

EXAM SCHEDULE ORGANISER

A COURSE PROJECT REPORT

By

PRIYANSHI DAVID

(RA2011028010039)

Under the Guidance of

Dr. M Jeyaselvi

for the course of

18CSC343T - Web Application Development



Department of Networking and Communications

Faculty of Engineering and Technology

SRM Institute of Science and Technology

SRM Nagar, Kattankulathur - 603203



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
SRM NAGAR, KATTANKULATHUR-603203**

BONAFIDE CERTIFICATE

Register No: RA2011028010039

Certified that the project title “EXAM SCHEDULE APP” is the bonafide work of **PRIYANSHI DAVID (RA2011028010039)**, who carried out the project work under my supervision for the course 18CSC343T - Web Application development during the academic Year 2022-23.

Date:

Signature

Dr. M Jeyaselvi

Assistant Professor

Department of Networking & Communications

SRM Institute of Science and Technology

SRM Nagar, Kattankulathur

TABLE OF CONTENTS

S. NO	CONTENT	PAGE
1.	Abstract	
2.	Introduction to Problem Statement	
3.	Design	
4.	Components	
5	Code Snippets	
6	Output Snippets	
7	Acknowledgement and References	

ABSTRACT:

This project is a web application for managing Exam Schedule. Students often compile a time table of subjects for their entire examination season. The objective is to organise and maintain an effective examination schedule for students to refer to whenever they need it.

PROBLEM STATEMENT:

Exam period is stressful. A lot of subjects have to be studied in very little time. This can lead to frustration and confusion among the students. This Exam Schedule List will help them organise their subjects according to date and time, preventing any problems in the future. It is always advisable to stick to an accurate pre-prepared exam time-table.

A simple Exam Schedule web application can be developed using a combination of Technologies, such as HTML, CSS, JavaScript, Database Management, Spring Boot Framework. All these technologies were taught in the mentioned course for the academic semester.

SCOPE OF THE PROJECT:

Web Design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design, interface design, including standardised code and proprietary software, user experience design, and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term web design is normally used to describe the

design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark-up then they are also expected to be up to date with web accessibility guidelines. Web Development is a broad term for the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing the simplest static single page of plain text to the most complex web-based internet applications, electronic businesses, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web design, web content development, client liaison, client-side/server-side scripting, web server and network security configuration, and e-commerce development. Smaller organisations may only require a single permanent or contracting webmaster, or secondary assignment to related job positions such as a graphic designer and/or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department

METHODOLOGY:

To implement the above goals, following methodology needs to be followed:

1. Specifying the application and various components of the Web application.
2. Specifying the Models, Repo, Controllers for the Spring Boot Framework Code.
3. Designing the Front end on Javascript for Basic Shopping List activity.
4. Specifying the Port Interconnections between the resources.

5. Analysis: Extracting the Data Required for analysis
6. Testing the Final environment under Various parameters through Postman Dashboard.

SYSTEM DESIGN:

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. It emphasises translating design. Specifications to performance specification.

INPUT DESIGN: Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimising the data entry and providing a multi-user facility. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organised into groups of similar data.

OUTPUT DESIGN: Computer output is the most important and direct source of information to the user. Output design is a very important phase since the out- put needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output.

DATABASE DESIGN: Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the Data stored.

SYSTEM COMPONENTS

1. FRONT END:

- a. HTML
- b. CSS
- c. JavaScript

2. BACK END:

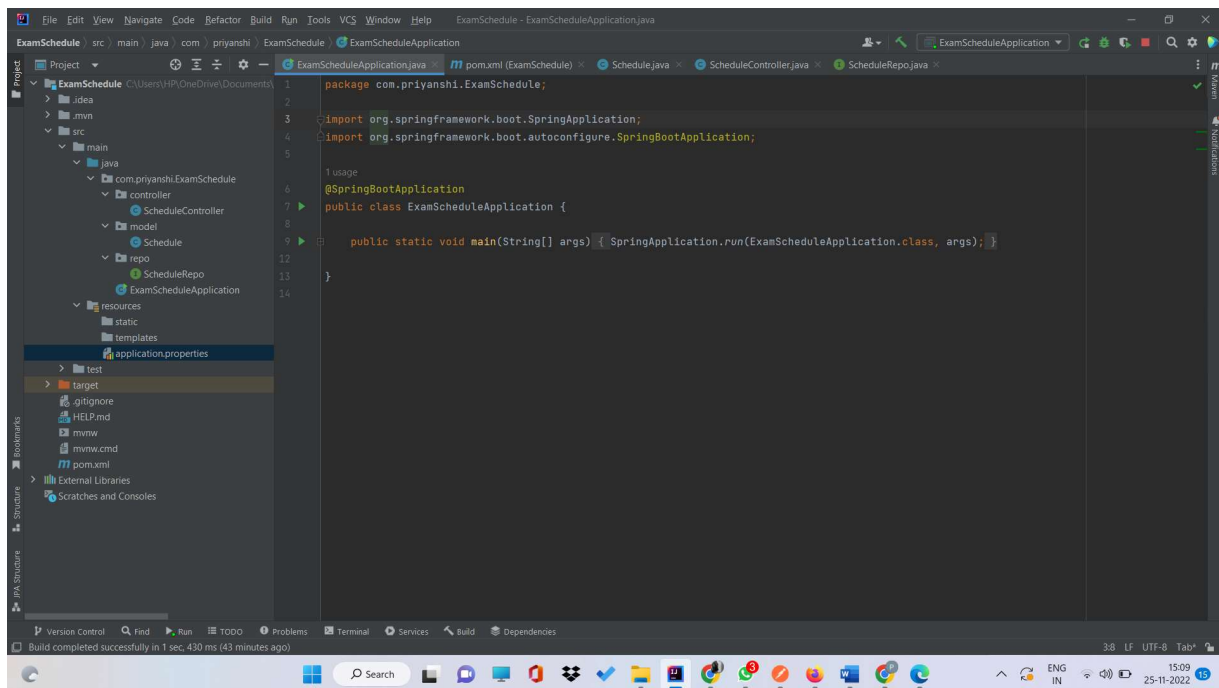
- a. Spring Boot Framework - Java Implementation
- b. Postman API - calling
 - i. PUT
 - ii. POST
 - iii. GET
 - iv. DELETE

3. DATABASE:

- a. MySQL
- b. MongoDB
- c. H2 Database - RDBMS for Java Applications

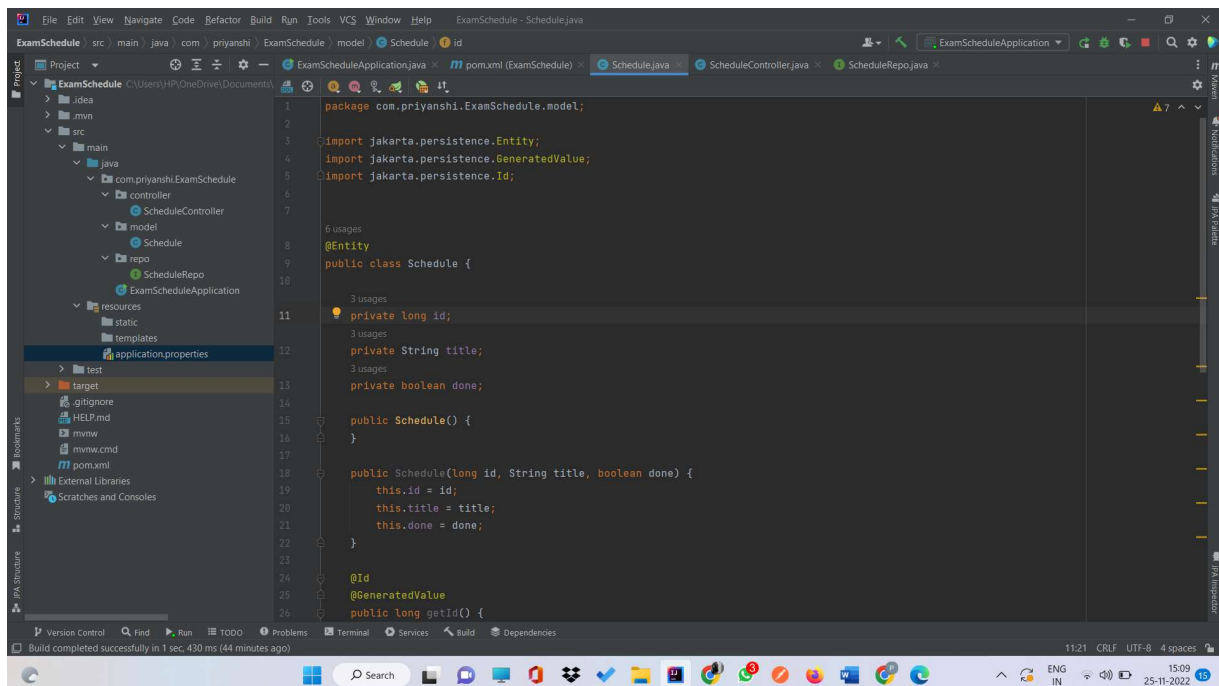
CODING SNIPPETS:

ExamScheduleApplication.Java

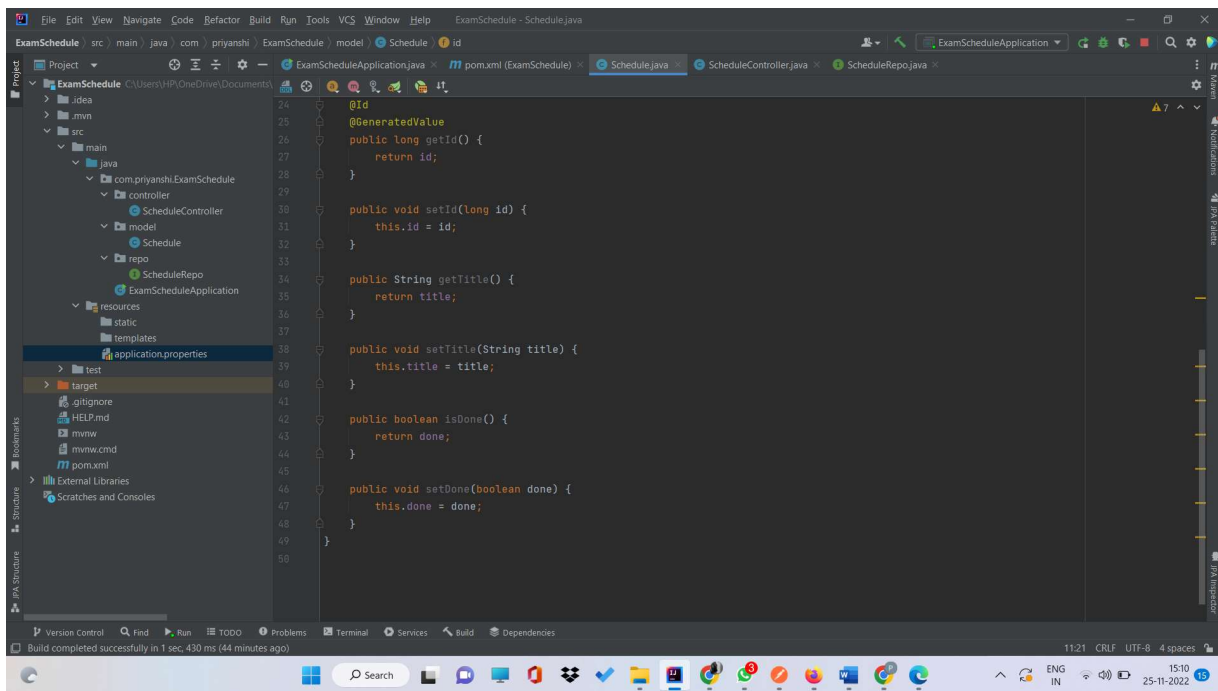


```
1 package com.priyanshi.ExamSchedule;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class ExamScheduleApplication {
8
9     public static void main(String[] args) { SpringApplication.run(ExamScheduleApplication.class, args); }
10 }
11
```

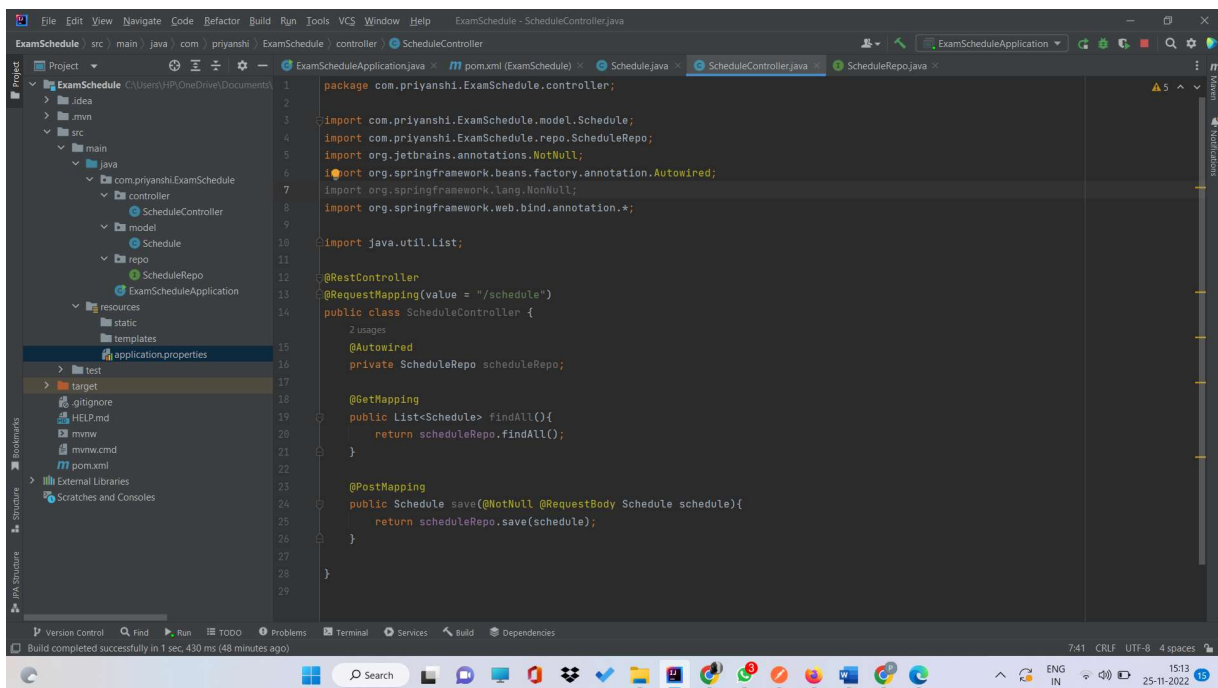
Schedule.Java



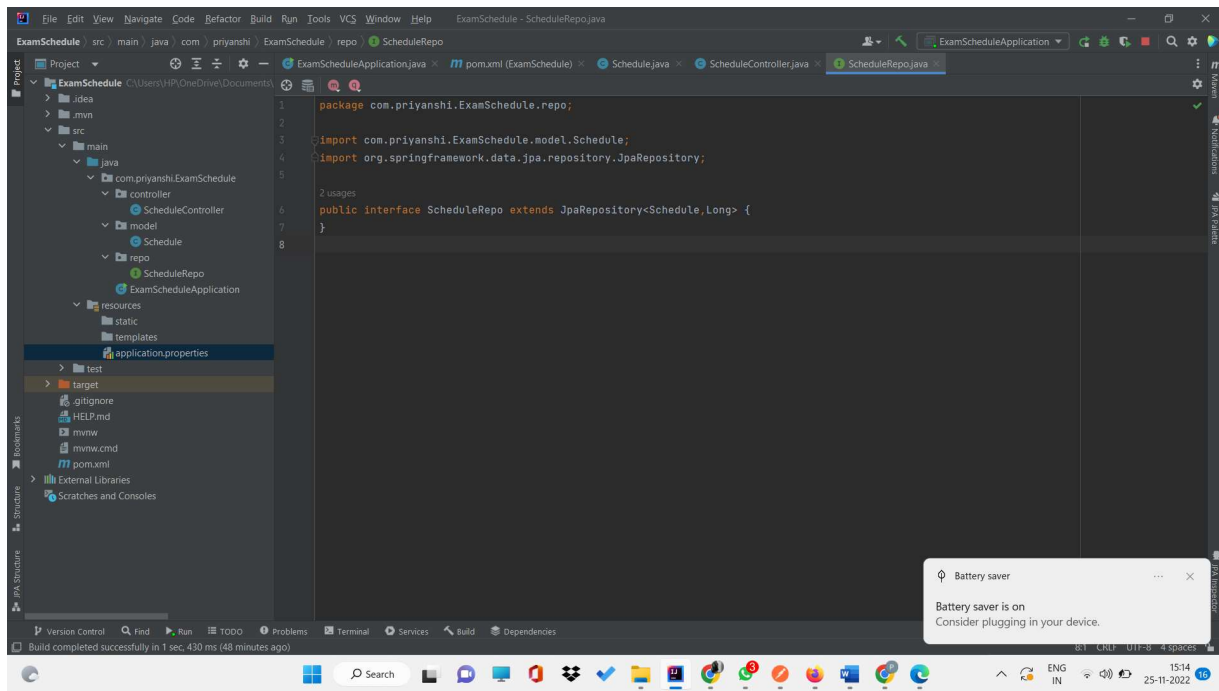
```
1 package com.priyanshi.ExamSchedule.model;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.Id;
6
7 @Entity
8 public class Schedule {
9
10     private long id;
11     private String title;
12     private boolean done;
13
14     public Schedule() {
15     }
16
17     public Schedule(long id, String title, boolean done) {
18         this.id = id;
19         this.title = title;
20         this.done = done;
21     }
22
23     @Id
24     @GeneratedValue
25     public long getId() {
26     }
27 }
28
```

ScheduleController.Java



ScheduleRepo.Java



Schedule.java CODE :

```
package com.priyanshi.ExamSchedule.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;

@Entity
public class Schedule {

    private long id;
    private String title;
    private boolean done;

    public Schedule() {
    }

    public Schedule(long id, String title, boolean done) {
        this.id = id;
        this.title = title;
        this.done = done;
    }

    @Id
    @GeneratedValue
    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }
}
```

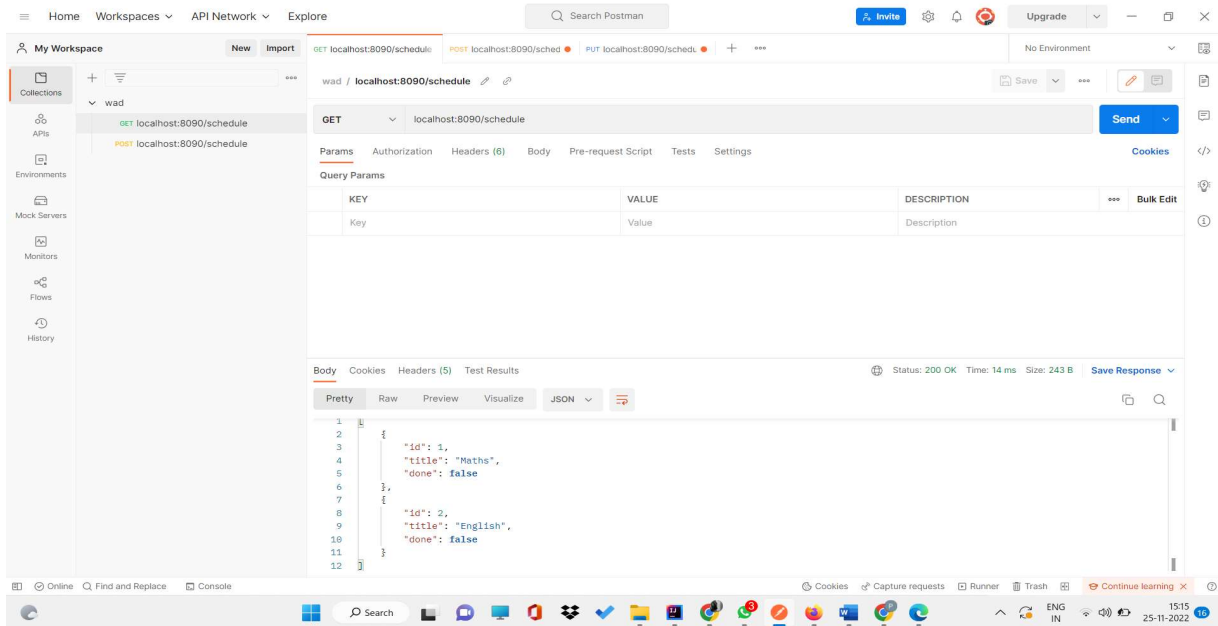
```
public boolean isDone() {  
    return done;  
}  
  
public void setDone(boolean done) {  
    this.done = done;  
}  
}
```

ScheduleController.Java

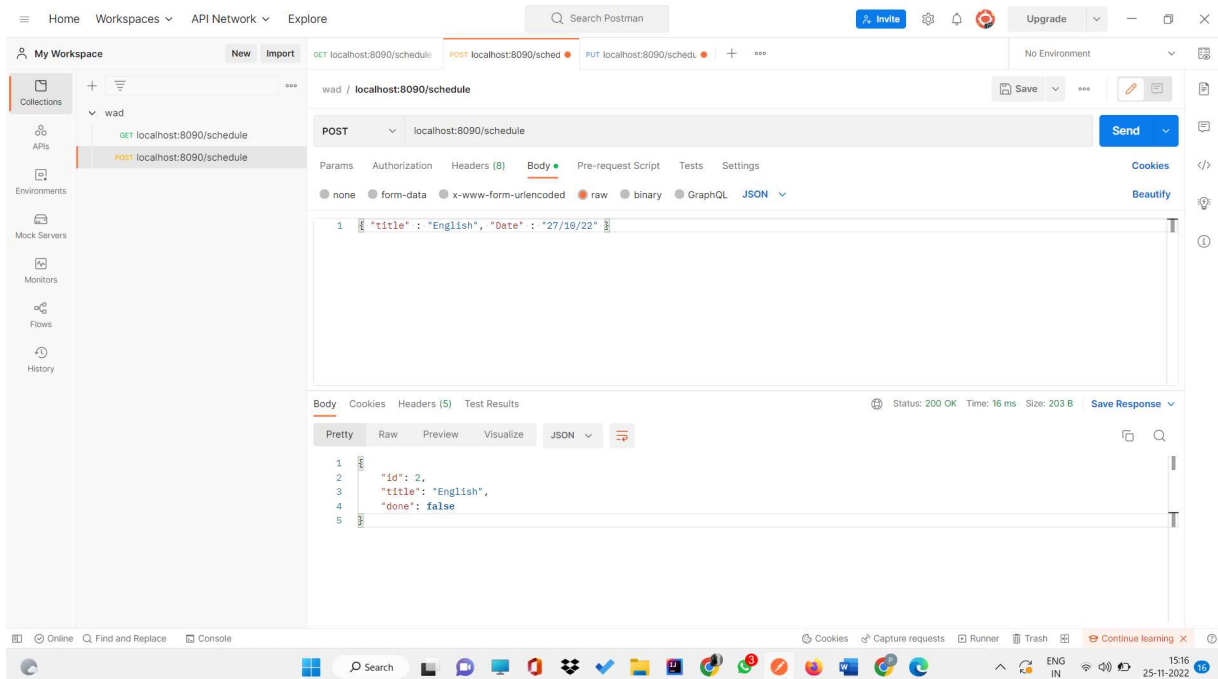
```
package com.priyanshi.ExamSchedule.controller;  
  
import com.priyanshi.ExamSchedule.model.Schedule;  
import com.priyanshi.ExamSchedule.repo.ScheduleRepo;  
import org.jetbrains.annotations.NotNull;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.lang.NonNull;  
import org.springframework.web.bind.annotation.*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping(value = "/schedule")  
public class ScheduleController {  
    @Autowired  
    private ScheduleRepo scheduleRepo;  
  
    @GetMapping  
    public List<Schedule> findAll(){  
        return scheduleRepo.findAll();  
    }  
  
    @PostMapping  
    public Schedule save(@NotNull @RequestBody Schedule schedule){  
        return scheduleRepo.save(schedule);  
    }  
}
```

POSTMAN OUTPUTS:

1. GET Method :



2. POST Method :



ACKNOWLEDGEMENT

I would like to thank Jeyaselvi Ma'am for giving me an opportunity to work on this project as well as constant encouragement towards its implementation.

I would also like to express my gratitude and appreciation towards all those who directly or indirectly contributed towards the completion of this project.

Without their help and guidance, this project would not have been completed.

REFERENCES

Tutorial followed: <https://www.youtube.com/watch?v=dyA-4QPSowA>

Documentation: <https://spring.io/guides>

Spring Boot Initialisation: <https://start.spring.io/>