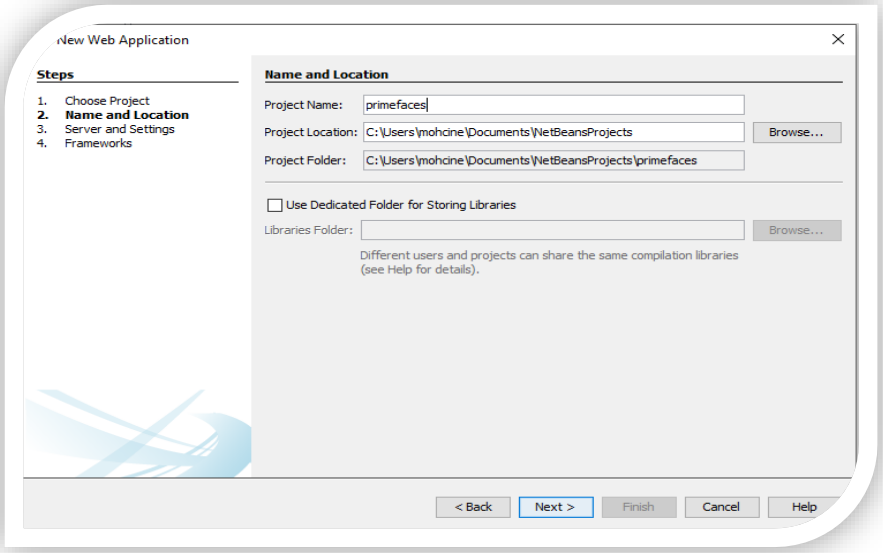
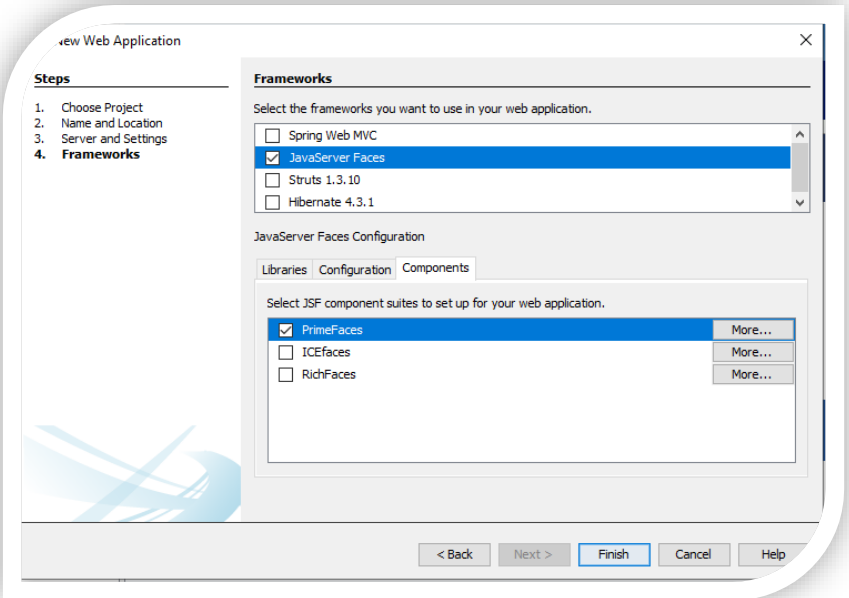
ATELIER 3 : Applications Web avec PrimeFaces

Objectif : développement d'une applications Web en utilisant JavaServer Faces et la Bibliothèque **PrimeFaces.**

Etape 1 : on commence par la création d'un projet java EE web on spécifiant Java server face comme Framework et Primafaces comment composant JSF.

Les deux images ci-dessous illustre cette étape.

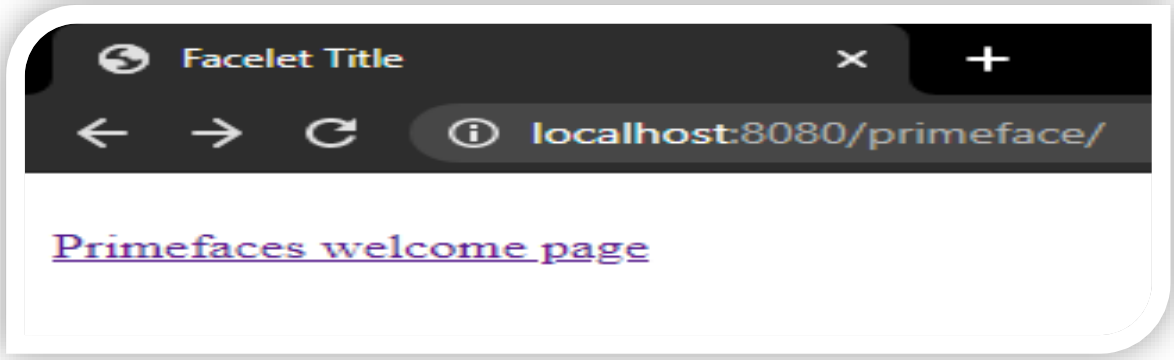




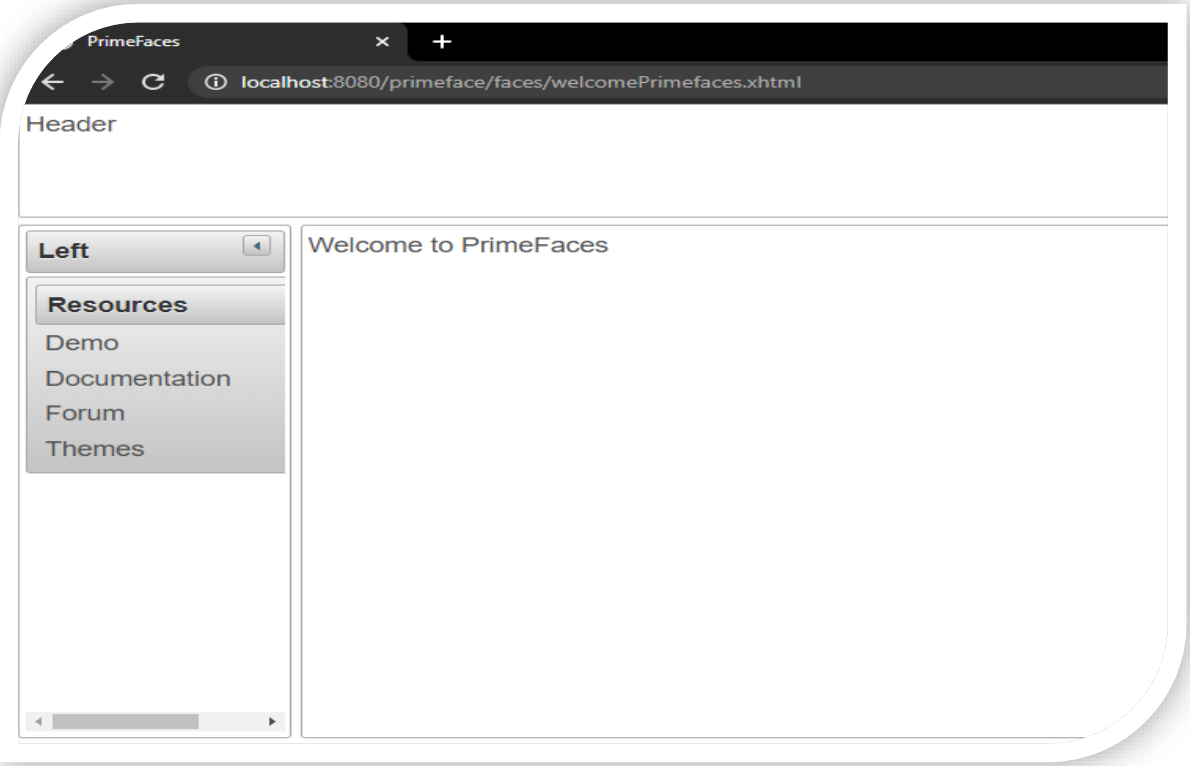
Après la création notre projet est présenté sous la structure suivante.

On clique sur Run pour tester l'application.

L'application est démarrée sur le lien : localhost:/8080/primeface

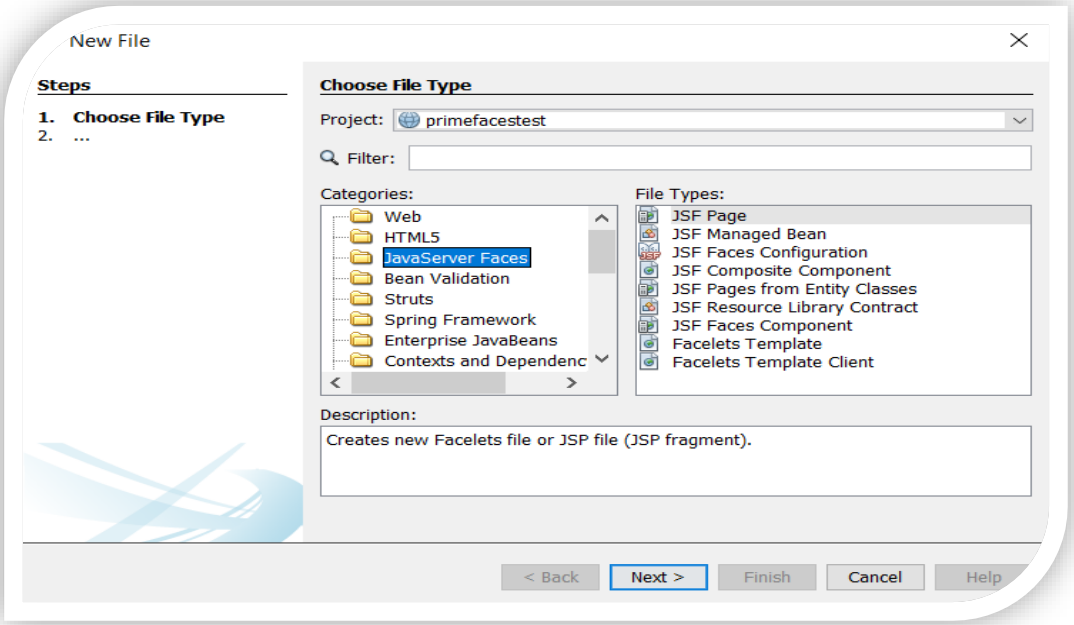


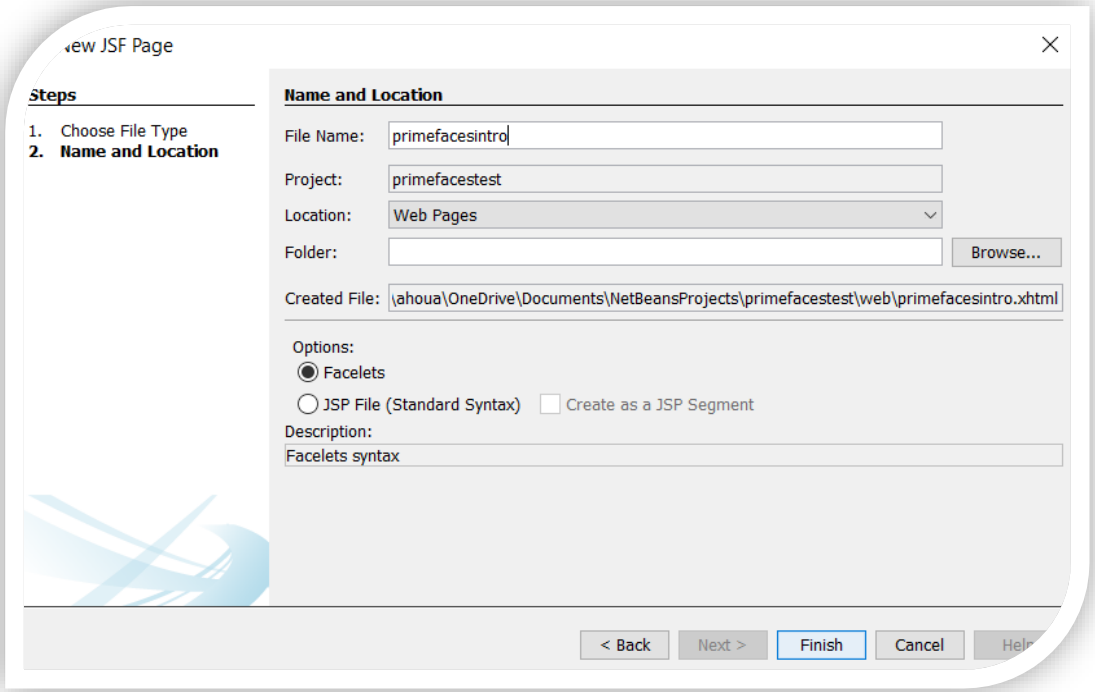
En cliquant sur le lien PrimeFaces welcome page on navigue vers une autre page sous le lien :
localhost :8080/primeface/face/welcomePrimeface.xhtml



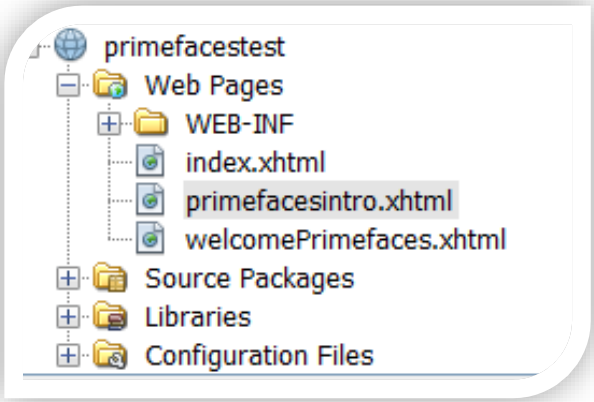
Etape 2 : utilisation des composants PrimeFaces.

Les composants PrimeFaces nous permettent de gagner beaucoup de temps lors du développement de nos applications JSF.





L’ajout du fichier :

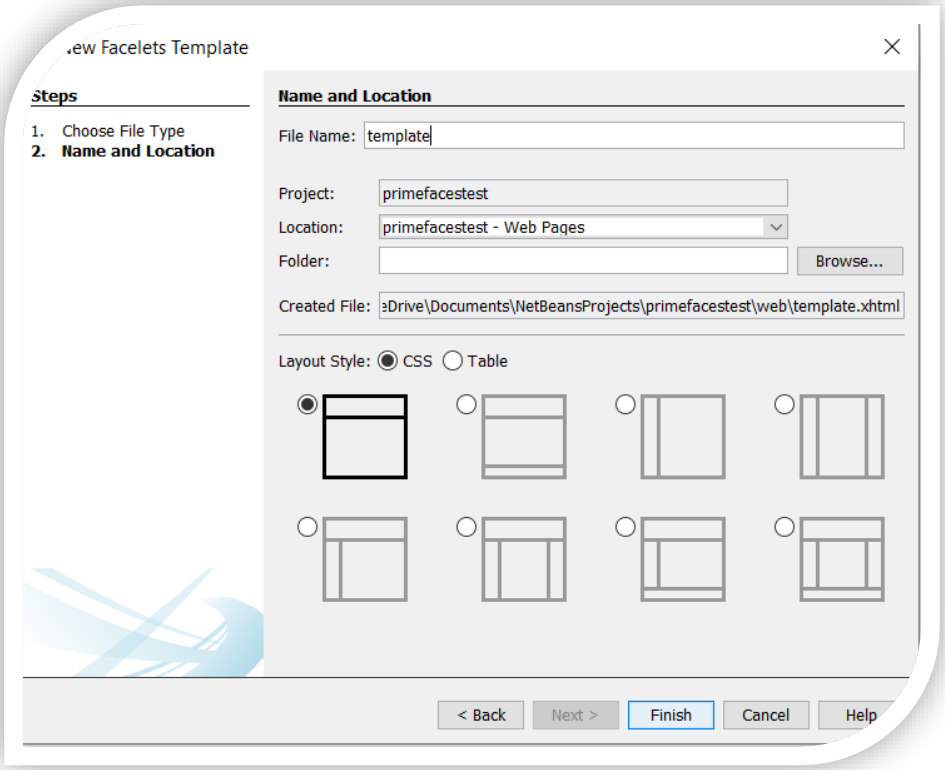


Le code du notre page :

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:p="http://primefaces.org/ui"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<ui:composition template="./template.xhtml">
<ui:define name="top">
<h2>Customer Information Data Entry</h2>
</ui:define>
<ui:define name="content">
<h:form>
<p:messages/>
<p:panel header="Enter Customer Information">
<h:panelGrid columns="2">
<h:outputLabel for="firstName" value="First Name"
styleClass="requiredLbl"/>
<h:inputText id="firstName" label="First Name"
value="#{customer.firstName}" required="true"/>

<h:outputLabel for="middleName" value="Middle
Name" styleClass="optionalLbl"/>
<h:inputText id="middleName" label="Middle Name"
value="#{customer.middleName}"/>
<h:outputLabel for="lastName" value="Last Name"
styleClass="requiredLbl"/>
<h:inputText id="lastName" label="Last Name"
value="#{customer.lastName}" required="true"/>
<h:outputLabel for="birthDate" value="Date of Birth"
styleClass="optionalLbl"/>
<p:calendar id="birthDate" value="#{customer.birthDate}"
showOn="button" navigator="true"/>
<h:panelGroup/>
<p:commandButton value="Submit" action="#{customerController.
saveCustomer}" ajax="false"/>
</h:panelGrid>
</p:panel>
</h:form>
</ui:define>
</ui:composition>
</html>
```

Etape 2 :

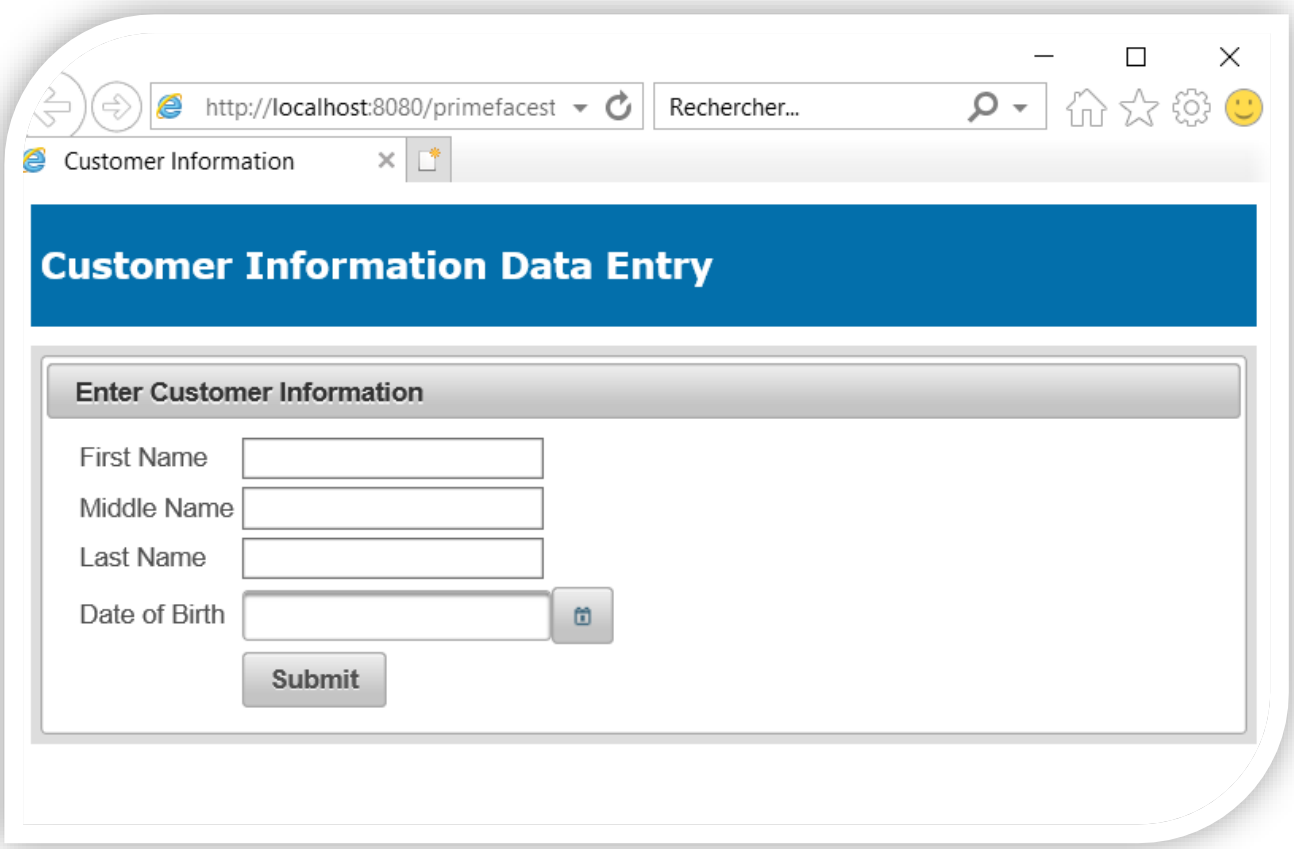
Crée **template.xhtml.**



On a utilisé le Template Facelets (**template.xhtml**) suivant, ce qui nous a permet d'obtenir de très bons styles CSS.

```xml
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:h="http://java.sun.com/jsf/html">
 <h:head>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 <h:outputStylesheet library="css" name="default.css"/>
 <h:outputStylesheet library="css" name="cssLayout.css"/>
 <h:outputStylesheet library="css" name="custom.css"/>
 <title>Customer Information</title>
 </h:head>
 <h:body>
 <div id="top" class="top">
 <ui:insert name="top"><h2>Customer Information</h2></ui:insert>
 </div>
 <div id="content" class="center_content">
 <ui:insert name="content">Content</ui:insert>
 </div>
 </h:body>
</html>
```

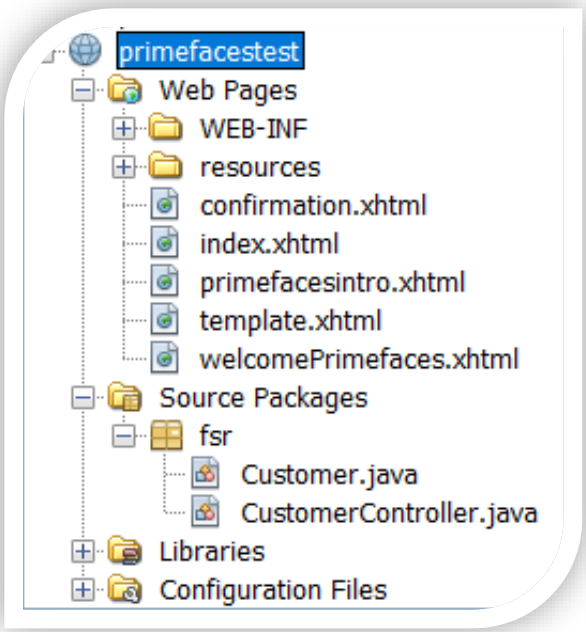
lorsque nous exécute le fichier **primefacesintro.xhtml**, le navigateur affiche le résultat suivant :



1.1. Les vues par onglets (Tabbed views)

En général, les formulaires HTML ont plusieurs champs ce qui complique la tâche de l'utilisateur. C'est pourquoi, on les divise en deux ou plusieurs onglets, pour alléger la page. Pour cet effet, PrimeFaces utilise un composant <p :tabView > qui permet de générer facilement des onglets. L'exemple suivant illustre comment utiliser ce composant:

La structure de projet est comme suit :



Index.xhtml :

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:p="http://primefaces.org/ui"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
xmlns:f="http://xmlns.jcp.org/jsf/core">
<ui:composition template="./template.xhtml">
<ui:define name="top">
<h2>Customer Information Data Entry</h2>
</ui:define>
<ui:define name="content">
<h:form>
<p:messages/>
<h:panelGrid columns="1" style="width: 100%">
<p:tabView>
<p:tab title="Personal Information">
<h:panelGrid columns="2">
<h:outputLabel for="firstName" value="First Name"
styleClass="requiredLbl"/>
<h:inputText id="firstName" label="First Name"
value="#{customer.firstName}" required="true"/>
<h:outputLabel for="middleName" value="Middle Name"
styleClass="optionalLbl"/>
<h:inputText id="middleName" label="Middle Name"
value="#{customer.middleName}"/>

<h:outputLabel for="lastName" value="Last Name"
styleClass="requiredLbl"/>
<h:inputText id="lastName" label="Last Name"
value="#{customer.lastName}" required="true"/>
<h:outputLabel for="birthDate" value="Date of Birth"
styleClass="optionalLbl"/>
<p:calendar id="birthDate" value="#{customer.birthDate}"
showOn="button" navigator="true"/>
</h:panelGrid>
</p:tab>
<p:tab title="Address">
<h:panelGrid columns="2">
<h:outputLabel for="line1" value="Line 1"
styleClass="requiredLbl"/>
<h:inputText id="line1" value="#{customer.addrLine1}"
required="true"/>
<h:outputLabel for="line2" value="Line 2"
styleClass="optionalLbl"/>
<h:inputText id="line2" value="#{customer.addrLine2}"/
<h:outputLabel for="city" value="City"
```

```
yleClass="requiredLbl"/>
<h:inputText id="city" value="#{customer.addrCity}"
required="true"/>
<h:outputLabel for="state" value="State"
styleClass="requiredLbl"/>
<h:selectOneMenu id="state" required="true"
value="#{customer.addrState}">
<f:selectItem itemValue="" itemLabel=""/>
<f:selectItem itemValue="AL" itemLabel="Alabama"/>
<f:selectItem itemValue="AK" itemLabel="Alaska"/>
<f:selectItem itemValue="AZ" itemLabel="Arizona"/>
<f:selectItem itemValue="AR" itemLabel="Arkansas"/>
<!-- other states omitted for brevity -->
</h:selectOneMenu>
<h:outputLabel for="zip" value="Zip"
styleClass="requiredLbl"/>
<h:inputText id="zip" value="#{customer.addrZip}"
required="true"/>
</h:panelGrid>
</p:tab>
<p:tab title="Phone Numbers">
<h:panelGrid columns="2">
<h:outputLabel for="homePhone" value="Home"/>
<p:inputMask id="homePhone" mask="(99)-99-99-99-99"
value="#{customer.homePhone}" size="14"
styleClass="optionalLbl"/>
<h:outputLabel for="mobilePhone" value="Mobile"/>
<p:inputMask id="mobilePhone" mask="(99)-99-99-99-99"
value="#{customer.mobilePhone}" size="14"
styleClass="optionalLbl"/>

<h:outputLabel for="workPhone" value="Work"/>
<p:inputMask id="workPhone" mask="(99)-99-99-99-99"
value="#{customer.workPhone}" size="14"
styleClass="optionalLbl"/>
</h:panelGrid>
</p:tab>
</p:tabView>
<p:commandButton value="Submit"
action="#{customerController.saveCustomer}" ajax="false"/>
</h:panelGrid>
</h:form>
</ui:define>
</ui:composition>
</html>
```

Customer.java

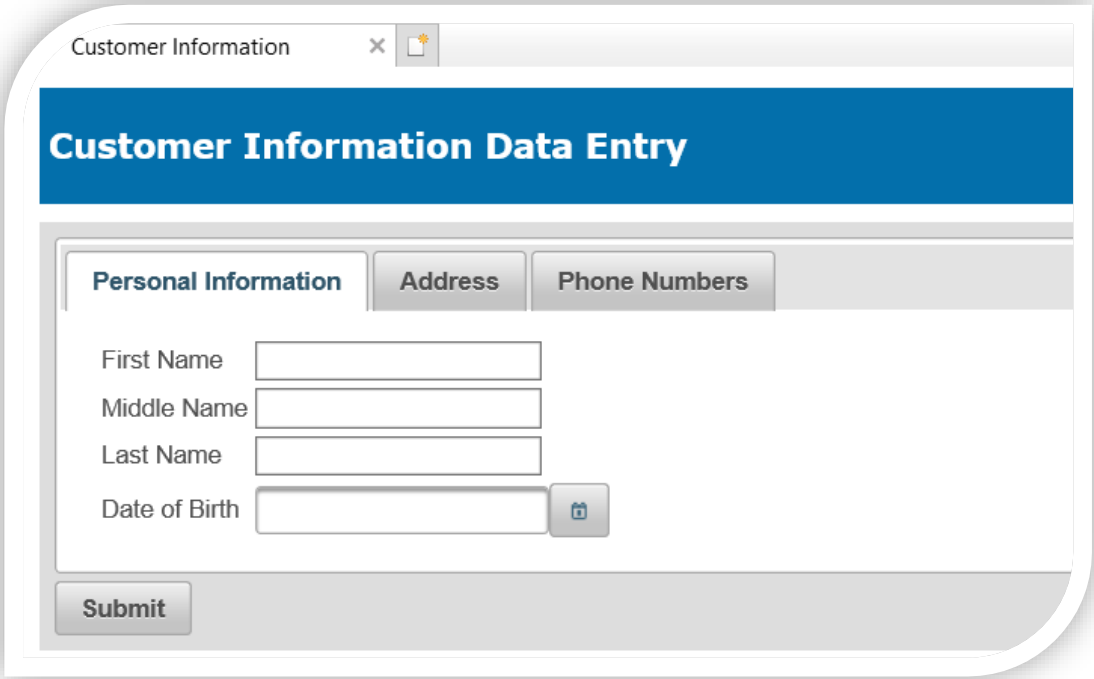
```
kage fsr;
mport java.io.Serializable;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
@SessionScoped
@Named("customer")
public class Customer implements Serializable {
 private String firstName;
 private String middleName;
 private String lastName;
 private Date birthDate;
 private SimpleDateFormat sdf = new
SimpleDateFormat("MM/dd/yyyy");
 private String addrLine1;
 private String addrLine2;
 private String addrCity;
 private String addrState;
 private String addrZip;
 private String homePhone;
 private String mobilePhone;
 private String workPhone;
 public Date getBirthDate() {
 return birthDate;
 }
```

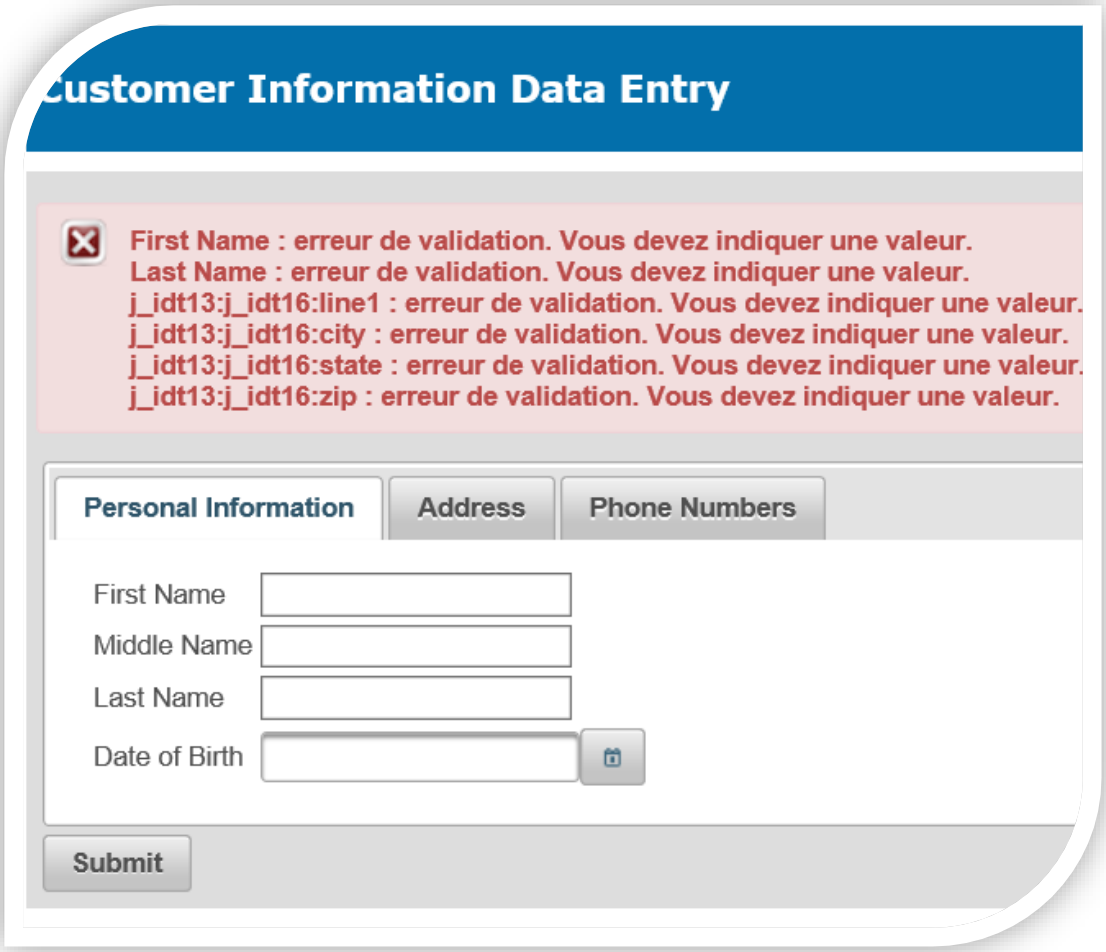
//ajouter getter & setter

CustomerControler.java

```java
package fsr;
import java.io.Serializable;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
@SessionScoped
@Named("customerController")
public class CustomerController implements Serializable {

    /** Creates a new instance of CustomerController */
    public CustomerController() {
    }
    public String saveCustomer() {
    return "confirmation";
    }
}
```

Lorsque nous exécute le projet, le navigateur affiche le résultat suivant :





Assistant des interfaces

Les assistants sont utiles chaque fois qu'on a besoin des utilisateurs pour remplir des champs de saisie dans un ordre spécifique. Dans l'exemple précédent, nous n'avions aucun moyen de forcer l'utilisateur à saisir l'adresse avant le numéro de téléphone. Cela nous empêche de valider les numéros de téléphone saisis correspondant bien à la zone géographique de l'adresse. Pour résoudre ce problème, on peut utiliser un assistant d'interfaces, lequel peut être réalisé facilement avec l'élément <p :wizard> de PrimesFaces.

Le code suivant montre comment utiliser ce composant.

Index.xhtml

```
    rsion='1.0' encoding='UTF-8' ?>
 TYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 ttp://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
html xmlns="http://www.w3.org/1999/xhtml"
xmlns:p="http://primefaces.org/ui"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core">
<ui:composition template="./template.xhtml">
<ui:define name="top">
<h2>Customer Information Data Entry</h2>
</ui:define>
<ui:define name="content">
<h:form id="form">
<h:panelGrid columns="1" style="width: 100%">
<p:wizard id="wizard">
<p:tab title="Personal Information" id="personalInfo">
<p:panel header="Personal Information">
<p:messages/>
<h:panelGrid columns="2">
<h:outputLabel for="firstName" value="First Name" styleClass="requiredLbl"/>
<h:inputText id="firstName" label="First Name" value="#{customer.firstName}" required="true"/>
<h:outputLabel for="middleName" value="Middle Name" styleClass="optionalLbl"/>
<h:inputText id="middleName" label="Middle Name" value="#{customer.middleName}"/>
<h:outputLabel for="lastName" value="Last Name" styleClass="requiredLbl"/>
<h:inputText id="lastName" label="Last Name" value="#{customer.lastName}" required="true"/>
<h:outputLabel for="birthDate" value="Date of Birth" styleClass="optionalLbl"/>
<p:calendar id="birthDate" value="#{customer.birthDate}" showOn="button" navigator="true"/>
</h:panelGrid>
</p:panel>
</p:tab>
<p:tab title="Phone Numbers" id="phone">
<p:panel header="Phone Numbers">
<p:messages/>
<h:panelGrid columns="2">
<h:outputLabel for="homePhone" value="Home"/>
<p:inputMask id="homePhone" mask="(99)-99-99-99" value="#{customer.homePhone}" size="12" styleClass="optionalLbl"/>
<h:outputLabel for="mobilePhone" value="Mobile"/>
<p:inputMask id="mobilePhone" mask="(99)-99-99-99" value="#{customer.mobilePhone}" size="12" styleClass="optionalLbl"/>
<h:outputLabel for="workPhone" value="Work"/>
<p:inputMask id="workPhone" mask="(99)-99-99-99" value="#{customer.workPhone}" size="12" styleClass="optionalLbl"/>
</h:panelGrid>
</p:panel>
</p:tab>
```

```
 title="Confirmation" id="confirmation">
:messages/>
<h:panelGrid columns="2">
<h:outputText value="First Name" styleClass="optionalLbl"/>
<h:outputText id="firstNameTxt" value="#{customer.firstName}" />
<h:outputText value="Middle Name" styleClass="optionalLbl"/>
<h:outputText id="middleNameTxt" value="#{customer.middleName}"/>
<h:outputText value="Last Name" styleClass="optionalLbl"/>
<h:outputText id="lastNameTxt" value="#{customer.lastName}" />
<h:outputText value="Date of Birth" styleClass="optionalLbl"/>
<h:outputText id="birthDateTxt" value="#{customer.formattedBirthDate}"/>
<h:outputText value="Home Phone" styleClass="optionalLbl"/>
<h:outputText id="homePhoneTxt" value="#{customer.homePhone}" />
<h:outputText value="Mobile Phone" styleClass="optionalLbl"/>
<h:outputText id="mobilePhoneTxt" value="#{customer.mobilePhone}" />
<h:outputText value="Work Phone" styleClass="optionalLbl"/>
<h:outputText id="workPhoneTxt" value="#{customer.workPhone}" />
<h:panelGroup/>
<h:inputHidden value="#{customer.addrState}"/>
<p:commandButton id="submitButton" value="Submit" actionListener="#{customerController.saveCustomer}" ajax="false"/>
</h:panelGrid>
</p:tab>
</p:wizard>
</h:panelGrid>
</h:form>
</ui:define>
</ui:composition>
</html>
```

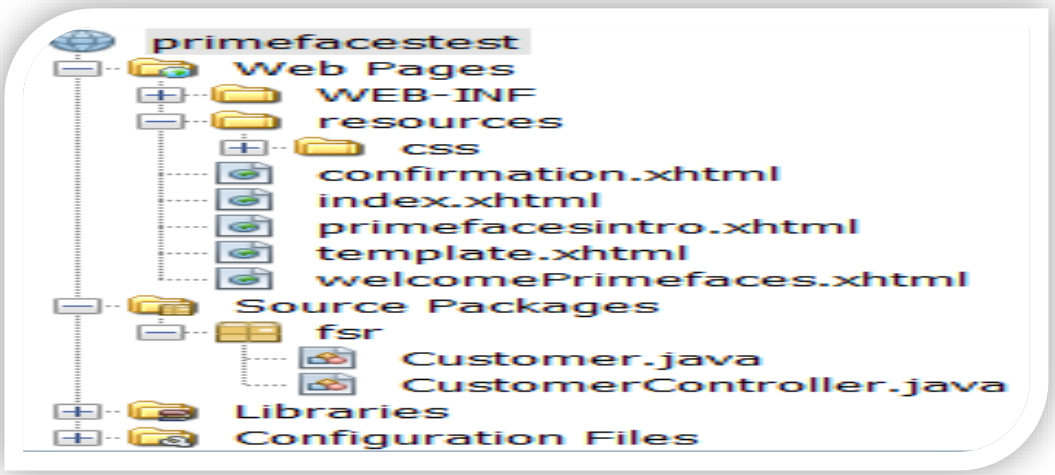
CustomerController.java

```
ckage fsr;
import java.io.Serializable;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
import javax.faces.event.ActionEvent;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
@SessionScoped
@Named("customerController")
public class CustomerController implements Serializable {
 /** Creates a new instance of CustomerController */
 public CustomerController() {
 }
 public void saveCustomer(ActionEvent actionEvent) {
 //In a real application, we would save the data,
 //In this example we simply show a message.
 FacesMessage facesMessage = new FacesMessage("Data SavedSuccessfully");

 facesMessage.setSeverity(FacesMessage.SEVERITY_INFO);
 FacesContext.getCurrentInstance().addMessage(null,
facesMessage);
 }
}
```

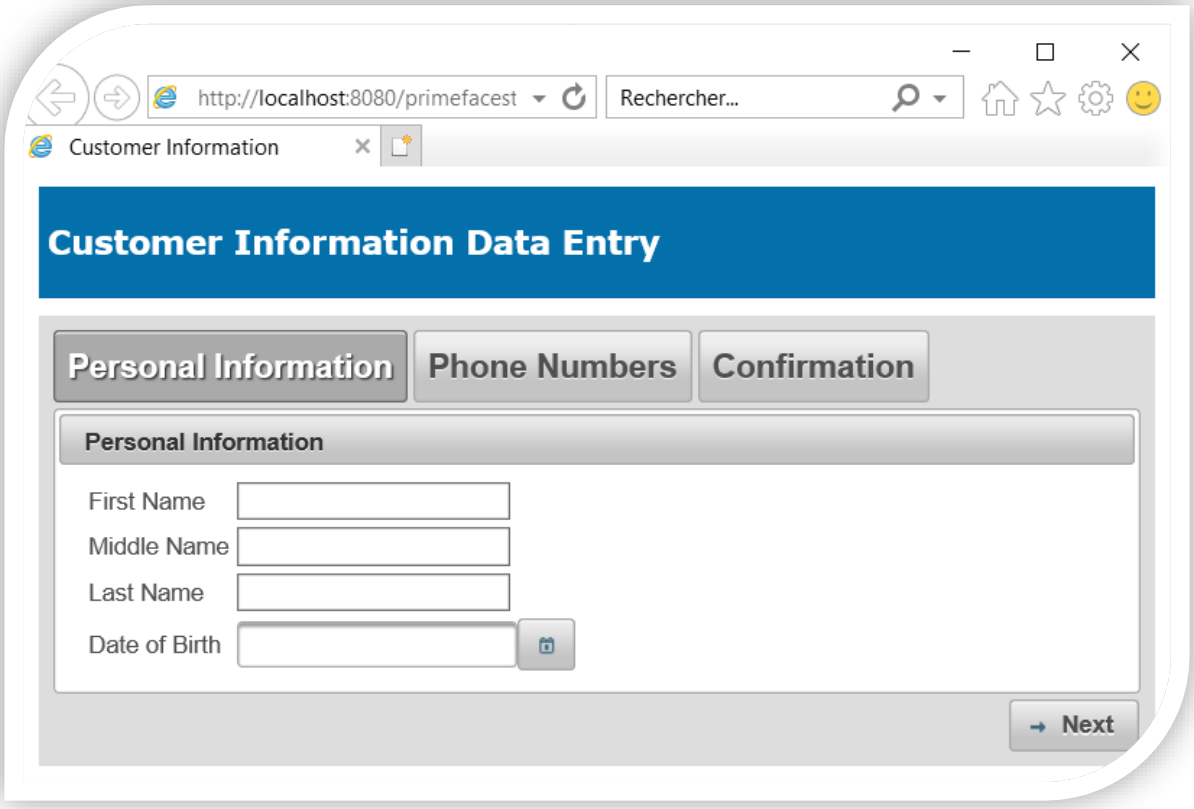
Confoirmation.xhtml

```
<ns="http://www.w3.org/1999/xhtml"
 s:ui="http://java.sun.com/jsf/facelets"
 lns:p="http://primefaces.org/ui"
 xmlns:h="http://java.sun.com/jsf/html"
 xmlns:f="http://java.sun.com/jsf/core">
 <body>
 <ui:composition template="./template.xhtml">
 <ui:define name="top">
 <h2>Customer Information Confirmation</h2>
 </ui:define>
 <ui:define name="content">
 <p:panel header="Customer Information">
 <h:panelGrid columns="2">
 <h:outputText value="First Name" styleClass="optionalLbl"/>
 <h:outputText id="firstName" value="#{customer.firstName}" />
 <h:outputText value="Middle Name" styleClass="optionalLbl"/>
 <h:outputText id="middleName" value="#{customer.middleName}"/>
 <h:outputText value="Last Name" styleClass="optionalLbl"/>
 <h:outputText id="lastName" value="#{customer.lastName}" />
 <h:outputText value="Date of Birth" styleClass="optionalLbl"/>
<h:outputText id="birthDate" value="#{customer.formattedBirthDate}" />
 <h:outputText value="Address" styleClass="optionalLbl"/>
 <h:panelGroup>
 <h:outputText value="#{customer.addrLine1}"/>
 <h:outputText value="," rendered="#{empty customer.addrLine2}"/>
 <br/>
 <h:outputText value="#{customer.addrLine2}," rendered="#{not empty customer.addrLine2}"/>
<f:verbatim rendered="#{not empty customer.addrLine2}"><br/>
</f:verbatim>
 <h:outputText value="#{customer.addrCity}, #{customer.addrState}, #{customer.addrZip}" />
 </h:panelGroup>
 <h:outputText value="Home Phone" styleClass="optionalLbl"/>
 <h:outputText id="homePhone" value="#{customer.homePhone}" />
 <h:outputText value="Mobile Phone" styleClass="optionalLbl"/>

 <h:outputText id="mobilePhone" value="#{customer.mobilePhone}" />
 <h:outputText value="Work Phone" styleClass="optionalLbl"/>
 <h:outputText id="homePhone" value="#{customer.workPhone}" />
 </h:panelGrid>
 </p:panel>
 </ui:define>
 </ui:composition>
 </body>
</html>
```

La structure de projet :

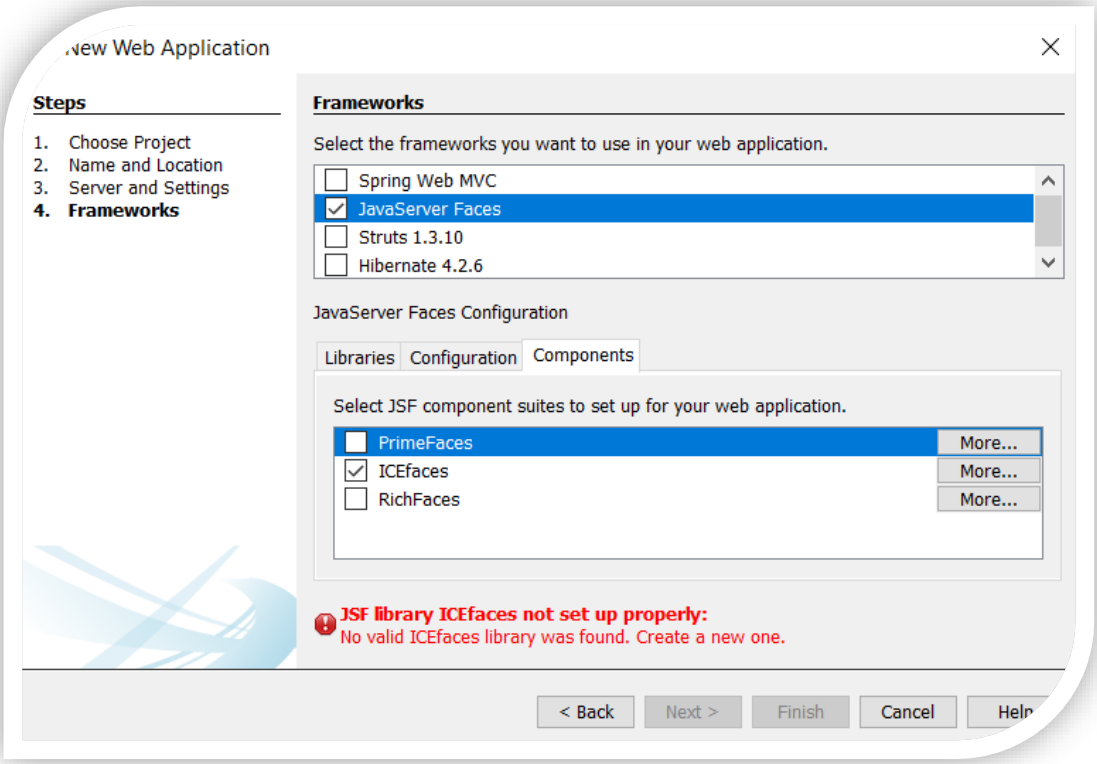


Les autres fichiers reste comme il est, et on exécute le projet, une fenêtre s'affiche comme suit :

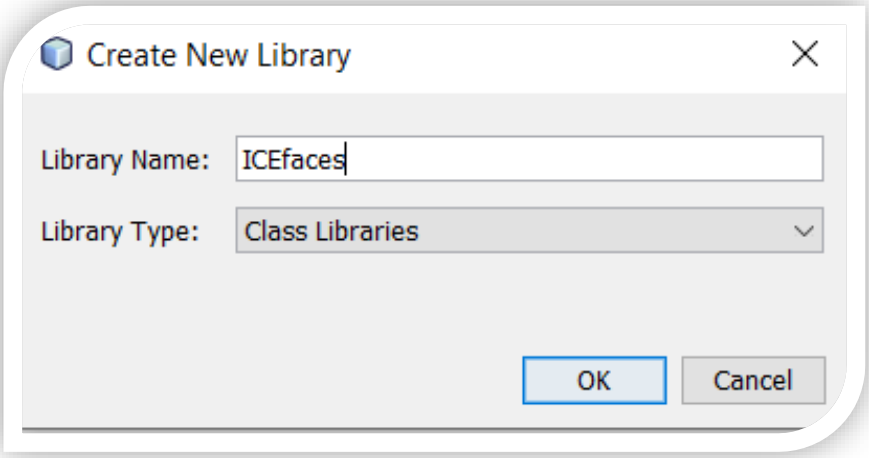


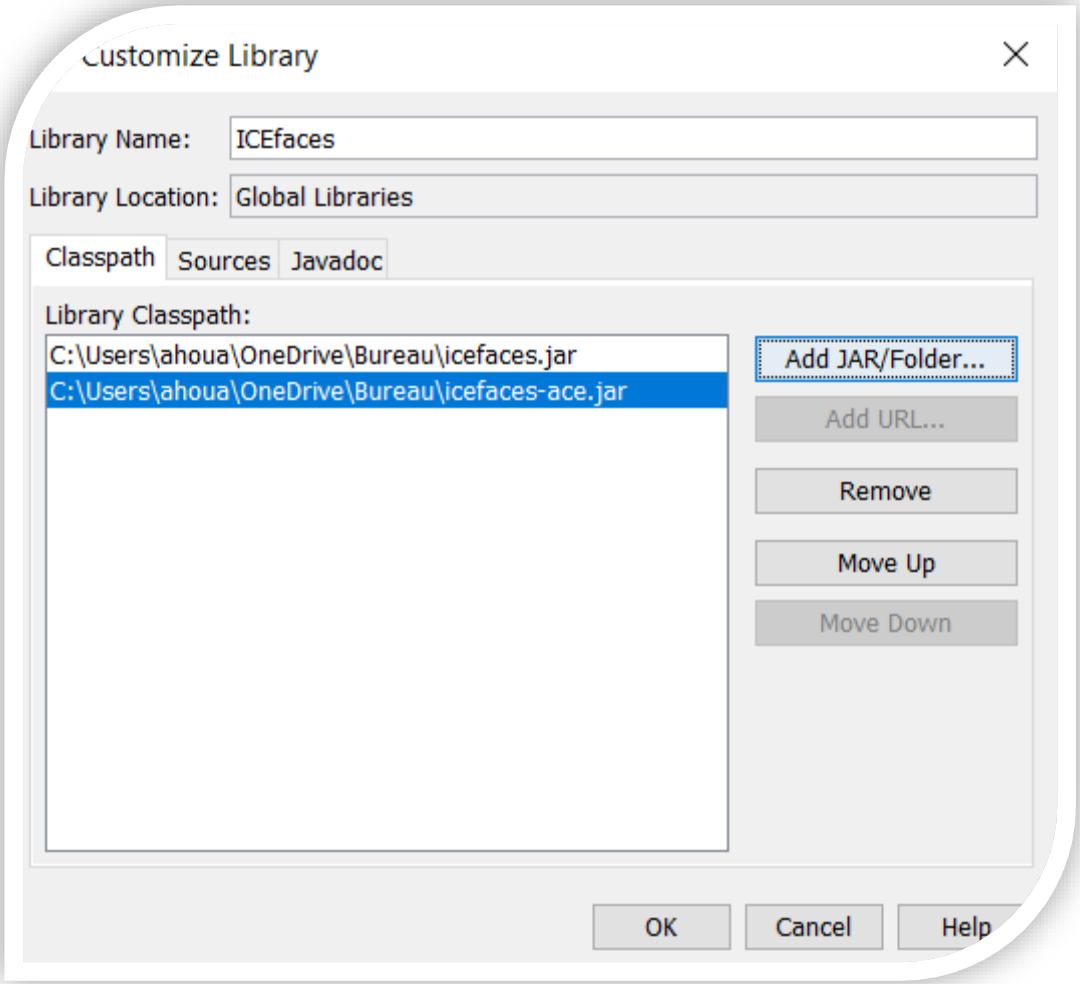
Les composants ICEfaces en JSF

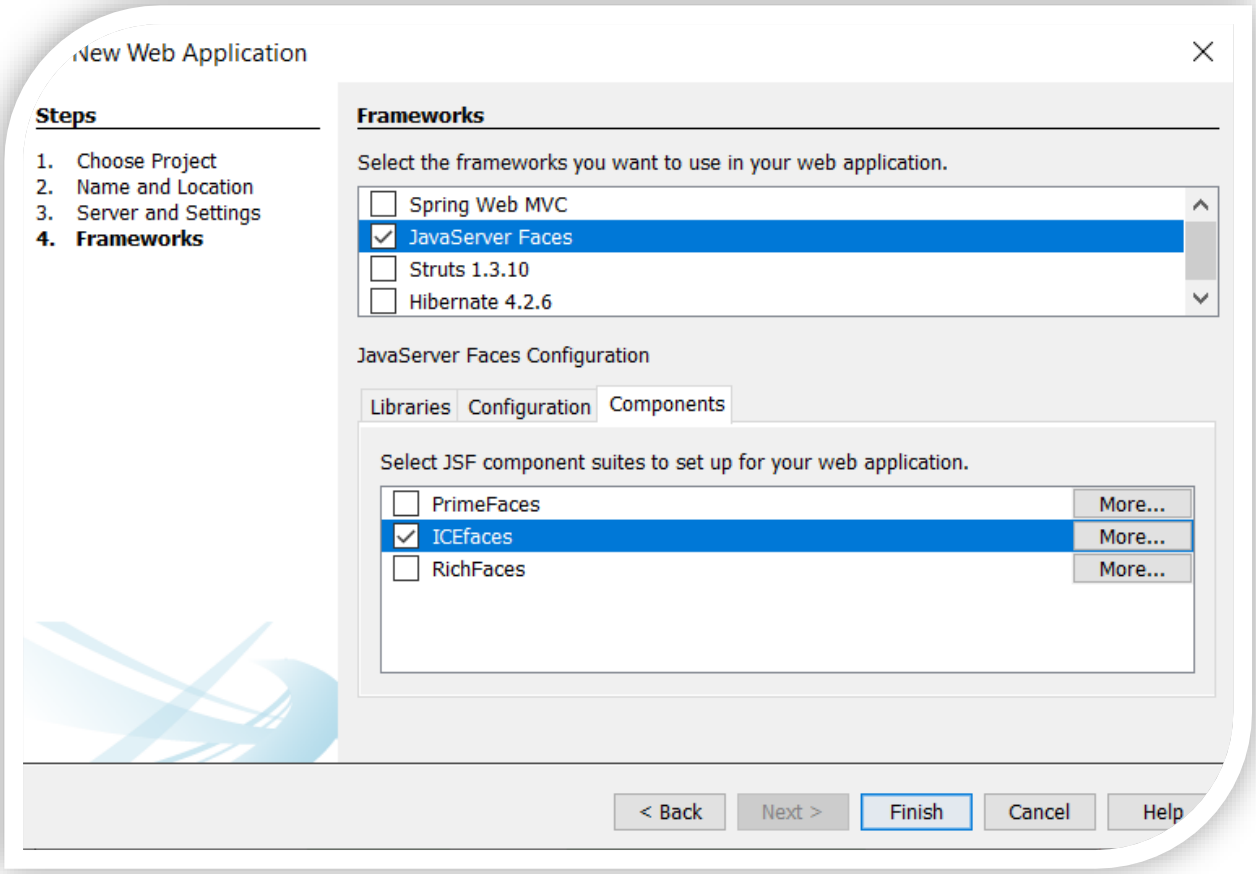
On crée un nouveau projet utilisont l'option ICEfaces



Contrairement à PrimeFacse, la librairie ICEfaces n'est pas intégrée dans NetBeans. Il faut la télécharger.




Customize Library
Library Name:
ICEfaces
Library Location:
Global Libraries
Classpath
Sources
Javadoc
Library Classpath:
C:\Users\ahoua\OneDrive\Bureau\icefaces.jar
C:\Users\ahoua\OneDrive\Bureau\icefaces-ace.jar
Add JAR/Folder...
Add URL...
Remove
Move Up
Move Down
OK
Cancel
Help


New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Frameworks
Select the frameworks you want to use in your web application.
Spring Web MVC
JavaServer Faces
Struts 1.3.10
Hibernate 4.2.6
JavaServer Faces Configuration
Libraries
Configuration
Components
Select JSF component suites to set up for your web application.
PrimeFaces
More...
ICEfaces
More...
RichFaces
More...
< Back
Next >
Finish
Cancel
Help