Assignment5

Module 5: Microservices with Springboot and Cloud

Objective:

- Design an application for student management system that includes all the functionalities and concepts learned in this course
- Follow the instructions to create solutions according to the concepts taught in each module

Problem Statement:

A popular education society "EduExcellence" is setting up a new school in the city. They want a "School Management Software" that should match the high standards of the school that they aspire to build. They have hired you to provide them with the software that should enable them to manage all aspects of running a school successfully, but they would want the application to grow gradually with the growth of the new school.

Module 5: Assignment

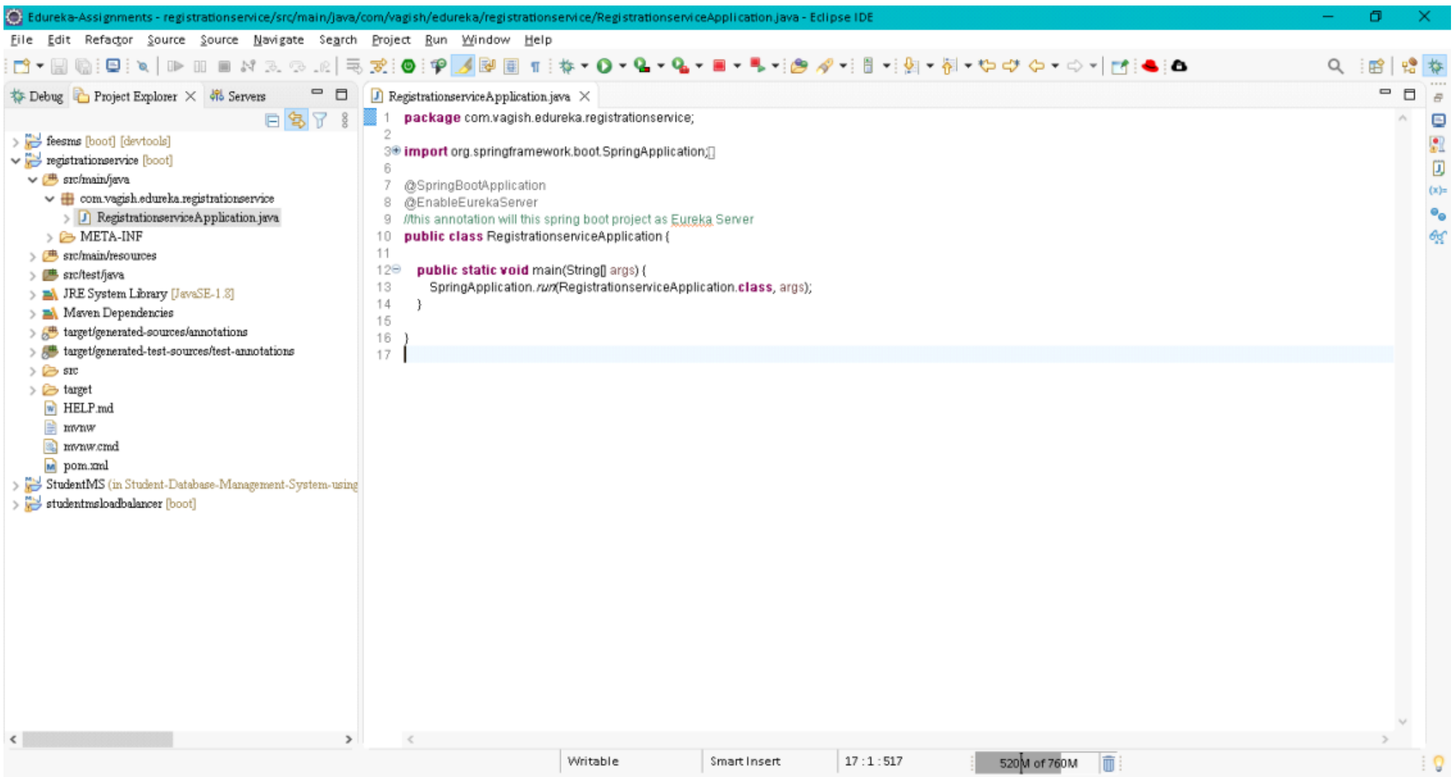
Use relevant microservice design patterns in your previous implementation:

1. Service registry/discovery
2. Client-side load balancing,
3. API Gateway

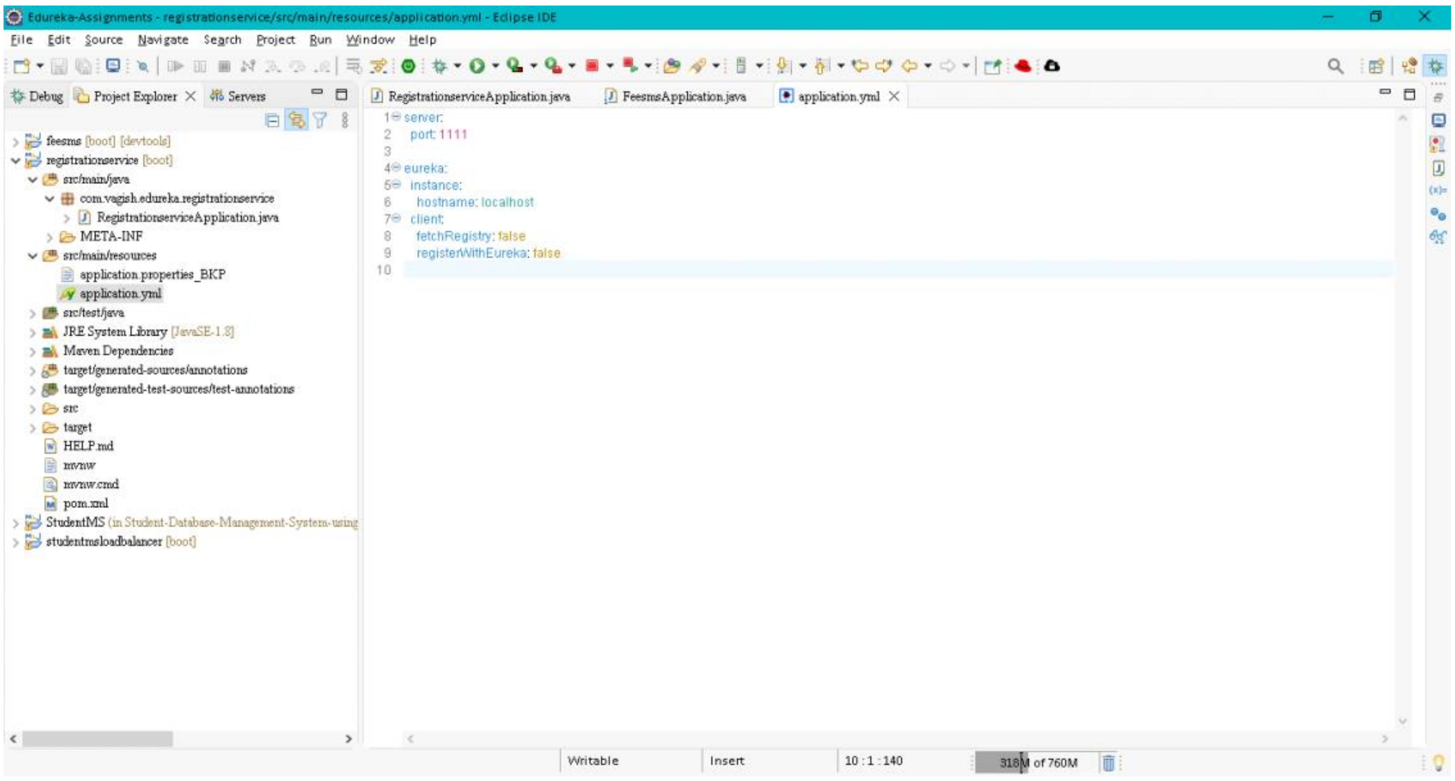
Hint: Use Spring Cloud projects.

Solution:

Registration Service

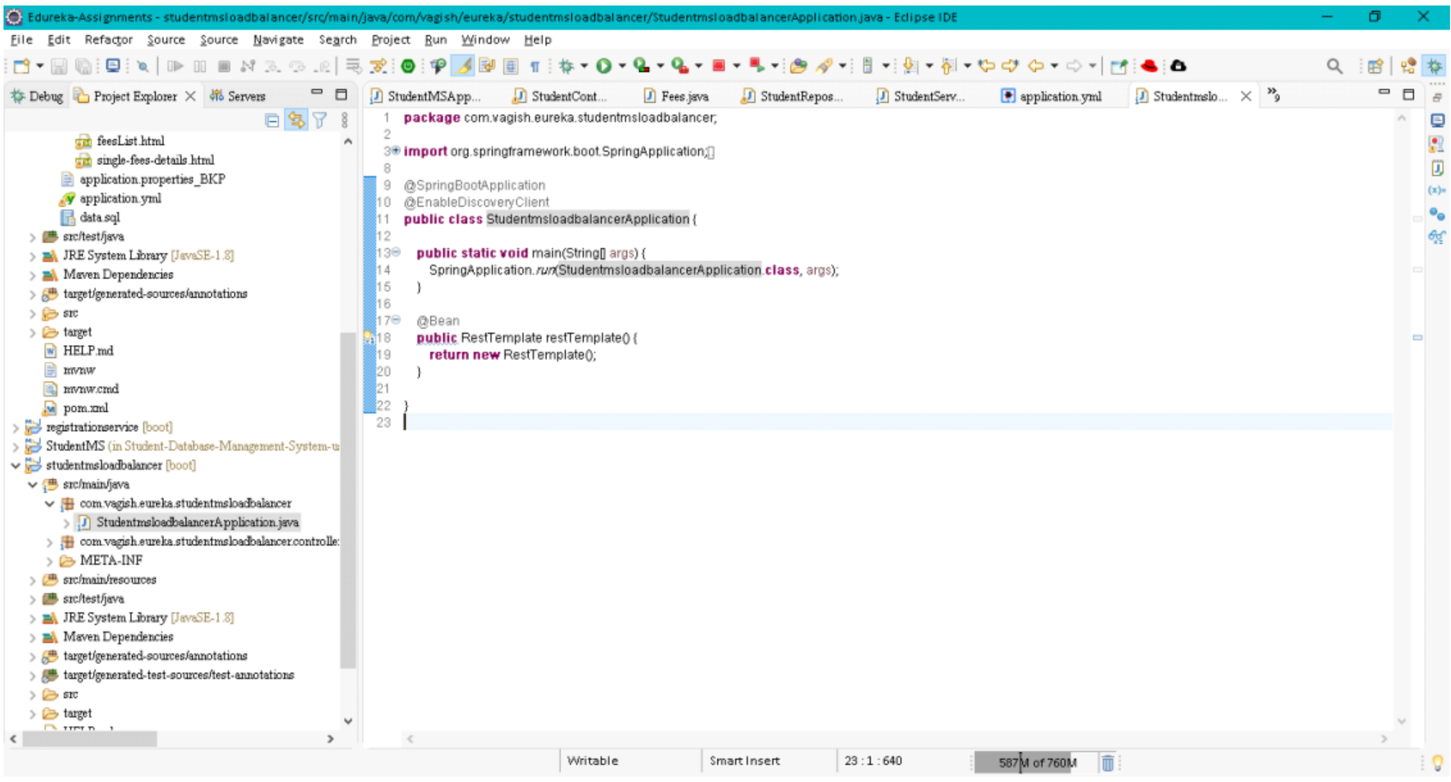

Edureka-Assignments - registrationservice/src/main/java/com/vagish/edureka/registrationservice/RegistrationserviceApplication.java - Eclipse IDE
File Edit Refactor Source Source Navigate Search Project Run Window Help
Debug Project Explorer Servers
feesms [boot] [devtools]
registrationservice [boot]
src/main/java
com.vagish.edureka.registrationservice
RegistrationserviceApplication.java
META-INF
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
StudentMS (in Student-Database-Management-System-using
studentmsloadbalancer [boot]
RegistrationserviceApplication.java
Writable
Smart Insert
17 : 1 : 517
520M of 760M

```java
package com.vagish.edureka.registrationservice;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableEurekaServer
//this annotation will this spring boot project as Eureka Server
public class RegistrationserviceApplication {

    public static void main(String[] args) {
        SpringApplication.run(RegistrationserviceApplication.class, args);
    }

}
```

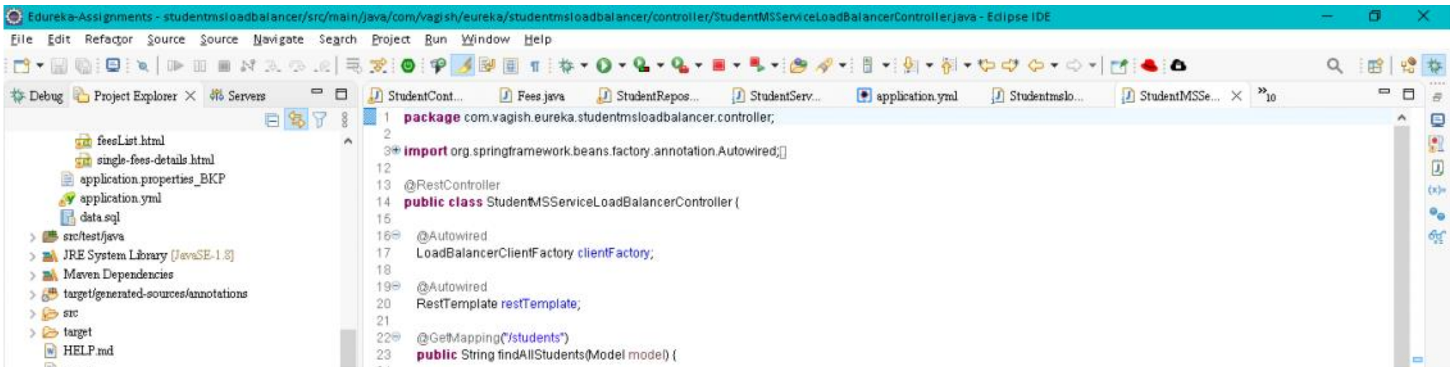


```yaml
server:
  port: 1111

eureka:
  instance:
    hostname: localhost
  client:
    fetchRegistry: false
    registerWithEureka: false
```

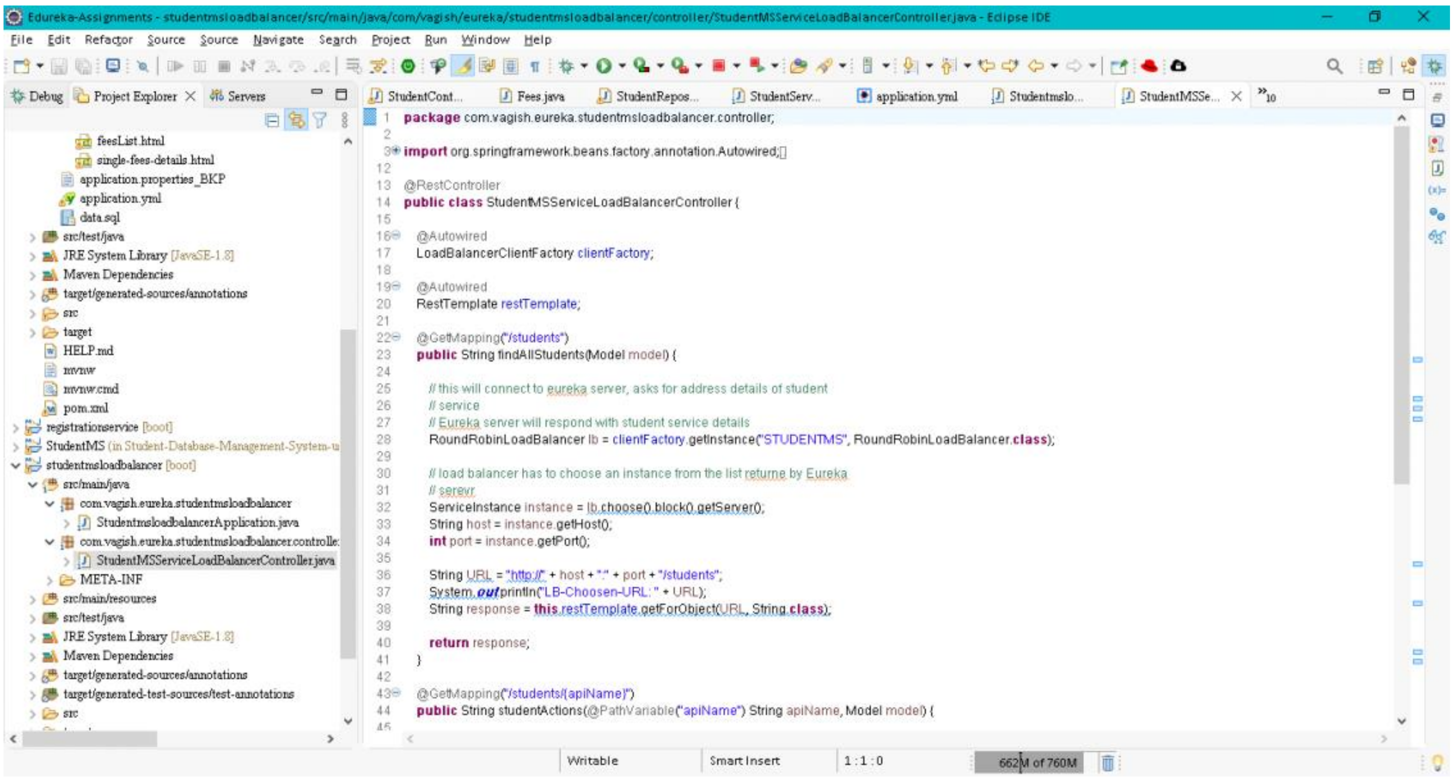
Student MS Client Side Load Balancer



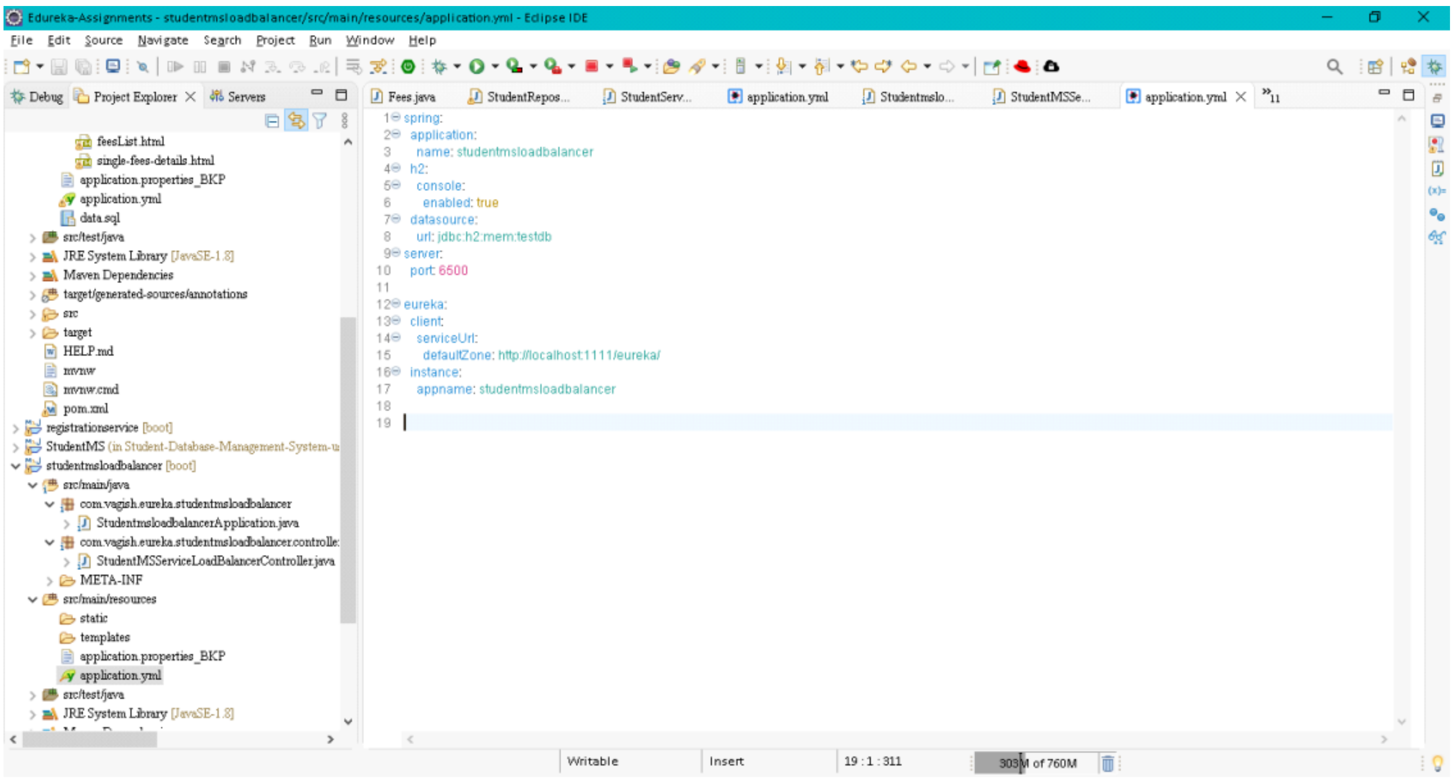
```java
package com.vagish.eureka.studentmsloadbalancer;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableDiscoveryClient
public class StudentmsloadbalancerApplication {

	public static void main(String[] args) {
		SpringApplication.run(StudentmsloadbalancerApplication.class, args);
	}

	@Bean
	public RestTemplate restTemplate() {
		return new RestTemplate();
	}

}
```



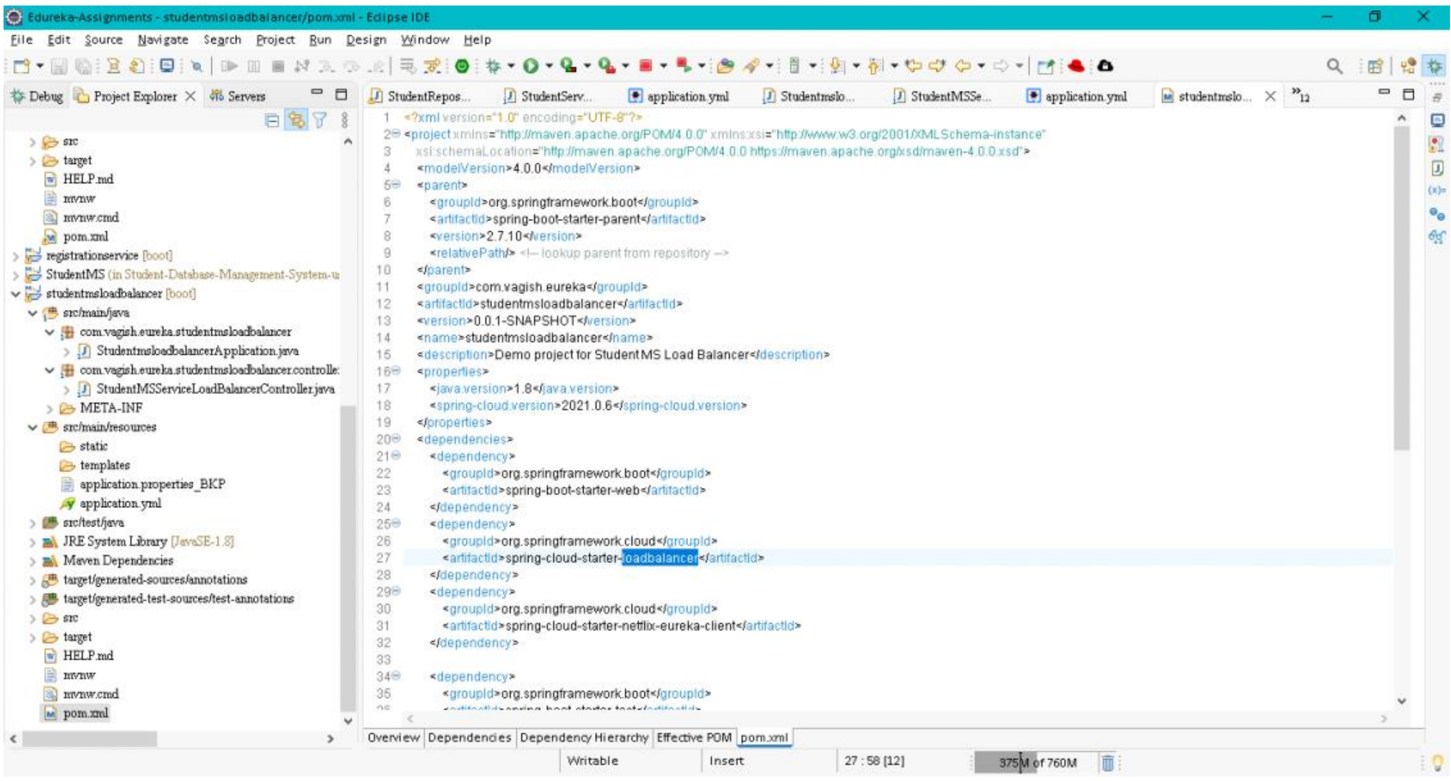
```java
package com.vagish.eureka.studentmsloadbalancer.controller;

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

@RestController
public class StudentMSServiceLoadBalancerController {

	@Autowired
	LoadBalancerClientFactory clientFactory;

	@Autowired
	RestTemplate restTemplate;

	@GetMapping("/students")
	public String findAllStudents(Model model) {
```


Edureka-Assignments - studentmsloadbalancer/src/main/java/com/vagish/eureka/studentmsloadbalancer/controller/StudentMSServiceLoadBalancerController.java - Eclipse IDE

```java
package com.vagish.eureka.studentmsloadbalancer.controller;

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

@RestController
public class StudentMSServiceLoadBalancerController {

  @Autowired
  LoadBalancerClientFactory clientFactory;

  @Autowired
  RestTemplate restTemplate;

  @GetMapping("/students")
  public String findAllStudents(Model model) {

    // this will connect to eureka server, asks for address details of student
    // service
    // Eureka server will respond with student service details
    RoundRobinLoadBalancer lb = clientFactory.getInstance("STUDENTMS", RoundRobinLoadBalancer.class);

    // load balancer has to choose an instance from the list returne by Eureka
    // serevr
    ServiceInstance instance = lb.choose().block().getServer();
    String host = instance.getHost();
    int port = instance.getPort();

    String URL = "http://" + host + ":" + port + "/students";
    System.out.println("LB-Choosen-URL: " + URL);
    String response = this.restTemplate.getForObject(URL, String.class);

    return response;
  }

  @GetMapping("/students/{apiName}")
  public String studentActions(@PathVariable("apiName") String apiName, Model model) {
```

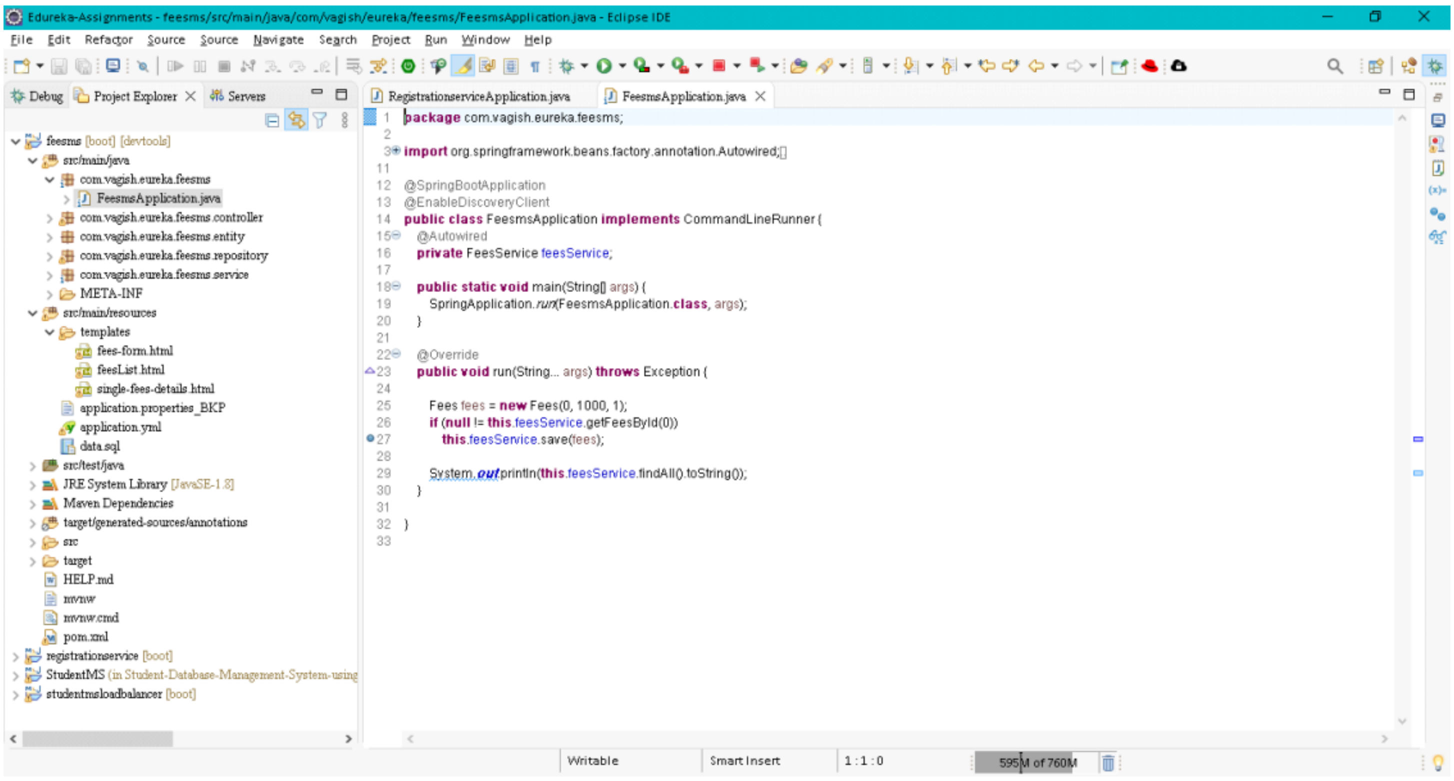

Edureka-Assignments - studentmsloadbalancer/src/main/resources/application.yml - Eclipse IDE

```yaml
spring:
  application:
    name: studentmsloadbalancer
  h2:
    console:
      enabled: true
  datasource:
    url: jdbc:h2:mem:testdb
server:
  port: 6500

eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:1111/eureka/
  instance:
    appname: studentmsloadbalancer

```

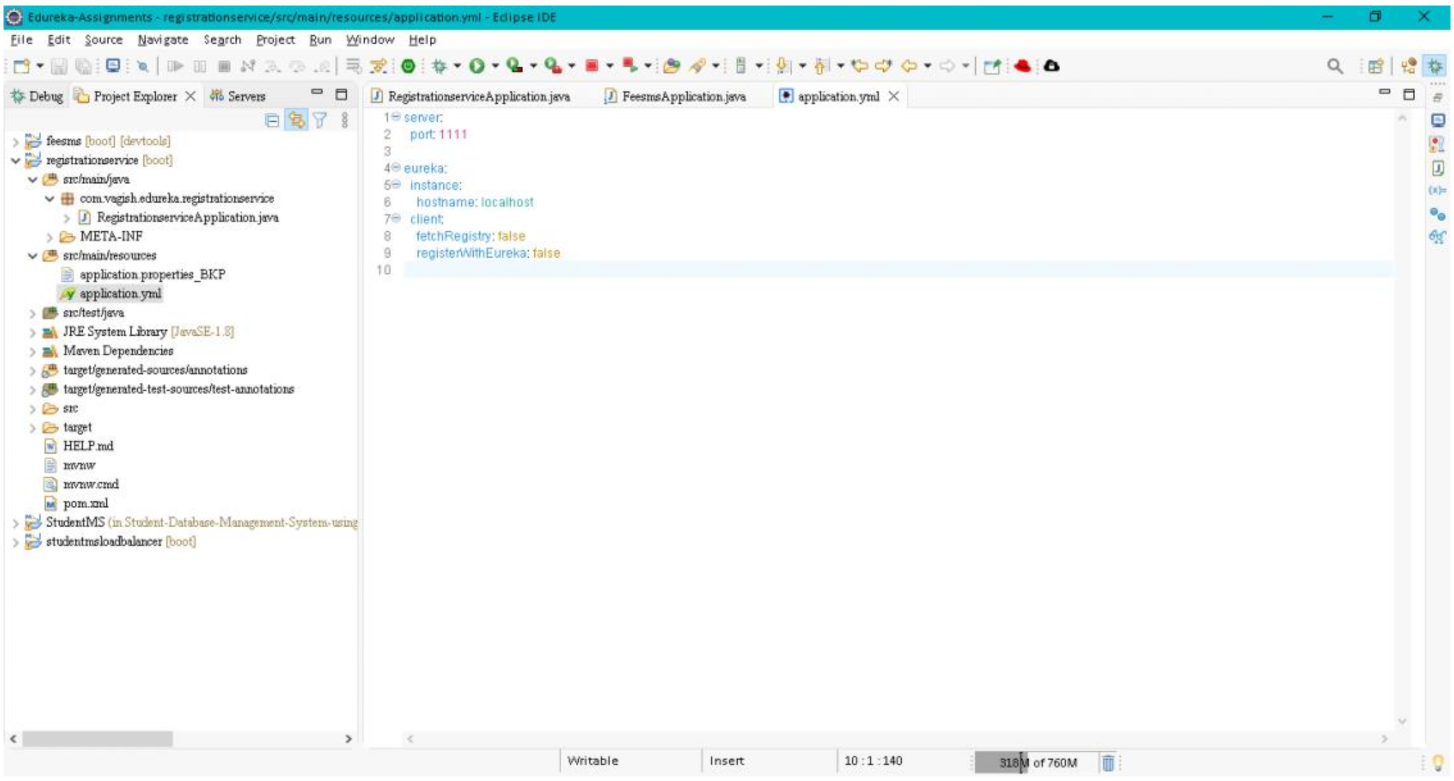

Edureka-Assignments - studentmsloadbalancer/pom.xml - Eclipse IDE

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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 https://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>2.7.10</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.vagish.eureka</groupId>
  <artifactId>studentmsloadbalancer</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>studentmsloadbalancer</name>
  <description>Demo project for Student MS Load Balancer</description>
  <properties>
    <java.version>1.8</java.version>
    <spring-cloud.version>2021.0.6</spring-cloud.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-loadbalancer</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
```

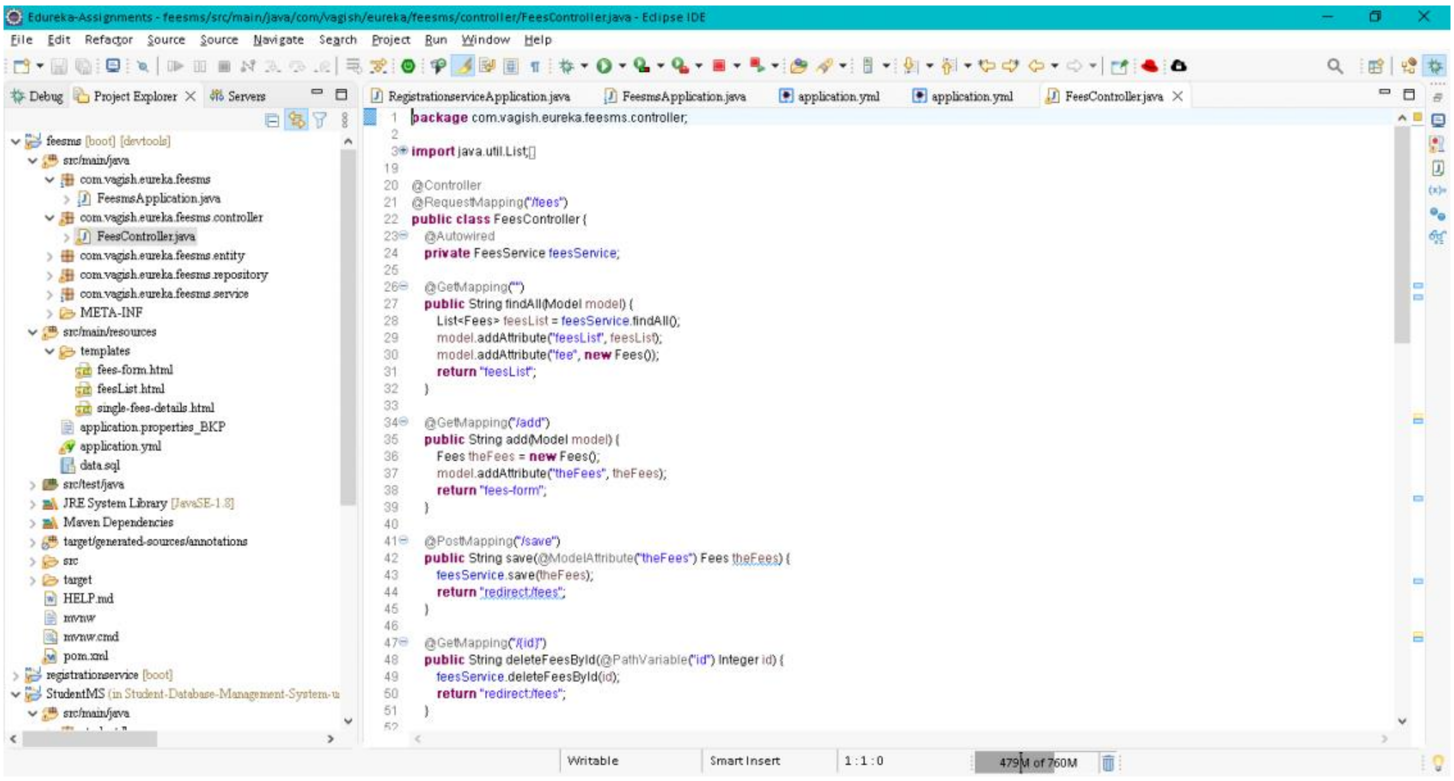
Fees MS


Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/FeesmsApplication.java - Eclipse IDE

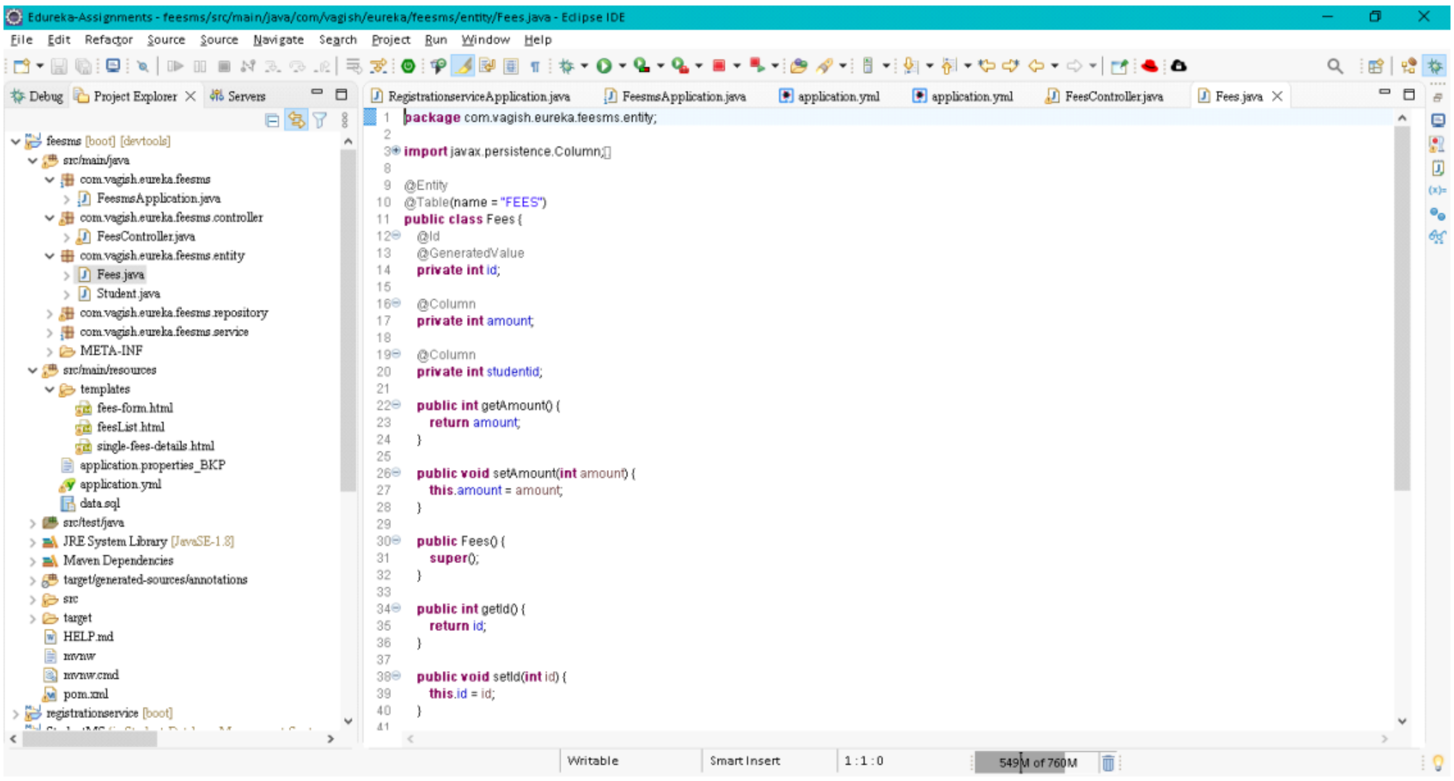
```java
package com.vagish.eureka.feesms;

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

@SpringBootApplication
@EnableDiscoveryClient
public class FeesmsApplication implements CommandLineRunner {
    @Autowired
    private FeesService feesService;

    public static void main(String[] args) {
        SpringApplication.run(FeesmsApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {

        Fees fees = new Fees(0, 1000, 1);
        if (null != this.feesService.getFeesById(0))
            this.feesService.save(fees);

        System.out.println(this.feesService.findAll().toString());
    }

}
```


Edureka-Assignments - registrationservice/src/main/resources/application.yml - Eclipse IDE

```yaml
server:
  port: 1111

eureka:
  instance:
    hostname: localhost
  client:
    fetchRegistry: false
    registerWithEureka: false
```


Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/controller/FeesController.java - Eclipse IDE

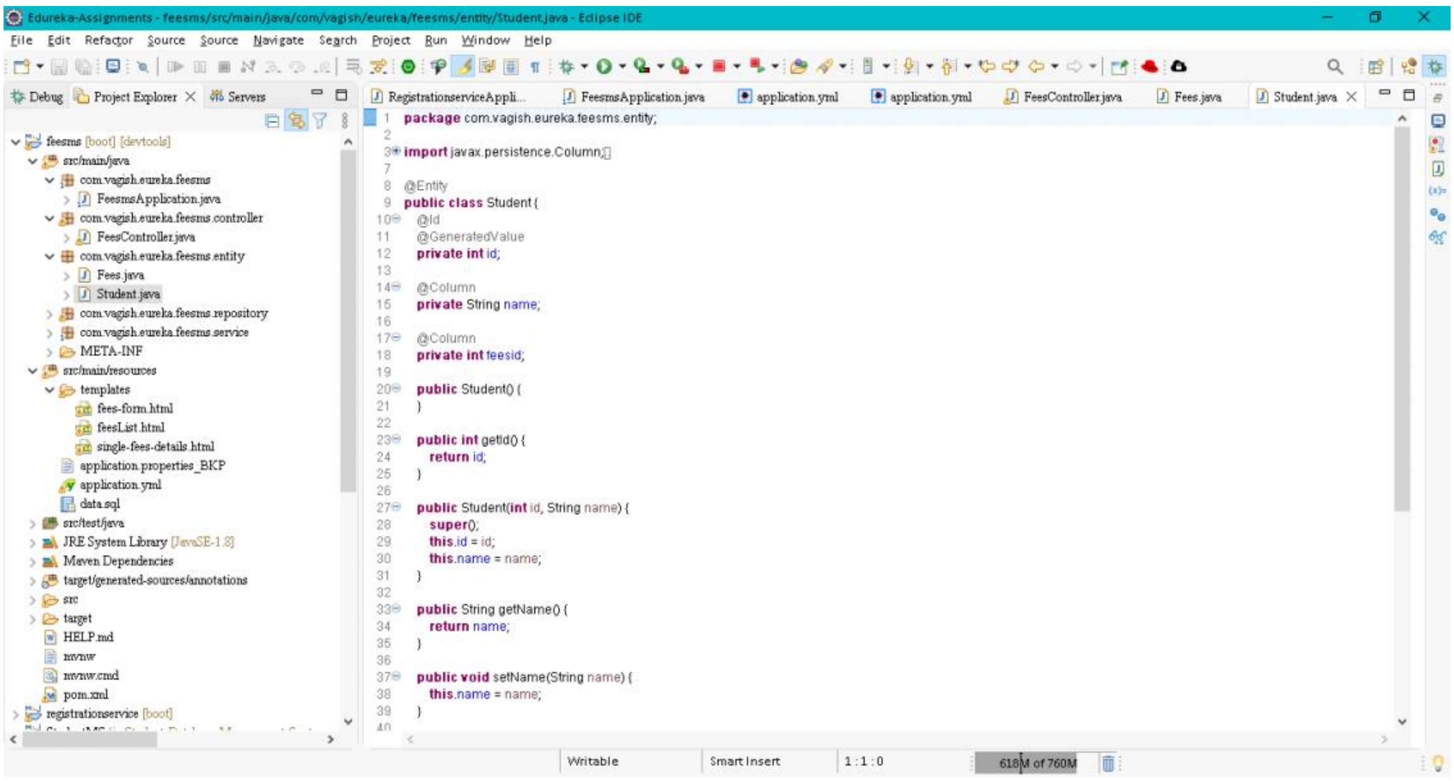
```java
package com.vagish.eureka.feesms.controller;

import java.util.List;

@Controller
@RequestMapping("/fees")
public class FeesController {
  @Autowired
  private FeesService feesService;

  @GetMapping("")
  public String findAll(Model model) {
    List<Fees> feesList = feesService.findAll();
    model.addAttribute("feesList", feesList);
    model.addAttribute("fee", new Fees());
    return "feesList";
  }

  @GetMapping("/add")
  public String add(Model model) {
    Fees theFees = new Fees();
    model.addAttribute("theFees", theFees);
    return "fees-form";
  }

  @PostMapping("/save")
  public String save(@ModelAttribute("theFees") Fees theFees) {
    feesService.save(theFees);
    return "redirect:/fees";
  }

  @GetMapping("/{id}")
  public String deleteFeesById(@PathVariable("id") Integer id) {
    feesService.deleteFeesById(id);
    return "redirect:/fees";
  }
```


Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/entity/Fees.java - Eclipse IDE
File Edit Refactor Source Source Navigate Search Project Run Window Help
Debug Project Explorer Servers
RegistrationserviceApplication.java FeesmsApplication.java application.yml application.yml FeesController.java Fees.java

```
feesms [boot] [devtools]
  src/main/java
    com.vagish.eureka.feesms
      FeesmsApplication.java
    com.vagish.eureka.feesms.controller
      FeesController.java
    com.vagish.eureka.feesms.entity
      Fees.java
      Student.java
    com.vagish.eureka.feesms.repository
    com.vagish.eureka.feesms.service
    META-INF
  src/main/resources
    templates
      fees-form.html
      feesList.html
      single-fees-details.html
    application.properties_BKP
    application.yml
    data.sql
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
registrationservice [boot]
```

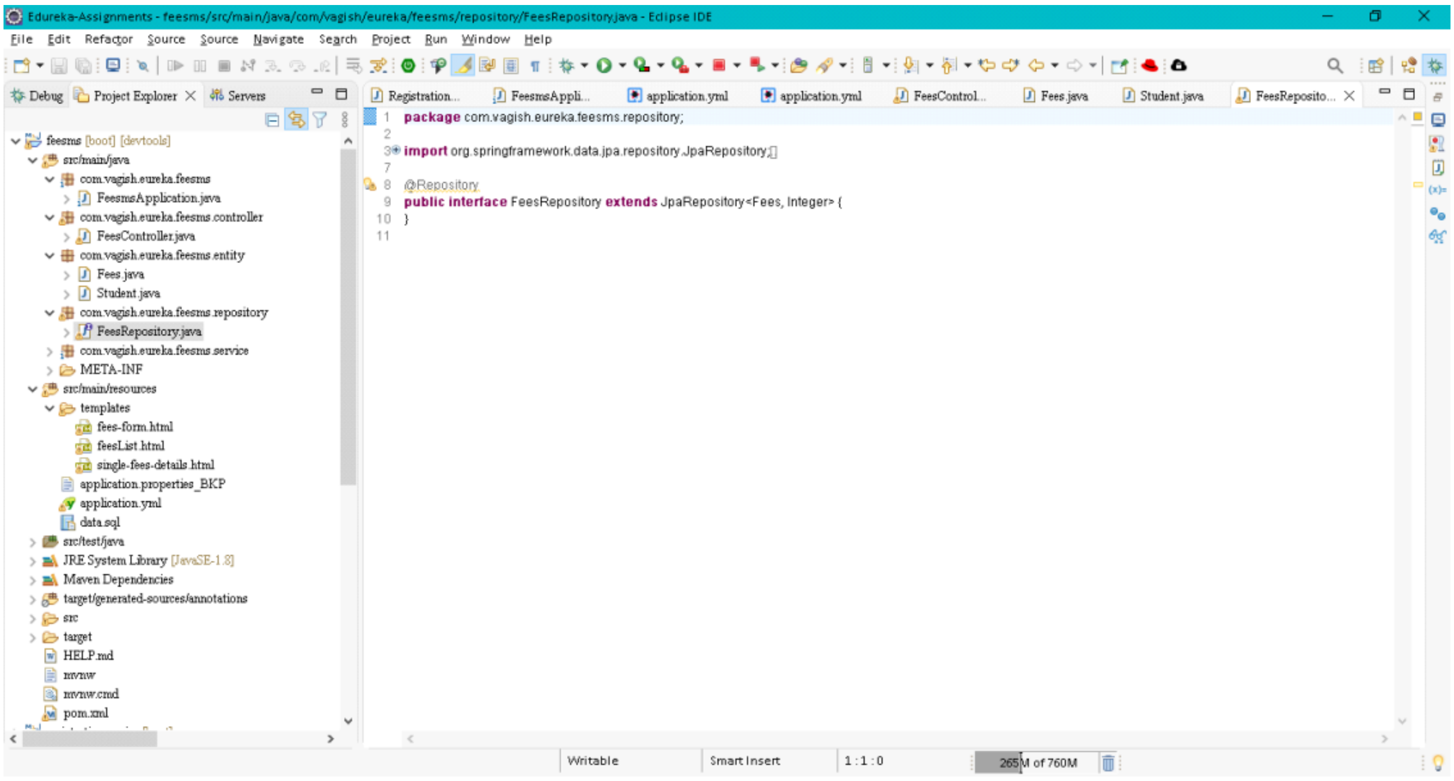
```java
package com.vagish.eureka.feesms.entity;

import javax.persistence.Column;

@Entity
@Table(name = "FEES")
public class Fees {
    @Id
    @GeneratedValue
    private int id;

    @Column
    private int amount;

    @Column
    private int studentid;

    public int getAmount() {
        return amount;
    }

    public void setAmount(int amount) {
        this.amount = amount;
    }

    public Fees() {
        super();
    }

    public int getId() {
        return id;
    }

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

Writable
Smart Insert
1 : 1 : 0
549M of 760M


Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/entity/Student.java - Eclipse IDE
File Edit Refactor Source Source Navigate Search Project Run Window Help
Debug Project Explorer Servers
RegistrationserviceAppli... FeesmsApplication.java application.yml application.yml FeesController.java Fees.java Student.java

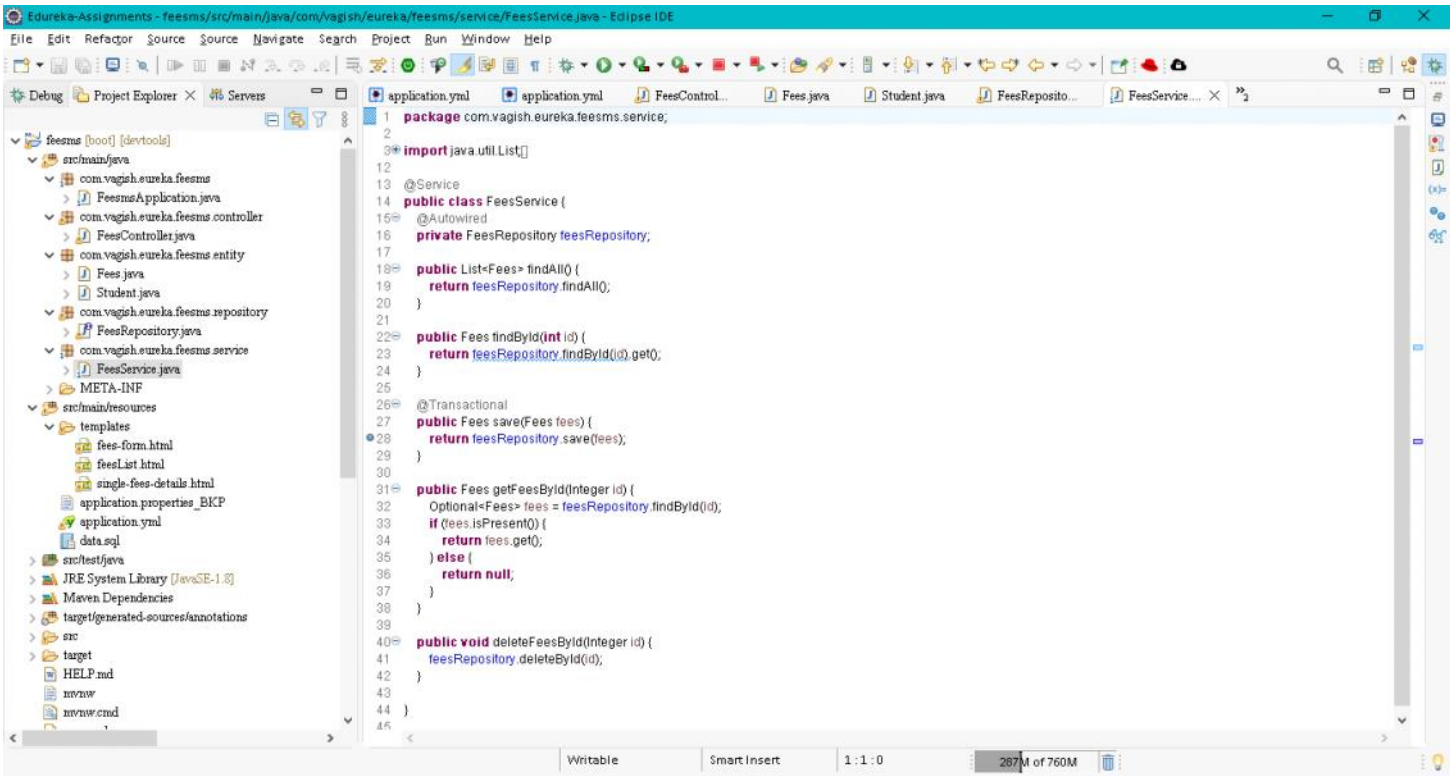
```
feesms [boot] [devtools]
  src/main/java
    com.vagish.eureka.feesms
      FeesmsApplication.java
    com.vagish.eureka.feesms.controller
      FeesController.java
    com.vagish.eureka.feesms.entity
      Fees.java
      Student.java
    com.vagish.eureka.feesms.repository
    com.vagish.eureka.feesms.service
    META-INF
  src/main/resources
    templates
      fees-form.html
      feesList.html
      single-fees-details.html
    application.properties_BKP
    application.yml
    data.sql
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
registrationservice [boot]
```

```java
package com.vagish.eureka.feesms.entity;

import javax.persistence.Column;

@Entity
public class Student {
    @Id
    @GeneratedValue
    private int id;

    @Column
    private String name;

    @Column
    private int feesid;

    public Student() {
    }

    public int getId() {
        return id;
    }

    public Student(int id, String name) {
        super();
        this.id = id;
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
```



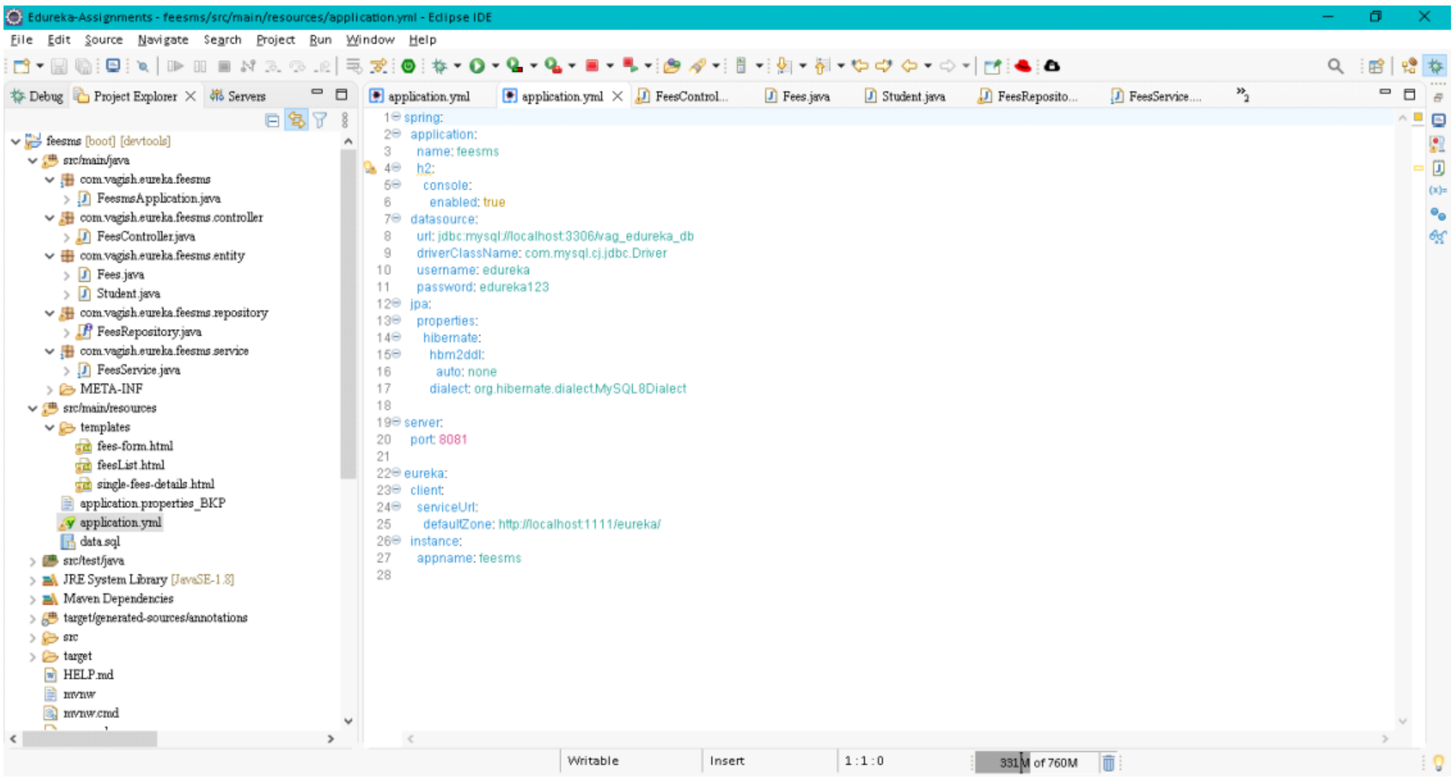
Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/repository/FeesRepository.java - Eclipse IDE

```java
package com.vagish.eureka.feesms.repository;

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

@Repository
public interface FeesRepository extends JpaRepository<Fees, Integer> {
}
```



Edureka-Assignments - feesms/src/main/java/com/vagish/eureka/feesms/service/FeesService.java - Eclipse IDE

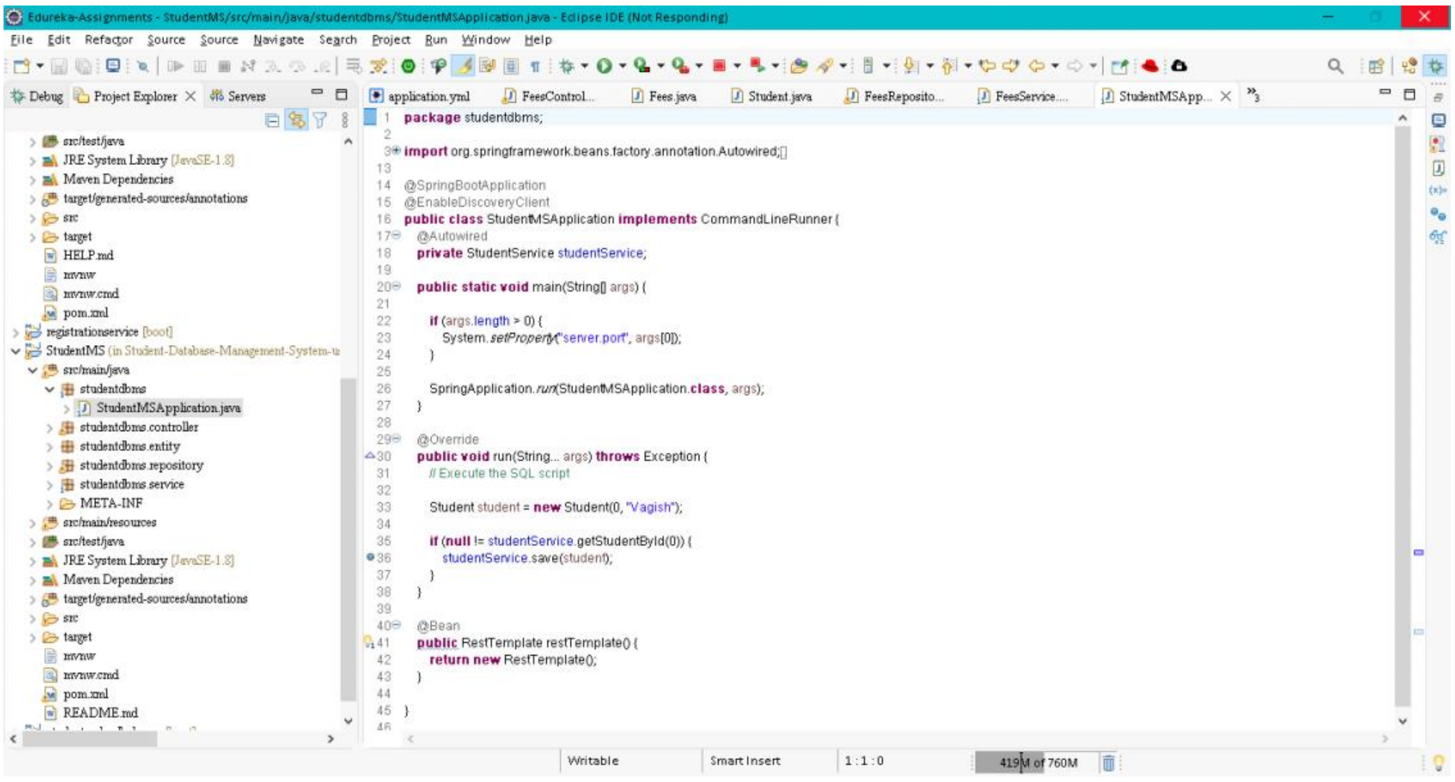
```java
package com.vagish.eureka.feesms.service;

import java.util.List;

@Service
public class FeesService {
    @Autowired
    private FeesRepository feesRepository;

    public List<Fees> findAll() {
        return feesRepository.findAll();
    }

    public Fees findById(int id) {
        return feesRepository.findById(id).get();
    }

    @Transactional
    public Fees save(Fees fees) {
        return feesRepository.save(fees);
    }

    public Fees getFeesById(Integer id) {
        Optional<Fees> fees = feesRepository.findById(id);
        if (fees.isPresent()) {
            return fees.get();
        } else {
            return null;
        }
    }

    public void deleteFeesById(Integer id) {
        feesRepository.deleteById(id);
    }

}
```


Edureka-Assignments - feesms/src/main/resources/application.yml - Eclipse IDE

```yaml
spring:
  application:
    name: feesms
  h2:
    console:
      enabled: true
  datasource:
   url: jdbc:mysql://localhost:3306/vag_edureka_db
   driverClassName: com.mysql.cj.jdbc.Driver
   username: edureka
   password: edureka123
  jpa:
   properties:
     hibernate:
      hbm2ddl:
        auto: none
      dialect: org.hibernate.dialect.MySQL8Dialect

server:
  port: 8081

eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:1111/eureka/
  instance:
    appname: feesms
```

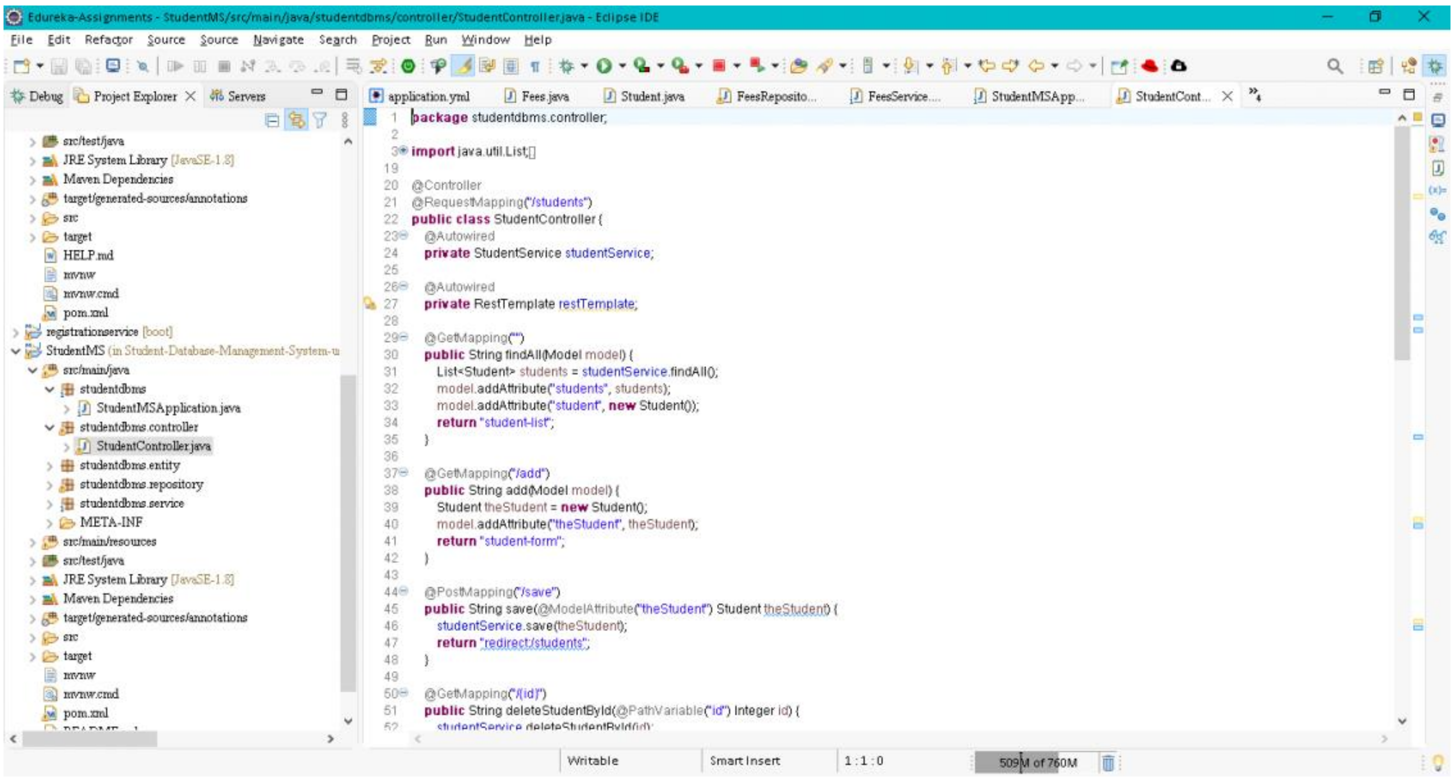
```
feesms [boot] [devtools]
  src/main/java
    com.vagish.eureka.feesms
      FeesmsApplication.java
    com.vagish.eureka.feesms.controller
      FeesController.java
    com.vagish.eureka.feesms.entity
      Fees.java
      Student.java
    com.vagish.eureka.feesms.repository
      FeesRepository.java
    com.vagish.eureka.feesms.service
      FeesService.java
    META-INF
  src/main/resources
    templates
      fees-form.html
      feesList.html
      single-fees-details.html
    application.properties_BKP
    application.yml
    data.sql
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
```

Student MS


Edureka-Assignments - StudentMS/src/main/java/studentdbms/StudentMSApplication.java - Eclipse IDE (Not Responding)

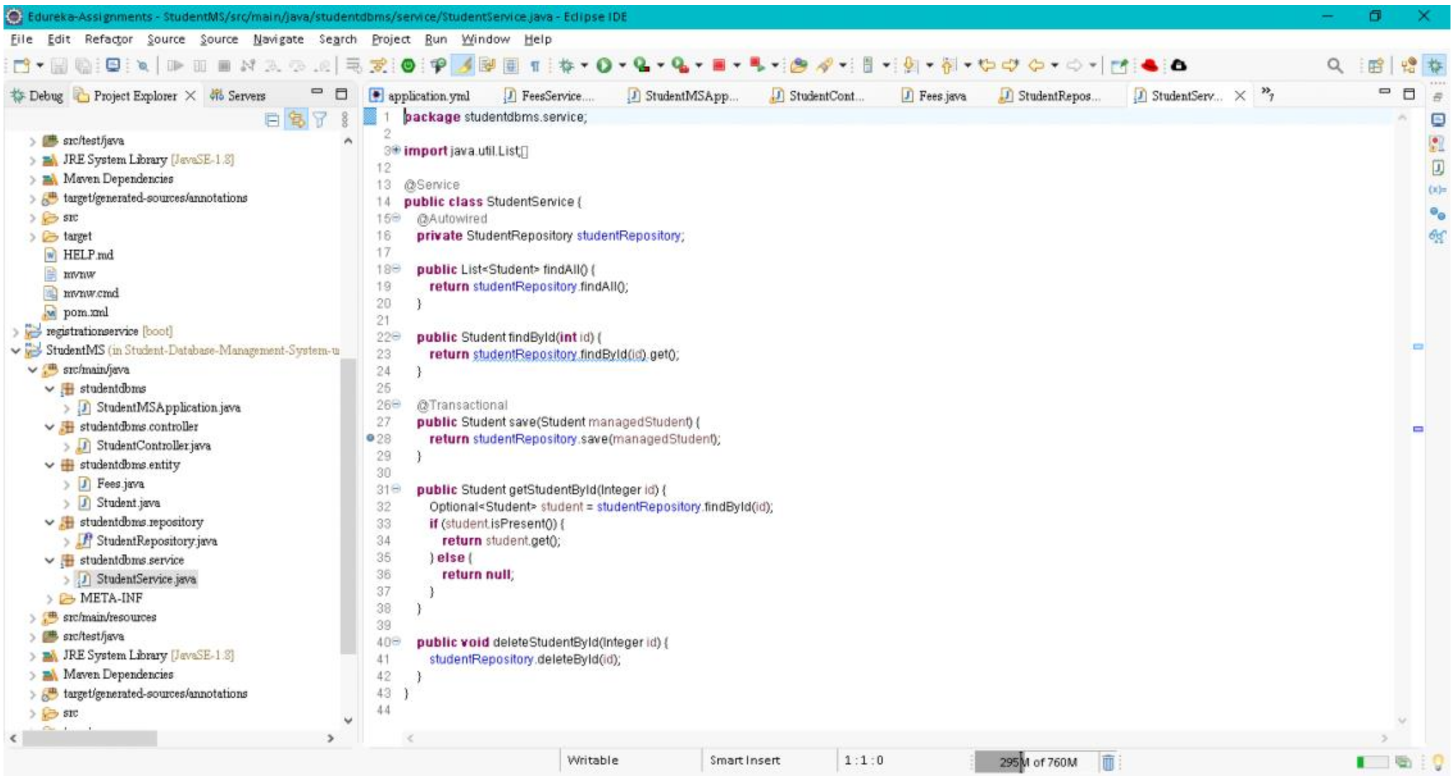
```java
package studentdbms;

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

@SpringBootApplication
@EnableDiscoveryClient
public class StudentMSApplication implements CommandLineRunner {
  @Autowired
  private StudentService studentService;

  public static void main(String[] args) {

    if (args.length > 0) {
      System.setProperty("server.port", args[0]);
    }

    SpringApplication.run(StudentMSApplication.class, args);
  }

  @Override
  public void run(String... args) throws Exception {
    // Execute the SQL script

    Student student = new Student(0, "Vagish");

    if (null != studentService.getStudentById(0)) {
      studentService.save(student);
    }
  }

  @Bean
  public RestTemplate restTemplate() {
    return new RestTemplate();
  }

}
```

```
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
registrationservice [boot]
StudentMS (in Student-Database-Management-System-u
  src/main/java
    studentdbms
      StudentMSApplication.java
    studentdbms.controller
    studentdbms.entity
    studentdbms.repository
    studentdbms.service
    META-INF
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  target/generated-sources/annotations
  src
  target
  mvnw
  mvnw.cmd
  pom.xml
  README.md
```



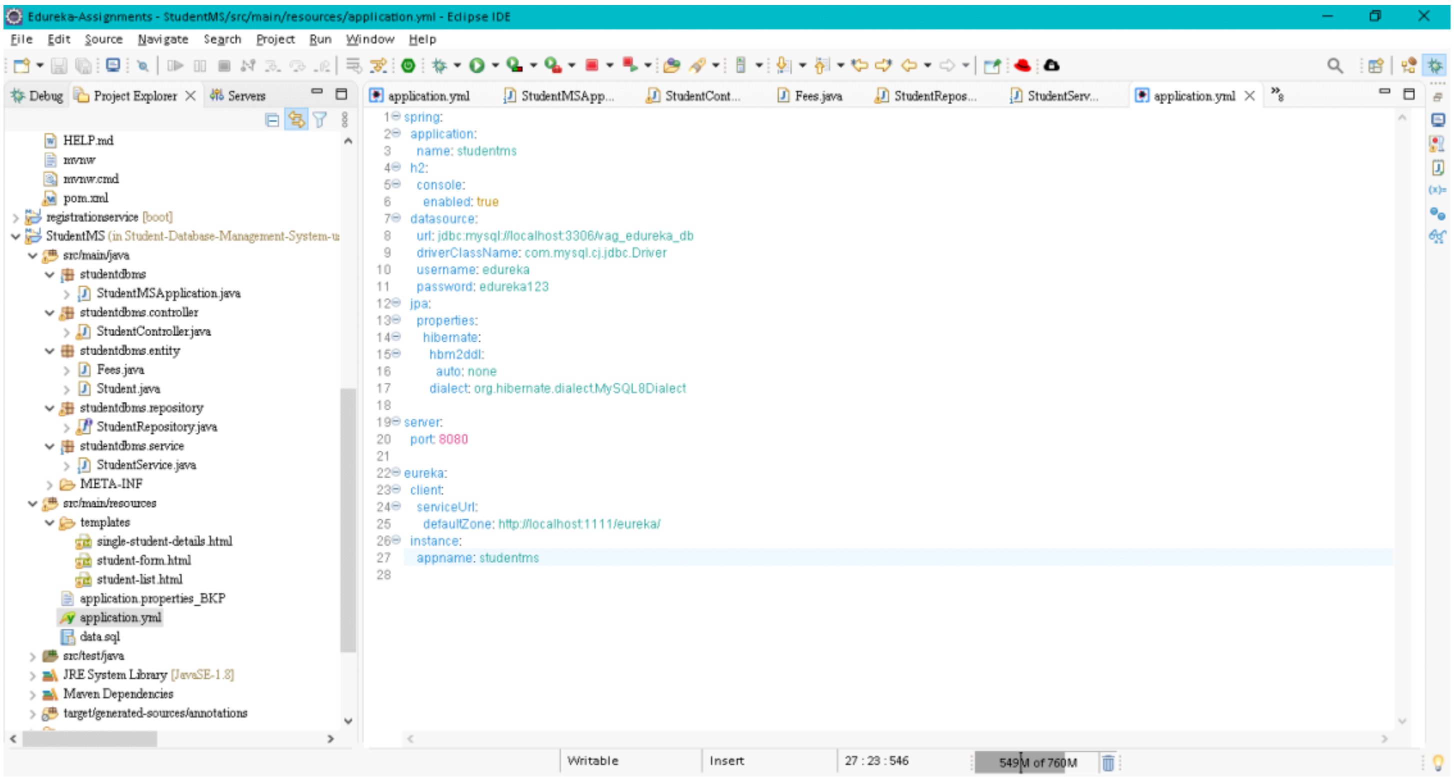
Edureka-Assignments - StudentMS/src/main/java/studentdbms/controller/StudentController.java - Eclipse IDE

```java
package studentdbms.controller;

import java.util.List;

@Controller
@RequestMapping("/students")
public class StudentController {
    @Autowired
    private StudentService studentService;

    @Autowired
    private RestTemplate restTemplate;

    @GetMapping("")
    public String findAll(Model model) {
        List<Student> students = studentService.findAll();
        model.addAttribute("students", students);
        model.addAttribute("student", new Student());
        return "student-list";
    }

    @GetMapping("/add")
    public String add(Model model) {
        Student theStudent = new Student();
        model.addAttribute("theStudent", theStudent);
        return "student-form";
    }

    @PostMapping("/save")
    public String save(@ModelAttribute("theStudent") Student theStudent) {
        studentService.save(theStudent);
        return "redirect:/students";
    }

    @GetMapping("/{id}")
    public String deleteStudentById(@PathVariable("id") Integer id) {
        studentService.deleteStudentById(id);
```



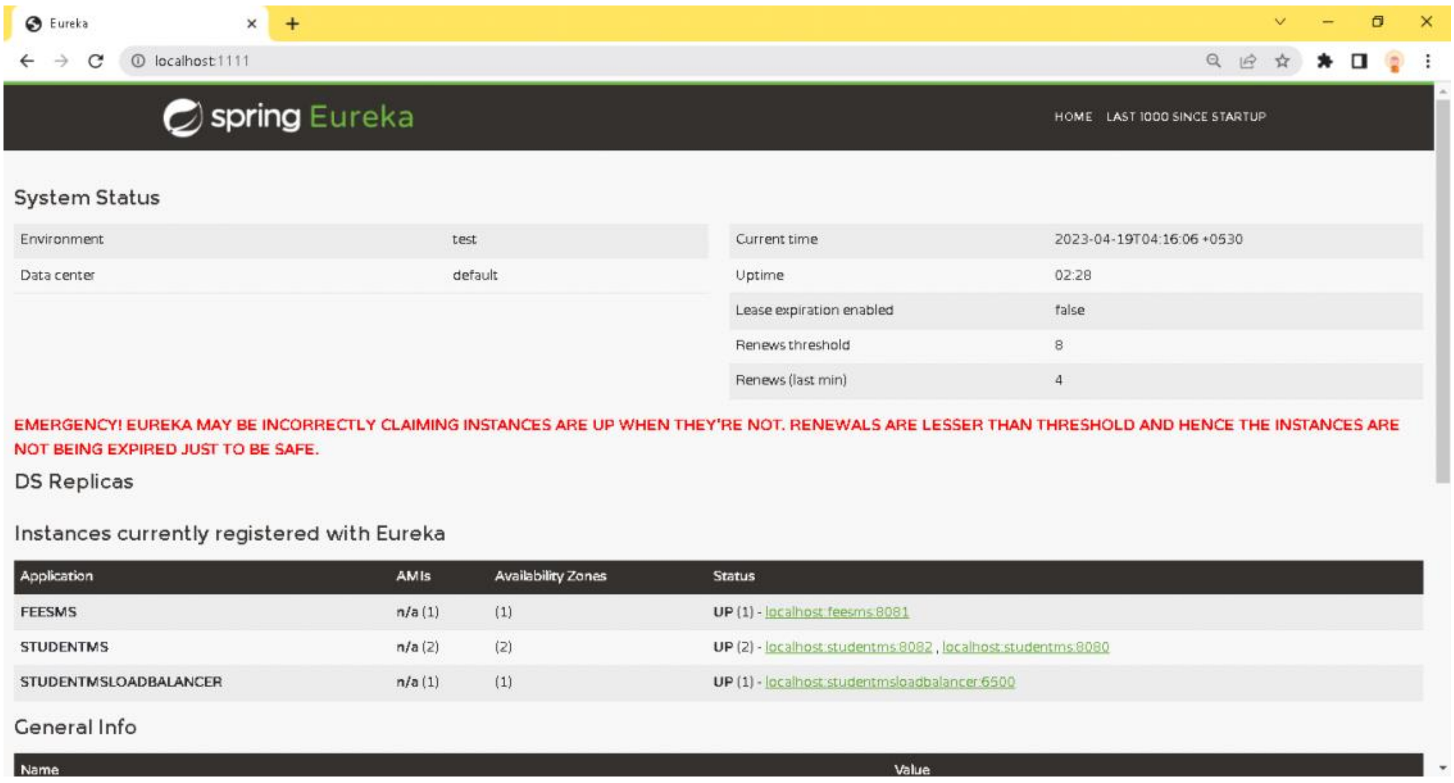
Edureka-Assignments - StudentMS/src/main/java/studentdbms/service/StudentService.java - Eclipse IDE

```java
package studentdbms.service;

import java.util.List;

@Service
public class StudentService {
    @Autowired
    private StudentRepository studentRepository;

    public List<Student> findAll() {
        return studentRepository.findAll();
    }

    public Student findById(int id) {
        return studentRepository.findById(id).get();
    }

    @Transactional
    public Student save(Student managedStudent) {
        return studentRepository.save(managedStudent);
    }

    public Student getStudentById(Integer id) {
        Optional<Student> student = studentRepository.findById(id);
        if (student.isPresent()) {
            return student.get();
        } else {
            return null;
        }
    }

    public void deleteStudentById(Integer id) {
        studentRepository.deleteById(id);
    }
}
```



```yaml
spring:
  application:
    name: studentms
  h2:
    console:
      enabled: true
  datasource:
    url: jdbc:mysql://localhost:3306/vag_edureka_db
    driverClassName: com.mysql.cj.jdbc.Driver
    username: edureka
    password: edureka123
  jpa:
    properties:
      hibernate:
        hbm2ddl:
          auto: none
        dialect: org.hibernate.dialect.MySQL8Dialect

server:
  port: 8080

eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:1111/eureka/
  instance:
    appname: studentms

```

<u>Eureka Server Output:</u>



System Status

Environment	test	Current time	2023-04-19T04:16:06 +0530
Data center	default	Uptime	02:28
		Lease expiration enabled	false
		Renews threshold	8
		Renews (last min)	4

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

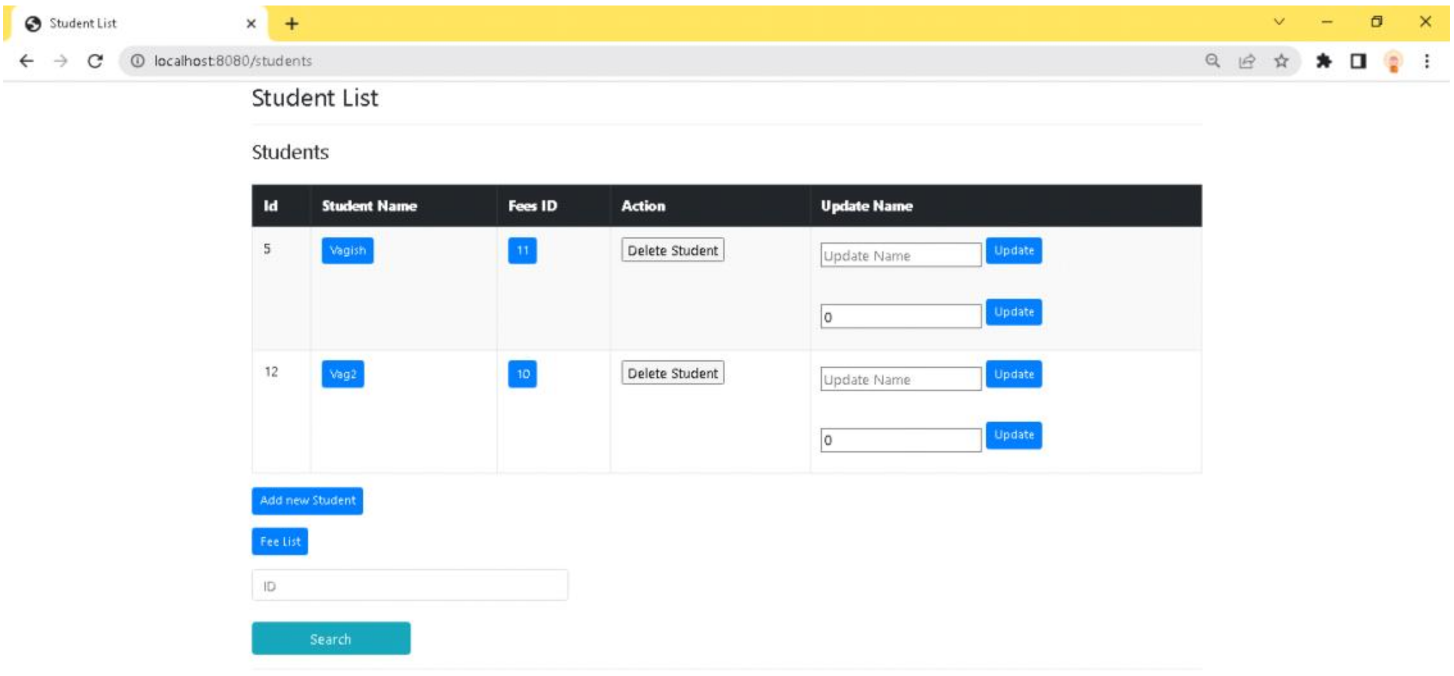
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
FEESMS	n/a (1)	(1)	UP (1) - localhost:feesms:8081
STUDENTMS	n/a (2)	(2)	UP (2) - localhost:studentms:8082 , localhost:studentms:8080
STUDENTMSLOADBALANCER	n/a (1)	(1)	UP (1) - localhost:studentmsloadbalancer:6500

General Info

Name	Value

<u>Student MS Output:</u>

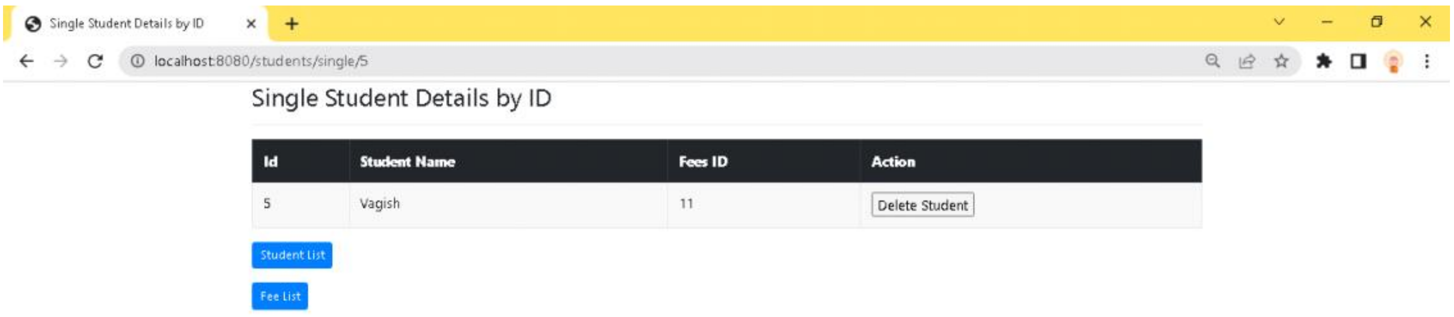

Student List

localhost:8080/students

Student List

Students

Id	Student Name	Fees ID	Action	Update Name
5	Vagish	11	Delete Student	Update Name Update 0 Update
12	Vag2	10	Delete Student	Update Name Update 0 Update

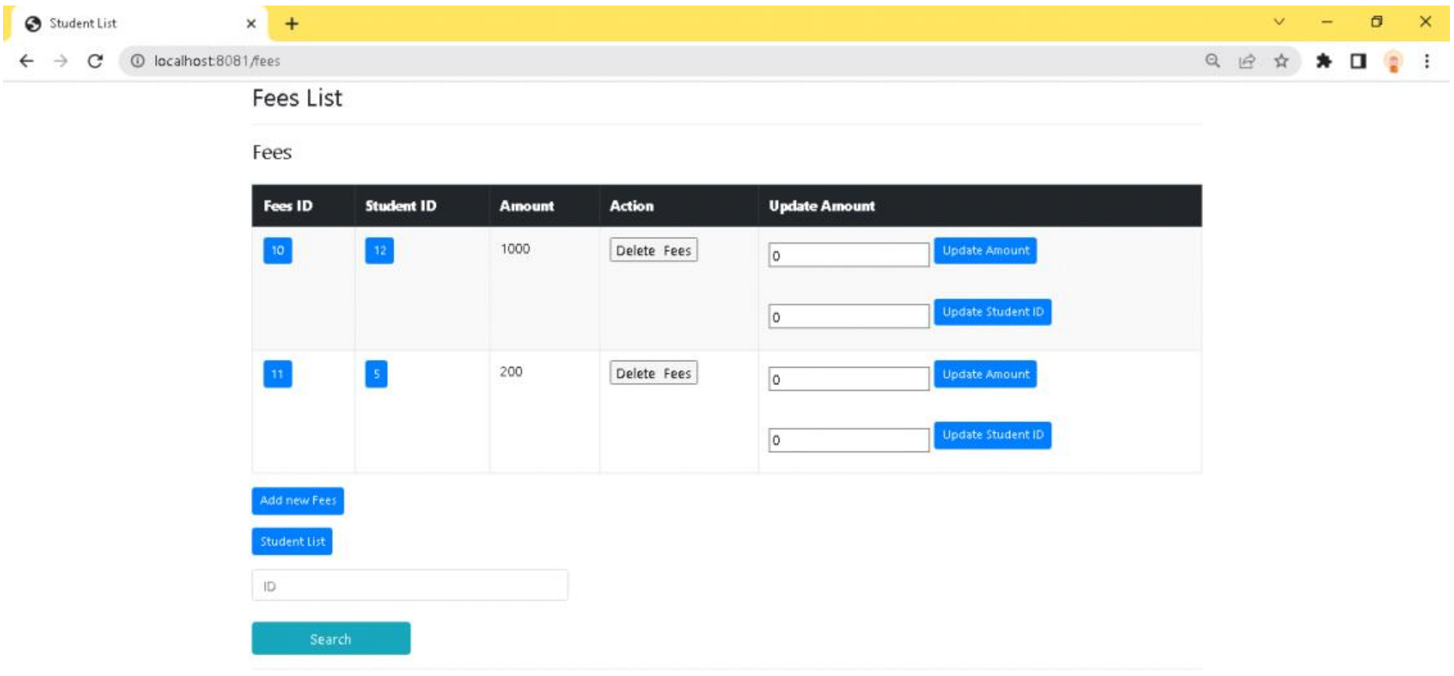

Single Student Details by ID

localhost:8080/students/single/5

Single Student Details by ID

Id	Student Name	Fees ID	Action
5	Vagish	11	Delete Student

Fees MS Output



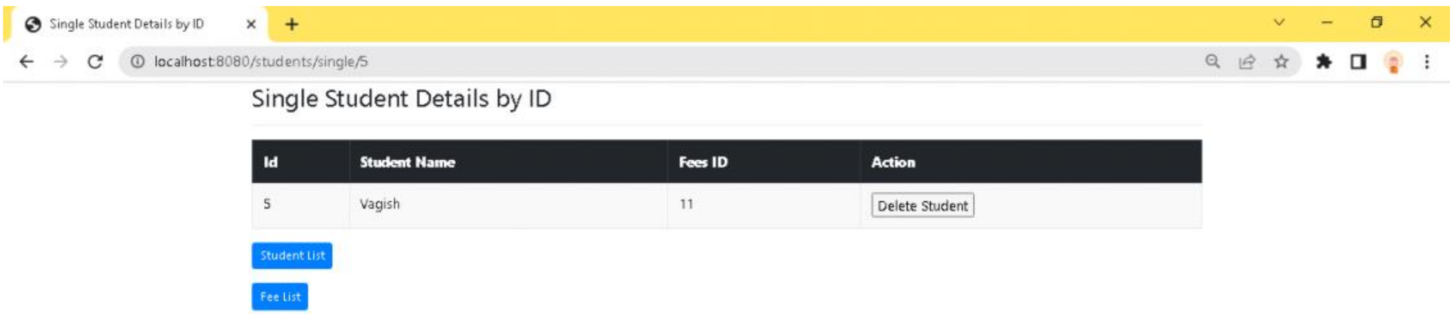
Student List

localhost:8081/fees

Fees List

Fees

Fees ID	Student ID	Amount	Action	Update Amount
10	12	1000	Delete Fees	0 Update Amount 0 Update Student ID
11	5	200	Delete Fees	0 Update Amount 0 Update Student ID



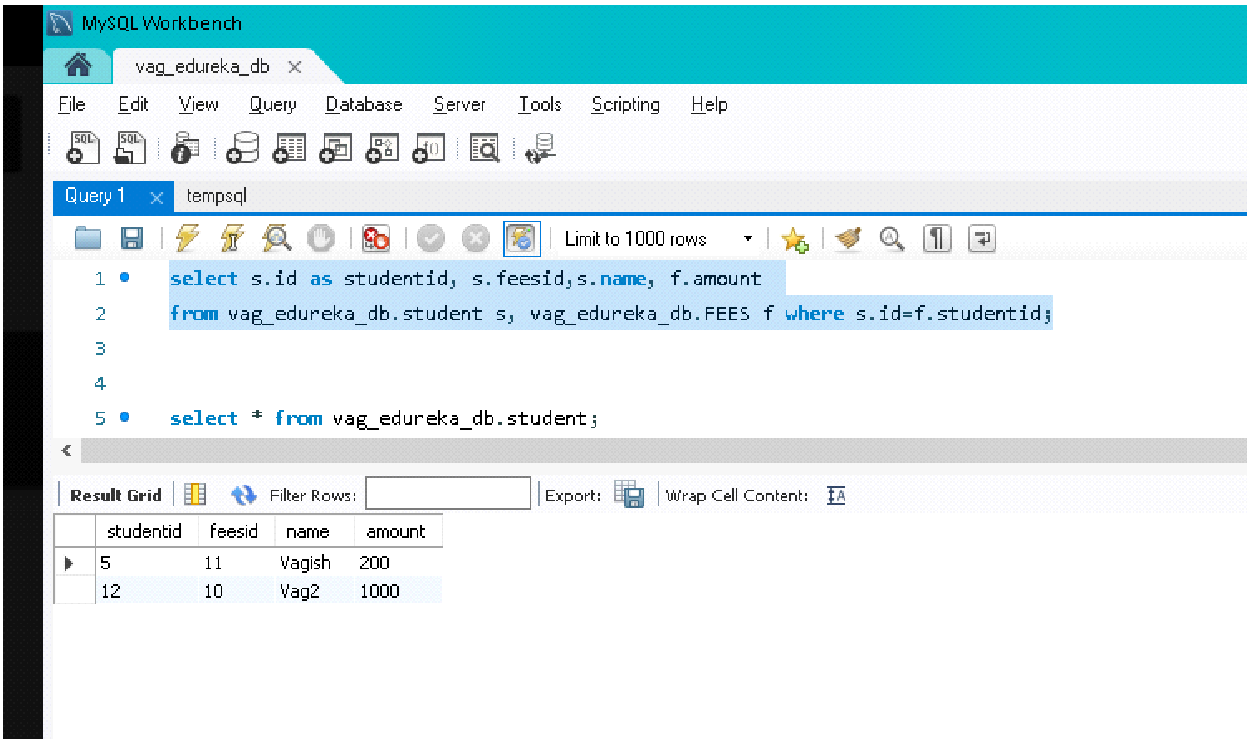
Single Student Details by ID

localhost:8080/students/single/5

Single Student Details by ID

Id	Student Name	Fees ID	Action
5	Vagish	11	Delete Student

Database Output:



```sql
select s.id as studentid, s.feesid,s.name, f.amount
from vag_edureka_db.student s, vag_edureka_db.FEES f where s.id=f.studentid;


select * from vag_edureka_db.student;
```

studentid	feesid	name	amount
5	11	Vagish	200
12	10	Vag2	1000

Uploaded PDF with consolidated Screenshots to create Student MS, Eureka Registry Service and Client Side Load Balancer to perform all the mentioned Tasks.

Also, all the related Maven Codes, result Screenshot PDF is saved on GIT Repo:
https://github.com/Vagish619/Edureka-Microservices.git

Branch Name:
Assignments_And_Projects