* Çakışan ders programları kontrol et
* Çakışma yoksa 'dersprogrami' tablosuna yeni bir kayıt ekleyin
* Başarı veya bir hatayı JSON formatında döndür
* })
* Rota: Bir ders programını sil
* app.delete('/dersprogrami/delete/:ProgramID', (req, res) => {
* Program ID'sini istek parametrelerinden çıkar
* 'dersprogrami' tablosunda bir DELETE sorgusu çalıştır
* Başarı veya bir hatayı JSON formatında döndür
* })
* Rota: Bir ders programını güncelle
* app.put('/dersprogrami/update/:ProgramID', (req, res) => {
* Program ID'sini ve güncellenmiş veriyi istek parametrelerinden ve gövdesinden çıkar
* Çakışan ders programları kontrol et
* Çakışma yoksa 'dersprogrami' tablosundaki ilgili kaydı güncelle
* # Başarı veya bir hatayı JSON formatında döndür
* })
* Sunucuyu 8800 portunda başlat
* app.listen(8800, () => {
* console.log("arka uç bağlandı!!!224422")
* })

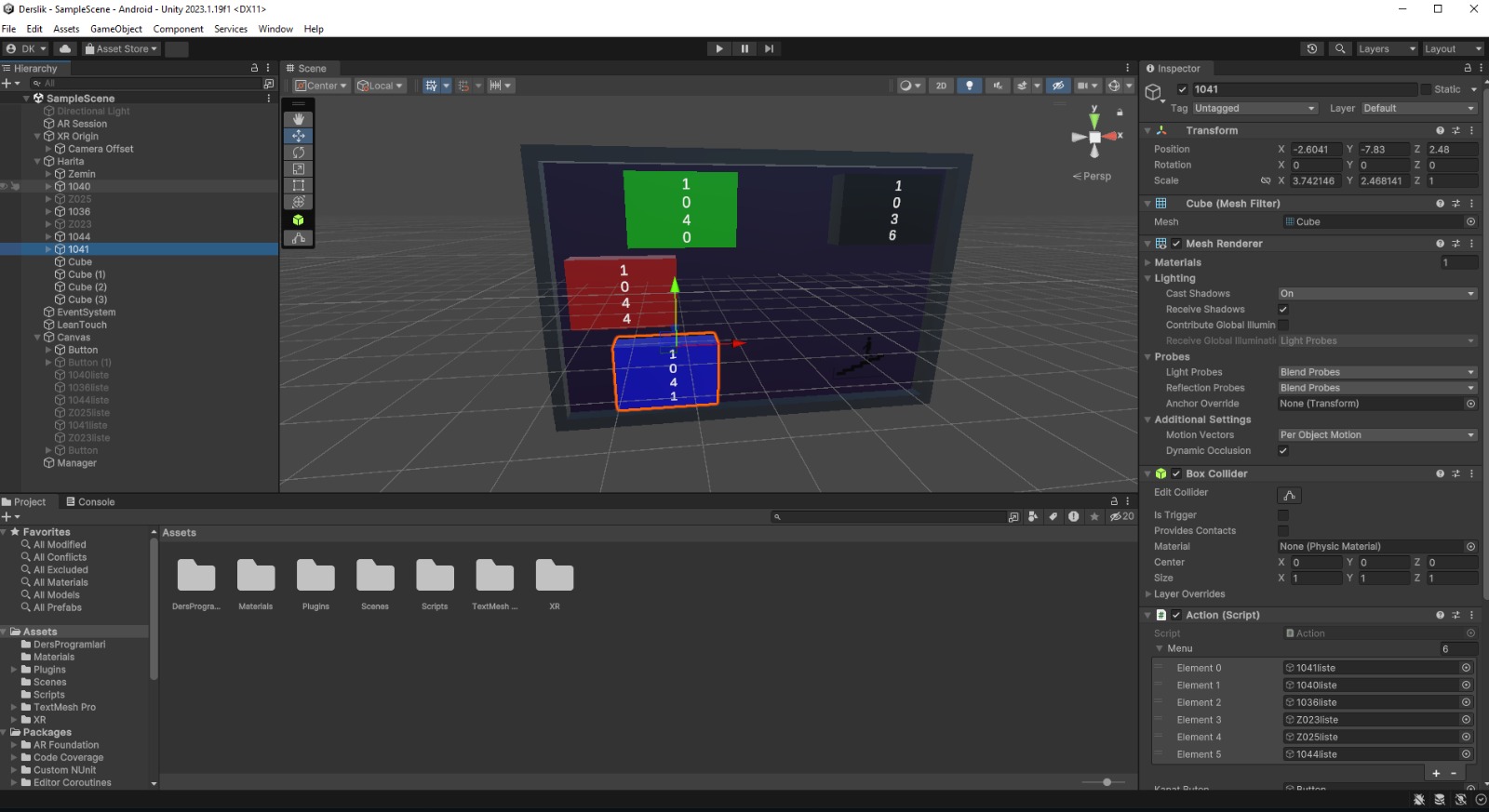
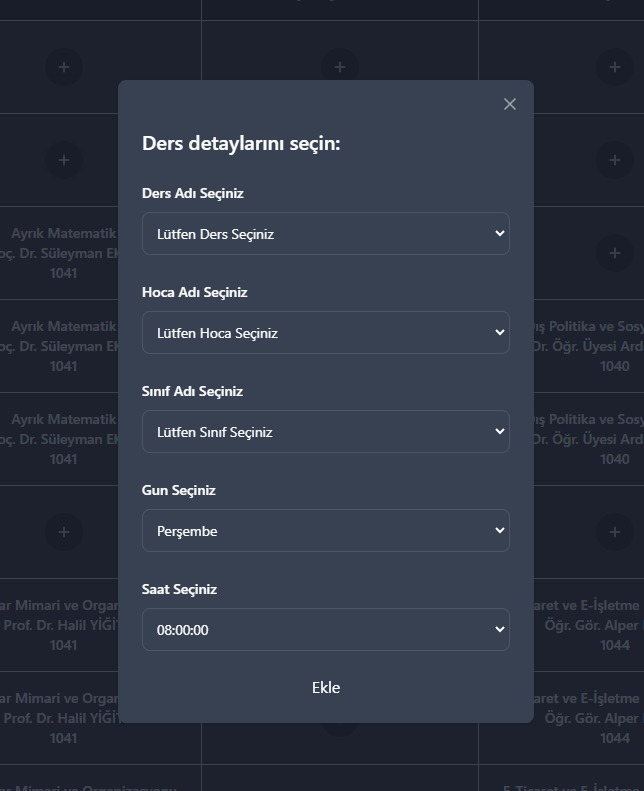


AR için Unity üzerinden bir uygulama yapıldı. Uygulamanın pseudo kodu:

* Bir MonoBehaviour sınıfı olan ButonYap oluştur.
* UnityEvent unityEvent adında bir public UnityEvent tanımla.
* GameObject button adında bir public GameObject tanımla.
* Start fonksiyonu:
* 'button' değişkenine bu scriptin bağlı olduğu GameObject'i ata.
* // Güncelleme fonksiyonu
* Update fonksiyonu:
* Eğer dokunma sayısı (Input.touchCount) 0'dan büyükse:
* İlk dokunma bilgisini al (Input.GetTouch(0)).
* Kameradan ekrandaki dokunma noktasına doğru bir ışın (Ray) oluştur.
* Bir RaycastHit nesnesi oluştur ve bu ışının dünya üzerinde çarptığı noktayı belirle.
* Eğer bu ışın, bir nesneye çarparsa:
* Eğer dokunma aşaması (touch.phase) Began (başladı) ise ve çarpılan nesne bu scriptin bağlı olduğu GameObject ise:
* Tanımlanan UnityEvent'i tetikle (unityEvent.Invoke()).



Oluşturulan programın son hali:



IV. Kaynakça

<https://youtu.be/eIjbSH3Imb8?si=KyTb4VrSG_PBHMnN> (28/12/2023)

<https://youtu.be/fPuLnzSjPLE?si=0WfoWQ0leFUtbCPn> (30/12/2023)

<https://youtu.be/SBmSRK3feww?si=BP9iIMNmGvezR3yP> (30/12/2023)

<https://youtu.be/7MPhxKHMi4c?si=kJvbK_Z1n_0uZAkL> (02/01/2024)

<https://youtu.be/OX-LkVWKSTg?si=xkUtv4zWDgMKIrpV> (03/01/2024)

<https://youtu.be/sX1aOBlwGWc?si=wpmMmqRF0fotY4JA> (03/01/2024)