Case Study Title: Online Course Enrollment System

Scenario: An educational startup wants to build a basic web application for students to view available courses and enroll online. The company has a small IT team familiar with Java and wants to use Spring MVC to ensure the application follows a clean, maintainable structure based on MVC architecture.

Objectives: 1. 2. 3. Display a list of available courses. Allow students to register by filling out an enrollment form. Confirm enrollment and store student details.

System Requirements: • Java 17 or later • Spring MVC framework • Apache Tomcat or embedded server • Maven for dependency management • JSP for frontend • Eclipse or Spring Tool Suite (STS) IDE

How Spring MVC Helps: Spring MVC allows the application to be divided into three main components: Layer Model Responsibility Represents the data (Course, Student, Enrollment info) View Displays the HTML pages for course listing and form input Controller Manages user requests and application logic

Application Flow: 1. User accesses the homepage → A controller handles this request and returns a list of available courses via the view.

2. User selects a course and proceeds to enroll→ A new view (HTML form) is presented to collect user data (name, email, etc.).

3.Form is submitted → The controller receives the form data, validates it, and passes it to the service layer or model to be processed.

4.Success page is shown → A confirmation view is displayed with enrollment details. 🧩 Components in Spring MVC: Component @Controller Description Handles web requests (e.g., show courses, process enrollment) @RequestMapping Model object Maps URLs to specific controller methods Holds the data to be passed to the view

Pom.xml:

Coursr.java:

```
package com.example.model;

public class Course {
    private String id;
    private String name;

    public Course() {}
    public Course(String id, String name) {
        this.id = id;
        this.name = name;
    }

    public String getId() { return id; }
    public String getName() { return name; }

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

Student.java:

```
package com.example.model;

public class Student {
    private String name;
    private String email;
    private String courseId;

    public String getName() { return name; }
    public String getEmail() { return email; }
    public String getCourseId() { return courseId; }

    public void setName(String name) { this.name = name; }
    public void setEmail(String email) { this.email = email; }
    public void setCourseId(String courseId) { this.courseId = courseId; }
}
```

CorseController.java:

```
package com.example.controller;

import com.example.model.Course;
import com.example.model.Student;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import java.util.Arrays;
import java.util.List;

@Controller
public class CourseController {

    @GetMapping("/courses")
    public String showCourses(Model model) {
        List<Course> courses = Arrays.asList(
            new Course("C101", "Java"),
            new Course("C102", "Spring MVC"),
            new Course("C103", "Hibernate")
            );
        model.addAttribute("courses", courses);
        return "courses";
    }

    @GetMapping("/enroll")
    public String showEnrollmentForm(Model model) {
        model.addAttribute("student", new Student());

        return "enroll";

    }


    @PostMapping("/submitEnrollment")

    public String submitEnrollment(@ModelAttribute("student") Student student, Model model) {

        model.addAttribute("student", student);

        return "success";

    }

}
```

Courses.jsp:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>

<head><title>Available Courses</title></head>

<body>

<h2>Available Courses</h2>

<ul>

  <c:forEach var="course" items="${courses}">

    <li>${course.name} - <a href="enroll">Enroll</a></li>

  </c:forEach>

</ul>

</body>

</html>
```

Enroll.jsp:

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>

<html>

<head><title>Enroll</title></head>
```

```
<body>
<h2>Enroll in a Course</h2>
<form:form method="POST" action="submitEnrollment" modelAttribute="student">
   Name: <form:input path="name"/><br/>
   Email: <form:input path="email"/><br/>
   Course ID: <form:input path="courseId"/><br/>
   <input type="submit" value="Submit"/>
</form:form>
</body>
</html>
```

Success.jsp:

```
<html>
<head><title>Success</title></head>
<body>
<h2>Enrollment Successful!</h2>
<p>Name: ${student.name}</p>
<p>Email: ${student.email}</p>
<p>Course ID: ${student.courseId}</p>
</body>
</html>
```

Dispatcher-servlet.xml:

```
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:context="http://www.springframework.org/schema/context"
      xmlns:mvc="http://www.springframework.org/schema/mvc"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="
```

```
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <context:component-scan base-package="com.example.controller" />
    <mvc:annotation-driven />

    <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views/" />
        <property name="suffix" value=".jsp" />
    </bean>
</beans>
```

Case study2:Online Shopping Portal

OrderService.java:

```
package com.example.service;

import org.springframework.stereotype.Service;

@Service
public class OrderService {

    public void addToCart(String product) {
        System.out.println("Adding product to cart: " + product);
    }
    }

    public void placeOrder(String orderId) {
        if ("INVALID".equals(orderId)) {
            throw new RuntimeException("Invalid order ID");
        }
        System.out.println("Order placed successfully: " + orderId);
    }

    public void cancelOrder(String orderId) {
        System.out.println("Order cancelled: " + orderId);
    }
}
```

OrderLoggingAspect.java:

```
package com.example.aspect;

import org.aspectj.lang.annotation.*;
import org.springframework.stereotype.Component;

@Aspect
@Component
public class OrderLoggingAspect {

    @Before("execution(* com.example.service.OrderService.*(..))")
    public void beforeAdvice() {
        System.out.println(" Method is starting...");
    }

    @AfterReturning("execution(* com.example.service.OrderService.*(..))")
    public void afterReturningAdvice() {
        System.out.println(" Method completed successfully.");
    }

    @AfterThrowing("execution(* com.example.service.OrderService.*(..))")
    public void afterThrowingAdvice() {
        System.out.println(" Exception occurred during method execution.");
    }

    @After("execution(* com.example.service.OrderService.*(..))")
    public void afterAdvice() {
        System.out.println(" Method execution ended.");
    }
}
```

AppConfig.java:

```
package com.example.config;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.EnableAspectJAutoProxy;

@Configuration
@ComponentScan(basePackages = "com.example")
@EnableAspectJAutoProxy
public class AppConfig {
}
```

MainApp.java:

```
package com.example;

import com.example.config.AppConfig;
import com.example.service.OrderService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);
        OrderService service = context.getBean(OrderService.class);

        service.addToCart("Laptop");
        service.placeOrder("ORD123");

        try {
            service.placeOrder("INVALID");
        } catch (Exception e) {
            System.out.println("Caught Exception: " + e.getMessage());
        }

        service.cancelOrder("ORD123");
    }
}
```