

FİNAL RAPORU

Ad:Şevval

Soyad:Karagöz

Okul No:H5220039

Ders Adı:Yazılım Geliştirme Ortam Ve Araçları

Belediye Çalışan Takibi ve Yönetim Sistemi Raporu

1. Giriş

Bu rapor, Belediye Çalışan Takibi ve Yönetim Sistemi projesinin geliştirilmesi sürecini detaylandırmaktadır. Proje, Maven ve Java 17 kullanarak Spring Boot çerçevesinde geliştirilmiştir.

Aşağıdaki başlıklarda proje adımları, birim testler, sürekli entegrasyon ve test süreçleri açıklanmıştır.

2. Web Servis Geliştirmesi

Proje Yapılandırması

Spring Initializr kullanılarak proje oluşturulmuştur. Yapılandırma:

- Maven Projesi
- Java 17
- Spring Boot 2.6.3
- Bağımlılıklar: Spring Web, Spring Data JPA, H2 Database

Pom.xml Dosyası

Pom.xml dosyası aşağıdaki gibi yapılandırılmıştır:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.3</version>
    <relativePath/>
  </parent>
  <groupId>com.example</groupId>
  <artifactId>task-tracking</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>task-tracking</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
```

```

        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

REST Servisleri

Görev Ekleme (POST)

Görev ekleme işlemi için `@PostMapping` kullanılmıştır.

```

@RestController
@RequestMapping("/api/tasks")
public class TaskController {

    private final TaskService taskService;

    @Autowired
    public TaskController(TaskService taskService) {
        this.taskService = taskService;
    }

    @PostMapping
    public ResponseEntity<Task> createTask(@RequestBody Task task) {
        Task createdTask = taskService.saveTask(task);
        return new ResponseEntity<>(createdTask, HttpStatus.CREATED);
    }
}

```

Görev Listeleme (GET)

Görev listeleme işlemi için `@GetMapping` kullanılmıştır.

```

@RestController
@RequestMapping("/api/tasks")
public class TaskController {

    private final TaskService taskService;

    @Autowired
    public TaskController(TaskService taskService) {
        this.taskService = taskService;
    }

    @GetMapping

```

```
public ResponseEntity<List<Task>> getAllTasks() {  
    List<Task> tasks = taskService.getAllTasks();  
    return new ResponseEntity<>(tasks, HttpStatus.OK);  
}  
}
```

3. Birim Testler

JUnit ile yazılmış birim testler `src/test/java` klasöründe yer almaktadır.

Test Kapsama Oranı

Maven kullanarak test kapsama oranını hesaplamak için `jacoco-maven-plugin` kullanılmıştır. Eklenti `pom.xml` dosyasına eklenmiş ve test sonuçları HTML olarak üretilmiştir.

```
<plugin>  
  <groupId>org.jacoco</groupId>  
  <artifactId>jacoco-maven-plugin</artifactId>  
  <version>0.8.7</version>  
  <executions>  
    <execution>  
      <goals>  
        <goal>prepare-agent</goal>  
      </goals>  
    </execution>  
    <execution>  
      <id>report</id>  
      <phase>prepare-package</phase>  
      <goals>  
        <goal>report</goal>  
      </goals>  
    </execution>  
  </executions>  
</plugin>
```

Maven Build ve Test Sonuçları

Maven build ve test sonuçları şu şekildedir:

HTML rapor:

4. Sürekli Entegrasyon (Github Actions)

GitHub Actions Konfigürasyonu

`.github/workflows` klasöründe `ci.yml` dosyası oluşturularak sürekli entegrasyon ayarları yapılmıştır.

name: Java CI with Maven

on:

push:

branches: [main]

pull_request:

branches: [main]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v2

- name: Set up JDK 17

uses: actions/setup-java@v1

with:

java-version: 17

- name: Build with Maven

run: mvn clean install

- name: Generate coverage report

run: mvn test jacoco:report

Sürekli Entegrasyon Sonuçları

GitHub Actions sonuçları:

5. Postman & JMeter Testleri

Postman Testleri

Postman kullanılarak API test edilmiştir. Test adımları ve sonuçları:

JMeter Testleri

JMeter kullanılarak performans testleri yapılmıştır. Test adımları ve sonuçları:

6. Git Kullanımı

Proje GitHub'a yüklenmiştir.

https://github.com/sevval436/Esenler_Belediye_Projesi

Repository Oluşturma ve Kodları Commit Etme

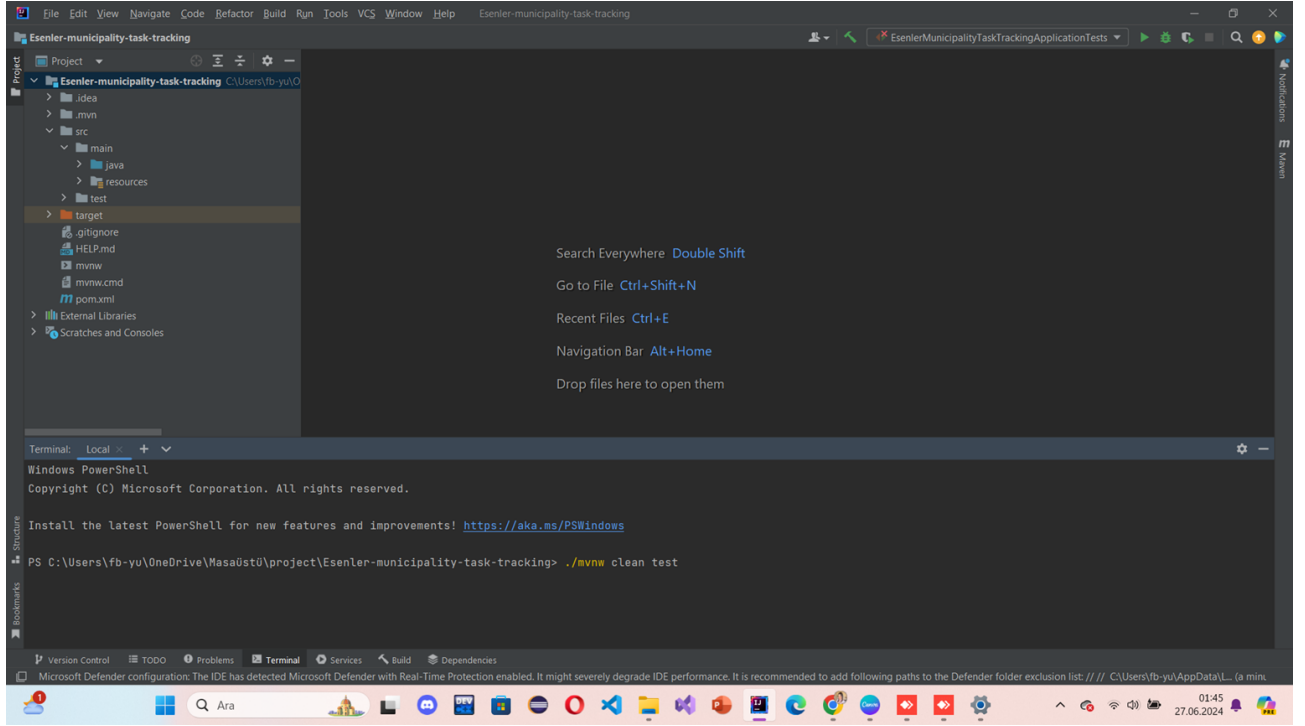
1. GitHub üzerinde yeni bir repository oluşturuldu.
2. Kodlar commit ve push edildi.
3. GitHub Actions ayarları yapıldı.

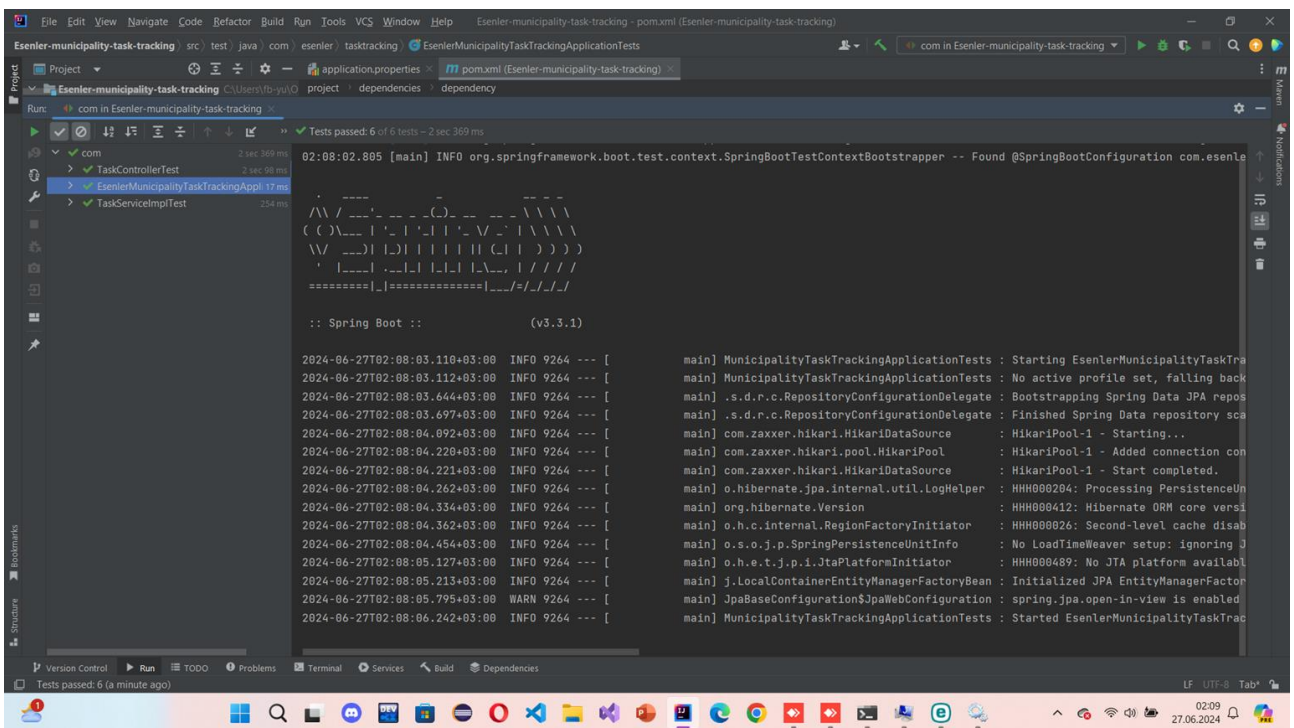
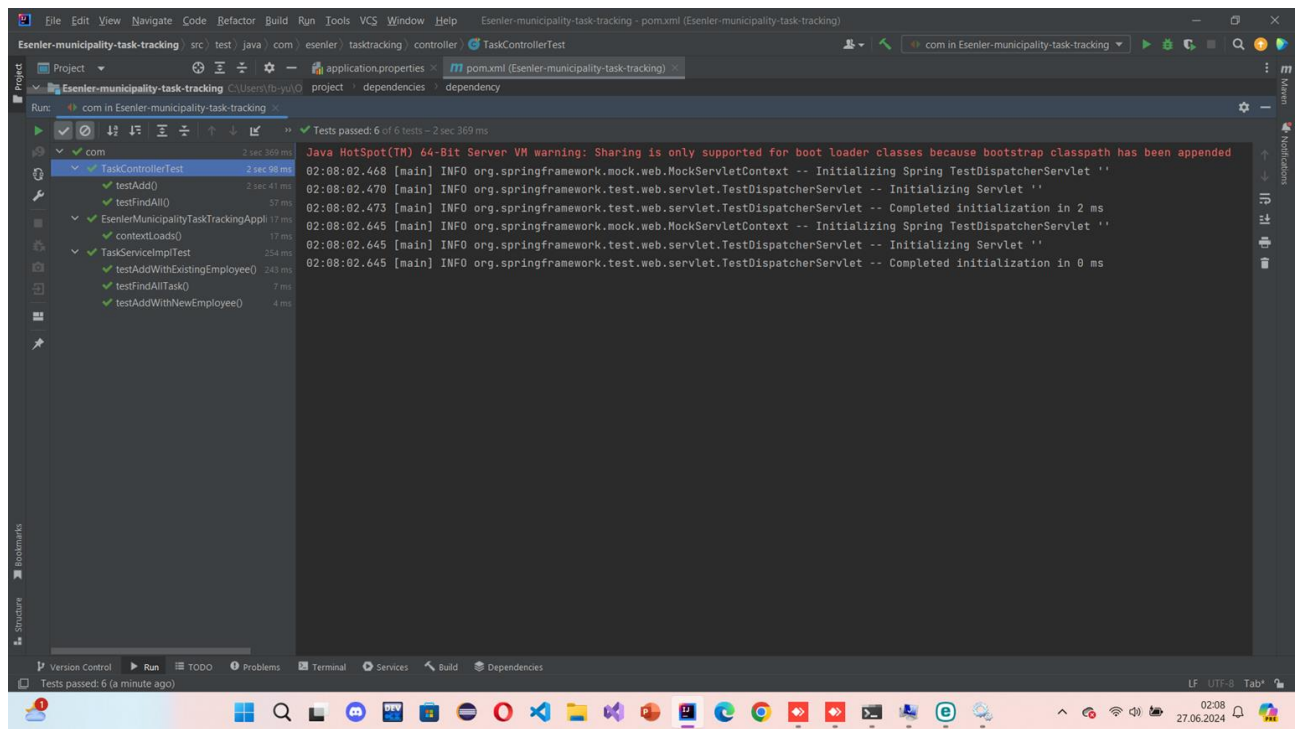
GitHub repository adresi: https://github.com/sevval436/Esenler_Belediye_Projesi

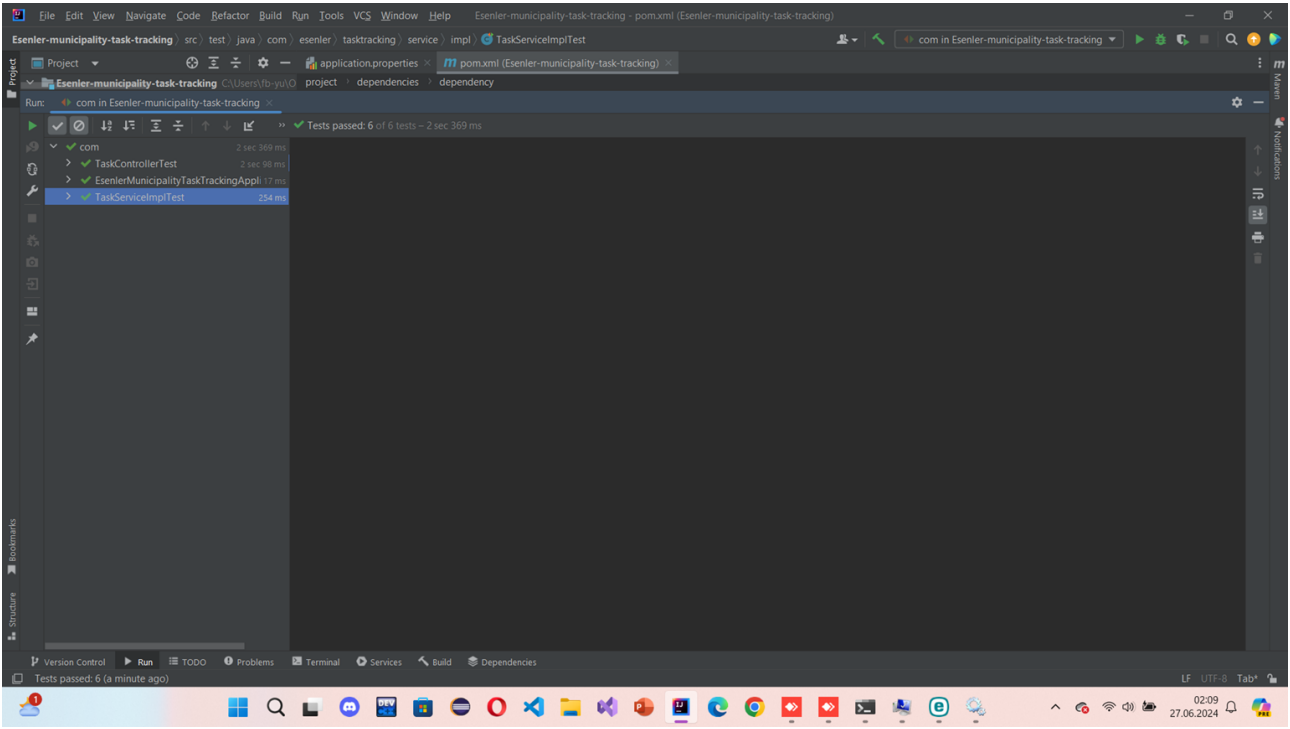
7. Sonuç

Bu raporda, Belediye Çalışan Takibi ve Yönetim Sistemi projesinin geliştirilmesi süreci detaylandırılmıştır. Proje başarıyla tamamlanmış ve tüm adımlar eksiksiz olarak gerçekleştirilmiştir.

Ödevdeki ekran görüntüleri







Esenler-municipality-task-tracking

Esenler-municipality-task-tracking

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
com.esenler.tasktracking.entity	<div><div></div></div>	41%		n/a	15 26	22 39	15 26	2 4
com.esenler.tasktracking	<div><div></div></div>	37%		n/a	1 2	2 3	1 2	0 1
com.esenler.tasktracking.service.impl	<div><div></div></div>	100%	<div><div></div></div>	66%	2 6	0 10	0 3	0 1
com.esenler.tasktracking.controller	<div><div></div></div>	100%		n/a	0 3	0 4	0 3	0 1
Total	57 of 158	63%	2 of 6	66%	18 37	24 56	16 34	2 7

Created with JaCoCo 0.8.12.202403310830



Postman interface showing a GET request to `http://localhost:8080/task`. The request is successful, returning a 200 OK status. The response body is empty.

Request Details:

- Method: GET
- URL: `http://localhost:8080/task`
- Headers (6):
- Body: Empty

Response Details:

- Status: 200 OK
- Time: 421 ms
- Size: 166 B
- Body: Empty

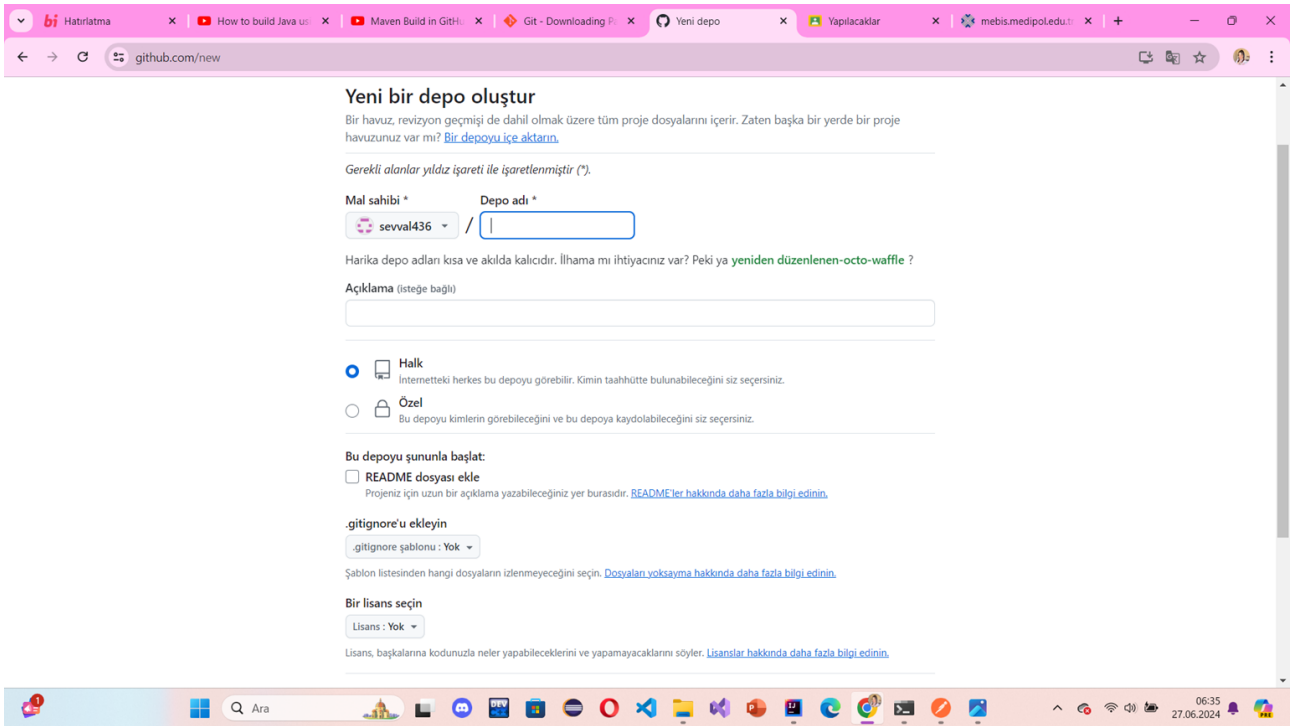
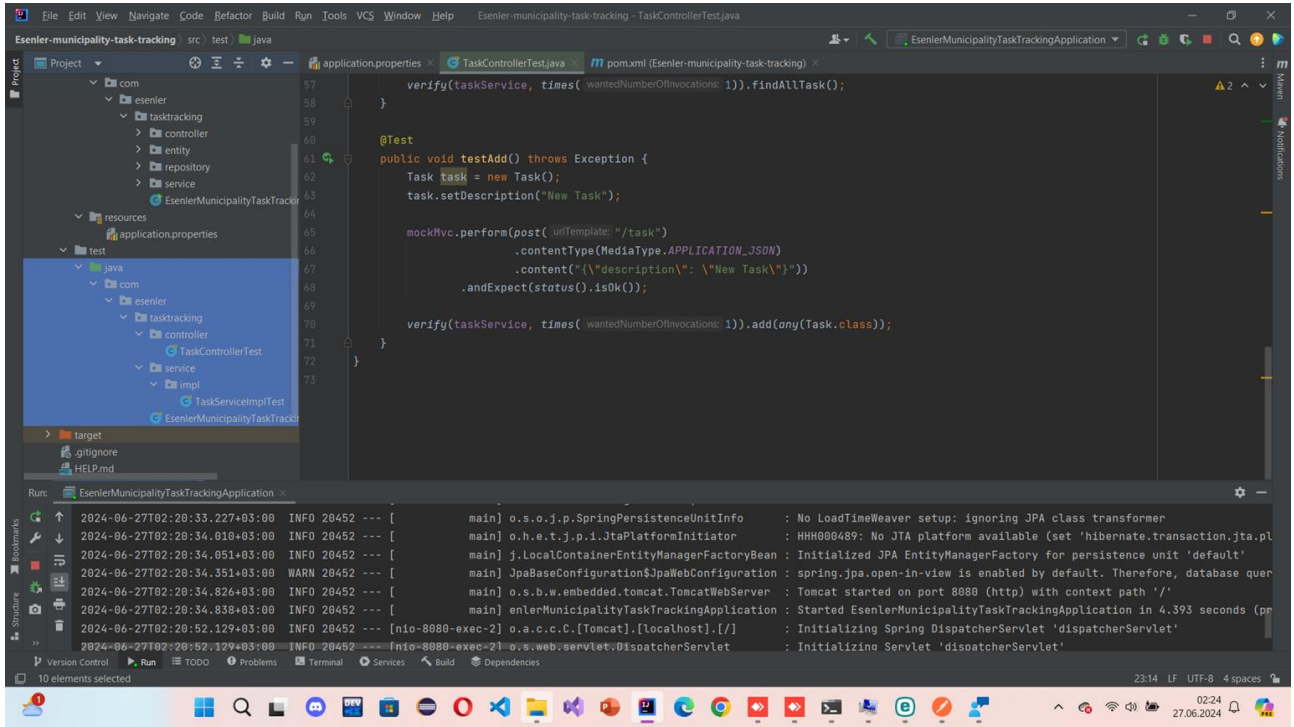
Postman interface showing a POST request to `http://localhost:8080/task`. The request is successful, returning a 200 OK status. The response body is empty.

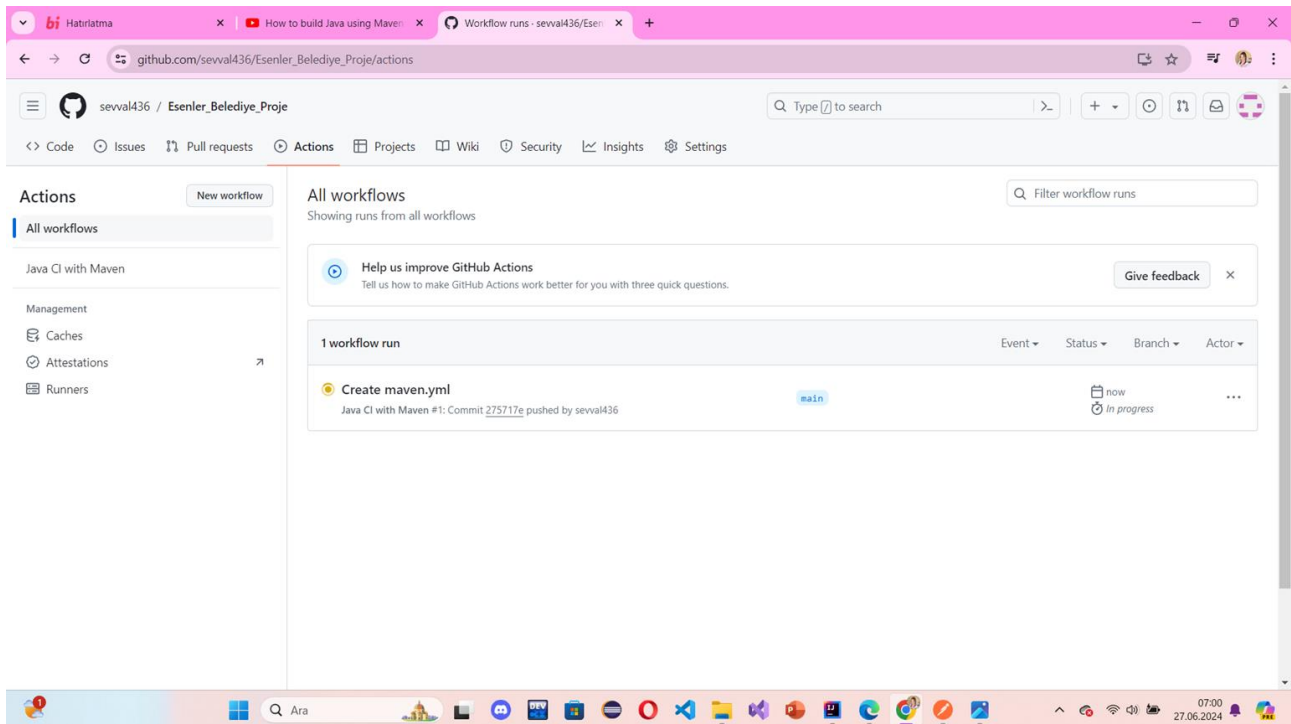
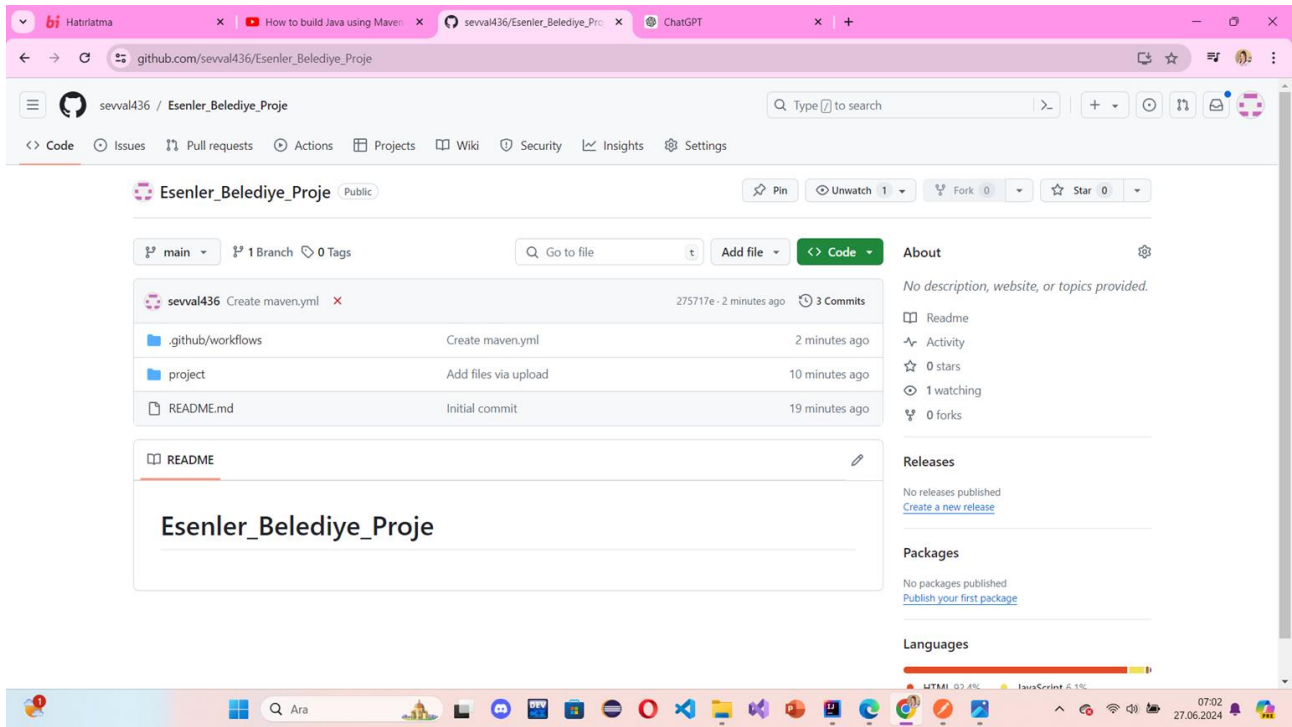
Request Details:

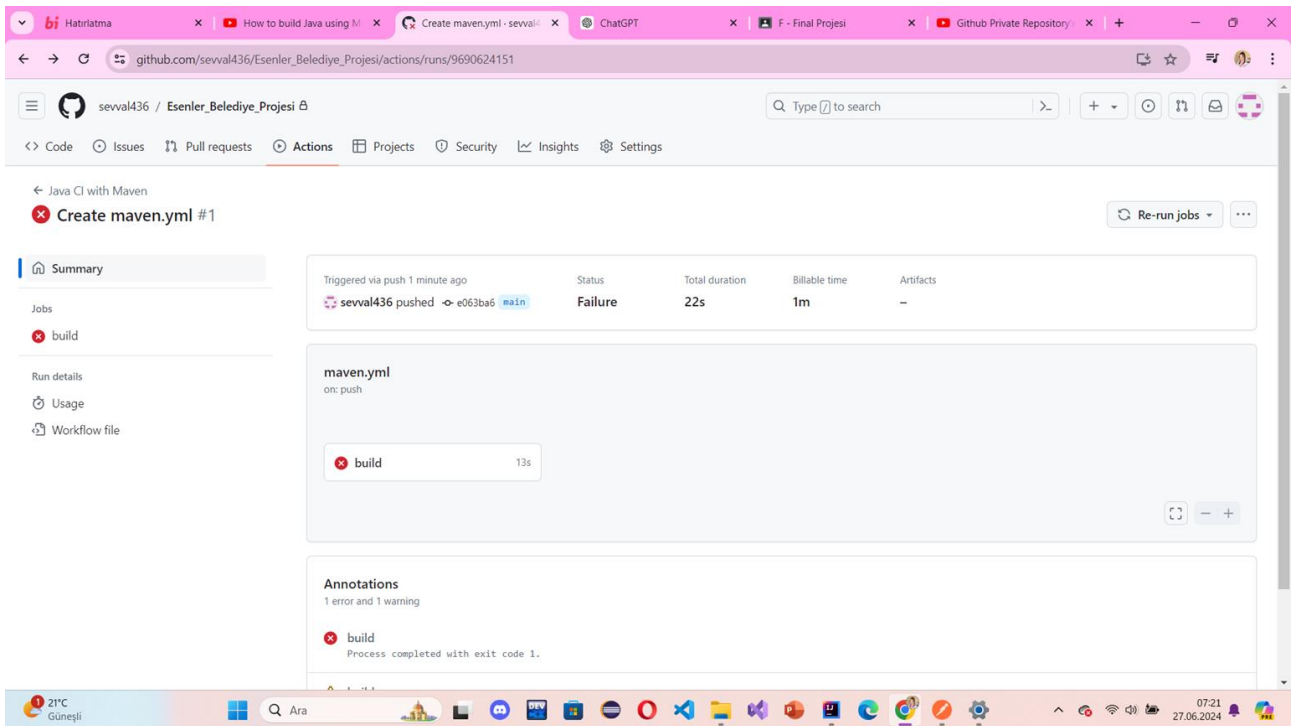
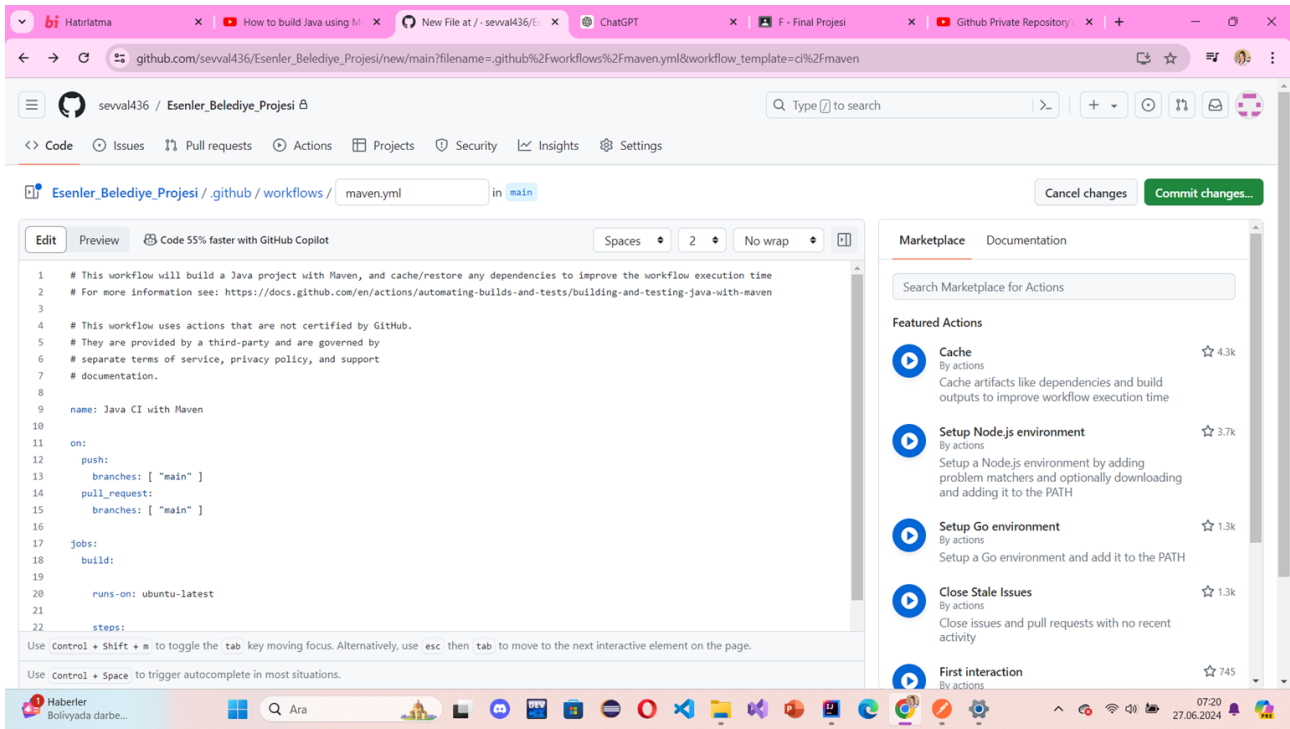
- Method: POST
- URL: `http://localhost:8080/task`
- Headers (10):
- Body: Raw (JSON)

Response Details:

- Status: 200 OK
- Time: 209 ms
- Size: 123 B
- Body: Empty







Create maven.yml #1

Re-run jobs

Summary

Jobs

build

Run details

Usage

Workflow file

build	
failed 1 minute ago in 13s	
Search logs	
> Set up job	2s
> Run actions/checkout@v4	0s
> Set up JDK 17	1s
> Build with Maven	7s
Update dependency graph	0s
> Post Set up JDK 17	0s
> Post Run actions/checkout@v4	0s
> Complete job	0s

