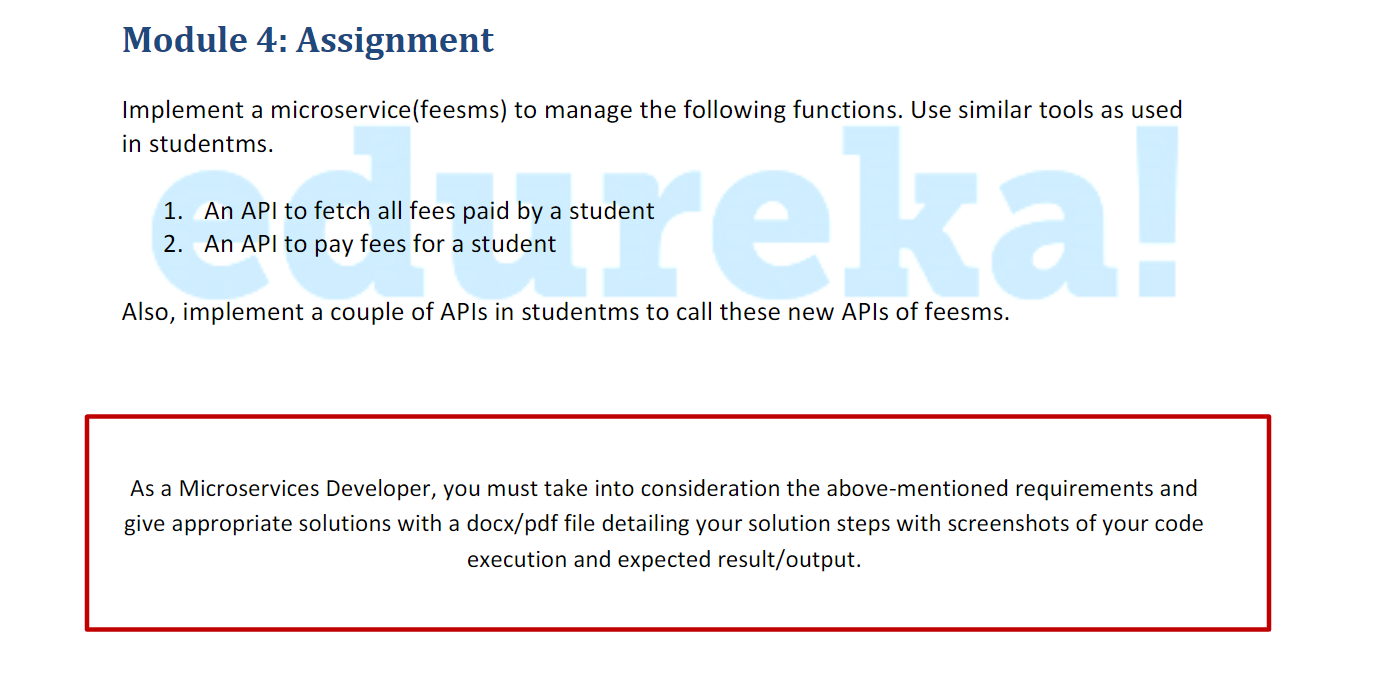
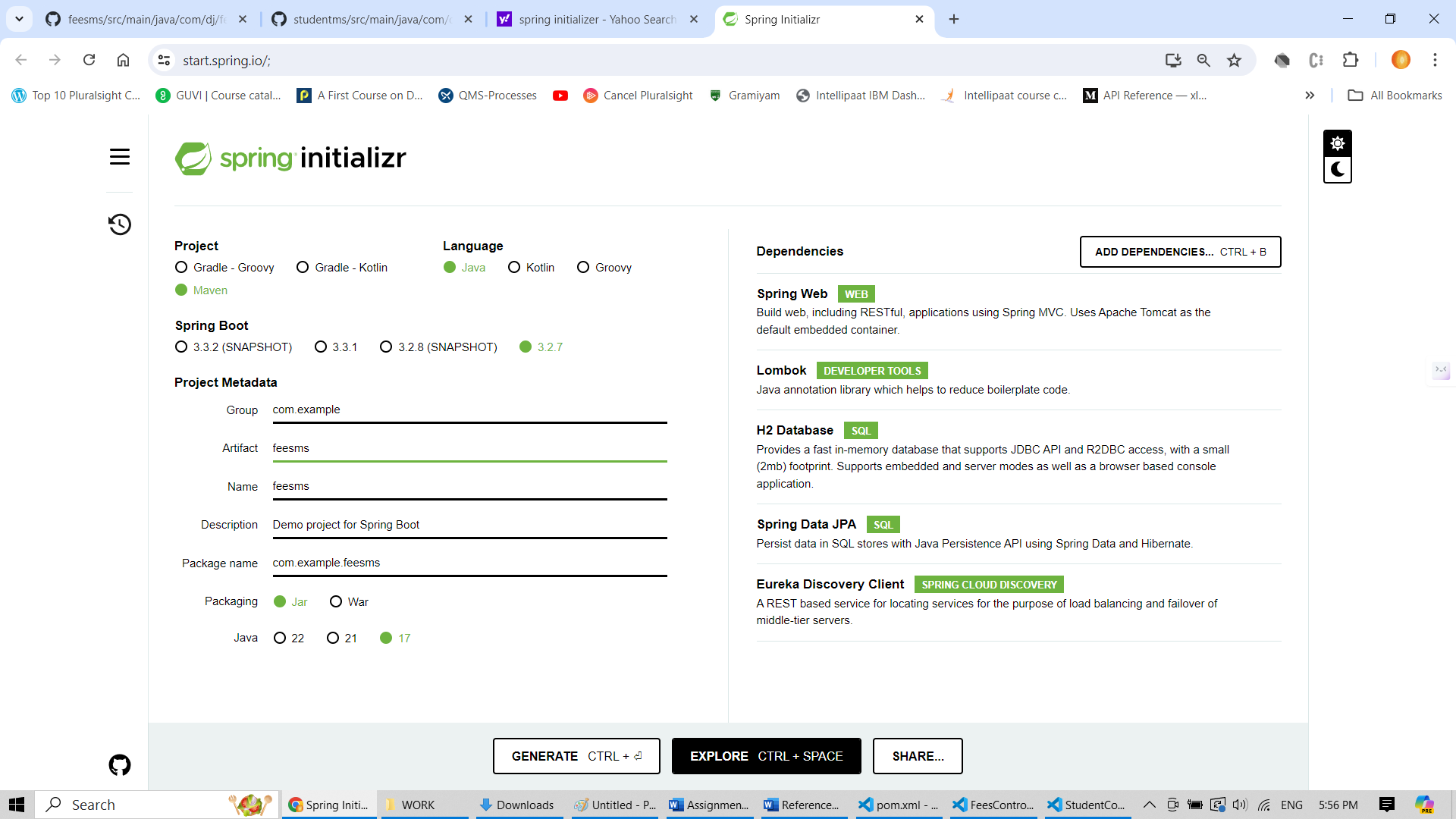
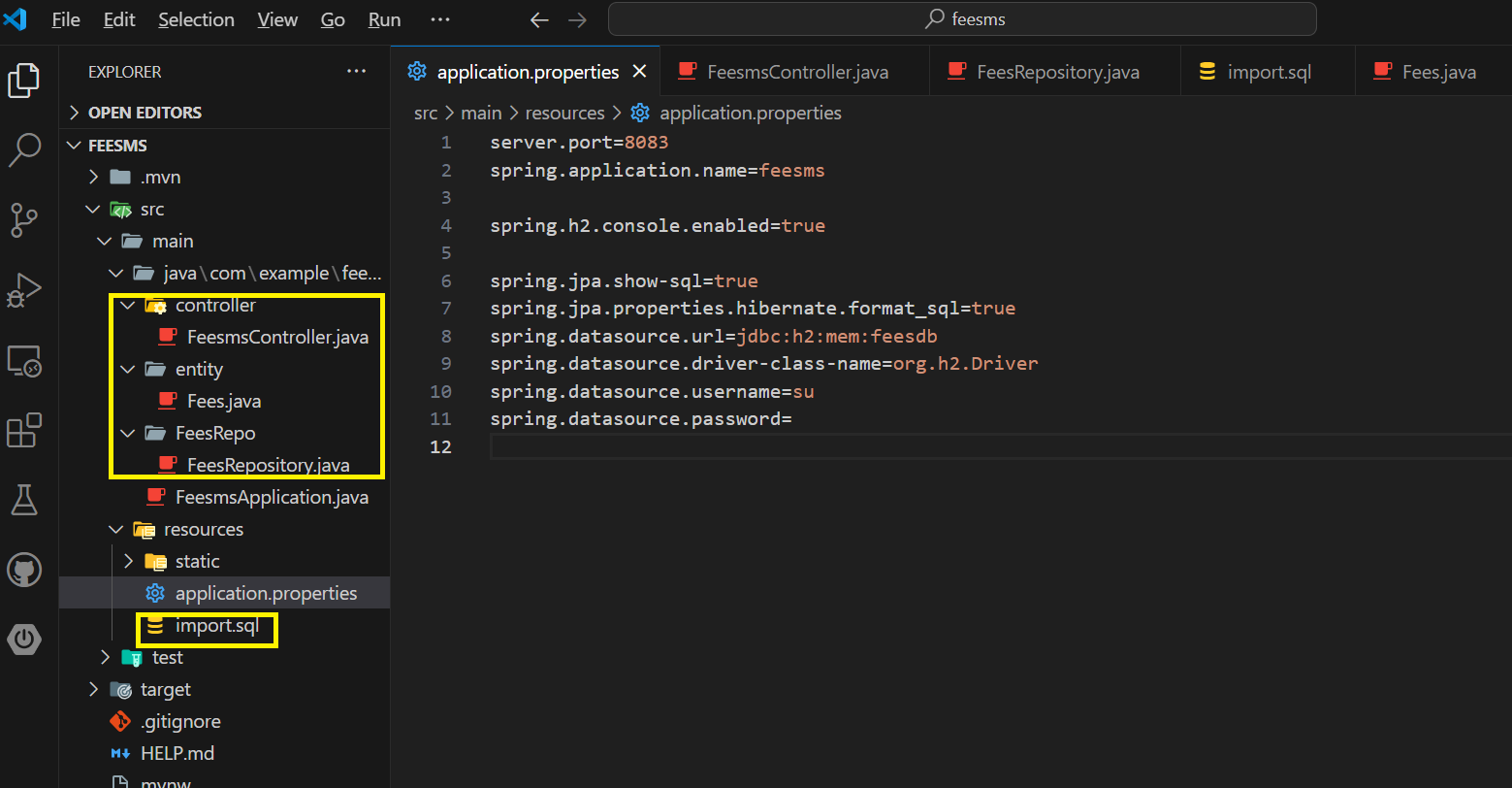
Problem 4:



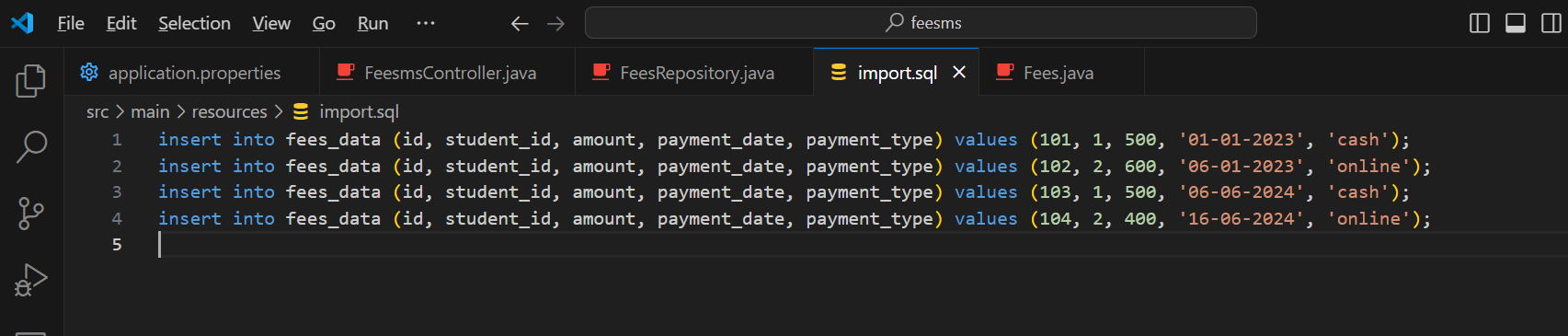
Step 1: create project feesms with following dependencies



2. Application.properties content for supporting H2, for configuring port



3. Default H2 data



4. Controller sources

package com.example.feesms.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.feesms.FeesRepo.FeesRepository;

import com.example.feesms.dto.FeesDto;

import com.example.feesms.entity.Fees;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

@RestController

@RequestMapping("/fees")

public class FeesmsController {

    @Autowired

    public FeesRepository feesrepo;

    // @GetMapping("/hello")

    // public String getMethodName() {

    //     return "Hello from feesms";

    // }

    //An API to fetch all fees paid by a student

    @GetMapping("fees/student/{studentId}")

    public ResponseEntity<List<Fees>> getStudentFees(@PathVariable("studentId") Integer studentId){

        List<Fees> studentFees = feesrepo.findByStudentId(studentId);

        return new ResponseEntity<>(studentFees, HttpStatus.OK);

    }

    //An API to pay fees for a student

    @PostMapping("fees/student/payfees/{studentId}")

    public ResponseEntity<Fees> payStudentFees(@PathVariable("studentId") Integer studentId, @RequestBody FeesDto feesDto) {

        Fees newFees = new Fees();

        newFees.setAmount(feesDto.getAmount());

        newFees.setStudentId(studentId);

        newFees.setPaymentDate(feesDto.getPaymentDate());

        newFees.setPaymentType(feesDto.getPaymentType());

        Fees savedfees = feesrepo.save(newFees);

        return new ResponseEntity<>(savedfees, HttpStatus.OK);

    }

}

6.Entity soruces

package com.example.feesms.entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.Getter;

import lombok.Setter;

@Entity

@Table(name = "fees\_data")

@Getter

@Setter

public class Fees {

    @Id

    @Column(name = "id")

    private Integer id;

    @Column(name = "student\_id")

    private Integer studentId;

    @Column(name = "amount")

    private double amount;

    @Column(name = "payment\_date")

    private String paymentDate;

    @Column(name = "payment\_type")

    private String paymentType;

}

7. Repository class soruces

package com.example.feesms.FeesRepo;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.example.feesms.entity.Fees;

@Repository

public interface FeesRepository extends JpaRepository<Fees, Integer> {

    List<Fees> findByStudentId(Integer studentId);

}

8.Dto class source

package com.example.feesms.dto;

import lombok.Getter;

import lombok.Setter;

@Getter

@Setter

public class FeesDto {

    private double amount;

    private String paymentDate;

    private String paymentType;

    // Optionally, you can include a constructor for convenience

    public FeesDto(double amount, String paymentDate, String paymentType) {

        this.amount = amount;

        this.paymentDate = paymentDate;

        this.paymentType = paymentType;

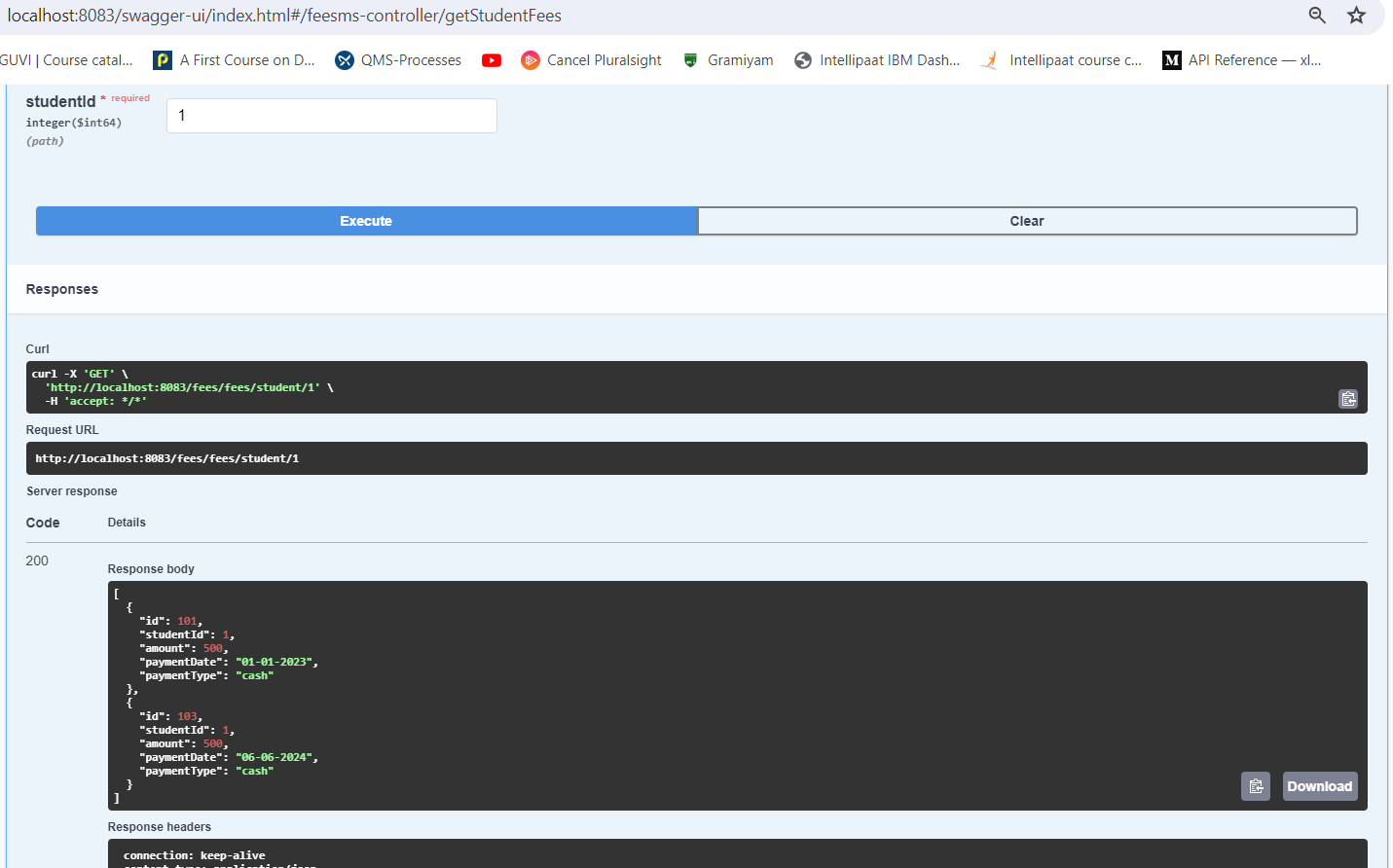
    }

    // Default constructor

    public FeesDto() {}

}

9: Output : An api to fetch all fees paid by a student



10. An api to pay fees for a student

