



2024-2025 Eğitim-Öğretim Yılı Bahar Dönemi

**YAZILIM GELİŞTİRME ve ORTAM ARAÇLARI FINAL PROJESİ**

Adı Soyadı/Name Surname: Nesibe İrem Sevim

Numarası/Number: HR210016

MYO/YO/Fakülte/Enstitü (School): MYO

Bölüm/Program (Dept.): Bilgisayar Programcılığı İ.Ö

Dersin Adı/Course Name: YAZILIM GELİŞTİRME ve ORTAM ORAÇLARI

Öğretim Elemanı/Instructor: ÖZKAN SARI

Öğrenci İmzası/Student Signature:

# STUDENT ASSIGNMENT MANAGEMENT SYSTEM - FINAL PROJE RAPORU

## PROJE BILGILERI

**Proje Adı:** Student Assignment Management System

**GitHub Repository Linki:** <https://github.com/iremmsvm/studentproject>

## PROJE SUREC VE MODELLERI

### Yazılım Süreç Modeli:

Bu projede **Agile** yaklaşımı benimsenmiştir.

- Proje adımları "sprint" benzeri küçük geliştirme aşamalarına ayrıldı.
- Her aşamada yeni bir özellik eklendi ve test edildi.

### Aşama Aşama Proje Süreci:

1. **Proje Planlaması:** Gerekli REST servislerin ve veritabanı yapısının belirlenmesi.
2. **Spring Boot Projesinin Oluşturulması:** <https://start.spring.io> üzerinden Maven ve Java 17 kullanarak proje yaratıldı.
3. **Model Katmanının Tasarımı:** Student ve Assignment entity sınıfları yazıldı.
4. **Controller ve Repository Katmanları:** CRUD fonksiyonlarını içeren REST servisler geliştirildi.
5. **Veritabanı Entegrasyonu:** H2 kullanılarak veritabanı yapılandırıldı.
6. **Testler:** JUnit ile birim testler yazıldı ve coverage raporlandı.
7. **CI/CD:** GitHub Actions ile Maven build entegrasyonu sağlandı.
8. **API Testleri:** Postman ve JMeter ile testler gerçekleştirildi.
9. **Rapor Hazırlama ve GitHub'a Ekleme:** Bu rapor README ile birlikte repoya yüklandı.

## WEB SERVISLER VE AÇIKLAMALAR

### AssignmentController.java:

- **POST /assignments:** Yeni bir ödev ekler.
- **GET /assignments:** Ödevleri tipiyle birlikte listeler.

### StudentController.java:

- **POST /students/assign:** Öğrenciye ödev atar (input: studentId, assignmentId).
- **GET /students:** Öğrencileri ve ödevlerini listeler.

## Kullanılan Annotation ve Yapı:

- `@RestController`, `@RequestMapping`, `@PostMapping`, `@GetMapping`, `@Autowired`

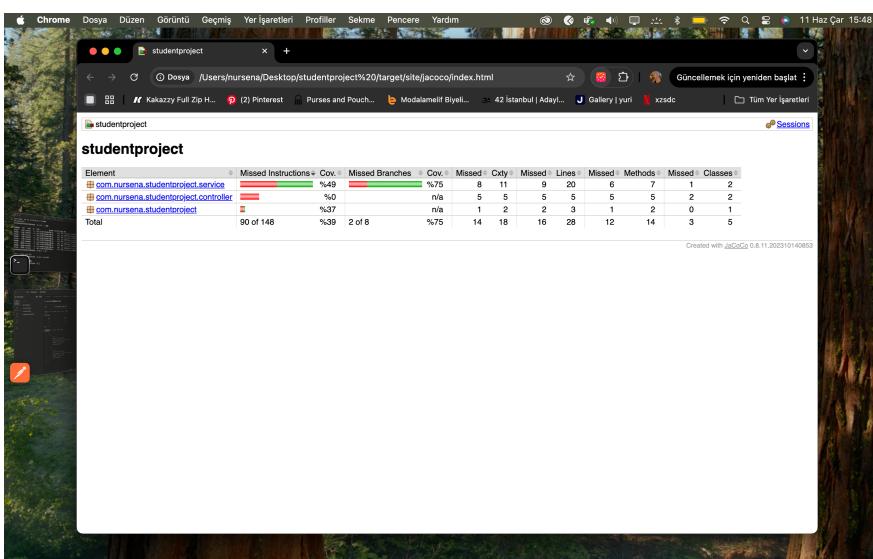
## H2 Database Yapısı:

- Console: `http://localhost:8080/h2-console`
- JDBC URL: `jdbc:h2:mem:testdb`

---

## BIRIM TESTLER & COVERAGE RAPORU

- JUnit kullanılarak Assignment ve Student servisleri test edildi.
- Maven aracılığıyla su komut çalıştırıldı:
  - `mvn test`
  - `mvn jacoco:report`
- Rapor `target/site/jacoco/index.html` olarak oluştu.
- Maven build başarı ekran görüntüsü ve coverage ekranı eklendi



## GITHUB ACTIONS - CI/CD

- `.github/workflows/ci.yml` dosyası tanımlandı.
- Her push veya PR için `mvn clean install` çalıştırılıyor.
- PR test edildi ve build başarıyla tamamlandı.

# POSTMAN & JMETER TESTLERİ

## Postman:

- Assignment ekleme, listeleme
- Student ekleme, ödev atama, listeleme
- Testler koleksiyon olarak kaydedildi

The screenshot shows the Postman interface with a collection named "Functional testing". A GET request is made to `http://localhost:9090/assignments`. The response status is 200 OK, and the JSON body contains the following data:

```
[{"id": 1, "topic": "Java Streams", "description": "Stream API soruları", "dueDate": "2025-06-30", "questionCount": 10}, {"id": 2, "topic": "Java Lambda", "description": "Fonksiyonel programlama", "dueDate": "2025-06-20", "questionCount": 10}]
```

The screenshot shows the Postman interface with a collection named "Functional testing". A POST request is made to `http://localhost:9090/assignments`. The request body is in `x-www-form-urlencoded` format with the following parameters:

Key	Value	Description
topic	Java Streams	
dueDate	2025-06-30	
type	exercise	
questionCount	10	
description	Stream API soruları	

The response status is 200 OK, and the JSON body shows the newly created assignment object.

```
{ "id": 1, "topic": "Java Streams", "description": "Stream API soruları", "dueDate": "2025-06-30", "questionCount": 10}
```

## JMeter:

- 100 kullanıcıyla GET istekleri test edildi
- Ortalama cevap süresi ve başarı oranı raporlandı

## **GIT VE GITHUB KULLANIMI**

### **Komutlar:**

```
git init  
git remote add origin https://github.com/iremmsvm/studentproject.git  
git add .  
git commit -m "Initial commit"  
git push -u origin main
```

- GitHub Actions, README.md ve bu rapor da repoya yüklandı.
- GitHub repo public hale getirildi.