

PROJE RAPORU

Proje Konusu: Öğrenci Ders Seçim Sistemi

Proje Başlangıç ve Bitiş Tarihi: 05.10.2023- 08.10.2023

Hazırlayanlar:

Sena Ceren Kurnaz, Zülal Sarioğlu, Gülsüm Öz, Ali Rıza Durak, Ali Enes Damat, Ömer Faruk Mahmutoglu

Proje Özeti;

Bu proje, öğrencilerin istedikleri derslere kayıt olmalarını ve ders programlarını oluşturmalarını sağlayan bir web uygulamasıdır.

Bu sistem;

- 1-Öğrencilerin sisteme kendi bilgileriyle giriş yapmasına,
- 2-Öğrencilerin ders sisteminden ders kataloglarını görüp dersleri seçmesine,
- 3- Seçilen derslere göre ders programı oluşturulmasına,
- 4-Ders programının onaylanarak öğrencinin kendi bilgi ekranında görmesine
- 5-Öğrencinin dilediği dersleri iptal etmesine olanak tanır.

GÖREVLER	YAPILAN İŞLEMLER
1-Sistem Mimarisi oluşturma	Sistemde kullanılacak birleşenler belirlendi ve bu bileşenlerin birbirleriyle olan bağlantıları diagram yardımıyla dizayn edildi.
2-Uygulamada kullanılacak CSS kütüphanesinin belirlenmesi	Grup üyelerinin CSS kütüphaneleri birer birer incelenerek uygulamada kullanılacak olan bileşenlere en uygun olan CSS kütüphanesi belirlendi.
3-Uygulamanın Arayüzünün Oluşturulması	Grup üyeleri ile birlikte çeşitli ders seçim arayüzleri incelenerek uygulamanın arayüz şablonu oluşturuldu. Daha sonra grup üyeleri arasında ilgili şablon paylaştırılarak HTML ve seçilen CSS kütüphanesi kullanılıp uygulama arayüzü oluşturuldu.
4-Projede kullanılacak fonksiyonların birleşen bazında belirlenmesi	Projenin ana sayfasında bulunan Login, Ders Seçim ve Öğrenci Bilgi bileşenleri için ilgili fonksiyonlar belirlendi.
5-Login Bileşeni için Fonksiyonalitenin oluşturulması	Öğrenci Bilgilerinin tutulacağı class oluşturulup bu classtan yeni öğrenciler üretildi. Öğrencinin kendi bilgileriyle sisteme giriş yapmasını sağlayan fonksiyon yazıldı.
6-Ders Seçimi için Fonksiyonalitenin Oluşturulması	Ders bilgilerinin tutulacağı class oluşturulup bu classtan yeni dersler üretildi. Öğrencinin istediği dersleri seçerek bu dersleri gün ve saat özelinde haftalık olarak ders programında görmesini ve bu listeye eklediği derslerden istediğini çıkarabilmesini sağlayan fonksiyonlar yazıldı. Eklediği dersleri almak istediğinde onaylamasını sağlayan fonksiyon yazıldı. Bu onay fonksiyonu vasıtasıyla öğrenci aldığı dersleri kendi bilgi ekranında görebilir duruma geldi ve ilgili derslerin kontenjanları güncellendi.
7-Öğrenci Bilgi Ekranı için Fonksiyonalitenin Oluşturulması	Ders seçim bileşeninde onaylanan derslerin ve öğrencinin kişisel bilgilerinin görüntülenmesi için fonksiyonlar oluşturuldu. İstenilmeyen dersin kaydının silinebilmesini sağlayan fonksiyon yazılarak bu fonksiyon yardımıyla dersin kontenjan bilgileri güncellendi.
8-Oluşturulan yapıların diğer grup üyeleri tarafından anlaşılmasının sağlanması	Proje kaynak kodlarında fonksiyon ve değişken isimlerinin anlaşılır belirlenmesi ve daha sonra ilgili yerlerde düzenleme yapabilmek için açıklama satırları vasıtasıyla yapıların anlaşılabilir hale getirildi.
9- Uygulamanın Arayüzü ve Fonksiyonalitesinin İsterler Kapsamında Test Edilmesi	İsterler baz alınarak uygulama süreç içerisinde her bir bileşen özelinde manuel olarak test edildi.
10- Hata Ayıklama	Oluşan arayüz ve fonksiyonalite hataları belirlenerek çözümler üretildi ve bu çözümler vasıtasıyla hatalar giderildi.

Proje Kaynak Kodlarına Örnekler;

```
// Ders classı oluşturuldu
class ders {
    constructor(kod, ad, ogretmenAd, sube, yer, kytogr, kont) {
        this.kod = kod;
        this.ad = ad;
        this.ogretmenAd = ogretmenAd;
        this.sube = sube;
        this.yer = yer;
        this.kytogr = kytogr;
        this.kont = kont;
    }
}

// Öğrenci classı oluşturuldu
class ogrenci {
    constructor(Name, OgrenciNo, eposta, dogumtarihi, iletisim, sifre) {
        this.Name = Name;
        this.OgrenciNo = OgrenciNo;
        this.eposta = eposta;
        this.dogumtarihi = dogumtarihi;
        this.iletisim = iletisim;
        this.sifre = sifre;
        this.dersListe = [];
    }
    // Öğrencinin aldığı dersleri almak için method
    ogrDersListe(dersListe) {
        this.dersListe = dersListe;
    }
}
```

- Yukarıda ders ve öğrenci yapılarını niteleyen classlar yer almaktadır.
-

```

function girisYap() {
    // Kullanıcı adı ve şifreyi inputlardan aldık
    var kullanıcıAdi = document.getElementById("userName").value;
    var sifre = document.getElementById("password").value;
    // Kullanıcılar arrayinden bilgileri eşleşen kullanıcıyı filtreledik
    const kullanıcı = kullanıcılar.filter(
        (i) => i.OğrenciNo == kullanıcıAdi && i.sifre == sifre
    );
    // Kullanıcı bilgileri eşleşmediği durumda kullanıcı değişkeninin uzunluğu olmayacağı için uyarı verildi
    if (kullanıcı.length < 1) {
        alert("Girilen kullanıcı bilgileri yanlış");
    } else {
        // Giriş yapan kullanıcının öğrenci numarası aktif değişkenine atandı
        aktif = kullanıcı[0].OğrenciNo;
        alert("Hoşgeldiniz");
        // Giriş yapıldığı için giriş formunun gizlenmesi sağlandı
        form.style.display = "none";
        // Kullanıcı bilgi ekranına geçmek için ilgili DOM elementi görünür hale getirildi
        bilgiDOM.style.display = "block";
        // İlgili DOM elemtinin içine ilgili kullanıcı bilgileri yazıldı
        bilgiDOM.innerHTML = `<table class="table-striped table-bordered" >

            <thead> Öğrenci Bilgileri:
            </thead>
            <br>
            <tbody>
                <tr>
                    <td>İsim :</td>
                    <td>${kullanıcı[0].Name}</td>
                </tr>

                <tr>
                    <td>Öğrenci Numarası :</td>
                    <td>${kullanıcı[0].OğrenciNo}</td>
                </tr>

                <tr>
                    <td>E-Posta Adresi :</td>
                    <td>${kullanıcı[0].eposta}</td>
                </tr>

                <tr>
                    <td>Doğum Tarihi :</td>
                    <td>${kullanıcı[0].dogumtarihi}</td>
                </tr>

                <tr>
                    <td>İletişim Bilgileri :</td>
                    <td>${kullanıcı[0].iletisim}</td>
                </tr>
            </tbody>
            </table>
            `;
    }
}

```

- Öğrencinin kendine ait bilgiler ile sisteme giriş yapmasını sağlayan fonksiyon yukarıda gösterilmiştir.

```
function dersEkle() {
  // Ders kodunu select value su içinde bulunan stringden almak için slice işlemi yaptık
  const value = select.value.slice(0, select.value.indexOf("|"));
  // Selectin seçili indexini aldık
  const sLessonValue = document.getElementById("dersler").selectedIndex;
  // Seçilen dersin bir daha seçilememesi için koşul
  if (
    secilenDersler.find((i) => i.kod == value)
    ? secilenDersler.find((i) => i.kod == value).kod
    : null == dersler[sLessonValue].kod
  ) {
    alert("Bu ders zaten ekli");
    console.log(secilenDersler);
  }
}
```

```
dersler.forEach((ders) => {
  // Kontenjan dolu mu değil mi diye kontrol yapıldı
  if (ders.kod == value && ders.kyogr >= ders.kont) {
    alert("Kontenjan Dolu");
  }
  // Dersin aşağıdaki koşullar sağlandığında listeye ve ders programına eklenmesini sağlayan koşul
  else if (
    ders.kod == value &&
    ders.kyogr < ders.kont &&
    secilenDersler.includes(dersler[sLessonValue]) == false
  ) {
    secilenDersler.push(ders);

    if (value == "MAT101") {
      var arr = [].slice.call(mat);
      arr.forEach((item) => {
        item.innerHTML += "Matematik";
      });
    }
    if (value == "FZK101") {
      var arr = [].slice.call(fizik);
      arr.forEach((item) => {
        item.innerHTML += "Fizik";
      });
    }
    if (value == "BİV101") {
      var arr = [].slice.call(biyo);
      arr.forEach((item) => {
        item.innerHTML += "Biyoloji";
      });
    }
    if (value == "KMY101") {
      var arr = [].slice.call(kimya);
      arr.forEach((item) => {
        item.innerHTML += "Kimya";
      });
    }
    if (value == "GEO101") {
      var arr = [].slice.call(geo);
      arr.forEach((item) => {
        item.innerHTML += "Geometri";
      });
    }
    if (value == "İSG101") {
      var arr = [].slice.call(ısg);
      arr.forEach((item) => {
        item.innerHTML += "İş Sağlığı";
      });
    }
    if (value == "ETK101") {
      var arr = [].slice.call(etik);
      arr.forEach((item) => {
        item.innerHTML += "Etik";
      });
    }
    if (value == "TRH101") {
      var arr = [].slice.call(tarih);
      arr.forEach((item) => {
        item.innerHTML += "Tarih";
      });
    }
  }
})
```

- Öğrencinin ders seçim ekranında istediği dersi daha önce seçilmemiş ise ve dersin açık kontenjanı bulunuyorsa bu dersin seçilebilmesini sağlayan fonksiyon yukarıda gösterilmiştir.

```
function dersListele() {  
  // `alınanDersler` tablosunu temizleyin.  
  const alınanDersler = document.getElementById("alınanDersler");  
  alınanDersler.innerHTML = "";  
  // `secilenDersler` dizisindeki her ders için bir satır oluşturun.  
  secilenDersler.forEach((item) => {  
    // Bir satır oluşturun.  
    const value = item.kod;  
    const tr = document.createElement("tr");  
  
    // Ders adını içeren bir hücre oluşturun.  
    const adTd = document.createElement("td");  
    adTd.textContent = item.ad;  
    tr.appendChild(adTd);  
  
    // Ders kodunu içeren bir hücre oluşturun.  
    const kodTd = document.createElement("td");  
    kodTd.textContent = item.kod;  
    tr.appendChild(kodTd);  
  
    // Sil düğmesini içeren bir hücre oluşturun.  
    const deleteTD = document.createElement("td");  
    const deleteBtn = document.createElement("button");  
    deleteBtn.textContent = "Delete";  
    deleteBtn.addEventListener("click", () => {  
      // Ders koduna göre dersi listeden silin.  
      secilenDersler = secilenDersler.filter((ders) => ders.kod !== item.kod);  
      item.kytogr -= 1;  
      liste();  
    });  
  });  
}
```

```
// Silinen ders ders progamından kaldırıldı
```

```
if (value == "MAT101") {  
  var arr = [].slice.call(mat);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "FZK101") {  
  var arr = [].slice.call(fizik);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "BİY101") {  
  var arr = [].slice.call(biyo);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "KMY101") {  
  var arr = [].slice.call(kimya);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "GE0101") {  
  var arr = [].slice.call(geo);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "ISG101") {  
  var arr = [].slice.call(isg);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "ETK101") {  
  var arr = [].slice.call(etik);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}  
  
if (value == "TRH101") {  
  var arr = [].slice.call(tarih);  
  arr.forEach((item) => {  
    item.innerHTML = "";  
  });  
}
```

```

// Silinen dersi tablodan kaldırın.
alınanDersler.removeChild(tr);
// çakışan ders olma durumu için dersler kaldırıldığında kırmızı uyarı renginin kaldırılmasını sağlar
for (let i = 10; i < 58; i++) {
    td[i].style.backgroundColor = "Inherit";
}
});
// Satıra Delete td si ve buttonunu ekleme
deleteTD.appendChild(deleteBtn);
tr.appendChild(deleteTD);

// Satırı tabloya ekleyin.
alınanDersler.appendChild(tr);
});
}

```

- Öğrencinin seçtiği dersi seçilen dersler listesinde ve dinamik olarak oluşturulmuş ders programında görmesini ve listelediği dersler arasından çıkarmak istediği dersi, ders programından çıkarabilmesini sağlayan fonksiyon yukarıda gösterilmiştir.

```

function ogrDersOnay() {
    if (!aktif) {
        alert("Lütfen önce giriş yapın.");
    }
    // Seçilen dersler öğrenci bilgi ekranına eklendi
    else {
        kullanicilar.find((i) => i.OgrenciNo == aktif).ogrDersListe(secilenDersler);

        secilenDersler.forEach((e) => e.kytogr++);
        alınanDersler.innerHTML = "";
        select.innerHTML = "";
        for (let i = 0; i < td.length; i++) {
            switch (td[i].innerHTML) {
                case "Tarih":
                    td[i].innerHTML = "";
                    break;
                case "İş Sağlığı":
                    td[i].innerHTML = "";
                    break;
                case "Etik":
                    td[i].innerHTML = "";
                    break;
                case "Geometri":
                    td[i].innerHTML = "";
                    break;
                case "Biyoloji":
                    td[i].innerHTML = "";
                    break;
                case "Fizik":
                    td[i].innerHTML = "";
                    break;
                case "Kimya":
                    td[i].innerHTML = "";
                    break;
                default:
                    break;
            }
        }
        // Selectin ui nı güncellemek için fonksiyon çağırıldı
        liste();
        alert("Dersleriniz bilgi ekranına eklendi!");
    }
}

```



```

const dersBilgi = kullanicilar
  .filter((i) => i.OgrenciNo == aktif)[0]
  .dersListe.map((i) => {
    // return `<tr>
    //   <td>Ders Kodu :${i.kod}</td>
    //   <td>Ders Adı :${i.ad}</td>
    //   <td>Öğretmen Adı :${i.ogretmenAd}</td>
    //   <td>Dersin Şubesi :${i.sube}</td>
    //   <td>Sınıf :${i.yer}</td>
    // </tr>`;
    // Ders bilgi ekranına eklenecek dersler için tablo elemanları oluşturuldu
    const table = document.createElement("table");
    const tr = document.createElement("tr");
    const kodTd = document.createElement("td");
    const adTd = document.createElement("td");
    const ogradTd = document.createElement("td");
    const subeTd = document.createElement("td");
    const yerTd = document.createElement("td");
    const tr2 = document.createElement("tr");
    const deleteTd = document.createElement("td");
    const deleteBtn = document.createElement("button");
    // Dersin bilgileri ilgili tablo elemanının içine yazıldı
    kodTd.textContent = i.kod;
    adTd.textContent = i.ad;
    ogradTd.textContent = i.ogretmenAd;
    subeTd.textContent = i.sube;
    yerTd.textContent = i.yer;
    deleteBtn.textContent = "Delete";
    // Tablo elemanları ui da ilgili yerlere yerleştirildi
    tr.appendChild(kodTd);
    tr.appendChild(adTd);
    tr.appendChild(ogradTd);
    tr.appendChild(subeTd);
    tr.appendChild(yerTd);
    tr2.appendChild(deleteTd);
    deleteTd.appendChild(deleteBtn);
    table.appendChild(tr);
    table.appendChild(tr2);
    bilgiDOM.appendChild(table);
    table.className = "table-striped table-bordered";
    deleteBtn.className = "btn btn-danger";
    // Eklenen dersin silinmesi için click fonksiyonu
    deleteBtn.addEventListener("click", () => {
      kullanicilar
        .find((i) => i.OgrenciNo == aktif)
        .dersListe.filter((j) => j.kod != i.kod);
      bilgiDOM.removeChild(table);
      i.kytogr--;
      alert(i.ad + " dersiniz başarıyla silindi!");
      // Kontenjanlar güncellensin diye gerekli işlemler
      select.innerHTML = "";
      liste();
    });
  });
});

```

- Öğrencinin seçtiği ve dinamik ders programında gördüğü dersleri onaylayarak kendi bilgi ekranında göstermesini sağlayan ve ders seçim arayüzünü güncelleyerek başlangıç haline döndüren fonksiyon yukarıda yer almaktadır.

```

// Giriş ekranının gösterilmesi için gerekli olan fonksiyon
function girisEkrani() {
  if (document.getElementById("form")) {
    form.style.display = "block";
  } else {
    console.log("form DOM'da yok");
  }
}

if (document.getElementById("bilgi")) {
  bilgiDOM.style.display = "none";
} else {
  console.log("bilgiDOM DOM'da yok");
}

if (document.getElementById("dersSecimEkrani")) {
  dersSecimEkraniDOM.style.display = "none";
} else {
  console.log("dersSecimEkrani DOM'da yok");
}
}

// Ders seçim ekranının gösterilmesi için gerekli fonksiyon
function dersSecimEkrani() {
  if (document.getElementById("dersSecimEkrani")) {
    dersSecimEkraniDOM.style.display = "block";
  } else {
    console.log("form DOM'da yok");
  }

  if (document.getElementById("bilgi")) {
    bilgiDOM.style.display = "none";
  } else {
    console.log("bilgiDOM DOM'da yok");
  }

  if (document.getElementById("form")) {
    form.style.display = "none";
  } else {
    console.log("dersSecimEkrani DOM'da yok");
  }
}

// Öğrenci bilgi ekranının gösterilmesi için gerekli olan fonksiyon
function bilgiEkrani() {
  if (document.getElementById("dersSecimEkrani")) {
    dersSecimEkraniDOM.style.display = "none";
  } else {
    console.log("form DOM'da yok");
  }

  if (document.getElementById("bilgi")) {
    bilgiDOM.style.display = "block";
  } else {
    console.log("bilgiDOM DOM'da yok");
  }

  if (document.getElementById("form")) {
    form.style.display = "none";
  } else {
    console.log("dersSecimEkrani DOM'da yok");
  }
}

```

- Kullanıcının navigasyon menüsünden seçtiği ekranı görüntüleyebilmesini sağlayan fonksiyonlar yukarıda yer almaktadır.

```
function navUpdate() {  
  if (!aktif) {  
    document.getElementById("dersSecimEkranıA").style.display = "none";  
    document.getElementById("bilgiEkranıA").style.display = "none";  
    document.getElementById("cikisyap").style.display = "none";  
  } else {  
    document.getElementById("dersSecimEkranıA").style.display = "block";  
    document.getElementById("bilgiEkranıA").style.display = "block";  
    document.getElementById("cikisyap").style.display = "block";  
  }  
}  
  
function cikisYap() {  
  aktif = null;  
  navUpdate();  
  girisEkranı();  
}
```

- Yukarıda, giriş ekranının navbar kısmında bulunan menü stilleri “giriş yap” kısmı hariç display none olarak tutulan fonksiyon ve sistemden çıkış yapılmasını sağlayan fonksiyon yer almaktadır.

Projede Karşılaşılan Problemler;

1-Eklenen derslerin silinmek istenmesi durumunda click eventinin silme butonu yerine ekleme butonunda çalışması.

Çözüm: Click eventi içerisine çağırılan silme fonksiyonu dışarıda yazılmak yerine click eventi içerisine yazılarak problem çözüme ulaşmıştır.

2- Eklenen derslerin ders programında ilgili yerlerde gösterilememesi:

Çözüm: Ders programı tablosunda ilgili hücrelere belirlenen class name’ler verilerek dersler bu class name’lere göre tablonun innerHTML yapısına yazdırılmıştır.

3- Onaylanan derslerin silinmesi durumunda dersin kontenjanının azalmaması:

Çözüm: Ders class’ı kullanılarak üretilen dersin “**kytogr**” property’sine bağlı value, “**ogrDersOnay**” fonksiyonunda oluşturulan delete butonunun click eventinde azaltılarak ve ders seçim ekranındaki ders seçim ekranı arayüzü bu değere göre güncellenerek çözüme kavuşturulmuştur.

Sonuç: Öğrencilerin sistemde ders seçmesi, seçtiği derslere göre ders programı oluşması ve bu ders programı ışığında öğrencinin dilediği dersi iptal etmesi ya da onaylamasını sağlayan bir uygulama oluşturulmuştur. Uygulamanın tüm safhaları kendi içerisinde test ve kontrollere tabi tutulup çıkan sorunlar çözüme kavuşturulmuştur. Öğrenci ders seçim sistemi problemsiz bir şekilde çalışmaktadır.

