Spring Boot & MVC CC

Delivery Agency

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.example</groupId>
	<artifactId>DeliveryAgency</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>DeliveryAgency</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>
```

Agency Controller.java

```
package com.controller;

import java.util.List;

import javax.validation.Valid;

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.ExceptionHandler;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.exception.NoOrdersFoundException;
import com.model.Order;
import com.service.AgencyService;

//use appropriate annotation to configure AgencyController as Controller
@Controller
public class AgencyController {

        //Use appropriate annotation
        @Autowired
        private AgencyService service;

        @RequestMapping(value = “/searchPage”, method = RequestMethod.GET)
        public String searchPage(@ModelAttribute(“orderBean”) Order orderBean) {
                return “searchPage”;
        }



        //invoke the service class – searchDeliveryOrders method.
        @RequestMapping(value = “/orderList”, method = RequestMethod.POST)
        public String searchDeliveryOrders(@ModelAttribute(“orderBean”) @Valid Order
orderBean, BindingResult result,
                        ModelMap model) throws NoOrdersFoundException  {
                //fill the code
                if (result.hasErrors()) {
                        return “searchPage”;
                } else {
                        List<Order> ordersFound = service.searchDeliveryOrders(orderBean);
                        if(ordersFound.isEmpty()) {
                                throw new NoOrdersFoundException(“No orders found for this
search criteria”);
                        } else {
                                model.addAttribute(“orders”, ordersFound);
                                return “deliveryListPage”;
                        }

                }
        }

        @ExceptionHandler(NoOrdersFoundException.class)
        public ModelAndView exceptionHandler(Exception e) {
                ModelAndView mav = new ModelAndView();
          mav.addObject(“exception”, e);
          mav.setViewName(“exceptionPage”);
          return mav;
        }
}
```

Delivery Agency Application.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.*"})
public class DeliveryAgencyApplication {

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

}
```

No Orders Found Exception.java

```
package com.exception;

public class NoOrdersFoundException extends Exception {
        private static final long serialVersionUID = 1L;

        public NoOrdersFoundException(String msg) {
                super(msg);
        }
}
```

Order.java

```
package com.model;

import javax.validation.constraints.NotEmpty;

import org.springframework.stereotype.Component;

//pojo class with required attributes,getters and setters
@Component
public class Order {
	private String orderId;
	private String orderFrom;
	private String deliveryTo;
	// fill the code

	@NotEmpty(message = "Delivery Date should be Mandatory")
	private String deliveryDate;

	private String deliveryStatus;
	public String getOrderId() {
		return orderId;
	}
	public void setOrderId(String orderId) {
		this.orderId = orderId;
	}
	public String getOrderFrom() {
		return orderFrom;
	}
	public void setOrderFrom(String orderFrom) {
		this.orderFrom = orderFrom;
	}
	public String getDeliveryTo() {
		return deliveryTo;
	}
	public void setDeliveryTo(String deliveryTo) {
		this.deliveryTo = deliveryTo;
	}
	public String getDeliveryDate() {
		return deliveryDate;
	}
	public void setDeliveryDate(String deliveryDate) {
		this.deliveryDate = deliveryDate;
	}

	public String getDeliveryStatus() {
		return deliveryStatus;
	}
	public void setDeliveryStatus(String deliveryStatus) {
		this.deliveryStatus = deliveryStatus;
	}
	public Order(String orderId, String orderFrom, String deliveryTo, String deliveryDate, String
deliveryStatus) {
		super();
		this.orderId = orderId;
		this.orderFrom = orderFrom;
		this.deliveryTo = deliveryTo;
		this.deliveryDate = deliveryDate;
		this.deliveryStatus = deliveryStatus;
	}
```

```
    public Order()   {}

}
```

Agency Service.java

```
package com.service;

import java.util.ArrayList;

import org.springframework.stereotype.Service;

import com.model.Order;

//use appropriate annotation to configure AgencyService as a Service
@Service
public class AgencyService {
    ArrayList<Order> list = new ArrayList<Order>();

    //search orders and return the ArrayList<Order>
    public ArrayList<Order> searchDeliveryOrders(Order orderBean) {
        ArrayList<Order> foundOrders = new ArrayList<>();
        for(Order order : list) {
            if(order.getDeliveryDate().equals(orderBean.getDeliveryDate()) && order.getDeliveryStatus().equals(orderBean.getDeliveryStatus())) {
                foundOrders.add(order);
            }
        }
        return foundOrders;
    }
```

```
            public AgencyService() {
                        addOrders();
            }


            public  void addOrders(){                         // don't change this code
                        list.add(new Order("DF1234233","AJIO trends","#412, Shipitha cascade, Ganapathy,
PIN-641001","10/02/2021","pending"));
                        list.add(new Order("FGF7788","Amazon india","#12, shardhalotus, Ram Nagar, PIN-
641001", "02/02/2021","pending"));
                        list.add(new Order("AF5546","Mynthra","#12, Anandha nilayam,Kuniamuthur, PIN-
641001","08/02/2021","pending"));
                        list.add(new Order("DF5543","Flip cart","#201, Shipitha cascade, Saravanampatti,
PIN-641001","10/02/2021","delivered"));


                        list.add(new Order("DX17788","AJIO trends","#34/45, RV Homes, 5th cross, Anssari
street, PIN-600001","01/02/2021","pending"));
                        list.add(new Order("RF55346","Amazon india","#212/4, 10th cross, T-nagar, PIN-
60008","03/02/2021","delivered"));
                        list.add(new Order("DF88994","Mynthra","#556/4, 5th cross, Thambaram, PIN-
60010","12/02/2021","pending"));
                        list.add(new Order("FT51122","AJIO trends","#556/4,Nila paradise, Thambaram,
PIN-60010","03/02/2021","delivered"));
                        list.add(new Order("LK44338","Flip cart","#412/4, 10th cross, Velacheri, PIN-
60009","03/02/2021","delivered"));


                        list.add(new Order("JK99008","Amazon india","#45, 12Th main , GM Palya, PIN-
560075","03/02/2021","pending"));
                        list.add(new Order("MJ99765","Flip cart","#67/2, 3rd cross , Indra nagar, PIN-
560077","08/02/2021","delivered"));
                        list.add(new Order("NK66754","Mynthra","#302, Sakthi enclave ,1st main ,
Malleshpalya, PIN-560075","08/02/2021","delivered"));



            }
}
```

Delivery List Page.jsp

```
<%@page isELIgnored="false"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<body bgcolor="lavender">
        <h1>
                <center>Delivery Agency</center>
        </h1>

        <table>
                <tr>
                        <td>OrderId</td>
                        <td>Order From</td>
                        <td>Delivery To</td>
                        <td>Delivery Date</td>
                        <td>Status</td>
                </tr>
                <c:forEach items="${orders}" var="order">
                        <tr>
                                <td>${order.orderId}</td>
                                <td>${order.orderFrom}</td>
                                <td>${order.deliveryTo}</td>
                                <td>${order.deliveryDate}</td>
                                <td>${order.deliveryStatus}</td>
                        </tr>
                </c:forEach>
        </table>
```

```
</body>
</html>
```

Exception Page.jsp

```
<%@page isELIgnored="false"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>
<html>
<body bgcolor="lavender">
        <h1>
                <center>Delivery Agency</center>
        </h1>
        <center>
                <h3>${exception.message}</h3>
                <a href="/searchPage">
                        <button>Go to Search</button>
                </a>
        </center>
</body>
</html>
```

Search Page.jsp

```
<%@ 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"%>
<%@ 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>Delivery Agency</center>

    </h1>

    <center>

        <sf:form action="orderList" name="form" modelAttribute="orderBean"

            method="POST">


            <fieldset>

                <label for='deliveryDate'>Search Delivery Date</label>

                <sf:input path="deliveryDate" id="deliveryDate"
name="deliveryDate"

                    placeholder="dd/mm/yyyy" />

                <sf:errors path="deliveryDate" />

                <br />

                <label for='deliveryStatus'>Select Delivery Status</label>

                <sf:select path="deliveryStatus" id="deliveryStatus"
name="deliveryStatus">

                    <sf:option label="delivered" value="delivered"/>

                    <sf:option label="pending" value="pending"/>

                </sf:select>

            </fieldset>

            <input type="submit" id="submit" value="Search" name="submit" />

        </sf:form>

    </center>

</body>

</html>
```

Digital Home Broker

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.example</groupId>

        <artifactId>DigitalHomeBroker</artifactId>

        <version>0.0.1-SNAPSHOT</version>

        <name>DigitalHomeBroker</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>
```

Broker Controller.java

```
package com.controller;

import javax.validation.Valid;

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.Home;
import com.service.BrokerService;

//use appropriate annotation to configure BrokerController as Controller
@Controller
public class BrokerController {

	//Use appropriate annotation above this property
	@Autowired
	private BrokerService service;

	@RequestMapping(value = "/searchPage", method = RequestMethod.GET)
	public String searchPage(@ModelAttribute("homeBean") Home homeBean)	{

		return "searchPage";
	}

	//invoke the service class - searchHome method.
```

```
        @RequestMapping(value = "/homeList", method = RequestMethod.POST)
        public String searchHome(@ModelAttribute("homeBean") @Valid Home homeBean,
BindingResult result,
                        ModelMap model) {
                if (result.hasErrors()) {
                        return "searchPage";
                }
                model.addAttribute("homes", service.searchHome(homeBean));
                return "homeListPage";
        }
}
```

Digital Home Broker Application.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.*"})
public class DigitalHomeBrokerApplication {


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


}
```

Home.java

```
package com.model;

import javax.validation.constraints.Min;

import org.springframework.stereotype.Component;

//pojo class with required attributes,getters and setters
@Component
public class Home {

        private String location;
        private String address;
        private String contactNumber;
        private String bhkType;
        private String contactPerson;
        @Min(value = 5000, message = "Minimum rent should be 5000")
        private double rent;
        public double getRent() {
                return rent;
        }
        public void setRent(double rent) {
                this.rent = rent;
        }
        public String getLocation() {
                return location;
        }
        public void setLocation(String location) {
                this.location = location;
        }
        public String getAddress() {
                return address;
        }
        public void setAddress(String address) {
                this.address = address;
        }
        public String getContactNumber() {
                return contactNumber;
        }
        public void setContactNumber(String contactNumber) {
                this.contactNumber = contactNumber;
        }
        public String getBhkType() {
                return bhkType;
        }
        public void setBhkType(String bhkType) {
                this.bhkType = bhkType;
        }
        public String getContactPerson() {
                return contactPerson;
        }
        public void setContactPerson(String contactPerson) {
                this.contactPerson = contactPerson;
        }


        public Home(String location, String address, String contactNumber, String bhkType, String
contactPerson,double rent) {
                super();
                this.location = location;
                this.address = address;
                this.contactNumber = contactNumber;
                this.bhkType = bhkType;
                this.contactPerson = contactPerson;

                this.rent=rent;

        }

        public Home()

        {}


}
```

Broker Service.java

```
package com.service;


import java.util.ArrayList;


import org.springframework.stereotype.Service;


import com.model.Home;


//use appropriate annotation to configure BrokerService as a Service

@Service

public class BrokerService {


        ArrayList<Home> list = new ArrayList<Home>();


        // search homes/apartments from the list and return it as a ArrayList

        public ArrayList<Home> searchHome(Home homeBean) {

                ArrayList<Home> resultList = new ArrayList<Home>();

                for (Home home : list) {

                        if (home.getLocation().equalsIgnoreCase(homeBean.getLocation())

                                        &&
home.getBhkType().equalsIgnoreCase(homeBean.getBhkType())
```

```
                    && home.getRent() == homeBean.getRent()) {
                resultList.add(home);
            }
        }
        return resultList;

    }

    public BrokerService() {
        addHomes();
    }

    // Dont modify this code
    public ArrayList<Home> addHomes() {
        list.add(new Home("Coimbatore", "#412, Shipitha cascade, Ganapathy, PIN-
641001", "9917678456", "1BHK",
                "Malavika", 8000));
        list.add(new Home("Coimbatore", "#12, shardhalotus, Ram Nagar, PIN-641001",
"9917100989", "2BHK", "Karthik",
                12000));
        list.add(new Home("Coimbatore", "#12, Anandha nilayam,Kuniamuthur, PIN-
641001", "9917100989", "2BHK", "Karthik",
                12000));
        list.add(new Home("Coimbatore", "#201, Shipitha cascade, Saravanampatti, PIN-
641001", "9917123456", "3BHK",
                "Sowparnika", 17000));

        list.add(new Home("Chennai", "#34/45, RV Homes, 5th cross, Anssari street, PIN-
600001", "7788900989", "1BHK",
                "Mukunthan", 6000));
        list.add(new Home("Chennai", "#212/4, 10th cross, T-nagar, PIN-60008",
"9917100989", "2BHK", "Annesh", 14000));
```

```
        list.add(new Home("Chennai", "#556/4, 5th cross, Thambaram, PIN-60010",
"9917100989", "2BHK", "Jisha", 14000));

        list.add(new Home("Chennai", "#556/4,Nila paradise, Thambaram, PIN-60010",
"9917100989", "2BHK", "Jisha",

            14000));

        list.add(

            new Home("Chennai", "#412/4, 10th cross, Velacheri, PIN-60009",
"9917156989", "3BHK", "Rakesh", 18000));


        list.add(

            new Home("Bangalore", "#45, 12Th main , GM Palya, PIN-560075",
"9917122119", "1BHK", "Senthil", 11000));

        list.add(new Home("Bangalore", "#67/2, 3rd cross , Indra nagar, PIN-560077",
"9944550989", "2BHK", "RamKumar",

            17000));

        list.add(new Home("Bangalore", "#302, Sakthi enclave ,1st main , Malleshpalya, PIN-
560075", "9917668889",

            "3BHK", "Saradha", 25000));

        return list;


    }

}
```

Home List Page.jsp

```
<%@page isELIgnored="false"%>

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>


<html>

<body bgcolor="lavender">

    <h1>

        <center>Digital Home Broker</center>
```

```
</h1>
<h2>
        <center>Search homes|Apartments !!!</center>
</h2>

<c:if test="${fn:length(homes) == 0}">
        <h2>
                <center>No home/Apartment found for this search</center>
        </h2>
</c:if>

<c:if test="${fn:length(homes) > 0}">
<table>
        <thead>
                <tr>
                        <td>Location</td>
                        <td>BhkType</td>
                        <td>Address</td>
                        <td>ContactPerson</td>
                        <td>ContactNumber</td>
                </tr>
        </thead>
        <tbody>
                <c:forEach items="${homes}" var="home">
                        <tr>
                                <td>${home.location}</td>
                                <td>${home.bhkType}</td>
                                <td>${home.address}</td>
                                <td>${home.contactPerson}</td>
                                <td>${home.contactNumber}</td>
                        </tr>
```

```
            </c:forEach>
        </tbody>
    </table>
    </c:if>

</body>
</html>
```

Search Page.jsp

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ 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>Digital Home Broker</center>
    </h1>
    <h2>
        <center>Search homes|Apartments !!!</center>
    </h2>
    <center>
        <table>
            <form:form action="homeList" modelAttribute="homeBean"
method="post">
                <tr>
                    <td><form:label path="location">Search
location</form:label></td>
```

```
                                        <td><form:input path="location" id="location"
name="location" />

                                        </td>

                                </tr>

                                <tr>

                                        <td><form:label path="bhkType">Search BHK
Type</form:label></td>

                                        <td><form:input path="bhkType" id="bhkType"
name="bhkType" />

                                        </td>

                                </tr>

                                <tr>

                                        <td><form:label path="rent">Preferred
Rent</form:label></td>

                                        <td><form:input path="rent" id="rent" name="rent" />

                                        <form:errors path="rent" />

                                        </td>

                                </tr>

                                <tr>

                                        <td><input type="submit" id="submit" name="submit"
value="Search" /></td>

                                </tr>

                        </form:form>

                </table>

        </center>

</body>

</html>
```

Farm Fresh Online store - V1

FarmFresh

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.example</groupId>

	<artifactId>FarmFresh</artifactId>

	<version>0.0.1-SNAPSHOT</version>

	<name>FarmFresh</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>
```

```xml
<!-- 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>
```

Shopping Controller.java

```
package com.controller;



import java.util.ArrayList;
import java.util.List;
import javax.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;

import com.exception.NoStockException;
import com.model.Product;
import com.service.ShoppingService;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

//use appropriate annotation to configure ShoppingController as Controller
@Controller
public class ShoppingController {

        @Autowired
        private ShoppingService shoppingService;

        @RequestMapping(value = "/showPage", method = RequestMethod.GET)
        public String showPage(@ModelAttribute("product")Product product) {
```

```
        return "showPage";

    }


    @ModelAttribute("productNames")

    public  List<String> populateProduct(){

        List<String> productNames = new ArrayList<String>();

            productNames.add("Apple");

            productNames.add("Orange");

            productNames.add("Grapes");

            productNames.add("Mango");

            productNames.add("JackFruit");

            return productNames;

    }


    //invoke the service class - calculateCostAndUpdate method.

    @RequestMapping(value = "/calculate", method = RequestMethod.POST)

    public String calculateCost(@ModelAttribute("product")@Valid Product product,
BindingResult result,ModelMap model) throws  NoStockException {


        if(result.hasErrors()){


            return ("showPage");

        }

        else{

            double cost=shoppingService.calculateCost(product);

            model.addAttribute("cost",cost);

            model.addAttribute("costPerKg",product.getCostPerKg());

            model.addAttribute("name",product.getProductName());

            model.addAttribute("quantity",product.getQuantity());
```

```
                }

                return ("billDesk");
        }



        @ExceptionHandler(NoStockException. class)
        public ModelAndView exceptionHandler(Exception e) {

                ModelAndView mav = new ModelAndView();
                mav.addObject("message", e.getMessage());
                mav.setViewName("exceptionPage");
                return mav;
        }

}
```

Farm Fresh Application.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.*"})
public class FarmFreshApplication {

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

}
```

No Stock Exception.java

```
package com.exception;

public class NoStockException extends Exception{
        private static final long serialVersionUID = 1L;
public NoStockException(String msg) {

        super(msg);
}

}
```

Product.java

```
package com.model;

import javax.validation.constraints.Min;
import javax.validation.constraints.NotEmpty;

import org.springframework.stereotype.Component;

@Component
public class Product {
        private String productName ;
        @Min(value=1,message="Minimum quantity should be 1Kg")
        private int quantity;
        private double costPerKg;
```

```
public String getProductName() {
                return productName;
        }

        public void setProductName(String productName) {
                this.productName = productName;
        }


        public int getQuantity() {
                return quantity;
        }


        public void setQuantity(int quantity) {
                this.quantity = quantity;
        }


        public double getCostPerKg() {
                return costPerKg;
        }


        public void setCostPerKg(double costPerKg) {
                this.costPerKg = costPerKg;
        }
```

```
public Product() {}
}
```

Shopping Service.java

```
package com.service;


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


import org.springframework.stereotype.Service;


import com.exception.NoStockException;
import com.model.Product;


//use appropriate annotation to configure ElectricityService as a Service
@Service
public class ShoppingService {


        public double calculateCost(Product product) throws  NoStockException {
        double totalCost=0.0;


        Map<String,Integer> stock=getProductStock();
        for (Entry<String, Integer> entry : stock.entrySet()) {
                if(product.getProductName().equalsIgnoreCase(entry.getKey())) {


                        if(product.getQuantity()>entry.getValue()) {
                                throw new NoStockException("No enough stock for product 
"+product.getProductName());
                        }
                }
        }

        if(product.getProductName().equals("Apple")) {
                product.setCostPerKg(250);
        }
        else if(product.getProductName().equals("JackFruit")) {
                product.setCostPerKg(75);
        }
        else if(product.getProductName().equals("Orange")) {
                product.setCostPerKg(90);
        }
        else if(product.getProductName().equals("Mango")) {
                product.setCostPerKg(60);
        }
        else if(product.getProductName().equals("Grapes")) {
                product.setCostPerKg(150);
        }

        totalCost=product.getQuantity()*product.getCostPerKg();

        return totalCost;
        }

        public  Map<String,Integer> getProductStock(){
                Map<String,Integer> stock = new HashMap<String,Integer>();
                stock.put("Apple",50);
                stock.put("Grapes",10);
                stock.put("Orange",30);
                stock.put("Mango",75);
                stock.put("JackFruit",25);
                        return stock;
            }
}
```

Bill Desk.jsp

```
            <%@page isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>
<body bgcolor="lavender"> <center>
<h2>Farm Fresh - A natural farm store!!</h2>

                                    <table  >
                                                <tr>
                                    <td>Product Name  <td>${name}</td>
                                    </tr>
                                    <tr>
                                     <td>Quantity  <td>${quantity}</td>
                                     </tr>
                                     <tr>
                                     <td>Cost Per Kg  <td>${costPerKg}</td>
                                     </tr>
                                     <tr>
                                     <td>Total cost Rs  <td>${cost}</td>
                                    </tr>
                        </table>
            </center>

            </body>
</html>
```

Exception Page.jsp

```
<%@page isELIgnored="false" %>

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<html>

<body bgcolor="lavender">


<center> <h2>Farm Fresh - A natural farm store!!</h2>

                                        <h3>

                              <%

                                        if (request.getAttribute("message") != null) {

                              %>

                              <%=request.getAttribute("message")%>

                              <%

                                        }

                              %>

                    </h3>

                    <div>

                    <center>            <a href="showPage"> Home</a></center>

                    </div>

                    </center>


</body>

</html>
```

Show Page.jsp

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

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

<%@ 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">


<!--  Design the page as per the requirements-->


<form:form method="post" action="calculate" modelAttribute="product">

<center><h2>Farm Fresh - A natural farm store!!</h2></center>

<table style="margin: 0px auto; margin-left: auto; margin-right:auto">

             <tr>

    <td>Select product</td>

                          <td>

                           <form:select path="productName" id="productName">

                                       <form:options items="${productNames}"/>

                                 </form:select>

                          </td>

             </tr>

    <tr>

             <td>Quantity</td>

    <td><form:input path="quantity" id="quantity"/></td>

    <td><form:errors path="quantity"/></td>

             </tr>




      </table>


      <center><input type="submit" value="Calculate Cost" id="submit"
name="submit"/></center>
```

```
	<br>

</form:form>



</body>
</html>
```

Flawless Academy - V1

FlawlessAcademy

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.example</groupId>
	<artifactId>FlawlessAcademy</artifactId>
	<version>0.0.1-SNAPSHOT</version>
```

```
        <name>FlawlessAcademy</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>
```

Academy Controller.java

```
package com.controller;


import java.util.HashMap;
import java.util.Map;
import javax.validation.Valid;


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.Academy;
import com.service.AcademyService;


@Controller
public class AcademyController {


    @Autowired
    private AcademyService service;


    @RequestMapping(value = "/enrollmentPage", method = RequestMethod.GET)
    public String showPage(@ModelAttribute("academyBean") Academy academyBean) {
        System.out.println("in controller");
        return "enrollmentPage";
    }
    @ModelAttribute("programList")
    public  Map<String, String> buildState(){


        Map<String, String> progMap = new HashMap<String, String>();
        progMap.put("ClassicalDance", "ClassicalDance");
        progMap.put("KarnaticVocals", "KarnaticVocals");
        progMap.put("WesternDance","WesternDance");
        progMap.put("Drawing", "Drawing");
        progMap.put("Instruments","Instruments");


        return progMap;


    }


    @RequestMapping(value = "/progEstimation", method = RequestMethod.POST)
    public String calculateProgramCost(@ModelAttribute("academyBean") @Valid Academy
academyBean, BindingResult result,
            ModelMap model) {
        if (result.hasErrors()) {
            System.out.println("Error : " + result.toString());
            return "enrollmentPage";
        }
        double cost=service.calculateProgramCost(academyBean);
        model.addAttribute("cost", cost);
```

```
                return "estimationPage";
        }



}
```

Flawless Academy Application.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.*"})
public class FlawlessAcademyApplication {

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

}
```

Academy.java

```
package com.model;

import javax.validation.constraints.Min;
import javax.validation.constraints.Max;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
```

```
@Component
public class Academy {

    private String program;

    private double costPerSession;
    private int weeksPerMonth;
    @Min(value = 2, message = "Minimum 2 Sessions/Week")
    @Max(value = 5, message = "Maximum 5 Sessions/Week")
    private int sessionsPerWeek;

    public int getSessionsPerWeek() {
        return sessionsPerWeek;
    }

    public void setSessionsPerWeek(int sessionsPerWeek) {
        this.sessionsPerWeek = sessionsPerWeek;
    }

    public Academy()
    {

    }

    public int getWeeksPerMonth() {
        return weeksPerMonth;
    }
    public void setWeeksPerMonth(int weeksPerMonth) {
        this.weeksPerMonth = weeksPerMonth;
    }
    public double getCostPerSession() {
        return costPerSession;
    }
    public void setCostPerSession(double costPerSession) {
        this.costPerSession = costPerSession;
    }

    public String getProgram() {
        return program;
    }
    public void setProgram(String program) {
        this.program = program;
    }




}
```

Academy Service.java

```
package com.service;

import org.springframework.stereotype.Service;

import com.model.Academy;

@Service
public class AcademyService {

    public double calculateProgramCost (Academy academyBean) {

        double cost=0.0;
        academyBean.setWeeksPerMonth(4);
        System.out.println(academyBean.getSessionsPerWeek()+"
"+academyBean.getProgram()+" "+academyBean.getWeeksPerMonth());

        if(academyBean.getProgram().equalsIgnoreCase("ClassicalDance") &&
academyBean.getSessionsPerWeek()>=1)

        {

            academyBean.setCostPerSession(150.0);


    cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();

        }

    else if(academyBean.getProgram().equalsIgnoreCase("KarnaticVocals") &&
academyBean.getSessionsPerWeek()>=1 )

    {

        academyBean.setCostPerSession(100.0);


    cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();

    }

    else if(academyBean.getProgram().equalsIgnoreCase("WesternDance") &&
academyBean.getSessionsPerWeek()>=1)

    {

        academyBean.setCostPerSession(125.0);


    cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();

    }

    if(academyBean.getProgram().equalsIgnoreCase("Drawing") &&
academyBean.getSessionsPerWeek()>=1)

    {

        academyBean.setCostPerSession(130.0);


    cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();

    }

else if(academyBean.getProgram().equalsIgnoreCase("Instruments") &&
academyBean.getSessionsPerWeek()>=1)
```

```
{
            academyBean.setCostPerSession(200.0);
            cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
}
            return cost;

            }

}
```

Enrollment Page.jsp

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

<!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> Flawless Academy</center></h1>
<form:form method="post" action="progEstimation" modelAttribute="academyBean">
<table style="margin: 0px auto; margin-left: auto; margin-right:auto">


                        <tr>
                                    <td>Select preferred program</td>
                                    <td>
                                     <form:select path="program" id="program">
                                                <form:options items="${programList}"/>
                                    </form:select>
```

```
                </td>
            </tr>

            <tr>
            <td>Number of sessions per week</td>
            <td><form:input path="sessionsPerWeek" id="sessionsPerWeek"/></td>
            <td><form:errors path="sessionsPerWeek" id="errormsg"/></td>
            <tr>
                <td><input type="submit" value="Program Cost" name="submit" /></td>

            </tr>

        </table>
</form:form>



</body>
</html>
```

Estimation Page.jsp

```
<%@page isELIgnored="false" %>
<html>
<body bgcolor="lavender">
<center><h2>Welcome to Flawless Academy. Your preferred program cost is Rs. ${cost}</h2></center>
</body>
</html>
```

--

SnapFitness

Snap Fitness Center - V1

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>2.4.2</version>

		<relativePath/> <!-- lookup parent from repository -->

	</parent>

	<groupId>com.example</groupId>

	<artifactId>SnapFitness</artifactId>

	<version>0.0.1-SNAPSHOT</version>

	<name>SnapFitness</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.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-validation</artifactId>
                        <version>2.5.2</version>
                </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>

<dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-test</artifactId>
             <version>5.1.7.RELEASE</version>
            <scope>test</scope>
        </dependency>



        <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>
```

Fitness Controller.java

```
package com.controller;


import java.util.HashMap;
import java.util.Map;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;


import javax.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;


import com.model.FitnessCenter;
import com.service.FitnessService;


//use appropriate annotation to configure SnapFitnessController as Controller
@Controller
public class FitnessController {


//          Use appropriate annotation
```

```
 @Autowired
        private FitnessService service;


        @ModelAttribute("packageList")
         public Map<String, String> buildState() {
                Map<String, String> pairs = new HashMap();
                pairs.put("Aerobics","Aerobics");
                pairs.put("Zumba","Zumba");
                pairs.put("PersonalTraining","PersonalTraining");
                pairs.put("Yoga","Yoga");
                pairs.put("RegularGYM","RegularGYM");
                return pairs;
        }
        @GetMapping("/snapEnquiryPage")
         public String snapEnquiryPage( @ModelAttribute("snap") FitnessCenter fitnessCenter){
                 return "snapEnquiryPage";
         }
 @PostMapping("/packEstimation")
        public String calculatePackageCost(@Valid @ModelAttribute("snap")FitnessCenter fitness,
BindingResult result,
                        ModelMap model) {
        if(result.hasErrors()){
             model.addAttribute("messege", "Number of months should be minimum one");
                        return "snapEnquiryPage";
                }
                model.addAttribute("cost",service.calculatePackageCost(fitness));
                return "snapEstimation";
        }



}
```

Snap Fitness Application.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.*"})
public class SnapFitnessApplication {

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

}
```

Fitness Center.java

```

package com.model;
import javax.validation.constraints.Min;
import org.springframework.stereotype.Component;

//pojo class with required attributes,getters and setters

//use appropriate annotation to make this class as component class
@Component
public class FitnessCenter {

    private String preferredPackage;
    private double costPerMonth;
    // use appropriate annotation for validating numberOfMonths attribute
    @Min(value= 1,message="Number of months should be minimum one")
    private int numberOfMonths;


    public String getPreferredPackage() {
        return preferredPackage;
    }
    public void setPreferredPackage(String preferredPackage) {
        this.preferredPackage = preferredPackage;
    }
    public double getCostPerMonth() {
        return costPerMonth;
    }
    public void setCostPerMonth(double costPerMonth) {
        this.costPerMonth = costPerMonth;
    }
    public int getNumberOfMonths() {
        return numberOfMonths;
    }
    public void setNumberOfMonths(int numberOfMonths) {
        this.numberOfMonths = numberOfMonths;
    }


}
```

Fitness Service.java

```
package com.service;
```

```
import com.model.FitnessCenter;

import org.springframework.stereotype.Service;

//use appropriate annotation to configure SnapFitnessService as a Service class
@Service
public class FitnessService {

    //calculate the Package cost and return the rent amount
    public double calculatePackageCost (FitnessCenter fitness) {

        double packageCost=0.0;
        // fill the code
        String serviceName = fitness.getPreferredPackage();
        int months = fitness.getNumberOfMonths();
        if(serviceName.equals("Aerobics"))
        {
            packageCost = months*1500.0;
        }
        else if(serviceName.equals("Zumba"))
        {
            packageCost = months*1750.0;
        }
        else if(serviceName.equals("RegularGYM"))
        {
            packageCost = months*1250.0;
        }
        else if(serviceName.equals("PersonalTraining"))
        {
            packageCost = months*2500.0;
        }
        else if(serviceName.equals("Yoga"))
        {
            packageCost = months*1300.0;
        }

        return packageCost;


    }

}
```

Snap Enquiry Page.jsp

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ 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> Snap Fitness Center </center></h1>
   <form:form modelAttribute="snap" method="post" action="/packEstimation">
    <center>
        <table>
            <tr>
                <td><form:label path="preferredPackage">Select
Package</form:label></td>
                <td><form:select path="preferredPackage" id="preferredPackage"
```

```
                        name="preferredPackage" items="${packageList}"
required="required" /></td>
            </tr>
            <tr>
                <td><form:label path="numberOfMonths">Package duration (In
months)</form:label></td>
                <td><form:input path="numberOfMonths" id="numberOfMonths"
                        name="numberOfMonths" required="required"
/><label>${messege}</label></td>
            </tr>
            <tr>
                <td><input type="submit" id="submit(Name)" name="submit"
                    value="Package Cost" /></td>
                <td></td>
            </tr>

        </table>
        </center>
    </form:form>



</body>
</html>
```

Snap Estimation.jsp

```
<%@page isELIgnored="false" %>
<html>
<body bgcolor="lavender">
<h3>Welcome to Snap Fitness Center.Your preferred package cost is Rs.${cost}</h3>
</body>
</html>
```

```
</html>
```

CarstoreApplication

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>2.2.6.RELEASE</version>

                <relativePath/> <!-- lookup parent from repository -->

        </parent>

        <groupId>com.cts</groupId>

        <artifactId>carstore</artifactId>

        <version>0.0.1-SNAPSHOT</version>

        <name>CarStoreSystem</name>

        <description>Car Store project using Spring Boot</description>


        <properties>

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

                <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>

        </properties>
```

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

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <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>
    <!--project specific dependencies  -->
    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>7.3.0</version>
        <scope>test</scope>
    </dependency>
```

```
               <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
               <dependency>
                      <groupId>org.seleniumhq.selenium</groupId>
                      <artifactId>selenium-java</artifactId>
                      <version>3.141.59</version>
               </dependency>
               <!-- JSTL tag lib -->
  <dependency>
   <groupId>javax.servlet.jsp.jstl</groupId>
   <artifactId>javax.servlet.jsp.jstl-api</artifactId>
   <version>1.2.1</version>
  </dependency>
  <dependency>
                      <groupId>taglibs</groupId>
                      <artifactId>standard</artifactId>
                      <version>1.1.2</version>
               </dependency>
               <!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>

               <!-- <dependency>
<groupId>jstl</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
 </dependency> -->
   <dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
 <!-- Tomcat for JSP rendering -->
  <dependency>
   <groupId>org.apache.tomcat.embed</groupId>
   <artifactId>tomcat-embed-jasper</artifactId>
   <scope>provided</scope>
  </dependency>
  <!-- <dependency>
                        <groupId>org.junit.jupiter</groupId>
                        <artifactId>junit-jupiter-api</artifactId>
                        <version>5.3.2</version>
                        <scope>test</scope>
                </dependency>  -->
  <!-- https://mvnrepository.com/artifact/org.junit.platform/junit-platform-runner -->

                <dependency>
                        <groupId>org.junit.platform</groupId>
                        <artifactId>junit-platform-runner</artifactId>
                        <scope>test</scope>
                </dependency>
        </dependencies>

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

```
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-surefire-plugin</artifactId>
                    <configuration>
                        <suiteXmlFiles>
                            <suiteXmlFile>testSuite.xml</suiteXmlFile>
                        </suiteXmlFiles>
                    </configuration>
                </plugin>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-site-plugin</artifactId>
                </plugin>
            </plugins>
        </build>
        <reporting>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-surefire-report-plugin</artifactId>
                    <version>2.22.0</version>
                </plugin>
            </plugins>
        </reporting>

</project>
```

CarStoreController.java

```
package com.cts.carstore.controller;


import java.util.ArrayList;
import java.util.List;
import javax.validation.Valid;

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.validation.Validator;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

import com.cts.carstore.exception.ApplicationException;
import com.cts.carstore.model.Car;
import com.cts.carstore.model.CarSearch;
import com.cts.carstore.service.CarStoreService;


@Controller
public class CarStoreController {

	@Autowired
	private CarStoreService service;

	@Autowired
	private Validator validator;

	public CarStoreController() {
		System.out.println("in default constr of controller");

	}
 @Autowired
 public CarStoreController(CarStoreService service, Validator validator) {
   this.service=service;
   this.validator=validator;
 }

@GetMapping("/showCarSearchForm")
public String showCarSearchForm(
                @ModelAttribute("carSearch") CarSearch carSearch) {

        //Add code here..
        carSearch=new CarSearch();
        return "carSearch"; //TODO, modify return value
}

@PostMapping("/getCarSearchResultPage")
public String getCarSearchResultForm(@Valid @ModelAttribute("carSearch") CarSearch carSearch
                ,BindingResult result,ModelMap map
)throws ApplicationException  {

        //Add code here..
        if(result.hasErrors())
        return "carSearch"; //TODO, modify return value

        List<Car> carList=service.getCarSearchResult(carSearch);
        map.addAttribute("carList", carList);

        return "carSearchResult";
}
```

```
@ModelAttribute("cities")
public List<String> populateCities() {
            List<String> cities = new ArrayList<String>();


             cities.add("Chennai");
              cities.add("Mumbai");
              cities.add("Delhi");
              cities.add("Bangalore");
              cities.add("Pune");
              cities.add("Kolkatta");



            return cities;
}
@ModelAttribute("brands")
public List<String> populateBrands() {
            List<String> brands = new ArrayList<String>();

              brands.add("Maruti Suzuki");
              brands.add("Honda");
              brands.add("Mahindra");
              brands.add("Toyota");
              brands.add("Hundai");

            return brands;
}
@ModelAttribute("budgetUpto")
public List<String> populateBudget() {
            List<String> budgetUpto = new ArrayList<String>();

             budgetUpto.add("Below 3 Lakh");
```

```
         budgetUpto.add("3 Lakh");
         budgetUpto.add("5 Lakh");
         budgetUpto.add("7 Lakh");
         budgetUpto.add("10 Lakh");



        return budgetUpto;
    }

    @ModelAttribute("fuelTypes")
    public List<String> populateFuelTypes() {
        List<String> fuelTypes = new ArrayList<String>();

        fuelTypes.add("Petrol");
         fuelTypes.add("Diesel");
         fuelTypes.add("CNG");

        return fuelTypes;
    }
}
```

ApplicationException.java

```
package com.cts.carstore.exception;



public class ApplicationException extends Exception {

    private static final long serialVersionUID = -9079454849611061074L;

    public String errorMessage;
}
```

ExceptionHandlerControllerAdvice.java

```
package com.cts.carstore.exception;



import java.time.LocalDateTime;

import javax.servlet.http.HttpServletRequest;

import org.springframework.http.HttpStatus;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.servlet.ModelAndView;




@ControllerAdvice
public class ExceptionHandlerControllerAdvice {

	@ExceptionHandler(ApplicationException.class)
	@ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
	public ModelAndView handleResourceNotFound(final ApplicationException exception,
			final HttpServletRequest request, final Model model) {

		//Add code here..
```

```
        ModelAndView mv=new ModelAndView();
        mv.setViewName("error");
        mv.addObject("errorMessage","Low Budget-No car available below 3 lakh price");
        //add attribute error and timestamp
        mv.addObject("errorTime",LocalDateTime.now());

        return mv; //TODO, change return value
    }
}
```

Car.java

```
package com.cts.carstore.model;

public class Car {
    private String brandName;
    private String modelName;
    private double price;   //price in lakh
    private String fuelType;
    private double mileage;
    private int seatingCapacity;

    public Car() {
        // TODO Auto-generated constructor stub
    }

    public Car(String brandName, String modelName, double price, String fuelType,double mileage, int seatingCapacity) {
        super();
        this.brandName = brandName;
        this.modelName = modelName;
        this.price = price;
        this.fuelType = fuelType;
        this.mileage = mileage;
        this.seatingCapacity = seatingCapacity;
}

public String getBrandName() {
        return brandName;
}

public void setBrandName(String brandName) {
        this.brandName = brandName;
}

public String getModelName() {
        return modelName;
}

public void setModelName(String modelName) {

        this.modelName = modelName;
}

public double getPrice() {
        return price;
}

public void setPrice(double price) {
        this.price = price;
}
```

```
	public String getFuelType() {
		return fuelType;
	}

	public void setFuelType(String fuelType) {
		this.fuelType = fuelType;
	}

	public int getSeatingCapacity() {
		return seatingCapacity;
	}

	public void setSeatingCapacity(int seatingCapacity) {
		this.seatingCapacity = seatingCapacity;
	}

	public double getMileage() {
		return mileage;
	}

	public void setMileage(double mileage) {
		this.mileage = mileage;
	}



}
```

Car Search.java

```
package com.cts.carstore.model;

import javax.validation.constraints.*;

public class CarSearch {
	// Use validation annotations as per the requirement
	@NotEmpty(message="Customer name is required")
	private String customerName;

	@NotEmpty(message="Gender is required")
	private String gender;

	@NotEmpty(message="Mobile number is required")
	@Pattern(regexp="[7-9]{1}[0-9]{9}",message="Mobile number should be 10 digits and starting with digit 7/8/9")
	private String mobileNumber;

	private String customerCity;
	private String brand;
	private String budgetUpto;
	private String fuelType;


	public CarSearch() {
		// TODO Auto-generated constructor stub
	}

	public String getCustomerName() {
		return customerName;
	}
```

```
public void setCustomerName(String customerName) {
            this.customerName = customerName;
}

public String getGender() {
            return gender;
}

public void setGender(String gender) {
            this.gender = gender;
}

public String getMobileNumber() {

            return mobileNumber;
}

public void setMobileNumber(String mobileNumber) {
            this.mobileNumber = mobileNumber;
}

public String getCustomerCity() {
            return customerCity;
}

public void setCustomerCity(String customerCity) {
            this.customerCity = customerCity;
}

public String getBrand() {
```

```
                return brand;
        }

        public void setBrand(String brand) {
                this.brand = brand;
        }

        public String getBudgetUpto() {
                return budgetUpto;
        }

        public void setBudgetUpto(String budgetUpto) {
                this.budgetUpto = budgetUpto;
        }

        public String getFuelType() {
                return fuelType;
        }

        public void setFuelType(String fuelType) {

                this.fuelType = fuelType;
        }

}
```

Error Response.java

```
package com.cts.carstore.model;
```

```
public class ErrorResponse {

        private String errorMessage;
        private String requestedURI;

        //add code here
}
```

Car Store Service.java

```
package com.cts.carstore.service;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Service;

import com.cts.carstore.exception.ApplicationException;
import com.cts.carstore.model.Car;
import com.cts.carstore.model.CarSearch;

@Service
public class CarStoreService {

        public List<Car> getCarSearchResult(CarSearch carSearch)
                                                                        throws
ApplicationException {
                //Add code here..
                List<Car> list=new ArrayList<Car>();
                List<Car> available=buildCars();
                if(carSearch.getBudgetUpto().startsWith("Below"))
        throw new ApplicationException();

    int budgetLimit=Integer.parseInt(carSearch.getBudgetUpto().substring(0,2).trim());
    for(Car car:available) {
        if(car.getBrandName().equalsIgnoreCase(carSearch.getBrand())&&

car.getFuelType().equalsIgnoreCase(carSearch.getFuelType())
                && car.getPrice()<=budgetLimit) {
            list.add(car);
        }
    }

    return list;  //TODO, modify this return value
}



// DO NOT CHANGE THIS METHOD
//DO NOT CHANGE CODE WITHIN METHOD
private List<Car> buildCars() {
    List<Car> cars = new ArrayList<Car>();
    // brand,modelName,price,fuelType,seatingCapacity
    Car car1 = new Car("Maruti Suzuki", "Swift", 5.20, "Petrol", 21.21, 5);
    Car car2 = new Car("Maruti Suzuki", "Alto", 3, "Petrol", 22.05, 5);
    Car car3 = new Car("Maruti Suzuki", "Ertiga LXi", 7.5, "Petrol", 19.01, 7);
    Car car4 = new Car("Maruti Suzuki", "Ertiga VXi", 8.9, "CNG", 26.2, 7);
    Car car5 = new Car("Maruti Suzuki", "WagonR", 4.5, "Petrol", 20.52, 5);
    Car car6 = new Car("Maruti Suzuki", "WagonR LXi", 5.20, "CNG", 32.52, 5);
    Car car7 = new Car("Honda", "Amaze", 6.20, "Petrol", 18.6, 5);
    Car car8 = new Car("Honda", "Amaze1.2E MT", 7.6, "Diesel", 24.7, 5);
    Car car9 = new Car("Honda", "Amaze1.5E MT", 8, "Diesel", 25.2, 5);
    Car car10 = new Car("Honda", "City", 9.30, "Petrol", 17.4, 5);
```

```
        Car car11 = new Car("Mahindra", "KUV100", 5.6, "Petrol", 15, 6);

        Car car12 = new Car("Toyota", "Yaris", 8.8, "Petrol", 18.1, 5);

        Car car13 = new Car("Hundai", "Aura", 7.30, "CNG", 20.5, 5);

        Car car14 = new Car("Hundai", "Creta", 9.9, "Petrol", 21.5, 5);

        cars.add(car1);

        cars.add(car2);

        cars.add(car3);

        cars.add(car4);

        cars.add(car5);

        cars.add(car6);

        cars.add(car7);

        cars.add(car8);

        cars.add(car9);

        cars.add(car10);

        cars.add(car11);

        cars.add(car12);

        cars.add(car13);

        cars.add(car14);


        return cars;

    }


}
```

Skeleton Validator.java

```
package com.cts.carstore.skeletonvalidator;


import java.lang.reflect.Method;

import java.util.logging.Level;

import java.util.logging.Logger;
```

```
public class SkeletonValidator {

        public SkeletonValidator() {
                validateClassName("com.cts.carstore.model.Car");
                validateClassName("com.cts.carstore.model.CarSearch");
                validateClassName("com.cts.carstore.model.ErrorResponse");
                validateClassName("com.cts.carstore.service.CarStoreService");
                validateClassName("com.cts.carstore.controller.CarStoreController");
                validateClassName("com.cts.carstore.exception.ApplicationException");

        validateClassName("com.cts.carstore.exception.ExceptionHandlerControllerAdvice");



                validateMethodSignature("getCarSearchResult:java.util.List",
                                "com.cts.carstore.service.CarStoreService");

        validateMethodSignature("showCarSearchForm:java.lang.String,populateCities:java.util.List,
populateBrands:java.util.List,populateBudget:java.util.List,populateFuelTypes:java.util.List",
                                "com.cts.carstore.controller.CarStoreController");
        }

        private static final Logger LOG = Logger.getLogger("SkeletonValidator");

        protected final boolean validateClassName(String className) {

                boolean iscorrect = false;
                try {
                        Class.forName(className);
                        iscorrect = true;
```

```
            LOG.info("Class Name " + className + " is correct");

        } catch (ClassNotFoundException e) {

            LOG.log(Level.SEVERE, "You have changed either the " + "class
name/package. Use the correct package "
                    + "and class name as provided in the skeleton");

        } catch (Exception e) {
            LOG.log(Level.SEVERE,
                    "There is an error in validating the " + "Class Name. Please
manually verify that the "
                            + "Class name is same as skeleton before
uploading");
        }
        return iscorrect;

    }



    protected final void validateMethodSignature(String methodWithExcptn, String className) {
        Class cls = null;
        try {

            String[] actualmethods = methodWithExcptn.split(",");
            boolean errorFlag = false;
            String[] methodSignature;
            String methodName = null;
            String returnType = null;

            for (String singleMethod : actualmethods) {
                boolean foundMethod = false;
                methodSignature = singleMethod.split(":");

                methodName = methodSignature[0];
                returnType = methodSignature[1];

                cls = Class.forName(className);
                Method[] methods = cls.getMethods();
                for (Method findMethod : methods) {
                    if (methodName.equals(findMethod.getName())) {
                        foundMethod = true;

                        if
(!(findMethod.getReturnType().getName().equals(returnType))) {

                            errorFlag = true;
                            LOG.log(Level.SEVERE, " You have changed
the " + "return type in '" + methodName
                                    + "' method. Please stick to
the " + "skeleton provided");

                        } else {
                            LOG.info("Method signature of " +
methodName + " is valid");
                        }

                    }
                }
                if (!foundMethod) {
                    errorFlag = true;
                    LOG.log(Level.SEVERE, " Unable to find the given public
method " + methodName
                            + ". Do not change the " + "given public
method name. " + "Verify it with the skeleton");
```

```
                }

            }
            if (!errorFlag) {
                LOG.info("Method signature is valid");
            }

        } catch (Exception e) {
            LOG.log(Level.SEVERE,
                    " There is an error in validating the " + "method structure. Please manually verify that the "
                            + "Method signature is same as the skeleton before uploading");
        }
    }

}
```

Car Store Application.java

```
package com.cts.carstore;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cts.carstore.skeletonvalidator.SkeletonValidator;

@SpringBootApplication
public class CarStoreApplication {

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

```
                new SkeletonValidator();
        }

}
```

Car Search.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <br>
        <br>
        <!-- Add code here.. -->
        <h1 id="heading">Windsor Car Showroom</h1>

        <h3 style="align:center">Search Cars</h3>
        <sf:form name="form" action="getCarSearchResultPage"
                                        modelAttribute="carSearch" method="post">
                <sf:label path="customerName">Customer Name:</sf:label>
                <sf:input path="customerName" id="customerName" />
                <sf:errors path="customerName" cssStyle="color:red;"/>
                <br />
                <sf:label path="mobileNumber">Mobile Number:</sf:label>
                <sf:input path="mobileNumber" id="mobileNumber" />
                <sf:errors path="mobileNumber" cssStyle="color:red;" />
                <br />
                <sf:label path="gender">Gender:</sf:label>
                <sf:radiobutton path="gender" value="Male" />Male
                <sf:radiobutton path="gender" value="Female" />Female
                <sf:errors path="gender" cssStyle="color:red;"/>
                <br />
                <sf:label path="customerCity">Customer City:</sf:label>
                <sf:select path="customerCity" id="customerCity">
                        <sf:options items="${cities}" />
                </sf:select>
                <br />
                <sf:label path="brand">Brand:</sf:label>
                <sf:select path="brand" id="brand">
                        <sf:options items="${brands}" />
                </sf:select>
                <br />
                <sf:label path="fuelType">Fuel Type:</sf:label>
                <sf:select path="fuelType" id="fuelType">
                        <sf:options items="${fuelTypes}" />
                </sf:select>
                <br />
                <sf:label path="budgetUpto">Budget Upto:</sf:label>
                <sf:select path="budgetUpto" id="budgetUpto">
                        <sf:options items="${budgetUpto}" />
                </sf:select>
                <br />
                <input type="submit" value="CarSearch" />
                <input type="reset" value="Clear" />
        </sf:form>
```

```
</body>
</html>
```

Car Search Result.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <br>
    <br>
    <!--Add code here  -->
    <sf:form id="searchResult">
    <c:if test="${carList.isEmpty()==false}">
        <h3>Here are cars matching your search criteria: </h3>
        <table border="1">
            <tr>
                <th>Brand Name</th>
                <th>Model Name</th>
```

```
                        <th>Price in lakh</th>
                        <th>Fuel/Transmission</th>
                        <th>Mileage</th>
                        <th>Seating Capacity</th>
                    </tr>
                    <c:forEach items="${carList}" var="car">
                        <tr>
                            <td>${car.brandName}</td>
                            <td>${car.modelName}</td>
                            <td>${car.price}</td>
                            <td>${car.fuelType}</td>
                            <td>${car.mileage}</td>
                            <td>${car.seatingCapacity}</td>
                        </tr>
                    </c:forEach>
                </table>
        </c:if>
        <c:if test="${carList.isEmpty()==true}">
                <h3 id="noResult">Sorry,No car available matching your profile.</h3>
        </c:if>
        </sf:form>
        <br/>
        <a href="showCarSearchForm" id="searchCars">Search Cars</a>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" isELIgnored="false" isErrorPage="true"%>
```

```
<%@ 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"%>

<!DOCTYPE HTML>
<html>

<head>
<title>Car Search Errors</title>
</head>

<body>
        <!--Add code here..  -->
        <sf:form id="error">
        <h3>Unable to retrieve car information.Below are the error details:</h3>
        <h3>${errorMessage}</h3>
        <h3>${errorTime}</h3>
        </sf:form>
        <br/>
        <a href="showCarSearchForm" id="searchCars">Search Cars</a>
</body>
</html>
```
