

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.

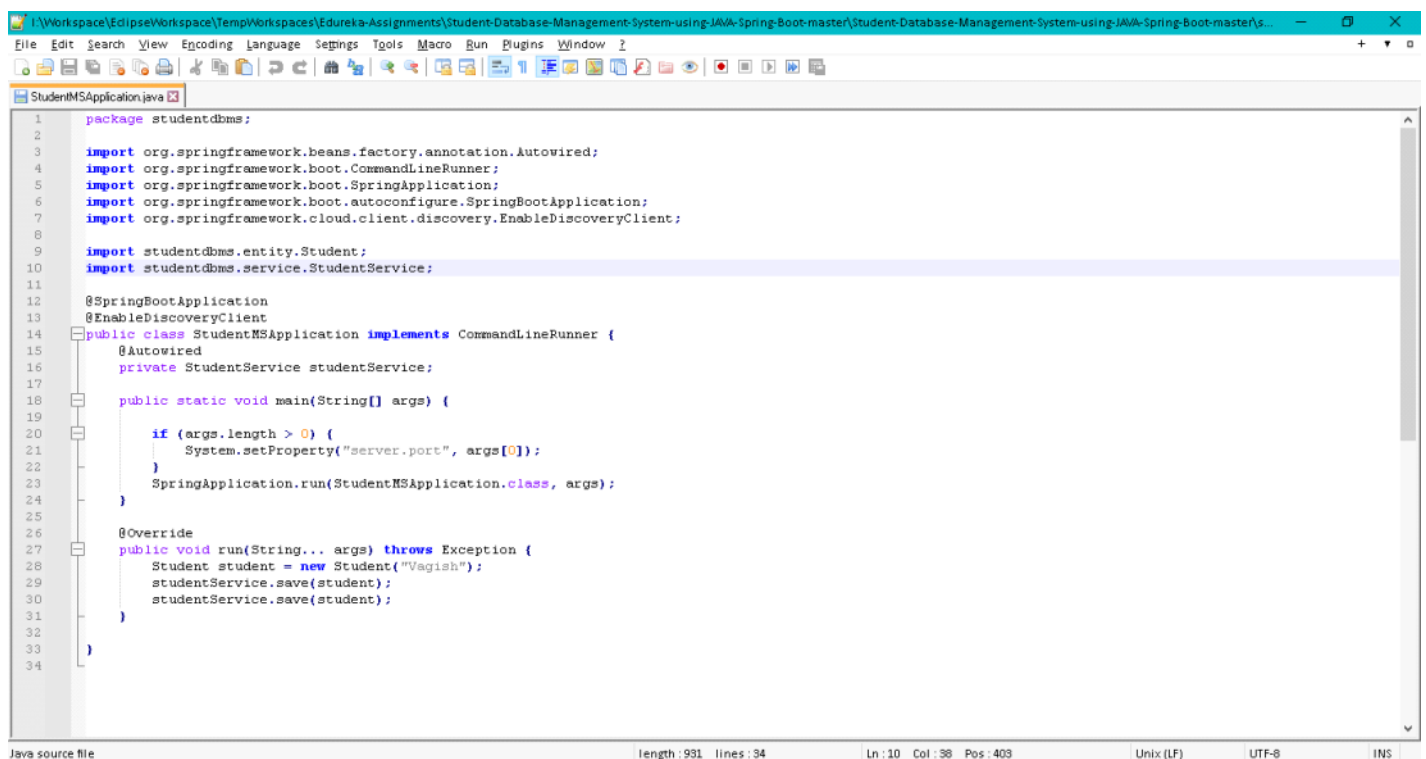
The design is complete, and it's time to implement.

Using Java, Spring Boot, Spring Web, Spring Data JPA, and H2(in-memory database),implement a microservice(students) to manage students with all CRUD operations.

- 1.An API to fetch all students
- 2.An API to fetch a single student
- 3.An API to create a new student
- 4.An API to delete a student
- 5.An API to update a student

Hint: Service registry/discovery, load balancing, circuit breaker.

Solution:



```
1 package studentdms;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.boot.CommandLineRunner;
5 import org.springframework.boot.SpringApplication;
6 import org.springframework.boot.autoconfigure.SpringBootApplication;
7 import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
8
9 import studentdms.entity.Student;
10 import studentdms.service.StudentService;
11
12 @SpringBootApplication
13 @EnableDiscoveryClient
14 public class StudentMSApplication implements CommandLineRunner {
15     @Autowired
16     private StudentService studentService;
17
18     public static void main(String[] args) {
19
20         if (args.length > 0) {
21             System.setProperty("server.port", args[0]);
22         }
23         SpringApplication.run(StudentMSApplication.class, args);
24     }
25
26     @Override
27     public void run(String... args) throws Exception {
28         Student student = new Student("Vagish");
29         studentService.save(student);
30         studentService.save(student);
31     }
32
33 }
34
```

```

1 package studentdbms.controller;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.ModelAttribute;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.PostMapping;
12 import org.springframework.web.bind.annotation.RequestMapping;
13
14 import studentdbms.entity.Student;
15 import studentdbms.service.StudentService;
16
17 @Controller
18 @RequestMapping("/students")
19 public class StudentController {
20     @Autowired
21     private StudentService studentService;
22
23     @GetMapping("/")
24     public String findAll(Model model) {
25         List<Student> students = studentService.findAll();
26         model.addAttribute("students", students);
27         model.addAttribute("student", new Student());
28         return "student-list";
29     }
30
31     @GetMapping("/add")
32     public String add(Model model) {
33         Student theStudent = new Student();
34         model.addAttribute("theStudent", theStudent);
35         return "student-form";
36     }
37
38     @PostMapping("/save")
39
40     @PostMapping("/save")
41     public String save(@ModelAttribute("theStudent") Student theStudent)
42     {
43         studentService.save(theStudent);
44         return "redirect:/students";
45     }
46
47     @GetMapping("/{id}")
48     public String deleteStudentById(@PathVariable("id") Integer id) {
49         studentService.deleteStudentById(id);
50         return "redirect:/students";
51     }
52
53     @GetMapping("/single/{id}")
54     public String getStudentById(@PathVariable Integer id, Model model)
55     {
56         Student singleStudent = studentService.getStudentById(id);
57         model.addAttribute("singleStudent", singleStudent);
58         return "single-student-details";
59     }
60
61     @PostMapping("/updateStudent/{id}")
62     public String updateStudentName(@PathVariable("id") int id,
63     @ModelAttribute("temp") Student student) {
64         Student updatedStudent = studentService.findById(id);
65         updatedStudent.setName(student.getName());
66         studentService.save(updatedStudent);
67         return "redirect:/students";
68     }
69 }

```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="
3 http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5 https://maven.apache.org/xsd/maven-4.0.0.xsd">
6 <modelVersion>4.0.0</modelVersion>
7 <parent>
8 <groupId>org.springframework.boot</groupId>
9 <artifactId>spring-boot-starter-parent</artifactId>
10 <version>2.6.3</version>
11 <relativePath><!-- lookup parent from repository -->
12 </relativePath>
13 </parent>
14 <groupId>StudentMS</groupId>
15 <artifactId>StudentMS</artifactId>
16 <version>0.0.1-SNAPSHOT</version>
17 <name>StudentMS</name>
18 <description>StudentMS</description>
19
20 <properties>
21 <java.version>1.8</java.version>
22 <spring-cloud.version>2021.0.6</spring-cloud.version>
23 </properties>
24
25 <dependencies>
26 <dependency>
27 <groupId>org.springframework.boot</groupId>
28 <artifactId>spring-boot-starter-actuator</artifactId>
29 </dependency>
30 <dependency>
31 <groupId>org.springframework.boot</groupId>
32 <artifactId>spring-boot-starter-data-jpa</artifactId>
33 </dependency>
34 <dependency>
35 <groupId>org.springframework.boot</groupId>
36 <artifactId>spring-boot-starter-web</artifactId>
37 </dependency>
38 <dependency>
39 <groupId>com.h2database</groupId>
40 <artifactId>h2</artifactId>
41 <scope>runtime</scope>
42 </dependency>
43 <dependency>
44 <groupId>org.springframework.boot</groupId>
45 <artifactId>spring-boot-starter-test</artifactId>
46 <scope>test</scope>
47 <exclusions>
48 <exclusion>
49 <groupId>org.junit.vintage</groupId>
50 <artifactId>junit-vintage-engine</artifactId>
51 </exclusion>
52 </exclusions>
53 </dependency>
54 <dependency>
55 <groupId>org.springframework.security</groupId>
56 <artifactId>spring-security-test</artifactId>
57 <scope>test</scope>
58 </dependency>
59 </dependencies>
60
61 </project>

```

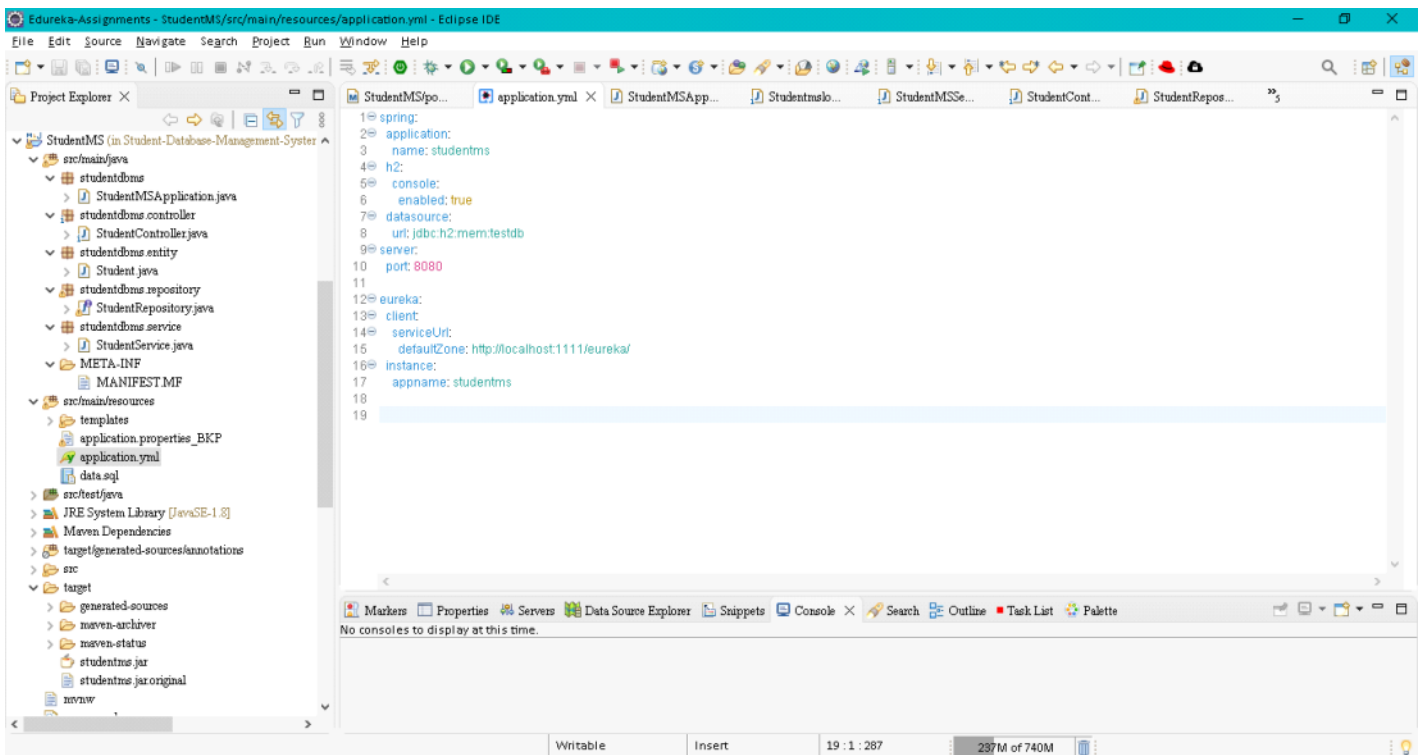
```
1 package studentdbas.entity;
2
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.Id;
7
8 @Entity
9 public class Student {
10     @Id
11     @GeneratedValue
12     private int id;
13
14     @Column
15     private String name;
16
17     public Student() {
18     }
19
20     public Student(String name) {
21         this.name = name;
22     }
23
24     public int getId() {
25         return id;
26     }
27
28     public String getName() {
29         return name;
30     }
31
32     public void setName(String name) {
33         this.name = name;
34     }
35
36     public void setId(int id) {
37         this.id = id;
38     }
39
40     @Override
41     public String toString() {
42         return "Student [id=" + id + ", name=" + name + "];"
43     }
44 }
```

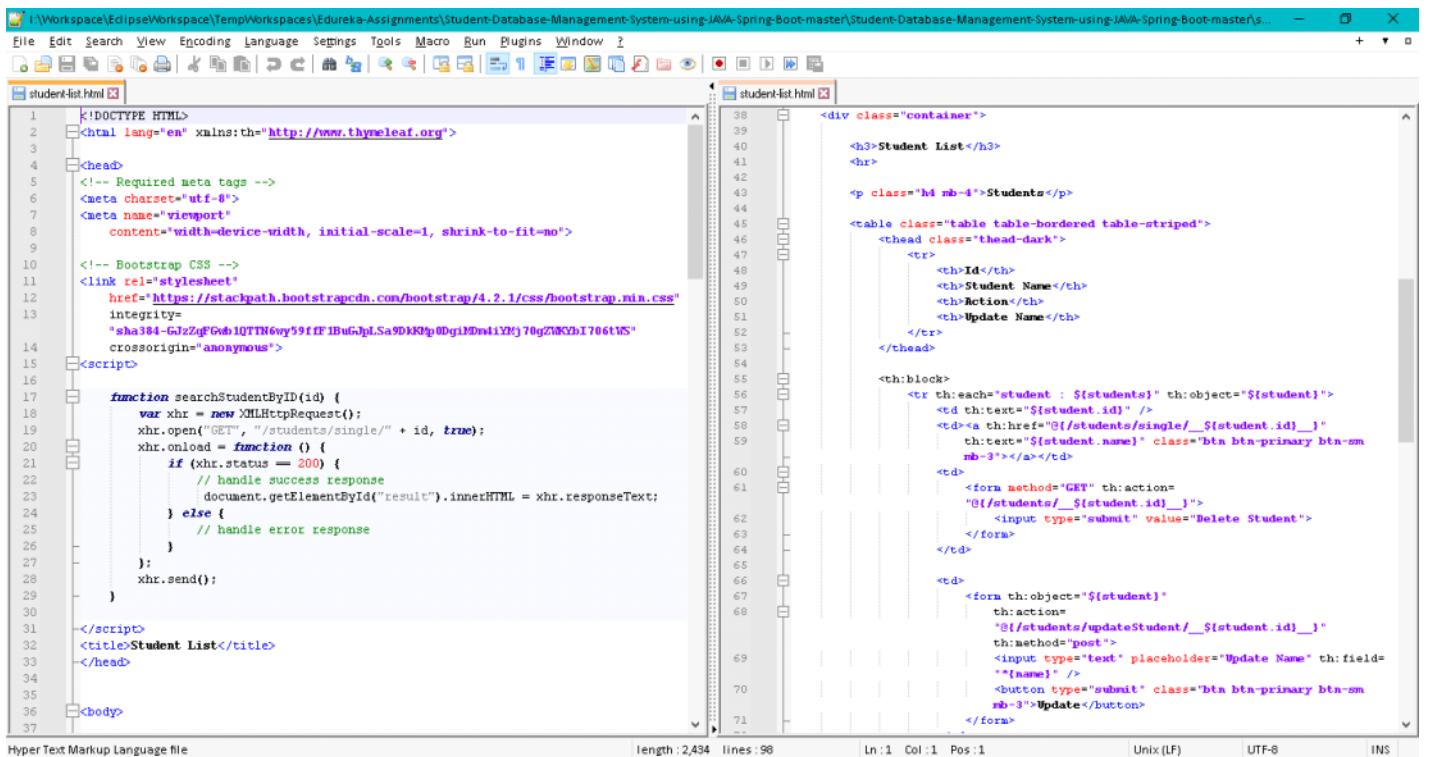
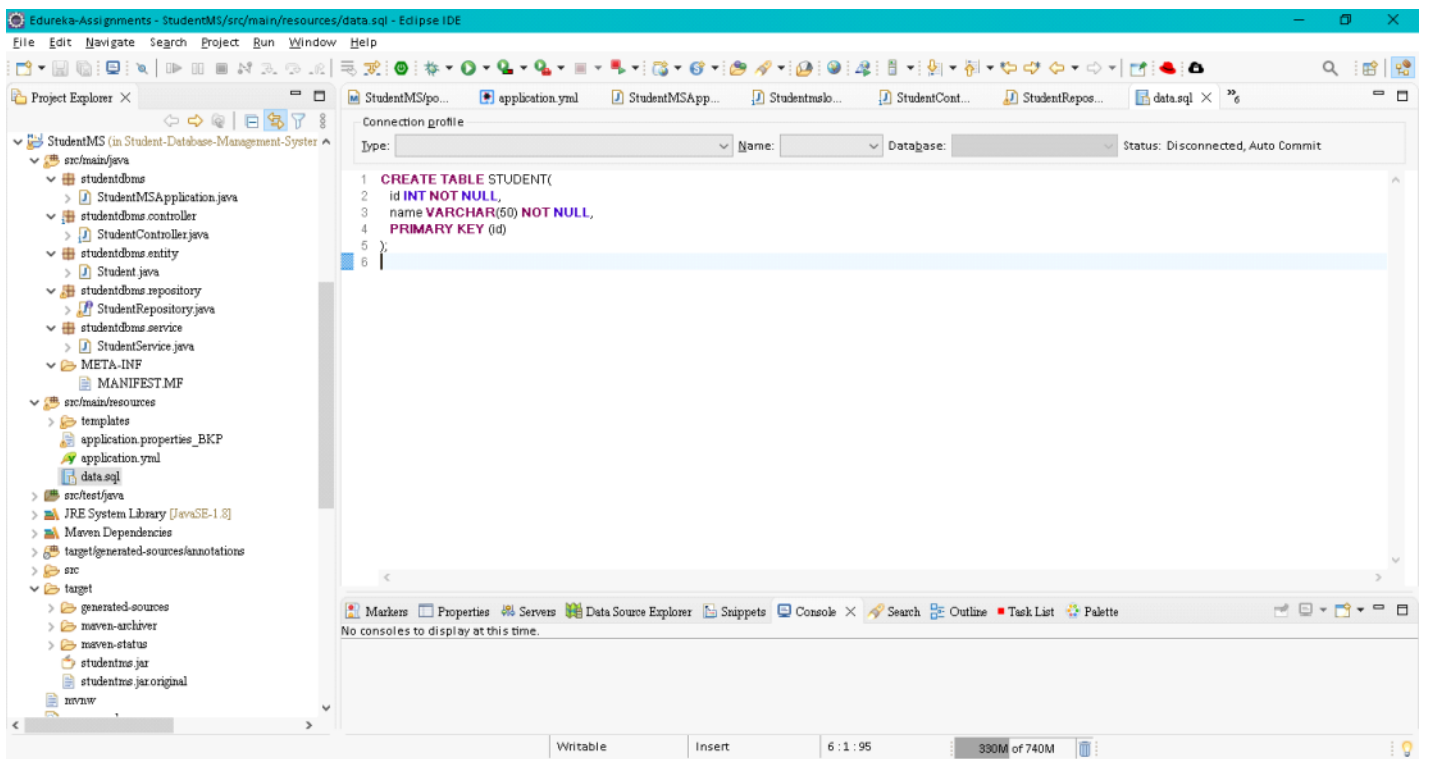
Java source file length: 641 lines: 45 Ln: 1 Col: 1 Pos: 1 Unix (LF) UTF-8 INS

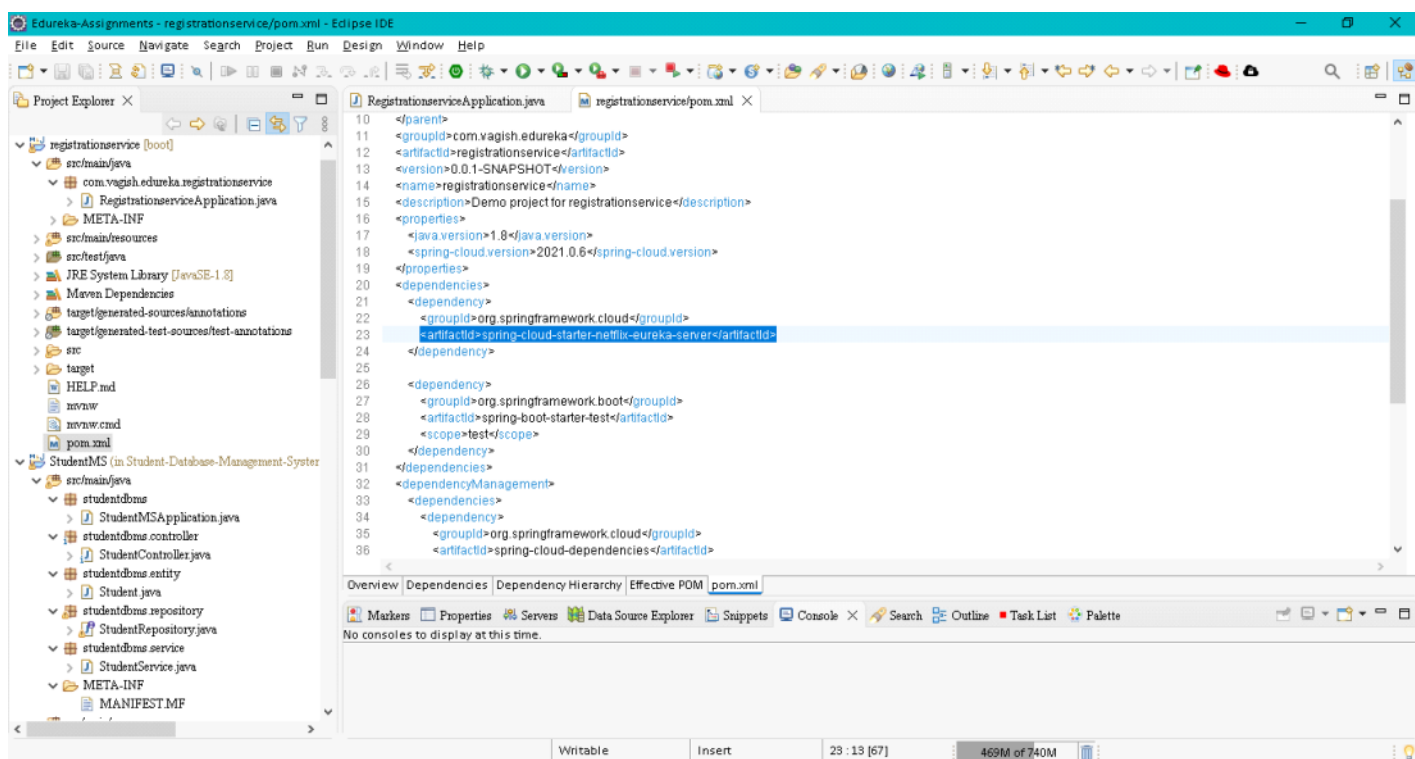
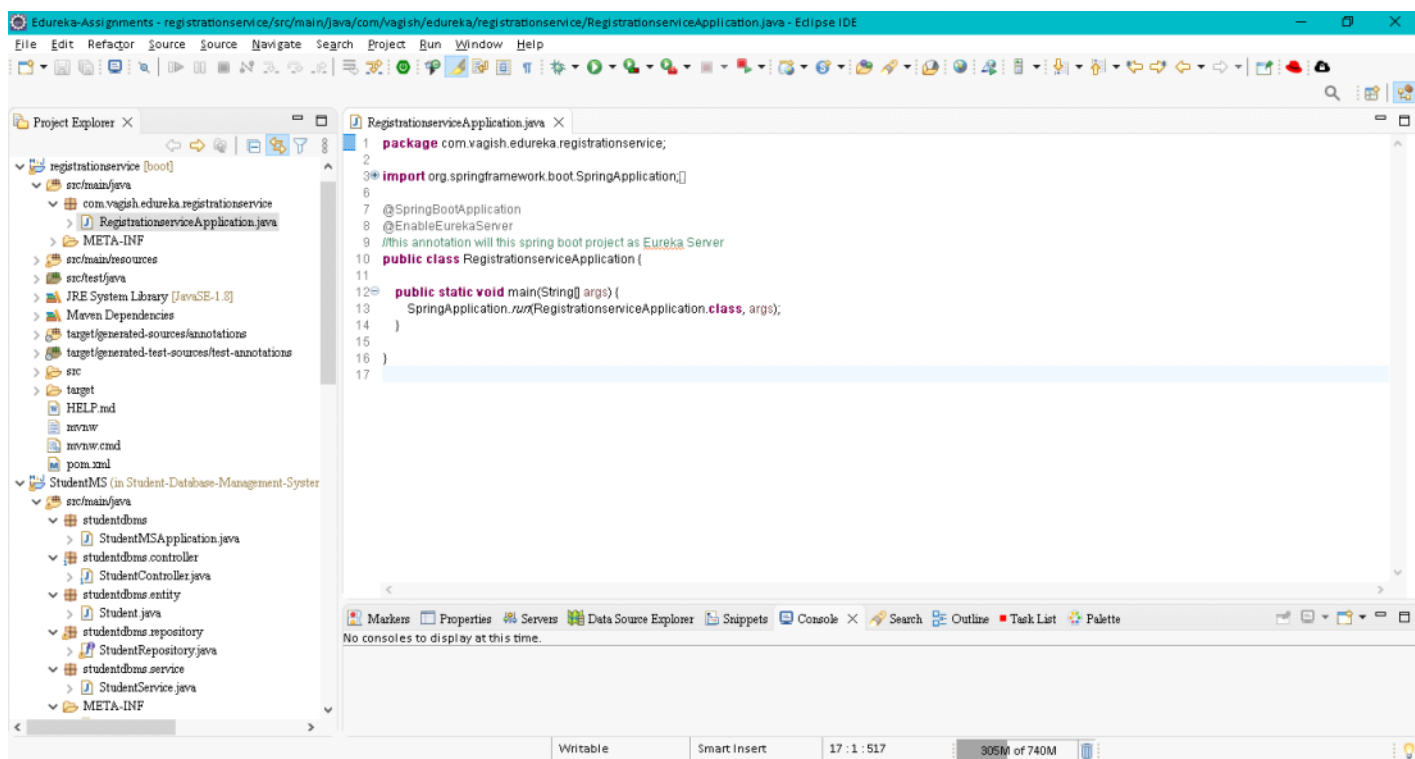
```
1 package studentdbas.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4 import org.springframework.stereotype.Repository;
5
6 import studentdbas.entity.Student;
7
8 @Repository
9 public interface StudentRepository extends JpaRepository<Student, Integer> {
10
11 }
12 }
```

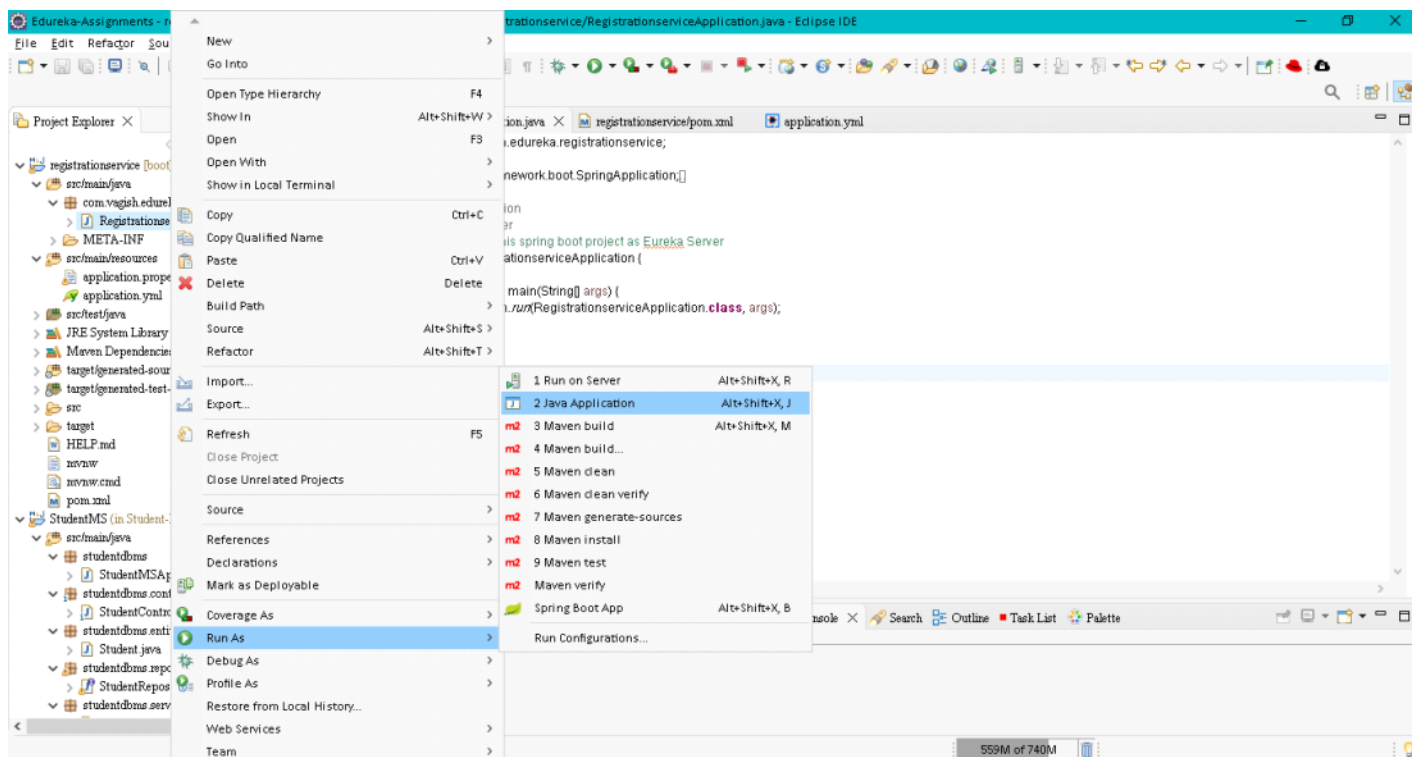
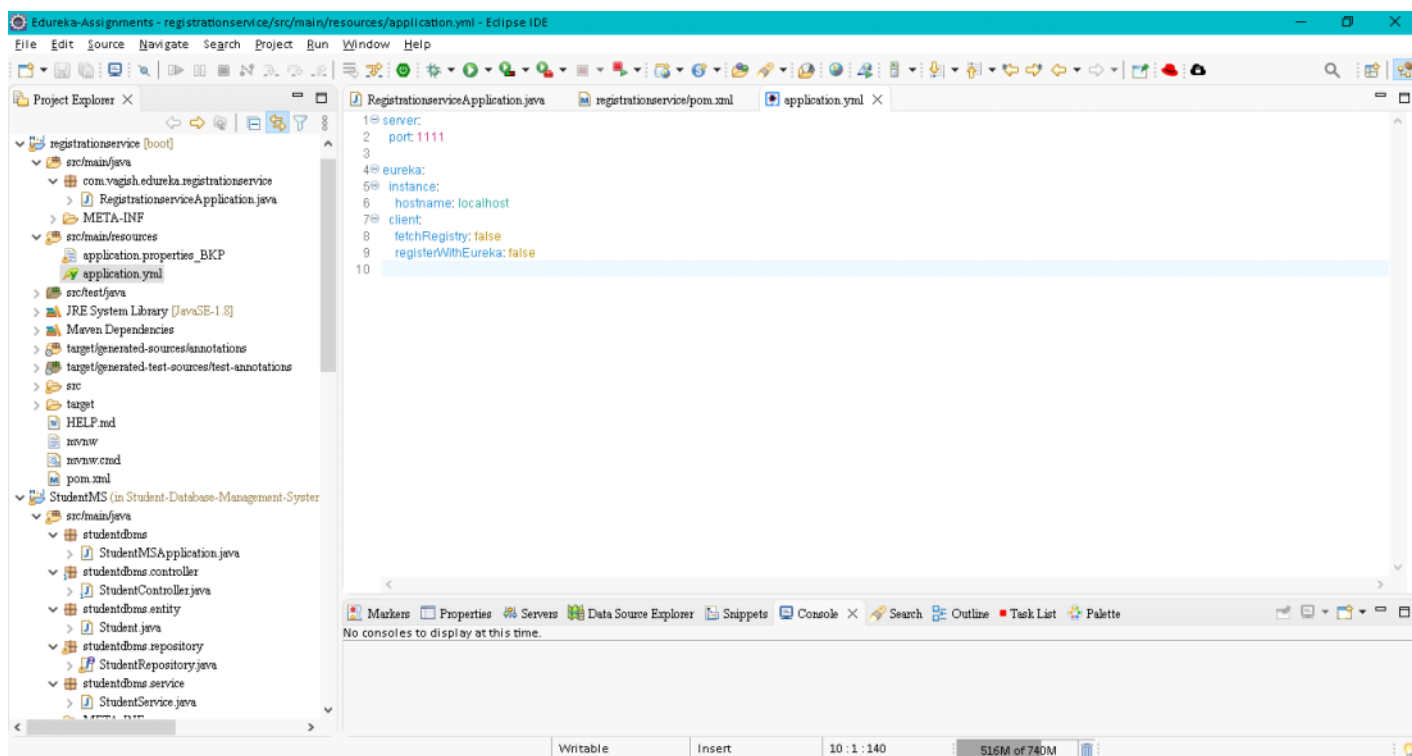
Java source file length: 274 lines: 12 Ln: 1 Col: 1 Pos: 1 Unix (LF) UTF-8 INS

```
1 package studentdbms.service;
2
3 import java.util.List;
4 import java.util.Optional;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8 import org.springframework.transaction.annotation.Transactional;
9
10 import studentdbms.entity.Student;
11 import studentdbms.repository.StudentRepository;
12
13 @Service
14 public class StudentService {
15     @Autowired
16     private StudentRepository studentRepository;
17
18     public List<Student> findAll() {
19         return studentRepository.findAll();
20     }
21
22     public Student findById(int id) {
23         return studentRepository.findById(id).get();
24     }
25
26     @Transactional
27     public Student save(Student student) {
28         return studentRepository.save(student);
29     }
30
31     public Student getStudentById(Integer id) {
32         Optional<Student> student = studentRepository.findById(id);
33         if (student.isPresent()) {
34             return student.get();
35         } else {
36             return null;
37         }
38     }
39
40     public void deleteStudentById(Integer id) {
41         studentRepository.deleteById(id);
42     }
43 }
```









Eureka

localhost:1111

HOME LAST 1000 SINCE STARTUP

spring Eureka

System Status

Environmenttest

Data centerdefault

Current time2023-04-11T13:06:32 +0530

Uptime00:00

Lease expiration enabledfalse

Renews threshold1

Renews (last min)0

DS Replicas

Instances currently registered with Eureka

ApplicationAMIsAvailability ZonesStatus

No instances available

General Info

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

I:\Workspace\Files\microservices-jars>dir
Volume in drive I is Workspace SSD
Volume Serial Number is A414-2FD1

Directory of I:\Workspace\Files\microservices-jars

11-04-2023  12:31    <DIR>          .
11-04-2023  12:31    <DIR>          ..
11-04-2023  12:29             63,078,901 studentms.jar
               1 File(s)        63,078,901 bytes
               2 Dir(s)  42,622,935,040 bytes free

I:\Workspace\Files\microservices-jars>java -jar studentms.jar 3000
```


Eureka

localhost:1111

Data centerdefault

Uptime00:01

Lease expiration enabledfalse

Renews threshold3

Renews (last min)0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
STUDENTMS	n/a (1)	(1)	UP (1) - localhost:studentms:3000

General Info

Name	Value
total-avail-memory	308mb
num-of-cpus	4
current-memory-usage	156mb (50%)
server-uptime	00:01

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

I:\Workspace\Files\microservices-jars>java -jar studentms.jar 5000
```

Eureka

localhost:1111

Data centerdefault

Uptime00:01

Lease expiration enabledfalse

Renews threshold5

Renews (last min)0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
STUDENTMS	n/a (2)	(2)	UP (2) - localhost-studentms:5000 - localhost-studentms:3000

General Info

Name	Value
total-avail-memory	308mb
num-of-cpus	4
current-memory-usage	160mb (51%)
server-uptime	00:01

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

File Edit Refactor Source Source Navigate Search Project Run Window Help

StudentmsloadbalancerApplication.java

StudentMSServiceLoadBalancerController.java

```
1 package com.vagish.eureka.studentmsloadbalancer;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @EnableDiscoveryClient
7 public class StudentmsloadbalancerApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(StudentmsloadbalancerApplication.class, args);
11     }
12
13     @Bean
14     public RestTemplate restTemplate() {
15         return new RestTemplate();
16     }
17 }
```

Writable Smart Insert 1:1:0 400M of 740M

```
Edureka-Assignments - studentmsloadbalancer/src/main/java/com/vagish/eureka/studentmsloadbalancer/controller/StudentMSServiceLoadBalancerController.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

StudentmsloadbalancerApplication.java StudentMSServiceLoadBalancerController.java
1 package com.vagish.eureka.studentmsloadbalancer.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 public class StudentMSServiceLoadBalancerController {
7
8     @Autowired
9     LoadBalancerClientFactory clientFactory;
10
11     @Autowired
12     RestTemplate restTemplate;
13
14     @GetMapping("/students")
15     public String findAllStudents(Model model) {
16
17         // this will connect to eureka server, asks for address details of student
18         // service
19         // Eureka server will respond with student service details
20         RoundRobinLoadBalancer lb = clientFactory.getInstance("STUDENTMS", RoundRobinLoadBalancer.class);
21
22         // load balancer has to choose an instance from the list returned by Eureka
23         // server
24         ServiceInstance instance = lb.choose().block().getServer();
25         String host = instance.getHost();
26         int port = instance.getPort();
27
28         String URL = "http://" + host + ":" + port + "/students";
29         System.out.println("LB-Choosen-URL: " + URL);
30         String response = restTemplate.getForObject(URL, String.class);
31
32         return response;
33     }
34
35     @GetMapping("/students/{apiName}")
36 }
```

```
Edureka-Assignments - studentmsloadbalancer/src/main/resources/application.yml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help

StudentmsloadbalancerApplication.java StudentMSServiceLoadBalancerController.java application.yml
1 spring:
2   application:
3     name: studentmsloadbalancer
4   h2:
5     console:
6       enabled: true
7   datasource:
8     url: jdbc:h2:mem:testdb
9   server:
10    port: 8080
11
12 eureka:
13   client:
14     serviceUrl:
15       defaultZone: http://localhost:1111/eureka/
16     instance:
17       appname: studentmsloadbalancer
18
19
```

Eureka

localhost:1111

Data centerdefault

Uptime00:03

Lease expiration enabledfalse

Renews threshold6

Renews (last min)4

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
STUDENTMS	n/a (2)	(2)	UP (2) - localhost:studentms:5000 , localhost:studentms:3000
STUDENTMSLOADBALANCER	n/a (1)	(1)	UP (1) - localhost:studentmsloadbalancer:6500

General Info

Name	Value
total-avail-memory	308mb
num-of-cpus	4
current-memory-usage	160mb (51%)

Eureka

Student List

localhost:6500/students

Student List

Students

Id	Student Name	Action	Update Name
1	<div>Vagish</div>	<div>Delete Student</div>	<div><div>Update Name</div><div>Update</div></div>

Add new Student

ID

Search

Eureka-Assignments - studentmsloadbalancer/src/main/java/com/vagish/eureka/studentmsloadbalancer/StudentmsloadbalancerApplication.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Markers Properties Servers Data Source Explorer Snippets Console Search Outline Task List Palette

```
StudentmsloadbalancerApplication [Java Application] C:\Java\jdk1.8.0_341\bin\java.exe (11-Apr-2023, 1:09:40 pm) [pid: 6216]
2023-04-11 13:09:17.492 INFO 6216 — [main] o.s.c.l.BootstrapLogger : Initializing Spring WebApplicationInitializer
2023-04-11 13:09:18.566 WARN 6216 — [main] i.guration.LoadBalancerCaffeineWarnLogger : Spring Cloud LoadBalancer is currently working with the default cache. While this cache implementation is useful for development and tests, it's recommended to use Caffeine cache in production. You can switch to using Caffeine cache, by adding it and org.springframework.cache.caffeine.CaffeineCacheManager to the classpath.
2023-04-11 13:09:18.600 INFO 6216 — [main] o.s.c.n.eureka.InstanceInfoFactory : Setting initial instance status as: STARTING
2023-04-11 13:09:18.699 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Initializing Eureka in region us-east-1
2023-04-11 13:09:18.716 INFO 6216 — [main] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2023-04-11 13:09:18.748 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Disable delta property : false
2023-04-11 13:09:18.748 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Single vip registry refresh property : null
2023-04-11 13:09:18.748 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Force full registry fetch : false
2023-04-11 13:09:18.748 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Application is null : false
2023-04-11 13:09:18.749 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Registered Applications size is zero : true
2023-04-11 13:09:18.749 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Application version is -1: true
2023-04-11 13:09:18.749 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Getting all instance registry info from the eureka server
2023-04-11 13:09:19.359 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : The response status is 200
2023-04-11 13:09:19.366 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Starting heartbeat executor: renew interval is: 30
2023-04-11 13:09:19.371 INFO 6216 — [main] c.n.discovery.InstanceInfoReplicator : InstanceInfoReplicator onDemand update allowed rate per min is 4
2023-04-11 13:09:19.378 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Discovery Client initialized at timestamp 1681198759377 with initial instances count: 2
2023-04-11 13:09:19.385 INFO 6216 — [main] o.s.c.n.e.s.EurekaServiceRegistry : Registering application STUDENTMSLOADBALANCER with eureka with status UP
2023-04-11 13:09:19.386 INFO 6216 — [main] com.netflix.discovery.DiscoveryClient : Saw local status change event StatusChangeEvent [timestamp=1681198759386, current=UP, previous=STARTING]
2023-04-11 13:09:19.389 INFO 6216 — [infoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_STUDENTMSLOADBALANCER/localhost:studentmsloadbalancer:6500 - registering service...
2023-04-11 13:09:19.438 INFO 6216 — [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 6500 (http) with context path '/'
2023-04-11 13:09:19.439 INFO 6216 — [main] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 6500
2023-04-11 13:09:19.473 INFO 6216 — [infoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_STUDENTMSLOADBALANCER/localhost:studentmsloadbalancer:6500 - registration status: 204
2023-04-11 13:09:20.596 INFO 6216 — [main] o.s.c.l.BootstrapLogger : Cannot determine local hostname
2023-04-11 13:09:20.610 INFO 6216 — [main] c.v.e.s.StudentmsloadbalancerApplication : Started StudentmsloadbalancerApplication in 9.062 seconds (JVM running for 9.651)
2023-04-11 13:10:30.957 INFO 6216 — [nio-6500-exec-1] o.a.c.c.C.[Tomcat] [localhost/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-04-11 13:10:30.957 INFO 6216 — [nio-6500-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-04-11 13:10:30.959 INFO 6216 — [nio-6500-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
LB-Choosen-URL: http://localhost:5000/students
LB-Choosen-URL: http://localhost:3000/students
LB-Choosen-URL: http://localhost:5000/students
LB-Choosen-URL: http://localhost:3000/students
LB-Choosen-URL: http://localhost:5000/students
LB-Choosen-URL: http://localhost:3000/students
```

284M of 734M

Eureka x Save Student x +

← → ↻ local:5000/students/add

Student Form

[Back to Student List](#)

Save Student

[Save](#)

Eureka

Student List

localhost5000/students

Student List

Students

Id	Student Name	Action	Update Name
1	Vagish	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>
2	Vag2	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>

Add new Student

ID

Search

Eureka

Student List

localhost5000/students

Student List

Students

Id	Student Name	Action	Update Name
1	Vagish	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>
2	Vag2	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>
3	StudentToBeDeleted	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>

Add new Student

ID

Search

Eureka

Student List

localhost:5000/students

Student List

Students

Id	Student Name	Action	Update Name
1	Vagish	Delete Student	<input type="text" value="Update Name"/> Update
2	Vag2	Delete Student	<input type="text" value="Rohan"/> Update

Add new Student

Search

Eureka

Student List

localhost:5000/students

Student List

Students

Id	Student Name	Action	Update Name
1	Vagish	Delete Student	<input type="text" value="Update Name"/> Update
2	Rohan	Delete Student	<input type="text" value="Update Name"/> Update

Add new Student

Search

Eureka

Student List

localhost5000/students

Student List

Students

Id	Student Name	Action	Update Name
1	Vagish	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>
2	Rohan	Delete Student	<input type="text" value="Update Name"/> <button>Update</button>

Add new Student

Search

Single Student Details by ID

Id	Student Name	Action
2	Rohan	Delete Student

Back to Student List

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