

## Day-9 Spring MVC and Spring AOP Assignment

### Case Study 1: Online Course Enrollment System (Spring MVC)

#### 1. Course.java

```
package com.example.model;

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

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

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

#### 2. Student.java

```
package com.example.model;

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

    public String getName() { return name; }
    public void setName(String name) { this.name = name; } public String getEmail() { return
    email; }
    public void setEmail(String email) { this.email = email; } public int getCourseId() { return
    courseId; }
    public void setCourseId(int courseId) { this.courseId = courseId; }
}
```

#### 3. CourseController.java

```
package com.example.controller;

import java.util.List;
import org.springframework.stereotype.Controller; import
org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*; import
com.example.model.Course;
import com.example.model.Student; @Controller
```

```

public class CourseController {

    List<Course> courseList = List.of( new Course(1,
        "Java"),
        new Course(2, "Python"),
        new Course(3, "Spring Boot")
    );

    @RequestMapping("/courses")
    public String showCourses(Model model) { model.addAttribute("courses",
        courseList); return "courses";
    }

    @RequestMapping("/enroll")
    public String enrollForm(Model model) { model.addAttribute("student",
        new Student()); model.addAttribute("courses", courseList); return
        "enroll";
    }

    @PostMapping("/submitEnrollment")
    public String submitForm(@ModelAttribute("student") Student student, Model model) { model.addAttribute("student",
        student);
        return "success";
    }
}

```

#### 4. dispatcher-servlet.xml

```

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    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">

    <context:component-scan base-package="com.example" />

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

```

#### 5. JSP Files

courses.jsp:

```
<h2>Available Courses</h2>
```

```

<ul>
    <c:forEach items="\${courses}" var="course">
        <li>\${course.id} - \${course.name}</li>
    </c:forEach>
</ul>

<a href="enroll">Enroll Now</a>

enroll.jsp:
<h2>Enrollment Form</h2>
<form action="submitEnrollment" method="post"> Name: <input
    type="text" name="name"/><br/> Email: <input type="email"
    name="email"/><br/> Course:
    <select name="courseId">
        <c:forEach items="\${courses}" var="course">
            <option value="\${course.id}">\${course.name}</option>
        </c:forEach>
    </select><br/>
    <input type="submit" value="Enroll"/>
</form>

success.jsp:
<h2>Enrollment Successful</h2>
<p>Thank you, \${student.name}, for enrolling in course ID: \${student.courseId}</p>

```

## Case Study 2: Online Shopping Portal – Spring AOP

### 1. OrderService.java

```

java
CopyEdit
package com.example.service;

import org.springframework.stereotype.Component;

@Component
public class OrderService {

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

    public void placeOrder(String orderId) {
        if (orderId.equals("INVALID_ID")) {
            throw new RuntimeException("OrderNotFoundException");
        }
        System.out.println("Order placed: " + orderId);
    }

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

```

## 2. OrderLoggingAspect.java

java

CopyEdit

```
package com.example.aspect;
```

```
import org.aspectj.lang.JoinPoint;
```

```
import org.aspectj.lang.annotation.*;
```

```
import org.springframework.stereotype.Component;
```

```
@Aspect
```

```
@Component
```

```
public class OrderLoggingAspect {
```

```
    @Before("execution(* com.example.service.OrderService.*(..))")
```

```
    public void beforeAdvice(JoinPoint jp) {
```

```
        System.out.println("Starting method: " + jp.getSignature());
```

```
    }
```

```
    @AfterReturning("execution(* com.example.service.OrderService.*(..))")
```

```
    public void afterReturningAdvice(JoinPoint jp) {
```

```
        System.out.println("Successfully executed: " + jp.getSignature());
```

```
    }
```

```
    @AfterThrowing(pointcut = "execution(* com.example.service.OrderService.*(..))", throwing = "ex")
```

```
    public void afterThrowingAdvice(JoinPoint jp, Throwable ex) {
```

```
        System.out.println("Exception in " + jp.getSignature() + ": " + ex.getMessage());
```

```
    }
```

```
    @After("execution(* com.example.service.OrderService.*(..))")
```

```
    public void afterAdvice(JoinPoint jp) {
```

```
        System.out.println("Method ended: " + jp.getSignature());
```

```
    }
```

```
}
```

## 3. AppConfig.java (Spring Java Config)

java

CopyEdit

```
package com.example.config;
```

```
import org.springframework.context.annotation.*;
```

```
import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
```

```
import org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator;
```

```
import com.example.service.OrderService;
```

```
import com.example.aspect.OrderLoggingAspect;
```

```
@Configuration
```

```
@ComponentScan(basePackages = "com.example")
```

```
@EnableAspectJAutoProxy
```

```
public class AppConfig {}
```

#### 4. MainApp.java

java

CopyEdit

```
package com.example;
```

```
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
```

```
import com.example.config.AppConfig;
```

```
import com.example.service.OrderService;
```

```
public class MainApp {
```

```
    public static void main(String[] args) {
```

```
        var context = new AnnotationConfigApplicationContext(AppConfig.class);
```

```
        OrderService service = context.getBean(OrderService.class);
```

```
        service.addToCart("iPhone");
```

```
        service.placeOrder("ORD123");
```

```
        try {
```

```
            service.placeOrder("INVALID_ID");
```

```
        } catch (Exception e) {}
```

```
        context.close();
```

```
    }
```

```
}
```