College Admission Fee Calculator - V1

Grade settings: Maximum grade: 100
Based on: College Admission Fee Calculator - V1
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 240 s **Maximum memory used**: 1.50 GiB **Maximum execution file size**: 1.50 GiB

College Admission Fee Calculator

One of the leading College in Tamilnadu approaches you to create a separate online application for calculating admission fees based on department (EEE/ECE/CSE) chosen by the applicant.

Create a Spring MVC Spring Boot Application for developing an Admission Fee Calculator Application. Design an Admission Fees Calculator Page to choose an appropriate department (EEE/ECE/CSE) and enter the other details like admissionId, hostelPreference and firstGraduateInfo.

On clicking Calculate button, the application should calculate the admissionFees based on the department chosen by the applicant. The applicant has to then be redirected to result.jsp page that displays the message "**Admission Fees to be paid is Rs.<<admissionFees>>"**

Application Work Flow:

- **AdmissionController** is the Controller class.
- **Admission** is the model class with 5 attributes admissionId, deptName, hostelPreference, tuitionFees and firstGraduateInfo along with its getters and setters.
- **AdmissionService** is the Service class which has a method called calculateAdmissionFees that takes Admission as its argument and returns double.

1. Calculate the admissionFees depending on the deptName and the other information provided by the customer.
2. This method should set the **tuitionFees** instance variable based on the value given in the below table for each department.
3. It should be returned as the double.

deptName	tuitionFees
EEE	45000.00
ECE	50000.00
CSE	60000.00

Initially, the applicant should be routed via the AdmissionController's **feesCalculatorPage** method to **feescalculatorpage.jsp** that allows applicant to choose the deptName and enter the other inputs.

[Note: feesCalculatorPage method has to be written inside the AdmissionController]

A method in the **AdmissionController** known as buildState should be annotated with the ModelAttribute "**deptNameList"**. This method should populate the depart name **(EEE/ECE/CSE)** the Map as key-value pair. Both key and value be the same department name. (Eg: Key- **EEE**, Value- **EEE**) and then return the Map. This should be then used to autopopulate the deptName in the **feescalculatorpage.jsp**

A method in the **AdmissionController** known as **getHostelPreference** should be annotated with the ModelAttribute "**preferenceList"**. This method should populate the depart name **(YES/NO)** the Map as key-value pair. Both key and value be the same hostel preference. (Eg: Key- **YES**, Value- **YES**) and then return the Map. This should be then used to autopopulate the hostel preference in the **feescalculatorpage.jsp**

A method in the **AdmissionController** known as **getFirstGraduateInfo** should be annotated with the ModelAttribute "**graduateList"**. This method should populate the graduate info **(YES/NO)** the Map as key-value pair. Both key and value be the same department name. (Eg: Key- **YES**, Value- **YES**) and then return the Map. This should be then used to autopopulate the first graduate info in the **feescalculatorpage.jsp**

[Note: buildState, getHostelPreference and getFirstGraduateInfo methods should be written inside the AdmissionController]

On clicking the Calculate button, AdmissionController's **calculateAdmissionFees** method should be called. This method takes three arguments - model attribute named "**admission"** which holds the form populated **Admission** Object, BindingResult and the ModelMap.

- This method should calculate the **admissionFees** by invoking the **calculateAdmissionFees** method of the **AdmissionService**.
- **admissionFees** has to be calculated as shown in the below example,
- **Example:**

If applicant chosen deptName as "CSE", hostelPreference ="yes" and firstGraduateInfo ="No" then,

admissionFees = packageFees + tuitionFees + hostelFees = 30000.00 +60000.00 + 75000.00 = 165000.00

If applicant chosen deptName as "CSE", hostelPreference ="yes" and firstGraduateInfo ="yes" then **packageFees is not included for that particular applicant**

admissionFees = tuitionFees + hostelFees = 60000.00 + 75000.00 = 135000.00

If applicant chosen deptName as "CSE", hostelPreference ="No" and firstGraduateInfo ="yes" then **hostelFees and packageFees are not included for that particular applicant**

admissionFees = tuitionFees = 60000.00 = 60000.00

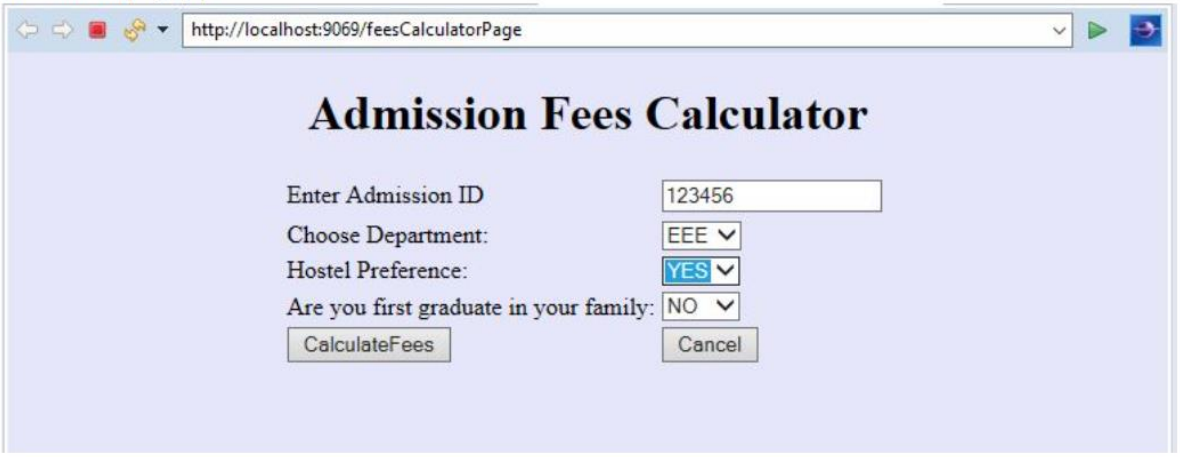
If applicant chosen deptName as "CSE", hostelPreference ="No" and firstGraduateInfo ="No" then **hostelFees is not included for that particular applicant**

admissionFees = tuitionFees + packageFees = 60000.00 + 30000.00 = 90000.00

Note: packageFees and hostelFees for each department is given in the below table

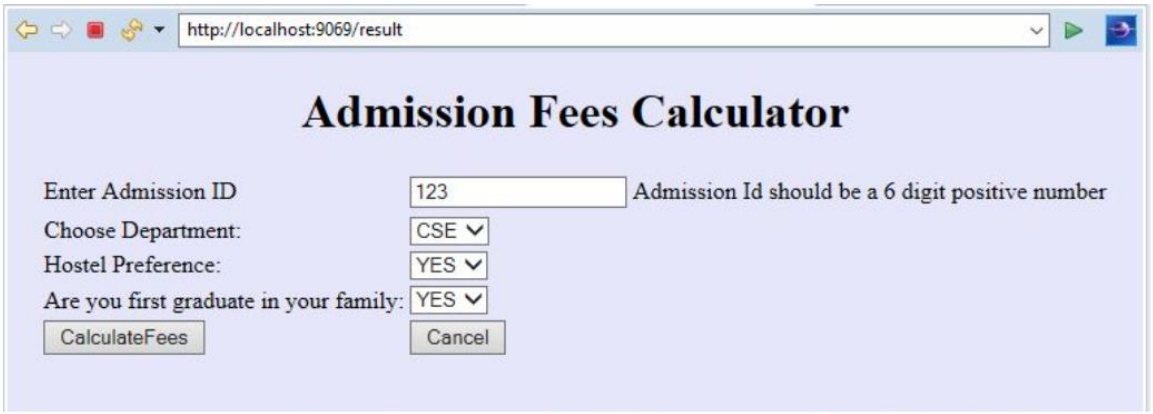
deptName	packageFees	hostelFees
EEE	20000.00	Rs.75000.00 for all the departments
ECE	25000.00	
CSE	30000.00	

feescalculatorpage.jsp



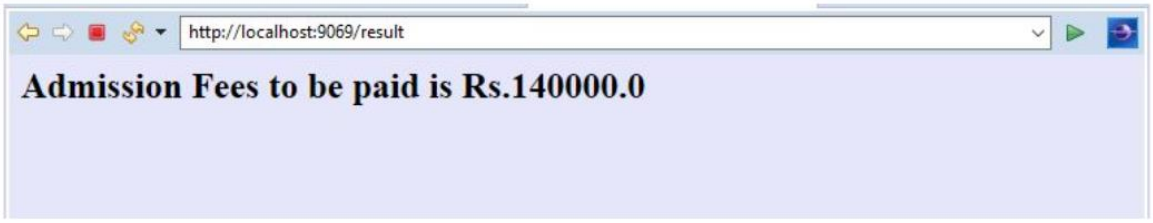
- If the admissionId entered by the applicant is less than 6 digits then an error message "**AdmissionId should be a 6 digit positive number"** has to be displayed.

feescalculatorpage.jsp



Redirect the customer to result.jsp page with a message "**Admission Fees to be paid is Rs.<<admissionFees>>**"

result.jsp



Design Constraints:

UI Design Constraints:

feescalculatorpage.jsp		
Component	**ID**	**Constraints**
Select	deptName	Should be auto populated using the model attribute written above

		the **buildState** method inside the **AdmissionController**. Do not hard code the values
Textbox	admissionId	Should be a 6 digit number
Select	preferHostel	Should be auto populated using the model attribute written above the **getHostelPreference** method inside the **AdmissionController**. Do not hard code the values
Select	firstGraduateInfo	Should be auto populated using the model attribute written above the **getFirstGraduateInfo** method inside the **AdmissionController**. Do not hard code the values
Submit	submit	-

Note:

- In **result.jsp**, the Result has to be rendered in the **<h2>** tag

Component Specification

Controller

AdmissionController			
AttributeName	**AttributeType**	**Access Specifier**	**Constraints**
service	AdmissionService	private	Use annotation to Autowire

AdmissionController			
Method Name	**Method Argument name: type**	**Return type**	**RequestMapping URL & Request Method**
feesCalculatorPage	modelAttribute "admission": Admission	String	/feesCalculatorPage & GET
calculateAdmissionFees	modelAttribute "admission": Admission, result:BindingResult, map:ModelMap	String	/result & POST
buildState		Map <String,String>	Should be annotated with ModelAttribute with name "**deptNameList"**
getHostelPreference		Map <String,String>	Should be annotated with ModelAttribute with name "**preferenceList"**
getFirstGraduateInfo		Map <String,String>	Should be annotated with ModelAttribute with name "**graduateList"**

Service

AdmissionService		
Method Name	**Method Argument name: type**	**Return type**
calculateAdmissionFees	admission : Admission	double

Model

Admission	
AttributeName	**AttributeType**
admissionId	int
deptName	String
preferHostel	String
firstGraduateInfo	String
tuitionFees	double

Overall Design Constraints:

- **AdmissionController** should be inside the package **com.controller**
- **AdmissionService** should be inside the package **com.service**
- **Admission** should be in the package **com.model**
- Use appropriate annotation to configure AdmissionService as a Service
- Use appropriate annotation to configure AdmissionController as a Controller
- AdmissionService should be autowired inside the AdmissionController.
- Use annotations to implement the business Validation as specified in the screen shot **[That is, when the admissionId is less than 6 digit then error message "Admission ID should be a 6 digit positive number" should be rendered in the UI]**
- Do not change the property name given in the application.properties files, you can change the value and you can include additional property if needed.
- In the pom.xml you are provided with all the dependencies needed for developing the application.
- You will not be evaluated based on the UI design (layout, color, formatting, etc.). You are free to have a basic UI with all the required UI components (input fields, buttons, labels, etc.). Expected components with the id alone should be designed as per the requirement.
- Adhere to the design specifications mentioned in the case study.
- Do not change or delete the class/method names or return types which are provided to you as a part of the base code skeleton.
- Please make sure that your code does not have any compilation errors while submitting your case study solution.

Result Description

Automatic evaluation[+]

AdmissionFeeCalculator/pom.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>2.1.8.RELEASE</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.controller</groupId>
	<artifactId>AdmissionFeeCalculator</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>AdmissionFeeCalculator</name>
	<description>Demo project for Spring Boot</description>

	<properties>
		<java.version>1.8</java.version>
	</properties>

	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>

		 <dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
   </dependency>
   <dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
 </dependency>
 <dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.1</version>
  <scope>provided</scope>
 </dependency>

 <dependency>
  <groupId>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
 </dependency>
 <dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
 </dependency>
 <!-- Spring test dependencies -->
	<dependency>
			<groupId>org.mockito</groupId>
			<artifactId>mockito-all</artifactId>
			<version>1.10.19</version>
			<scope>test</scope>
```

```
                </dependency>

                <!-- <dependency>
                        <groupId>org.seleniumhq.selenium</groupId>
                        <artifactId>selenium-java</artifactId>
                        <version>2.53.0</version>
                </dependency>-->

                <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>htmlunit-driver</artifactId>
    <version>2.26</version>
</dependency>

        <!-- https://mvnrepository.com/artifact/org.w3c.css/sac -->
<dependency>
    <groupId>org.w3c.css</groupId>
    <artifactId>sac</artifactId>
    <version>1.3</version>
</dependency>


<dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-test</artifactId>
                         <version>5.1.7.RELEASE</version>  <!-- 4.0.5 -->
                        <scope>test</scope>
                </dependency>
<!-- End of spring test dependencies -->



                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-test</artifactId>
                        <scope>test</scope>
                        <exclusions>
                                <exclusion>
                                        <groupId>org.junit.vintage</groupId>
                                        <artifactId>junit-vintage-engine</artifactId>
                                </exclusion>
                        </exclusions>
                </dependency>
        </dependencies>

        <build>
                <plugins>
                        <plugin>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-plugin</artifactId>
                        </plugin>
                </plugins>
        </build>

</project>


```

AdmissionFeeCalculator/src/main/java/com/controller/AdmissionController.java

```
package com.controller;

import java.util.HashMap;
import java.util.Map;

import javax.validation.Valid;
```

```java

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.model.Admission;
import com.service.AdmissionService;

@Controller
public class AdmissionController {

	@Autowired
	private AdmissionService service;

	@RequestMapping(value = "/feesCalculatorPage", method = RequestMethod.GET)
	public String feesCalculatorPage (@ModelAttribute("admission") Admission admission)
	{
			return "feescalculatorpage";
	}
	@ModelAttribute("deptNameList")
	public  Map<String, String> buildState(){

			Map<String, String> serviceMap = new HashMap<String, String>();
			serviceMap.put("EEE", "EEE");
			serviceMap.put("ECE", "ECE");
			serviceMap.put("CSE", "CSE");
			return serviceMap;
	}

	@ModelAttribute("preferenceList")
	public  Map<String, String> getHostelPreference(){

			Map<String, String> serviceMap = new HashMap<String, String>();
			serviceMap.put("YES", "YES");
			serviceMap.put("NO", "NO");
			return serviceMap;
	}

	@ModelAttribute("graduateList")
	public  Map<String, String> getFirstGraduateInfo(){

			Map<String, String> serviceMap = new HashMap<String, String>();
			serviceMap.put("YES", "YES");
			serviceMap.put("NO", "NO");
			return serviceMap;
	}

	@RequestMapping(value = "/result", method = RequestMethod.POST)
	public String calculateAdmissionFees(@Valid @ModelAttribute("admission") Admission admission,
				BindingResult result,ModelMap map)
	{
			if(result.hasErrors()) {
				return "feescalculatorpage";
	}
			else
			{
			double cost=service.calculateAdmissionFees(admission);
			map.addAttribute("cost",cost);
			return "result";

			}
	}

```

```
}

```

AdmissionFeeCalculator/src/main/java/com/example/demo/AdmissionFeeCalculatorApplication.java

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.controller","com.model","com.service"})
public class AdmissionFeeCalculatorApplication {

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

}

```

AdmissionFeeCalculator/src/main/java/com/model/Admission.java

```
package com.model;


import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;

import org.springframework.stereotype.Component;

@Component
public class Admission {

    @NotNull
	@Size(min=6, max=6, message = "Admission Id should be a 6 digit positive number")
	private String admissionId;
	private String deptName;
	private String preferHostel;
	private String firstGraduateInfo;
	private double tuitionFees;



	public String getAdmissionId() {
		return admissionId;
	}
	public void setAdmissionId(String admissionId) {
		this.admissionId = admissionId;
	}
	public String getDeptName() {
		return deptName;
	}
	public void setDeptName(String deptName) {
		this.deptName = deptName;
	}
	public String getPreferHostel() {
		return preferHostel;
	}
	public void setPreferHostel(String preferHostel) {
		this.preferHostel = preferHostel;
	}
	public String getFirstGraduateInfo() {
		return firstGraduateInfo;
	}
	public void setFirstGraduateInfo(String firstGraduateInfo) {
		this.firstGraduateInfo = firstGraduateInfo;
```

```
45          }
46          public double getTuitionFees() {
47                      return tuitionFees;
48          }
49          public void setTuitionFees(double tuitionFees) {
50                      this.tuitionFees = tuitionFees;
51          }
52
53
54
55  }
56
```

AdmissionFeeCalculator/src/main/java/com/service/AdmissionService.java

```
1   package com.service;
2
3   import org.springframework.stereotype.Service;
4
5   import com.model.Admission;
6
7   @Service
8   public class AdmissionService {
9
10          public double calculateAdmissionFees(Admission obj)
11          {
12                      double admissionFees;
13                      if(obj.getDeptName().equalsIgnoreCase("EEE"))
14                      {
15                                  obj.setTuitionFees(45000.00);
16                      }
17                      else if(obj.getDeptName().equalsIgnoreCase("ECE"))
18                      {
19                                  obj.setTuitionFees(50000.00);
20                      }
21                      else
22                      {
23                                  obj.setTuitionFees(60000.00);
24                      }
25
26                      if(obj.getDeptName().equalsIgnoreCase("EEE"))
27                      {
28                                  if(obj.getPreferHostel().equalsIgnoreCase("Yes"))
29                                  {
30                                              if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
31                                              {
32                                                          admissionFees=obj.getTuitionFees()+75000.00;
33                                              }
34                                              else
35                                              {
36
        admissionFees=obj.getTuitionFees()+75000.00+20000.00;
37                                              }
38                                  }
39                                  else
40                                  {
41                                              if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
42                                              {
43                                                          admissionFees=obj.getTuitionFees();
44                                              }
45                                              else
46                                              {
47                                                          admissionFees=obj.getTuitionFees()+20000.00;
48                                              }
49                                  }
50                      }
51                      else if(obj.getDeptName().equalsIgnoreCase("ECE"))
52                      {
```

```
53                              if(obj.getPreferHostel().equalsIgnoreCase("Yes"))
54                              {
55                                      if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
56                                      {
57                                              admissionFees=obj.getTuitionFees()+75000.00;
58                                      }
59                                      else
60                                      {
61
        admissionFees=obj.getTuitionFees()+75000.00+25000.00;
62                                      }
63                              }
64                              else
65                              {
66                                      if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
67                                      {
68                                              admissionFees=obj.getTuitionFees();
69                                      }
70                                      else
71                                      {
72                                              admissionFees=obj.getTuitionFees()+25000.00;
73                                      }
74                              }
75                      }
76
77                      else
78                      {
79                              if(obj.getPreferHostel().equalsIgnoreCase("Yes"))
80                              {
81                                      if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
82                                      {
83                                              admissionFees=obj.getTuitionFees()+75000.00;
84                                      }
85                                      else
86                                      {
87
        admissionFees=obj.getTuitionFees()+75000.00+30000.00;
88                                      }
89                              }
90                              else
91                              {
92                                      if(obj.getFirstGraduateInfo().equalsIgnoreCase("Yes"))
93                                      {
94                                              admissionFees=obj.getTuitionFees();
95                                      }
96                                      else
97                                      {
98                                              admissionFees=obj.getTuitionFees()+30000.00;
99                                      }
100                             }
101                     }
102                     return admissionFees;
103     }
104
105 }
106
107
```

AdmissionFeeCalculator/src/main/resources/application.properties

```
1 server.port=9069
2 spring.mvc.view.prefix = /WEB-INF/views/
3 spring.mvc.view.suffix = .jsp
4 spring.mvc.static-path-pattern=/resources/**
```

AdmissionFeeCalculator/src/main/webapp/WEB-INF/views/feescalculatorpage.jsp

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

```
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<body style="background-color:lavender">
<h1><center> Admission Fees Calculator </center></h1>
<form:form method="post" action="result" modelAttribute="admission">

<table style="margin: 0px auto; margin-left: auto; margin-right:auto">
                        <tr>
                                <td>Enter Admission ID</td><td><form:input path="admissionId"
id="admissionId"/></td><td><form:errors path="admissionId"/></tr>
                                <tr>
                                <tr>
                                        <td>Choose Department:</td>
                                        <td>
                                         <form:select path="deptName" id="deptName">
                                                <form:options items="${deptNameList}"/>
                                        </form:select>

                                        </td>
                                </tr>

                                <tr>
                                        <td>Hostel Preference:</td>
                                        <td>
                                         <form:select path="preferHostel" id="preferHostel">
                                                <form:options items="${preferenceList}"/>
                                        </form:select>

                                        </td>
                                </tr>

                                <tr>
                                        <td>Are you first graduate in your family:</td>
                                        <td>
                                         <form:select path="firstGraduateInfo" id="firstGraduateInfo">
                                                <form:options items="${graduateList}"/>
                                        </form:select>

                                        </td>
                                </tr>

                                        <td><input type="submit" value="CalculateFees" id="submit"
/></td>
                                        <td><input type="reset" value="Cancel"/></td>
                                </tr>

                </table>
</form:form>


</body>
</html>

```

AdmissionFeeCalculator/src/main/webapp/WEB-INF/views/result.jsp

```
<%@page isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<body bgcolor="lavender">
```

```
5 <h2>Admission Fees to be paid is Rs.${cost}</h2>
6 </body>
7 </html>
```

Grade

Reviewed on Friday, 18 February 2022, 12:27 AM by Automatic grade
Grade 98.75 / 100
Assessment report
[-]**SOURCE CODE ANALYZER REPORT**

Error Msg: A method should have only one exit point, and that should be the last statement in the method
Variable Name:
Class Name: AdmissionController

Assessment Completed Successfully
[-]**Grading and Feedback**

Feature Test - 15.00 / 15.00(Success)
** check whether Service is configured in Controller class with the required Annotation - 4.50 / 4.50*
** check whether the Controller class with the required annotation is implemented - 3.50 / 3.50*
** check whether the Service class with the required annotation is implemented - 3.50 / 3.50*
** check whether the model class with the required annotation is implemented - 3.50 / 3.50*

Functional testing - 15.00 / 15.00(Success)
** check whether the admission fees is calculated as per the requirement for CSE department - 5.00 / 5.00*
** check whether the admission fees is calculated as per the requirement for ECE department - 5.00 / 5.00*
** check whether the admission fees is calculated as per the requirement for EEE department - 5.00 / 5.00*

Spring Testing in the Controller - 35.00 / 35.00(Success)
** check whether the request is mapped for feesCalculatorPage and redirected to feescalculatorpage.jsp - 5.00 / 5.00*
** check whether the model attribute is specified above the buildState method that holds the Map containing the deptName - 5.00 / 5.00*
** check whether the model attribute is specified above the getHostelPreference method that holds the Map containing the hostel preference - 2.50 / 2.50*
** check whether the model attribute is specified above the getFirstGraduateInfo method that holds the Map containing the first graduate info - 2.50 / 2.50*
** check whether the model attribute with the name deptNameList is specified above the buildState method that holds the Map containing the department name - 5.00 / 5.00*
** check whether the model attribute with the name preferenceList is specified above the getHostelPreference method that holds the Map containing the hostel preference - 2.50 / 2.50*
** check whether the model attribute with the name graduateList is specified above the getFirstGraduateInfo method that holds the Map containing the first graduate info - 2.50 / 2.50*
** check whether layering is followed that is whether admission fees is calculated by invoking the service method inside the Controller - 5.00 / 5.00*
** check whether the validations are working correctly - 5.00 / 5.00*

UI Testing - 30.00 / 30.00(Success)
** check whether deptName in feescalculatorpage is auto populated as per the requirement - 3.00 / 3.00*
** check whether preferHostel in feescalculatorpage is auto populated as per the requirement - 3.00 / 3.00*

*   * check whether firstGraduateInfo in feescalculatorpage is auto populated as per the requirement - 3.00 / 3.00*
*   * check whether the type attribute of admissionId in feescalculatorpage is as per the requirement - 1.00 / 1.00*
*   * check whether UI from feescalculatorpage to result page is redirected after calculating the admission fees for EEE department - 5.00 / 5.00*
*   * check whether UI from feescalculatorpage to result page is redirected after calculating the admission fees for ECE department - 5.00 / 5.00*
*   * check whether UI from feescalculatorpage to result page is redirected after calculating the admission fees for CSE department - 5.00 / 5.00*
*   * check whether UI from feescalculatorpage to result page is redirected after calculating the admission fees for CSE department with first graduate offer - 5.00 / 5.00*

Comments and best practices/standards - 3.75 / 5.0(Partial)