

Buffet Restaurant

BookController.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.BookRestaurant;
import com.service.BookService;


@Controller
public class BookController {

    @Autowired
    private BookService bookService;

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

```

        return "showpage";
    }

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

    public String calculateTotalCost(@Valid @ModelAttribute("book") BookRestaurant
book, BindingResult result,ModelMap model) {

        if(result.hasErrors()) {

            return "showpage";

        }

        else {

            double totalcost=bookService.calculateTotalCost(book);

            model.addAttribute("totalcost",totalcost);

        }

        return "billdesk";
    }

    @ModelAttribute("hallList")

    public Map<String, String> populateHallType(){

        Map<String,String> m=new HashMap<>();

        m.put("AC", "AC");

        m.put("NonAC", "NonAC");

        return m;

    }

}

```

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.WebMvcConfigurerAdapter;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
import org.springframework.web.servlet.i18n.SessionLocaleResolver;

@Configuration
public class InternationalizationConfig extends WebMvcConfigurerAdapter {

    //Create a SessionLocaleResolver object and set the default locale to English return the
    SessionLocaleResolver object

    @Bean
    public LocaleResolver localeResolver() {
        SessionLocaleResolver resolver = new SessionLocaleResolver();
        resolver.setDefaultLocale(Locale.ENGLISH);
        return resolver;
    }
}
```

```

// 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 MessageSource messageSource() {

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

@Bean
public LocalValidatorFactoryBean validator(MessageSource messageSource) {

    LocalValidatorFactoryBean bean = new LocalValidatorFactoryBean();
    bean.setValidationMessageSource(messageSource);
    return bean;
}

// register the LocaleChangeInterceptor

@Override

```

```
        public void addInterceptors(InterceptorRegistry registry) {  
  
            registry.addInterceptor(localeChangeInterceptor());  
  
        }  
    }  
}
```

BuffetRestaurantApplication

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 BuffetRestaurantApplication {

public static void main(String[] args) {

SpringApplication.run(BuffetRestaurantApplication.class, args);

}

}

BookRestaurant.java

```
package com.model;

import javax.validation.constraints.Min;
import javax.validation.constraints.NotEmpty;
import org.springframework.stereotype.Component;

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

    @NotEmpty( message="{error.customerName.blank}")
    private String customerName;

    @NotEmpty(message="{error.contactNumber.blank}")
    private String contactNumber;

    private String hallType;

    @Min(value=0,message="{error.adultCount.forVeg.positive}")
    private int adultVegCount;

    @Min(value=0,message="{error.kidsCount.forVeg.positive}")
    private int kidsVegCount;

    @Min(value=0,message="{error.adultCount.forNonVeg.positive}")
    private int adultNonVegCount;

    @Min(value=0,message="{error.kidsCount.forNonVeg.positive}")
    private int kidsNonVegCount;

    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 getHallType() {  
    return hallType;  
}  
  
public void setHallType(String hallType) {  
    this.hallType = hallType;  
}  
  
public int getAdultVegCount() {  
    return adultVegCount;  
}  
  
public void setAdultVegCount(int adultVegCount) {  
    this.adultVegCount = adultVegCount;  
}  
  
public int getKidsVegCount() {  
    return kidsVegCount;  
}  
  
public void setKidsVegCount(int kidsVegCount) {  
    this.kidsVegCount = kidsVegCount;  
}
```

```

    public int getAdultNonVegCount() {
        return adultNonVegCount;
    }

    public void setAdultNonVegCount(int adultNonVegCount) {
        this.adultNonVegCount = adultNonVegCount;
    }

    public int getKidsNonVegCount() {
        return kidsNonVegCount;
    }

    public void setKidsNonVegCount(int kidsNonVegCount) {
        this.kidsNonVegCount = kidsNonVegCount;
    }
}

```

BookService.java

```

package com.service;

import org.springframework.stereotype.Service;

import com.model.BookRestaurant;

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

    //calculate the totalCost and return the cost
    public double calculateTotalCost(BookRestaurant book) {

```



```

        double cost=0.0;

cost=book.getAdultVegCount()*599+book.getKidsVegCount()*249+book.getAdultNonVegCou
nt()*699+book.getKidsNonVegCount()*349;

        String s=book.getHallType();

            if(s.equalsIgnoreCase("AC")) {

                cost+=500;

            }

        return cost;

    }

}

```

application.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/**

```

messages_de.properties

```

label.customerName=Customer Name in German
label.contactNumber=Contact Number in German
label.hallType=Hall Type in German
label.adultVegCount=Adult Count For Veg in German
label.kidsVegCount=Kids Count For Veg in German

```

label.adultNonVegCount=[Adult Count For NonVeg in German](#)
label.kidsNonVegCount=[Kids Count For NonVeg in German](#)

error.customerName.blank=[Customer Name cannot be empty in German](#)
error.contactNumber.blank=[Contact Number cannot be empty in German](#)
error.adultCount.forVeg.positive=[Adult count for Veg should not be a negative number in German](#)
error.kidsCount.forVeg.positive=[Kids count for Veg should not be a negative number in German](#)
error.adultCount.forNonVeg.positive=[Adult count for NonVeg should not be a negative number in German](#)
error.kidsCount.forNonVeg.positive=[Kids count for NonVeg should not be a negative number in German](#)

messages_fr.properties

label.customerName=[Customer Name in French](#)
label.contactNumber=[Contact Number in French](#)
label.hallType=[Hall Type in French](#)
label.adultVegCount=[Adult Count For Veg in French](#)
label.kidsVegCount=[Kids Count For Veg in French](#)
label.adultNonVegCount=[Adult Count For NonVeg in French](#)
label.kidsNonVegCount=[Kids Count For NonVeg in French](#)

error.customerName.blank=[Customer Name cannot be empty in French](#)
error.contactNumber.blank=[Contact Number cannot be empty in French](#)
error.adultCount.forVeg.positive=[Adult count for Veg should not be a negative number in French](#)
error.kidsCount.forVeg.positive=[Kids count for Veg should not be a negative number in French](#)
error.adultCount.forNonVeg.positive=[Adult count for NonVeg should not be a negative number in French](#)
error.kidsCount.forNonVeg.positive=[Kids count for NonVeg should not be a negative number in French](#)

messages.properties

label.customerName=[Customer Name in English](#)
label.contactNumber=[Contact Number in English](#)
label.hallType=[Hall Type in English](#)
label.adultVegCount=[Adult Count For Veg in English](#)
label.kidsVegCount=[Kids Count For Veg in English](#)
label.adultNonVegCount=[Adult Count For NonVeg in English](#)
label.kidsNonVegCount=[Kids Count For NonVeg in English](#)

error.customerName.blank=Customer Name cannot be empty in English
error.contactNumber.blank=Contact Number cannot be empty in English
error.adultCount.forVeg.positive=Adult count for Veg should not be a negative number in English
error.kidsCount.forVeg.positive=Kids count for Veg should not be a negative number in English
error.adultCount.forNonVeg.positive=Adult count for NonVeg should not be a negative number in English
error.kidsCount.forNonVeg.positive=Kids count for NonVeg should not be a negative number in English

billdesk.jsp

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

<h2>Thanks for Booking!!! Total Amount to be paid is 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>
<body style="background-color: lavender">
    <h1>
        <center>Delicious Buffet Restaurant</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 method="post" action="/billDesk" modelAttribute="book">
    <table align="center">
        <tr>
            <td><spring:message code="label.customerName" /></td>
            <td><form:input path="customerName" id="customerName"
/></td>

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

            <td><form:errors path="contactNumber" /></td>
        </tr>
        <tr>
            <td><spring:message code="label.hallType" /></td>
            <td><form:select path="hallType" items="${hallList} "
id="hallType" /></td>

        </tr>
        <tr>
            <td><spring:message code="label.adultVegCount" /></td>
            <td><form:input path="adultVegCount" id="adultVegCount"
/></td>

            <td><form:errors path="adultVegCount" /></td>
        </tr>
        <tr>
            <td><spring:message code="label.kidsVegCount" /></td>
            <td><form:input path="kidsVegCount" id="kidsVegCount"
/></td>

            <td><form:errors path="kidsVegCount" /></td>
        </tr>
        <tr>
            <td><spring:message code="label.adultNonVegCount" /></td>
            <td><form:input path="adultNonVegCount"
id="adultNonVegCount" /></td>

            <td><form:errors path="adultNonVegCount" /></td>
        </tr>
        <tr>
            <td><spring:message code="label.kidsNonVegCount" /></td>
            <td><form:input path="kidsNonVegCount"
id="kidsNonVegCount" /></td>

            <td><form:errors path="kidsNonVegCount" /></td>
        </tr>
    </table>
</form>

```

```
        <tr>
            <td><input type="submit" value="CaluclateCost" id="submit"
                name="submit"></td>
            <td><input type="reset" value="cancel" /></td>
        </tr>
    </table>
```

```
</form:form>
```

```
</body>
```

```
</html>
```