

Offimart/pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.1.8.RELEASE</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.example</groupId>
12    <artifactId>Offimart</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>Offimart</name>
15    <description>Demo project for Spring Boot</description>
16    <properties>
17        <java.version>1.8</java.version>
18    </properties>
19
20    <dependencies>
21        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-web</artifactId>
24        </dependency>
25
26        <dependency>
27            <groupId>org.apache.tomcat.embed</groupId>
28            <artifactId>tomcat-embed-jasper</artifactId>
29            <scope>provided</scope>
30        </dependency>
31
32        <dependency>
33            <groupId>javax.servlet</groupId>
34            <artifactId>servlet-api</artifactId>
35            <version>2.5</version>
36            <scope>provided</scope>
37        </dependency>
38    </dependencies>
39
40    <groupId>javax.servlet.jsp</groupId>
41    <artifactId>jsp-api</artifactId>
42    <version>2.1</version>
43    <scope>provided</scope>
44    </dependency>
45
46    <groupId>taglibs</groupId>
47    <artifactId>standard</artifactId>
48    <version>1.1.2</version>
49    </dependency>
50
51    <groupId>javax.servlet</groupId>
52    <artifactId>jstl</artifactId>
53    <version>1.2</version>
54    </dependency>
55    <!-- Spring test dependencies -->
56    <dependency>
57        <groupId>org.mockito</groupId>
58        <artifactId>mockito-all</artifactId>
59        <version>1.10.19</version>
```

```

60             <scope>test</scope>
61         </dependency>
62
63         <!-- <dependency>
64             <groupId>org.seleniumhq.selenium</groupId>
65             <artifactId>selenium-java</artifactId>
66             <version>2.53.0</version>
67         </dependency>-->
68
69         <dependency>
70 <groupId>org.seleniumhq.selenium</groupId>
71 <artifactId>htmlunit-driver</artifactId>
72 <version>2.26</version>
73 </dependency>
74
75         <!-- https://mvnrepository.com/artifact/org.w3c.css/sac -->
76 <dependency>
77 <groupId>org.w3c.css</groupId>
78 <artifactId>sac</artifactId>
79 <version>1.3</version>
80 </dependency>
81
82
83 <dependency>
84             <groupId>org.springframework</groupId>
85             <artifactId>spring-test</artifactId>
86             <version>5.1.7.RELEASE</version> <!-- 4.0.5 -->
87             <scope>test</scope>
88         </dependency>
89 <!-- End of spring test dependencies -->
90
91
92
93         <dependency>
94             <groupId>org.springframework.boot</groupId>
95             <artifactId>spring-boot-starter-test</artifactId>
96             <scope>test</scope>
97             <exclusions>
98                 <exclusion>
99                     <groupId>org.junit.vintage</groupId>
100                     <artifactId>junit-vintage-engine</artifactId>
101                 </exclusion>
102             </exclusions>
103         </dependency>
104     </dependencies>
105
106     <build>
107         <plugins>
108             <plugin>
109                 <groupId>org.springframework.boot</groupId>
110                 <artifactId>spring-boot-maven-plugin</artifactId>
111             </plugin>
112         </plugins>
113     </build>
114
115 </project>

```

116

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

```

1 package com.controller;
2
3 import java.util.Locale;

```

```

4
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.context.annotation.Configuration;
7 import org.springframework.context.support.ResourceBundleMessageSource;
8 import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;
9 import org.springframework.web.servlet.LocaleResolver;
10 import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
11 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
12 import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
13 import org.springframework.web.servlet.i18n.SessionLocaleResolver;
14
15 @Configuration
16 public class InternationalizationConfig implements WebMvcConfigurer {
17
18 //Create a SessionLocaleResolver object and set the default locale to English return the
SessionLocaleResolver object
19     @Bean
20     public LocaleResolver localeResolver() {
21
22         SessionLocaleResolver slr=new SessionLocaleResolver();
23         slr.setDefaultLocale(Locale.US);
24         return slr;
25
26
27     }
28
29 //Create LocaleChangeInterceptor object and set the parameter name as language and return the
localeChangeInterceptor
30     @Bean
31     public LocaleChangeInterceptor localeChangeInterceptor() {
32         LocaleChangeInterceptor lci=new LocaleChangeInterceptor();
33         lci.setParamName("language");
34         return lci;
35
36
37     }
38
39
40 //register the LocaleChangeInterceptor
41     @Override
42     public void addInterceptors(InterceptorRegistry registry) {
43         registry.addInterceptor(localeChangeInterceptor());
44
45     }
46
47     @Bean
48     public ResourceBundleMessageSource messageSource()
49     {
50         ResourceBundleMessageSource rms=new ResourceBundleMessageSource();
51         rms.setBasenames("messages","messages_fr","messages_de");
52         return rms;
53
54
55     }
56
57     @Bean
58     public LocalValidatorFactoryBean getValidator() {
59         LocalValidatorFactoryBean bean = new LocalValidatorFactoryBean();
60         bean.setValidationMessageSource(messageSource());
61         return bean;
62
63     }
64 }
65

```

Offimart/src/main/java/com/controller/ItemController.java

```
1 package com.controller;
2
3 import java.util.ArrayList;
4 import javax.validation.Valid;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7
8 import org.springframework.validation.BindingResult;
9
10 import com.exception.InvalidItemException;
11 import com.model.Item;
12 import com.service.ItemService;
13 import org.springframework.web.bind.annotation.ExceptionHandler;
14 import org.springframework.web.bind.annotation.ModelAttribute;
15 import org.springframework.web.bind.annotation.RequestMapping;
16 import org.springframework.web.bind.annotation.RequestMethod;
17 import org.springframework.web.servlet.ModelAndView;
18
19 //use appropriate annotation to configure ItemController as Controller
20 @Controller
21 public class ItemController {
22
23     @Autowired
24     private ItemService itemService;
25
26     @RequestMapping(value = "/showPage", method = RequestMethod.GET)
27     public String showPage(@ModelAttribute("item")Item item) {
28         return "showPage";
29     }
30
31     @ModelAttribute("itemList")
32     public ArrayList<String> populateItems(){
33         ArrayList<String> itemList = new ArrayList<String>();
34         itemList.add("IDcard holder");
35         itemList.add("Whiteboard");
36         itemList.add("Pen");
37         itemList.add("Binder clips");
38         itemList.add("Cutters");
39         itemList.add("stamp Pad");
40         return itemList;
41     }
42
43
44     //invoke the service class - calculateCostAndUpdate method.
45     @RequestMapping(value = "/billDesk", method = RequestMethod.POST)
46     public ModelAndView calculateCostAndUpdate(@ModelAttribute("item")@Valid Item item,
47 BindingResult result,ModelAndView mv)throws InvalidItemException {
48
49         if(result.hasErrors()){
50
51             mv.setViewName("showPage");
52         }
53         else{
54             Item obj = itemService.getItemFromList(item.getItemName());
55             boolean flag= itemService.performUpdate(obj, item.getQuantity());
56             if(flag) {
57                 double cost=itemService.calculateCost(obj.getCostPerItem(),
58 item.getQuantity());
59
60                 mv.addObject("cost", cost);
61             }
62         }
63     }
64 }
```

```

59             mv.addObject("costPerItem", obj.getCostPerItem());
60             mv.addObject("itemName", item.getItemName());
61             mv.addObject("quantity", item.getQuantity());
62         }
63         mv.setViewName("billDesk");
64     }
65
66     //itemService.calculateCostAndUpdate(item,mv);
67
68     return mv;
69 }
70 @ExceptionHandler(InvalidItemException.class)
71 public ModelAndView exceptionHandler(Exception e) {
72
73     ModelAndView mav = new ModelAndView();
74     mav.addObject("message", e.getMessage());
75     mav.setViewName("exceptionPage");
76     return mav;
77 }
78
79 }

```

80

Offimart/src/main/java/com/example/demo/OffimartApplication.java

```

1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.context.annotation.ComponentScan;
6
7 @SpringBootApplication
8 @ComponentScan({"com.*"})
9 public class OffimartApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(OffimartApplication.class, args);
13     }
14
15 }
16

```

Offimart/src/main/java/com/exception/InvalidItemException.java

```

1 package com.exception;
2
3 public class InvalidItemException extends Exception{
4     private static final long serialVersionUID = 1L;
5     public InvalidItemException(String msg) {
6
7         super(msg);
8     }
9
10 }
11

```

Offimart/src/main/java/com/model/Item.java

```

1 package com.model;
2
3 import javax.validation.constraints.Min;
4
5 import org.springframework.stereotype.Component;

```

```

6
7 @Component
8 public class Item {
9
10 private String itemId;
11 private String itemName;
12 @Min(value=1,message="{error.quantity}")
13 private int quantity;
14 private double costPerItem;
15 public String getItemId() {
16     return itemId;
17 }
18 public void setItemId(String itemId) {
19     this.itemId = itemId;
20 }
21 public String getItemName() {
22     return itemName;
23 }
24 public void setItemName(String itemName) {
25     this.itemName = itemName;
26 }
27 public int getQuantity() {
28     return quantity;
29 }
30 public void setQuantity(int quantity) {
31     this.quantity = quantity;
32 }
33 public double getCostPerItem() {
34     return costPerItem;
35 }
36 public void setCostPerItem(double costPerItem) {
37     this.costPerItem = costPerItem;
38 }
39
40 public Item(String itemId, String itemName, int quantity, double costPerItem) {
41     super();
42     this.itemId = itemId;
43     this.itemName = itemName;
44     this.quantity = quantity;
45     this.costPerItem = costPerItem;
46 }
47 public Item(){}
48
49

```

Offimart/src/main/java/com/service/ItemService.java

```

1 package com.service;
2
3 import java.util.ArrayList;
4 import org.springframework.stereotype.Service;
5
6 import com.exception.InvalidItemException;
7 import com.model.Item;
8
9 //use appropriate annotation to configure ElectricityService as a Service
10 @Service
11 public class ItemService {
12     private ArrayList<Item> itemList=new ArrayList<Item>();
13
14     //calculate the totalCost and return the cost
15     public double calculateCost(double costPerKg, int quantity){
16         return (costPerKg*quantity);
17     }
18 }

```

```

17         }
18
19         //Fetch the product by using productName
20         public Item getItemFromList(String itemName) throws InvalidItemException {
21
22             Item obj1=null;
23             for(Item obj: itemList) {
24                 if(obj.getItemName().equalsIgnoreCase(itemName)) {
25
26                     obj1= obj;
27                 }
28             }
29             return obj1;
30         }
31         // update Item quantity in the productList
32         public boolean performUpdate(Item item,int quantity) throws InvalidItemException {
33
34             boolean flag=false;
35             if(item.getQuantity()>=quantity) {
36                 item.setQuantity(item.getQuantity()-
quantity);
37
38                 flag=true;
39             }
40
41             else {
42
43                 throw new InvalidItemException("No stocks
are available for the item "+item.getItemName());
44             }
45
46             return flag;
47         }
48
49         //calculate the totalCost and return the cost
50
51         /*public void calculateCostAndUpdate(Item item,ModelAndView mv) throws InvalidItemException {
52             //Model mv=new Model();
53             double result=0.0;
54             for(Item obj: itemList) {
55                 if(obj.getItemName().equalsIgnoreCase(item.getItemName())) {
56
57                     if(obj.getQuantity()>item.getQuantity()) {
58                         obj.setQuantity(obj.getQuantity()-item.getQuantity());
59                         item.setCostPerItem(obj.getCostPerItem());
60                         result=obj.getCostPerItem()*item.getQuantity();
61
62                     if(obj.isOrganic()) {
63                         double discount=(result*10)/100;
64                         item.setDiscountedAmount(discount);
65                         result=result-discount;
66                     }
67                     model.addAttribute("cost", result);
68                     model.addAttribute("costPerItem",
obj.getCostPerItem());
69                     mv.addObject("cost", result);
70                     mv.addObject("costPerItem", obj.getCostPerItem());
71                     mv.addObject("itemName", item.getItemName());
72                     mv.addObject("quantity", item.getQuantity());
73                     mv.setViewName("billDesk");
74                     //item.setItemCost(result);
75                 }
76             }
77         }

```

```

74                                     else {
75                                     throw new InvalidItemException("No stocks are
available for the item "+obj.getItemName());
76                                     }
77                                     }
78                                     }
79                                     }
80                                     //return mv;
81
82                                     */
83
84                                     public ItemService()
85                                     {
86                                     addItem();
87                                     }
88                                     public void addItem() {
89
90                                     itemList.add(new Item("AP101","IDcard holder",200,75));
91                                     itemList.add(new Item("AP102","Whiteboard",75,250));
92                                     itemList.add(new Item("AP103","Pen",150,10));
93                                     itemList.add(new Item("AP104","Binder clips",100,2));
94                                     itemList.add(new Item("AP105","Cutter",50,25));
95                                     itemList.add(new Item("AP106","Stamp pad",45,50));
96                                     itemList.add(new Item("AP107","Notepad",100,15));
97                                     itemList.add(new Item("AP108","Marker",100,20));
98                                     itemList.add(new Item("AP109","Notebook",100,20));
99                                     itemList.add(new Item("AP109","Marker",100,20));
100                                    }
101                                    public ArrayList<Item> getItemList() {
102                                    return itemList;
103                                    }
104
105                                    public void setItemList(ArrayList<Item> list) {
106                                    this.itemList = list;
107                                    }
108
109
110 }

```

111

Offimart/src/main/resources/application.properties

```

1 server.port=9088
2 spring.mvc.view.prefix = /WEB-INF/views/
3 spring.mvc.view.suffix = .jsp
4 spring.mvc.static-path-pattern=/resources/**
5

```

Offimart/src/main/resources/messages.properties

```

1 label.itemName=Select Item in English
2 label.quantity=Quantity in English
3 error.quantity=Quantity should not be negative in English
4 error.itemName=Product should be mandatory in English
5

```

Offimart/src/main/resources/messages_de.properties

```

1 label.itemName=Select Item in German
2 label.quantity=Quantity in German
3 error.quantity=Quantity should not be negative in German
4

```


Offimart/src/main/resources/messages_fr.properties

```
1 label.itemName=Select Item in French
2 label.quantity=Quantity in French
3 error.quantity=Quantity should not be negative in French
4 error.itemName=Product should be mandatory in French
```

Offimart/src/main/webapp/WEB-INF/views/billDesk.jsp

```
1 <%@page isELIgnored="false" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <html>
4 <body bgcolor="lavender">
5 <h1><center> Organic Agro Industries </center></h1>
6 <center>
7
8             <table border="1" cellpadding="1">
9
10                    <tr>
11                            <td>Item Name</td> <td>${itemName}</td>
12                    </tr>
13                    <tr>
14                            <td>Quantity</td> <td>${quantity}</td>
15                    </tr>
16                    <tr>
17                            <td>Cost Per Item Rs:</td> <td>${costPerItem}</td>
18                    </tr>
19                    <tr>
20                            <td>Total price Rs:</td><td>${cost}</td>
21                    </tr>
22            </table></center>
23
24 </body>
25 </html>
```

28

Offimart/src/main/webapp/WEB-INF/views/exceptionPage.jsp

```
1 <%@page isELIgnored="false" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
4 <html>
5 <body bgcolor="lavender">
6 <h1><center> Online Office Mart </center></h1>
7 <center>
8
9             <h3>
10                    <%
11                            if (request.getAttribute("message") != null) {
12                                    <%=request.getAttribute("message")%>
13                            %>
14                    }
15                    %>
16            </h3>
17            <div>
18                    <form id="viewPageForm" action="/showPage" method="GET">
19                            <input type="submit" name="submit" id="submit" value="Back to
home" />
20                    </form>
21            </div>
22            </center>
```

```
23
24 </body>
25 </html>
```

Offimart/src/main/webapp/WEB-INF/views/showPage.jsp

```
1 <%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
2 <%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form" %>
3 <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
4 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
5   pageEncoding="ISO-8859-1" isELIgnored="false"%>
6
7
8 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
9 <html>
10 <body style="background-color:lavender">
11
12
13 <!-- Do not change the below line -->
14 <a href="/showPage?language=en">English</a>|<a href="/showPage?language=de">German</a>|<a
href="/showPage?language=fr">French</a></align>
15 <!-- Design the page as per the requirements-->
16 <h1><center> Online office mart </center></h1>
17 <form:form method="post" action="/billDesk" modelAttribute="item">
18
19 <table style="margin: 0px auto; margin-left: auto; margin-right:auto">
20
21   <tr>
22
23     <td><spring:message code="label.itemName"/></td>
24       <td>
25         <form:select path="itemName" id="itemName">
26           <form:options items="${itemList}"/>
27         </form:select>
28
29       </td>
30     <tr>
31       <td><spring:message code="label.quantity"/></td>
32     <td><form:input path="quantity" id="quantity"/></td>
33     <td><form:errors path="quantity"/></td>
34   </tr>
35   <tr>
36     <td><input type="submit" value="Calculate" id="submit" /></td>
37   </tr>
38 </table>
39 </form:form>
40
41 </body>
42 </html>
43
44
45
46
47
```

Grade

Reviewed on Monday, 2 May 2022, 8:05 PM by Automatic grade

Grade 100 / 100

Assessment report

Assessment Completed Successfully

[.]Grading and Feedback

Feature Test - 20.00 / 20.00(Success)

- * check whether the service is autowired inside the ItemController - 5.00 / 5.00
- * check whether ExceptionHandler annotation is present above the exceptionHandler method - 5.00 / 5.00
- * check whether the Controller annotation is present above the ItemController - 5.00 / 5.00
- * check whether Service annotation is present above the ItemService - 5.00 / 5.00

Spring Testing in the Controller - 17.50 / 17.50(Success)

- * check whether the request is mapped for showPage and redirected to showpage.jsp page - 5.00 / 5.00
- * check whether the request is mapped for billDesk and redirected to billDesk.jsp page - 5.00 / 5.00
- * check whether layering is followed that is whether item cost is calculated by invoking the service method inside the Controller - 7.50 / 7.50

Functional testing - 44.00 / 44.00(Success)

- * check whether the item cost for IDcard holder item is calculated properly - 6.00 / 6.00
- * check whether the item cost for Whiteboard is calculated properly - 6.00 / 6.00
- * check whether the discount amount for Binder clips is calculated properly - 6.00 / 6.00
- * check whether the quantity is updated properly for item Pen - 6.00 / 6.00
- * check whether the quantity is updated properly for item Cutter - 6.00 / 6.00
- * check whether the quantity is updated properly for item StampPad - 7.00 / 7.00
- * check whether User defined exception raised properly - 7.00 / 7.00

UI Testing - 13.50 / 13.50(Success)

- * check whether itemName field auto-populated in showPage UI properly for item IDcard holder - 2.00 / 2.00
- * check whether itemName field auto-populated in showPage UI properly for item Binder clips - 2.00 / 2.00
- * check whether itemName field auto-populated in showPage UI properly for item Whiteboard - 1.00 / 1.00
- * check whether itemName field auto-populated in showPage UI properly for item Pen - 1.00 / 1.00
- * check whether the component quantity configured properly in showPage.jsp - 0.50 / 0.50
- * check whether the output is rendered properly for item Pen in billDesk.jsp page - 1.50 / 1.50
- * check whether the output is rendered properly for item Whiteboard in billDesk.jsp page - 0.50 / 0.50
- * check whether exception page is rendered properly when insufficient quantity - 1.50 / 1.50
- * check whether showPage is rendered with English language - 0.50 / 0.50
- * check whether showPage is rendered with German language - 0.50 / 0.50
- * check whether showPage is rendered with French language - 0.50 / 0.50
- * check whether error message for quantity from the property file is correctly displayed for German - 1.00 / 1.00
- * check whether error message for quantity from the property file is correctly displayed for French - 1.00 / 1.00

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

Test Case Passed