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


import com.model.Order;
import com.service.BillService;


//use appropriate annotation to configure BillController as Controller
@Controller
public class BillController {
	@Autowired
	BillService service;


	@ModelAttribute("productList")
	public Map<String, String> populateProductType() {
		Map<String, String> serviceMap = new HashMap<String, String>();


		serviceMap.put("HeadPhone", "HeadPhone");
```

```
        serviceMap.put("TravelAdapter", "TravelAdapter");
        serviceMap.put("MemoryCard", "MemoryCard");
        serviceMap.put("PenDrive", "PenDrive");
        serviceMap.put("USBCable", "USBCable");

        return serviceMap;
    }

    @GetMapping("/showPage")
    public String showPage(@ModelAttribute("order") Order order) {
        return "showpage";
    }

    // invoke the service class - calculateTotalCost method.
    @PostMapping("/billDesk")
    public String calculateTotalCost(@Valid @ModelAttribute("order") Order order,
BindingResult result,
            ModelMap model) {

        // fill the code
        if (result.hasErrors()) {
            return "showpage";
        } else {
            double totalCost = service.calculateTotalCost(order);
            model.put("customerName", order.getCustomerName());
            model.put("totalCost", totalCost);
            return "billdesk";

        }

    }
```

```
}
```

INTERNATIONALIZATIONCONFIG.JAVA

```
package com.controller;

import java.util.Locale;

import org.springframework.context.MessageSource;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.support.ReloadableResourceBundleMessageSource;
import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;
import org.springframework.web.servlet.LocaleResolver;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
import org.springframework.web.servlet.i18n.SessionLocaleResolver;

@Configuration
public class InternationalizationConfig implements WebMvcConfigurer {

//Create a SessionLocaleResolver object and set the default locale to English
return the SessionLocaleResolver object
        @Bean
        public LocaleResolver localeResolver() {
                SessionLocaleResolver localeResolver = new SessionLocaleResolver();
                localeResolver.setDefaultLocale(Locale.US);
                return localeResolver;
        }

        // Create LocaleChangeInterceptor object and set the parameter name as
language
        // and return the localeChangeInterceptor
        @Bean
        public LocaleChangeInterceptor localeChangeInterceptor() {
                LocaleChangeInterceptor interceptor = new LocaleChangeInterceptor();
                interceptor.setParamName("language");
                return interceptor;

        }
        @Bean
        public LocalValidatorFactoryBean getValidator() {
            LocalValidatorFactoryBean bean = new LocalValidatorFactoryBean();
            bean.setValidationMessageSource(messageSource());
            return bean;
        }
        @Bean
        public MessageSource messageSource() {
                ReloadableResourceBundleMessageSource messageSource = new
ReloadableResourceBundleMessageSource();
```

```
        messageSource.setBasename("classpath:messages");
        messageSource.setDefaultEncoding("UTF-8");
        return messageSource;
    }

    // register the LocaleChangeInterceptor
    @Override
    public void addInterceptors(InterceptorRegistry registry) {

        registry.addInterceptor(localeChangeInterceptor());
    }
}
```

MOBILEACCESSORIESSHOPALLICATION

```
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 MobileAccessoriesShopApplication {

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

}
```

ORDER.JAVA

```
package com.model;

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

//pojo class with required attributes,getters and setters
public class Order {
    @NotBlank(message="{error.customerName.blank}")
    private String customerName;
    @NotBlank(message="{error.contactNumber.blank}")
    private String contactNumber;
    private String productName;
    @Positive(message="{error.quantity.negative}")
    private int quantity;

    public String getCustomerName() {
        return customerName;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public String getContactNumber() {
        return contactNumber;
    }
    public void setContactNumber(String contactNumber) {
        this.contactNumber = contactNumber;
    }
    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;
    }

}
```

BILLSERVICE.JAVA

```
package com.service;

import org.springframework.stereotype.Service;

import com.model.Order;

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

    // calculate the totalCost and return the cost
    public double calculateTotalCost(Order order) {
        double totalCost = 0.0;
        double cost = 0.0;
        int perProductPrice = 0;
        if (order.getProductName().equalsIgnoreCase("HeadPhone")) {
            perProductPrice = 450;
        }
        else if (order.getProductName().equalsIgnoreCase("TravelAdapter")) {
            perProductPrice = 1000;
        }
        else if (order.getProductName().equalsIgnoreCase("MemoryCard")) {
            perProductPrice = 300;
        }
        else if (order.getProductName().equalsIgnoreCase("PenDrive")) {
            perProductPrice = 650;
        }
        else if (order.getProductName().equalsIgnoreCase("USBCable")) {
            perProductPrice = 800;
        }
        cost = order.getQuantity() * perProductPrice ;
        totalCost = cost +(cost*3)/100;
```

```java
        // fill the code
        return totalCost;
    }

}
package com.service;

import org.springframework.stereotype.Service;

import com.model.Order;

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

    // calculate the totalCost and return the cost
    public double calculateTotalCost(Order order) {
        double totalCost = 0.0;
        double cost = 0.0;
        int perProductPrice = 0;
        if (order.getProductName().equalsIgnoreCase("HeadPhone")) {
            perProductPrice = 450;
        }
        else if (order.getProductName().equalsIgnoreCase("TravelAdapter")) {
            perProductPrice = 1000;
        }
        else if (order.getProductName().equalsIgnoreCase("MemoryCard")) {
            perProductPrice = 300;
        }
        else if (order.getProductName().equalsIgnoreCase("PenDrive")) {
            perProductPrice = 650;
        }
        else if (order.getProductName().equalsIgnoreCase("USBCable")) {
            perProductPrice = 800;
        }
        cost = order.getQuantity() * perProductPrice ;
        totalCost = cost +(cost*3)/100;
        // fill the code
        return totalCost;
    }

}
```

APPLICATION.PROPERTIES

```properties
#Do not change the configurations
#value for server.port can be changed
#include other needed configurations
server.port=9095
spring.mvc.view.prefix = /WEB-INF/views/
spring.mvc.view.suffix = .jsp
spring.mvc.static-path-pattern=/resources/**
```

Message_de.properties

```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
```

message_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
```

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
```

MOBILEACCESSORIESSHOPAPPLICATIONTEST

```
//package com.example.demo;
//
//import org.junit.jupiter.api.Test;
//import org.springframework.boot.test.context.SpringBootTest;
//
//@SpringBootTest
//class MobileAccessoriesShopApplicationTests {
//
//	@Test
//	void contextLoads() {
//	}
//
//}
```

BILLDESK.JSP

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

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


</body>
</html>
```

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>
<head>
<style>
.center {
    margin-left: auto;
    margin-right: auto;
}
</style>

</head>
<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-->


    <form:form modelAttribute="order" method="post" action="/billDesk">
```

```
        <table class="center">

            <tr>
                <td id="id1"><spring:message code="label.customerName"
/></td>
                <td><form:input path="customerName" id="customerName"
                        name="customerName" /></td>
                <td><form:errors
path="customerName"></form:errors></td>

            </tr>
            <tr>
                <td id="id2"><spring:message code="label.contactNumber"
/></td>
                <td><form:input path="contactNumber" id="contactNumber"
                        name="contactNumber" /></td>
                <td><form:errors
path="contactNumber"></form:errors></td>
            </tr>

            <tr>
                <td id="id3"><spring:message code="label.productName"
/></td>
                <td><form:select path="productName" id="productName"
name="productName"
                        items="${productList}" /></td>
            </tr>
            <tr>
                <td id="id2"><spring:message code="label.quantity"
/></td>
                <td><form:input path="quantity" id="quantity"
                        name="quantity" /></td>
                <td><form:errors path="quantity"></form:errors></td>
            </tr>

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

        </table>
    </form:form>
</body>
</html>
```