Automatic evaluation[-]

Proposed grade: 100.0 / 100
Result Description
[-]**Grading and Feedback**

```
Feature Test  - 30.0 / 30.0(Success)
         * check whether the controller annotation is present above the Controller - 6.0 /
6.0
         * check whether the ModelAttribute annotation is present above the Method - 6.0 /
6.0
         * check whether the Service is autowired inside the Controller - 6.0 / 6.0
         * check whether Service annotation is present above the Service - 6.0 / 6.0
         * check whether the ModelAttribute annotation has the appropriate value above the
Method - 6.0 / 6.0

Functional testing - 20.0 / 20.0(Success)
         * check whether the Total cost is calculated correctly for the product MemoryCard
- 4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product USBCable -
4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product PenDrive -
4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product HeadPhone
- 4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product
TravelAdapter - 4.0 / 4.0

Spring Testing in the Controller - 20.0 / 20.0(Success)
         * check whether the request is mapped for showPage and redirected to showpage -
5.0 / 5.0
         * check whether layering is followed that is whether total cost is calculated by
invoking the service method inside the Controller - 10.0 / 10.0
         * check whether the request is mapped for calculateTotalCost and redirected to
result page - 5.0 / 5.0

UI Testing - 25.0 / 25.0(Success)
         * check whether the showpage jsp is rendered for the product MemoryCard - 1.0 /
1.0
         * check whether the showpage jsp is rendered for the product USBCable - 1.0 / 1.0
         * check whether the showpage jsp is rendered for the product PenDrive - 1.0 / 1.0
         * check whether the showpage jsp is rendered for the product HeadPhone - 1.0 /
1.0
         * check whether the showpage jsp is rendered for the product TravelAdapter - 1.0
/ 1.0
         * check whether the showpage jsp is rendered with proper constraints for
customerName - 1.0 / 1.0
         * check whether the showpage jsp is rendered with proper constraints for
contactNumber - 1.0 / 1.0
         * check whether the showpage jsp is rendered with proper constraints for quantity
- 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product memoryCard - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product USBCable - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product PenDrive - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product HeadPhone - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product TravelAdapter - 1.0 / 1.0
         * check whether label names from the property file is correctly displayed for
English - 2.0 / 2.0
```

```
	* check whether label names from the property file is correctly displayed for
German - 2.0 / 2.0
	* check whether error message for blank from the property file is correctly
displayed for French - 2.0 / 2.0
	* check whether error message for Customer Name in blank from the property file
is correctly displayed for German - 2.0 / 2.0
	* check whether error message for Contact Number in blank from the property file
is correctly displayed for German - 2.0 / 2.0
	* check whether error message for Quality is negative from the property file is
correctly displayed for French - 2.0 / 2.0

Comments and best practices/standards - 5.0 / 5.0(Success)
```

Test Case Passed

MobileAccessoriesShop/src/main/webapp/WEB-INF/views/billdesk.jsp

```
<%@page isELIgnored="false" %>
<html>
<body bgcolor="lavender">

<h2>Hi ${customerName}!!! You need to pay Rs.${totalCost}</h2>

</body>
</html>
```

MobileAccessoriesShop/src/main/webapp/WEB-INF/views/showpage.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> Welcome to Mobile Accessories Store </center></h1>
<!--  Do not change the below line -->
<a href="/showPage?language=en">English</a>|<a href="/showPage?language=de">German</a>|<a href="/showPage?language=fr">French</a></align>
<!--  Design the page as per the requirements-->
<center>
<form:form modelAttribute="order" method="POST" action="billDesk">
<table>
	<tr>
		<td><spring:message code="label.customerName"></spring:message>
		<td><form:input path="customerName" id="customerName" /></td>
		<td><form:errors path="customerName" ><spring:message code="error.customerName.blank" /></form:errors></td>
	</tr>
	<tr>
		<td><spring:message code="label.contactNumber"></spring:message>
		<td><form:input path="contactNumber" id="contactNumber" /></td>
		<td><form:errors path="contactNumber" ><spring:message code="error.contactNumber.blank" /></form:errors></td>
	</tr>
	<tr>
		<td><spring:message code="label.productName"></spring:message>
		<td><form:select items="${productList}" path="productName" id="productName" /></td>
	</tr>
	<tr>
		<td><spring:message code="label.quantity"></spring:message>
		<td><form:input path="quantity" id="quantity" /></td>
```

```
        <td><form:errors path="quantity"  ><spring:message code="error.quantity.negative" /></form:errors></td>
    </tr>
    <tr>
        <td><input type="submit" value="PlaceOrder" id="submit"></td>
        <td><input type="reset" value="Cancel" id="cancel"></td>
    </tr>
</table>
</form:form>
</center>
</body>
</html>

```

MobileAccessoriesShop/pom.xml

```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</groupId>
    <artifactId>MobileAccessoriesShop</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>MobileAccessoriesShop</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</artifactId>
        </dependency>


    <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>
        <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>

```

MobileAccessoriesShop/src/main/java/com/controller/BillController.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.Order;
import com.service.BillService;

/**
 * The Class BillController is the main controller for the application.
 */
@Controller
public class BillController {

    /** The instance of bill service class. */
    @Autowired
    private BillService billService;

    /**
     * Populate product type.
     *
     * @return the map containing the product types
     */
    @ModelAttribute("productList")
    public Map<String, String> populateProductType() {
        Map<String, String> productMap = new HashMap<>();
        productMap.put("HeadPhone", "HeadPhone");
        productMap.put("TravelAdapter", "TravelAdapter");
        productMap.put("MemoryCard", "MemoryCard");
        productMap.put("PenDrive", "PenDrive");
        productMap.put("USBCable", "USBCable");
        return productMap;
    }

    /**
     * Show the default order page.
     *
     * @param order
     *            the order contains the details of the order
     * @param result
     *            the result shows if any form validation errors are present
     * @return the string is the view to redirect to
     */
    @RequestMapping(method = RequestMethod.GET, value = "/showPage")
    public String showPage(@ModelAttribute("order") Order order, BindingResult result) {
        return "showpage";
    }

    /**
     * Calculate total cost for the order.
     *
     * @param order
     *            the order contains the details for the order
     * @param result
     *            the result shows if there are any form validation errors
     * @param model
     *            the model is used to add the required attributes while redirecting
     *            to the next view
     * @return the string is the view to be redirected to
     */
    @RequestMapping(method = RequestMethod.POST, value = "/billDesk")
    public String calculateTotalCost(@Valid @ModelAttribute("order") Order order, BindingResult result,
            ModelMap model) {

        // fill the code
        String redirect = "";
        if (result.hasErrors()) {
            redirect = "showpage";
        } else {
            double cost = billService.calculateTotalCost(order);
            model.addAttribute("totalCost", cost);
            model.addAttribute("customerName", order.getCustomerName());
            redirect = "billdesk";
        }
        return redirect;
    }

}

```

MobileAccessoriesShop/src/main/java/com/controller/InternationalizationConfig.java

```
package com.controller;

import java.util.Locale;

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

```java
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.LocaleResolver;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
import org.springframework.web.servlet.i18n.SessionLocaleResolver;

/**
 * The Class InternationalizationConfig is used to configure the
 * internationalization aspect of the application.
 */
@Configuration
public class InternationalizationConfig extends WebMvcConfigurerAdapter {

    /**
     * Locale resolver is used to set the default locale.
     *
     * @return the locale resolver is the default locale
     */
    @Bean
    public LocaleResolver localeResolver() {
        SessionLocaleResolver sessionLocaleResolver = new SessionLocaleResolver();
        sessionLocaleResolver.setDefaultLocale(Locale.ENGLISH);
        return sessionLocaleResolver;
    }

    /**
     * Locale change interceptor is used to change the locale while parsing the
     * request.
     *
     * @return the locale change interceptor intercepts the request and parses the
     *         paramater "language"
     */
    @Bean
    public LocaleChangeInterceptor localeChangeInterceptor() {
        LocaleChangeInterceptor localeChangeInterceptor = new LocaleChangeInterceptor();
        localeChangeInterceptor.setParamName("language");
        return localeChangeInterceptor;

    }

    /**
     * @see org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter#addInterceptors(org.springframework.web.servlet.config.annotation.InterceptorRegistry)
     */
    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(localeChangeInterceptor());

    }
}
```

MobileAccessoriesShop/src/main/java/com/example/demo/MobileAccessoriesShopApplication.java

```java
package com.example.demo;

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

/**
 * The Class MobileAccessoriesShopApplication is the default Spring Boot
 * Application class.
 */
@SpringBootApplication
```

```java
@ComponentScan({ "com.controller", "com.model", "com.service" })
public class MobileAccessoriesShopApplication {

    /**
     * The main method.
     *
     * @param args
     *            the arguments
     */
    public static void main(String[] args) {
        SpringApplication.run(MobileAccessoriesShopApplication.class, args);
    }

}
```

MobileAccessoriesShop/src/main/java/com/model/Order.java

```java
package com.model;

import javax.validation.constraints.NotBlank;
import javax.validation.constraints.Positive;

/**
 * The Class Order is the model class for this application containing all the
 * necessary details for placing an order.
 */
public class Order {

    /** The customer name. */
    @NotBlank
    private String customerName;

    /** The contact number. */
    @NotBlank
    private String contactNumber;

    /** The product name. */
    private String productName;

    /** The quantity. */
    @Positive
    private int quantity;

    /**
     * Gets the customer name.
     *
     * @return the customer name
     */
    public String getCustomerName() {
        return customerName;
    }

    /**
     * Sets the customer name.
     *
     * @param customerName
     *            the new customer name
     */
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    /**
     * Gets the contact number.
     *
     * @return the contact number
     */
	public String getContactNumber() {
		return contactNumber;
	}

	/**
	 * Sets the contact number.
	 *
	 * @param contactNumber
	 *            the new contact number
	 */
	public void setContactNumber(String contactNumber) {
		this.contactNumber = contactNumber;
	}

	/**
	 * Gets the product name.
	 *
	 * @return the product name
	 */
	public String getProductName() {
		return productName;
	}

	/**
	 * Sets the product name.
	 *
	 * @param productName
	 *            the new product name
	 */
	public void setProductName(String productName) {
		this.productName = productName;
	}

	/**
	 * Gets the quantity.
	 *
	 * @return the quantity
	 */
	public int getQuantity() {
		return quantity;
	}

	/**
	 * Sets the quantity.
	 *
	 * @param quantity
	 *            the new quantity
	 */
	public void setQuantity(int quantity) {
		this.quantity = quantity;
	}

}

```

MobileAccessoriesShop/src/main/java/com/service/BillService.java

```
package com.service;

import org.springframework.stereotype.Service;

import com.model.Order;

/**
 * The Class BillService is the service class for this application and is used
 * to perform the business logic.
 */
```

```
@Service
public class BillService {

	/**
	 * Calculate total cost from the given order.
	 * 
	 * @param order
	 *            the order is the order whose total cost is to be calculated
	 * @return the double is the cost of the order the user has entered
	 */
	public double calculateTotalCost(Order order) {
		double cost = 0.0;
		// fill the code
		if (order.getProductName().equalsIgnoreCase("HeadPhone")) {
			cost = 450 * order.getQuantity();
		} else if (order.getProductName().equalsIgnoreCase("TravelAdapter")) {
			cost = 1000 * order.getQuantity();
		} else if (order.getProductName().equalsIgnoreCase("MemoryCard")) {
			cost = 300 * order.getQuantity();
		} else if (order.getProductName().equalsIgnoreCase("PenDrive")) {
			cost = 650 * order.getQuantity();
		} else if (order.getProductName().equalsIgnoreCase("USBCable")) {
			cost = 800 * order.getQuantity();
		}
		cost += cost * 0.03;
		return cost;
	}

}

```

MobileAccessoriesShop/src/main/resources/application.properties

```
#Do not change the configurations
#value for server.port can be changed
#include other needed configurations
server.port=9090
spring.mvc.view.prefix = /WEB-INF/views/
spring.mvc.view.suffix = .jsp
spring.mvc.static-path-pattern=/resources/**
```

MobileAccessoriesShop/src/main/resources/messages.properties

```
label.customerName=Customer Name in English
label.contactNumber=Contact Number in English
label.productName=Product Name in English
label.quantity=Quantity in English

error.customerName.blank=Customer Name cannot be empty in English
error.contactNumber.blank=Contact Number cannot be empty in English
error.quantity.negative=Quantity should not be zero or a negative number in English
```

MobileAccessoriesShop/src/main/resources/messages_de.properties

```
label.customerName=Customer Name in German
label.contactNumber=Contact Number in German
label.productName=Product Name in German
label.quantity=Quantity in German

error.customerName.blank=Customer Name cannot be empty in German
error.contactNumber.blank=Contact Number cannot be empty in German
error.quantity.negative=Quantity should not be zero or a negative number in German
```

MobileAccessoriesShop/src/main/resources/messages_fr.properties

```
label.customerName=Customer Name in French
label.contactNumber=Contact Number in French
label.productName=Product Name in French
label.quantity=Quantity in French

error.customerName.blank=Customer Name cannot be empty in French
error.contactNumber.blank=Contact Number cannot be empty in French
error.quantity.negative=Quantity should not be zero or a negative number in French
```

Grade

Reviewed on Thursday, 20 January 2022, 11:05 PM by Automatic grade

Grade 100 / 100

Assessment report

[-]**Grading and Feedback**

```
Feature Test  - 30.0 / 30.0(Success)
         * check whether the controller annotation is present above the Controller - 6.0 /
6.0
         * check whether the ModelAttribute annotation is present above the Method - 6.0 /
6.0
         * check whether the Service is autowired inside the Controller - 6.0 / 6.0
         * check whether Service annotation is present above the Service - 6.0 / 6.0
         * check whether the ModelAttribute annotation has the appropriate value above the
Method - 6.0 / 6.0

Functional testing - 20.0 / 20.0(Success)
         * check whether the Total cost is calculated correctly for the product MemoryCard
- 4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product USBCable -
4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product PenDrive -
4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product HeadPhone
- 4.0 / 4.0
         * check whether the Total cost is calculated correctly for the product
TravelAdapter - 4.0 / 4.0

Spring Testing in the Controller - 20.0 / 20.0(Success)
         * check whether the request is mapped for showPage and redirected to showpage -
5.0 / 5.0
         * check whether layering is followed that is whether total cost is calculated by
invoking the service method inside the Controller - 10.0 / 10.0
         * check whether the request is mapped for calculateTotalCost and redirected to
result page - 5.0 / 5.0

UI Testing - 25.0 / 25.0(Success)
         * check whether the showpage jsp is rendered for the product MemoryCard - 1.0 /
1.0
         * check whether the showpage jsp is rendered for the product USBCable - 1.0 / 1.0
         * check whether the showpage jsp is rendered for the product PenDrive - 1.0 / 1.0
         * check whether the showpage jsp is rendered for the product HeadPhone - 1.0 /
1.0
         * check whether the showpage jsp is rendered for the product TravelAdapter - 1.0
/ 1.0
         * check whether the showpage jsp is rendered with proper constraints for
customerName - 1.0 / 1.0
         * check whether the showpage jsp is rendered with proper constraints for
contactNumber - 1.0 / 1.0
         * check whether the showpage jsp is rendered with proper constraints for quantity
- 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product memoryCard - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product USBCable - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product PenDrive - 1.0 / 1.0
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product HeadPhone - 1.0 / 1.0
```

```
         * check whether UI from showpage  to result page is redirected after calculating
the total cost for the product TravelAdapter - 1.0 / 1.0
         * check whether label names from the property file is correctly displayed for
English - 2.0 / 2.0
         * check whether label names from the property file is correctly displayed for
German - 2.0 / 2.0
         * check whether error message for blank from the property file is correctly
displayed for French - 2.0 / 2.0
         * check whether error message for Customer Name in blank from the property file
is correctly displayed for German - 2.0 / 2.0
         * check whether error message for Contact Number in blank from the property file
is correctly displayed for German - 2.0 / 2.0
         * check whether error message for Quality is negative from the property file is
correctly displayed for French - 2.0 / 2.0

Comments and best practices/standards - 5.0 / 5.0(Success)
```

Test Case Passed