

# Ders Programı Otomasyonu

*Fevzi Eren iek, 211307018*

*Biliřim Sistemleri Mühendislięi*

*Kocaeli Üniversitesi*

*Kocaeli, Türkiye*

*fevzierencicek@gmail.com*

*Muhammet Arda Ünsalan, 211307077*

*Biliřim Sistemleri Mühendislięi*

*Kocaeli Üniversitesi*

*Kocaeli, Türkiye*

*ardaunsalan832@gmail.com*

*Metecan ıkırıkıoęlu, 211307041*

*Biliřim Sistemleri Mühendislięi*

*Kocaeli Üniversitesi*

*Kocaeli, Türkiye*

[metel7444@gmail.com](mailto:metel7444@gmail.com)

## Özet: Ders Programı Otomasyonu

Bu proje, öğrenci ve akademisyenlerin ders programlarını düzenlemelerine ve yönetmelerine yardımcı olmak amacıyla geliştirilmiş bir otomasyon sistemini içermektedir. Proje, birinci sınıftan başlayarak tüm akademik kademelerde kullanılabilir. Front-End tarafında React, Back-End tarafında Java Spring, ve veritabanı yönetimi için SQL kullanılarak tasarlanmıştır.

Anahtar Kelimeler: Front-End, React, Back-End, Java Spring, Veritabanı, SQL

## Giriş:

Dijital çağın hızla gelişen teknolojisi, eğitim sektöründe de önemli değişikliklere neden olmuş ve bu alandaki süreçleri daha etkili bir hale getirmek adına yeni çözümleri zorunlu kılmıştır. Bu bağlamda, ders programlarının düzenlenmesi ve yönetilmesi, öğrenci ve akademisyenler için kritik bir öneme sahiptir. Bu ihtiyaca yönelik olarak geliştirilmiş olan "Ders Programı Otomasyonu" projesi, kullanıcı dostu bir arayüz ve güçlü bir teknolojik altyapı ile ders programlarının kolayca oluşturulmasını, güncellenmesini ve takip edilmesini sağlamayı hedefler.

Bu otomasyon projesi, Front-End tarafında React, Back-End tarafında Java Spring, ve veritabanı yönetimi için SQL kullanılarak geliştirilmiştir. Kullanıcıların diledikleri dersi, ilgili hoca ve sınıf bilgileriyle birlikte sisteme ekleyebilmeleri, ders programlarını kişiselleştirebilmeleri ve bu bilgileri güvenli bir veritabanında saklayabilmeleri projenin temel işlevselliğini oluşturur.

Bu giriş bölümünde, projenin genel amacı, kullanılan teknolojiler ve beklenen faydalar hakkında bir genel bakış sunulmuştur. İlerleyen bölümlerde, proje detayları, teknik özellikler ve projenin sonuçları daha ayrıntılı bir şekilde ele alınacaktır. Bu otomasyon projesi, eğitim kurumlarında ve öğrenciler arasında daha etkili ve organize bir ders planlama deneyimi sağlamayı amaçlamaktadır.

## Otomasyon Özellikleri

- **Ders Ekleme ve Güncelleme:**

Kullanıcılar, ders eklemek veya güncellemek için kullanıcı dostu bir arayüz üzerinden işlemleri gerçekleştirirler. Dersin adı, hocası, günü, saati ve sınıf bilgileri gibi temel veriler girilir.

- **Ders Silme ve Yeni Ders Ekleme:**

Kullanıcılar, mevcut bir dersi sistemden silebilir ve yerine yeni bir ders ekleyebilir. Bu sayede, programdaki değişiklikleri anında güncelleme imkanı sağlanır.

- **Veritabanı İşlemleri:**

Kullanıcıdan alınan veriler, Back-End tarafındaki Java Spring tarafından işlenir. Veritabanı yönetimi için SQL kullanılarak, ders programı verileri güvenli bir şekilde depolanır ve güncellenir.

- **Ders Programı Görselleştirmesi:**

Front-End tarafında React kullanılarak oluşturulan arayüz, kullanıcılara ders programlarını görsel olarak takip etme imkanı sunar. Dersler, gün ve saat sıralamasına göre düzenlenmiş bir formda görüntülenir.

- **Güvenlik ve Veri Bütünlüğü:**

Proje, kullanıcı bilgilerini ve ders programı verilerini güvenli bir şekilde işlemek için güçlü bir güvenlik altyapısına sahiptir. Veritabanında tutulan veriler, SQL'nin sağladığı bütünlük mekanizmaları ile korunur.

- **İyileştirme ve Gelecek Geliştirmeler:**

Projede kullanıcı geri bildirimleri göz önünde bulundurularak ve teknoloji standartlarına uygun olarak geliştirmeler yapılır. Yeni özellikler eklenir ve sistem sürekli olarak kullanıcı deneyimini artırmak üzere güncellenir.

## Otomasyonun Çalışması

Ders Kaydet

Hoca Kaydet

Bir Hoca Seçin  
\*Hoca Seçiniz\*

Ders Seçiniz  
\*Bir ders seçiniz\*

Ders Günü Seçiniz  
\*Bir Gün seçiniz\*

Dersin Sınıfını seçiniz  
\*Dersin Sınıfını seçiniz\*

--:-- ⌚ --:-- ⌚

Kaydet

Ders Tablosunu Gör

- Ders ekleyiniz kısmına eklemek istediğimiz dersin ismini gireriz ve Ders Kaydet butonuna basarız.
- Hoca ismi ve soyismi kısmına istediğimiz hocanın ismini gireriz ve Hoca Kaydet butonuna basarız.
- 3. kısımda kaydettiğimiz hoca ve dersler arasında seçim yaparız örneğin hoca kısmında Ahmet Yavuz, Necip Yılmaz ve Selin Durmuş adlı hocalarımız ve ders

kısımında ise Matematik, Web Tasarım ve Fizik II kayıtlı derslerimiz olsun. Biz buradan Ahmet Yavuz adlı hocamızın Web Tasarım adlı dersi hangi gün, hangi saat ve hangi sınıfta vereceğini sisteme kaydebiliriz ve bu eklediğimiz saatte eğer başka bir ders aynı sınıfta aynı saate varsa dersler çakışacağından o dersi ekleyemeyiz bun sorguyuda SQL ile sağlarız bu sorguyu yaptığımız ve ders bilgilerini girdilerden aldığımız SQL kodlarını aşağıda paylaştık.

```
//Ders Programı bilgilerini alan Fonksiyon
1 usage  ▲ Fevzierenn *
public List<TabloDegerleri> dersPrograminiGetir() {
    Query q = entityManager.createNativeQuery( s:"select \n" +
        "program_id,dersler.ders_isim, hocalar.isim, hocalar.soyisim, gunler.gun, siniflar.sinif_id, " +
        "baslangic_saat,bitis_saat\n" +
        " from ders_programi D\n" +
        " inner join dersler on dersler.ders_id = D.ders_id \n" +
        " inner join hocalar on hocalar.hoca_id = D.hoca_id\n" +
        " inner join gunler on gunler.gun_id = D.gun_id\n" +
        " inner join siniflar on siniflar.sinif_id = D.sinif_id order by gunler.gun_id , baslangic_saat asc");
    return q.getResultList();
}
1. }
```

```
//Ders kaydını programa ekleyen fonksiyon
1 usage  ▲ Fevzierenn *
@Transactional
public void dersKaydiniYap(DersProgrami dersProgrami) {
    Query q =entityManager.createNativeQuery( s:"" +
        "insert into ders_programi (ders_id,hoca_id,sinif_id,gun_id,baslangic_saat,bitis_saat) values " +
        "( :dersid,:hocaaid, :sinifid,:gunid,:start,:finish)");
    q.setParameter( s:"dersid",dersProgrami.getDers_id());
    q.setParameter( s:"hocaaid",dersProgrami.getHoca_id());
    q.setParameter( s:"sinifid",dersProgrami.getSinif_id());
    q.setParameter( s:"gunid",dersProgrami.getGun_id());
    q.setParameter( s:"start", dersProgrami.getBaslangic_saat());
    q.setParameter( s:"finish", dersProgrami.getBitis_saat());
    q.executeUpdate();
}
2. }
```

```
//Hocanın Ders saatinde başka bir dersinin olup olmadığını kontrol eden kod
1 usage  ▲ Fevzierenn *
public List hocaDersiBul(HocaSaatleri hoca){
    Query q =entityManager.createNativeQuery( s:"" +
        "select count(*) from ders_programi\n" +
        "where hoca_id= :id and gun_id= :gunid and " +
        "( bitis_saat between :start and :finish or baslangic_saat between :start and :finish)");
    q.setParameter( s:"id",hoca.getHoca_id());
    q.setParameter( s:"start",hoca.getBaslangic_saat());
    q.setParameter( s:"finish", hoca.getBitis_saat());
    q.setParameter( s:"gunid",hoca.getGun_id());
    return q.getResultList();
}
3. }
```

4.

```
//Dersin işleneceği sınıfta başka bir dersin olup olmadığını belirleyen kod
1 usage  Fevzierenn *
public List sınıfDersiBul(SinifSaatleri tempSinif) {
    Query q = entityManager.createNativeQuery("select count(*) from ders_programi\n" +
        " where gun_id = :gunid and sinif_id = :sinifid and " +
        "(bitis_saat between :start and :finish or baslangic_saat between :start and :finish)");
    q.setParameter("sinifid", tempSinif.getSınıf_id());
    q.setParameter("start", tempSinif.getBaslangic_saat());
    q.setParameter("finish", tempSinif.getBitis_saat());
    q.setParameter("gunid", tempSinif.getGun_id());
    return q.getResultList();
}
```

- En altta yer alan Ders Tablosunu Gör butonuna basarak o zamana kadar eklenmiş tüm dersleri, o dersleri hangi hocaların hangi gün, hangi saat ve sınıflarda verdiğini görüntüleyebiliriz, bu sayfaya ait görüntüleri aşağıda paylaştık.

Gün	Ders	İsim	Soyisim	Sınıf	Başlangıç	Bitiş	
Pazartesi	Web Tasarımı	Önder	Yakut	bsm-2	10:01:00	12:01:00	SİL
Salı	Ayrık Matematik	Süleyman	Eken	bsm-3	10:01:00	12:01:00	SİL
Salı	Bilgisayar Mimarisi	Halil	Yigit	bsm-1	14:01:00	17:01:00	SİL
Çarşamba	Sistem Analizi	Alper	Metin	bsm-4	12:01:00	13:01:00	SİL

- Bu kısımda istersek daha önceden kaydetmiş olduğumuz dersi programdan silebiliriz.

## Projede Kullanılan teknolojiler

### React(Front-End):

React, kullanıcı arayüzü oluşturmak için geliştirilmiş bir JavaScript kütüphanesidir. Facebook tarafından başlatılmış ve sürdürülmekte olan bu açık kaynaklı kütüphane, özellikle tek sayfa uygulamaları (Single Page Applications - SPA) için etkili bir şekilde kullanılmaktadır. React, kullanıcı arayüzündeki bileşenleri organize etmek ve yönetmek için kullanılan bir "component-based" (bileşen tabanlı) yaklaşımı benimser.

### **Java Spring(Back-End):**

Java Spring (kısaca Spring), Java tabanlı bir framework'tür ve özellikle büyük ölçekli, karmaşık ve ölçeklenebilir uygulamaların geliştirilmesine odaklanır. Spring Framework, Java uygulamalarının geliştirilmesini ve sürdürülmesini kolaylaştırmak için bir dizi modül ve kütüphane sunar. Pek çok farklı bileşeni içerir ve geniş bir yelpazede uygulama geliştirme konularına odaklanır.

### **SQL(Database):**

SQL (Structured Query Language), ilişkisel veritabanı yönetim sistemlerinde (RDBMS) veritabanlarını yönetmek, sorgulamak ve işlemek için kullanılan bir programlama dilidir. SQL, özellikle ilişkisel veritabanlarıyla etkileşimde bulunmak için tasarlanmış bir dil olarak öne çıkar. Temel amacı, veritabanları üzerinde sorgular oluşturarak veri manipülasyonu yapmak, veri eklemek, güncellemek, silmek ve sorgularla veri çekmek olan bir dilidir.

### **Sonuç:**

"Ders Programı Otomasyonu" projesi, eğitim kurumlarında ve bireylerde ders organizasyonunu optimize etmeyi amaçlayan bir çözüm sunmaktadır. Bu projenin geliştirilmesi ve uygulanması, ders programlarını düzenleme ve takip etme süreçlerini daha etkili ve verimli hale getirmiştir. Aşağıda, projenin ana sonuçları özetlenmiştir:

- **Kullanıcı Dostu Arayüz ve Etkileşim:**  
React ve Java Spring kullanılarak oluşturulan bu otomasyon, kullanıcı dostu bir arayüz ve etkileşim sunar. Bu sayede kullanıcılar, ders programlarını kolayca düzenleyebilir ve takip edebilir.
- **Veri Güvenliği ve Bütünlüğü:**  
Projede kullanılan güçlü güvenlik altyapısı, kullanıcı bilgilerini ve ders programı verilerini güvenli bir şekilde saklar. SQL veritabanı, veri bütünlüğünü korumak için etkili bir şekilde kullanılır.
- **Esneklik ve Kişiselleştirme:**  
Kullanıcılar, ders programlarını kişiselleştirme imkanına sahiptir. Ders ekleyebilme, güncelleyebilme, silme ve yerine yeni bir ders ekleyebilme gibi esneklik, kullanıcıların ihtiyaçlarına uygun çözümler sunar.
- **Geliştirme ve Gelecek İlerlemeler:**  
Projede kullanıcı geri bildirimleri dikkate alınarak sürekli geliştirmeler yapılır. İlerleyen dönemlerde, yeni özelliklerin eklenmesi ve kullanıcı deneyiminin daha da iyileştirilmesi planlanmaktadır.

### ***Kaynakça***

1. <https://www.baeldung.com/spring-data-jpa-query-by-date> 05.01.2024
2. <https://stackoverflow.com/questions/50956554/create-native-sql-query-without-creating-entity-class-in-springboot> 02.01.2024
3. <https://www.javatpoint.com/java-tutorial> 28.12.2023
4. <https://docs.spring.io/spring-data/data-jpa/docs/current/api/org/springframework/data/jpa/repository/JpaRepository.html>
5. <https://www.baeldung.com/spring-data-jpa-delete> 25.12.2023
6. <https://upmostly.com/tutorials/how-to-refresh-a-page-or-component-in-react#:~:text=import%20React%20from%20'react'%3B,refreshPage%7D%3EClick%20to%20reload!> 20.12.2023