

Budowa aplikacji wielowarstwowych zastosowanie szablonów

Laboratorium 2

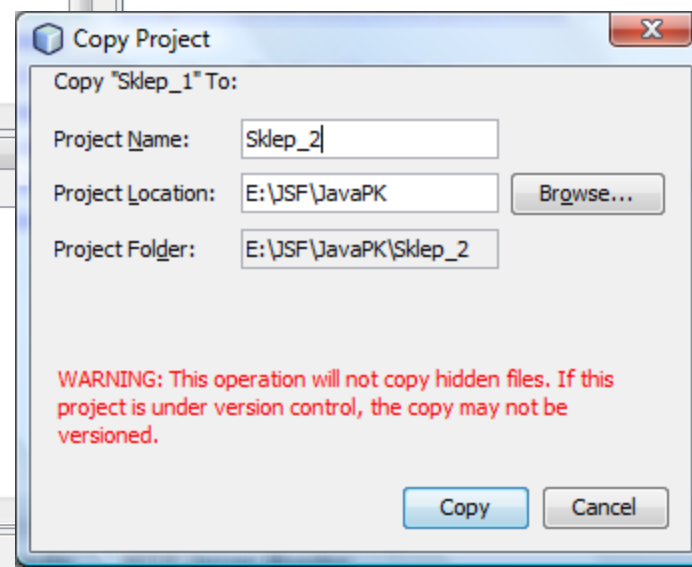
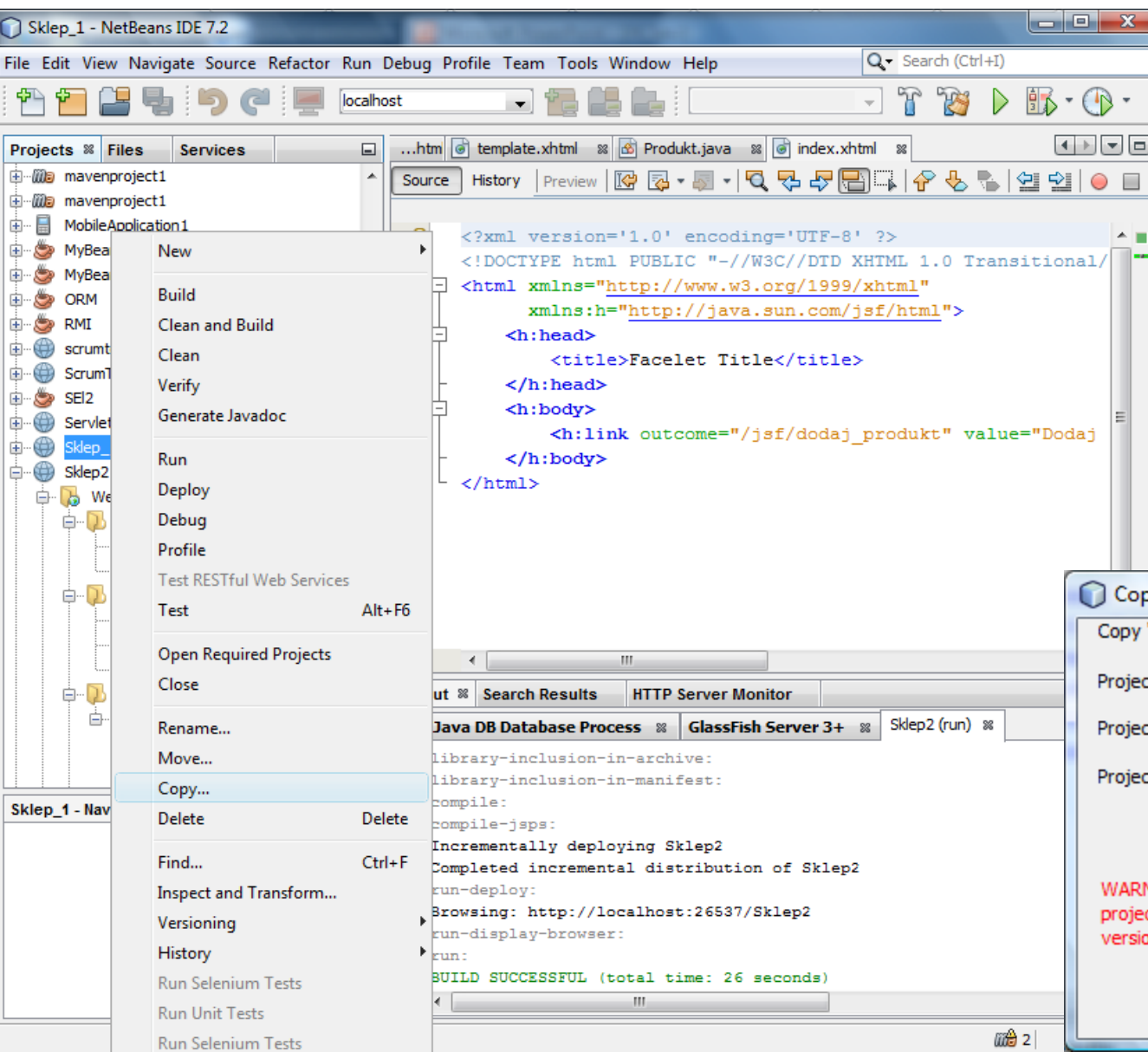
Programowanie komponentowe

Zofia Kruczkiewicz

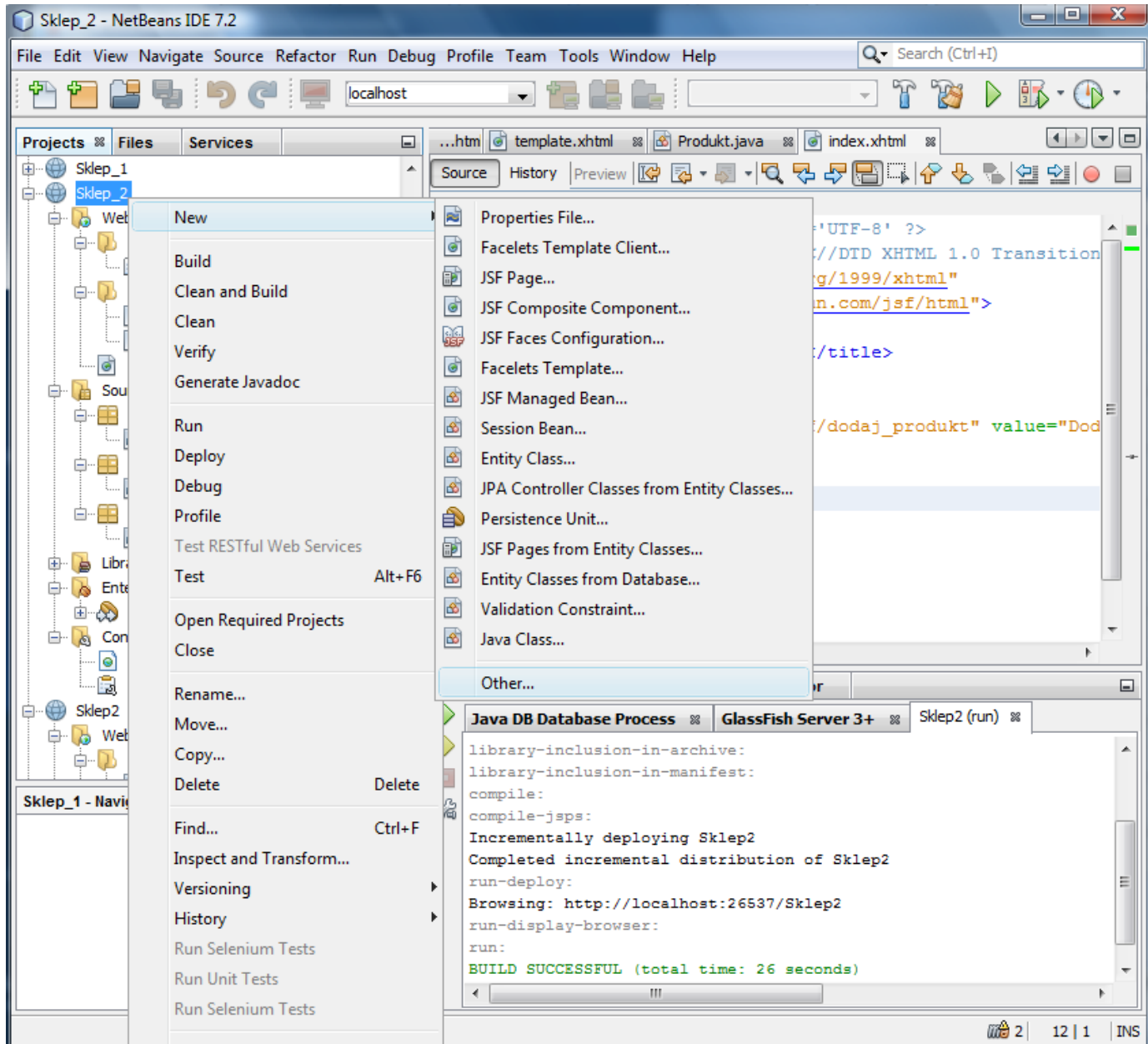
Przykład 1

Zastosowanie szablonów

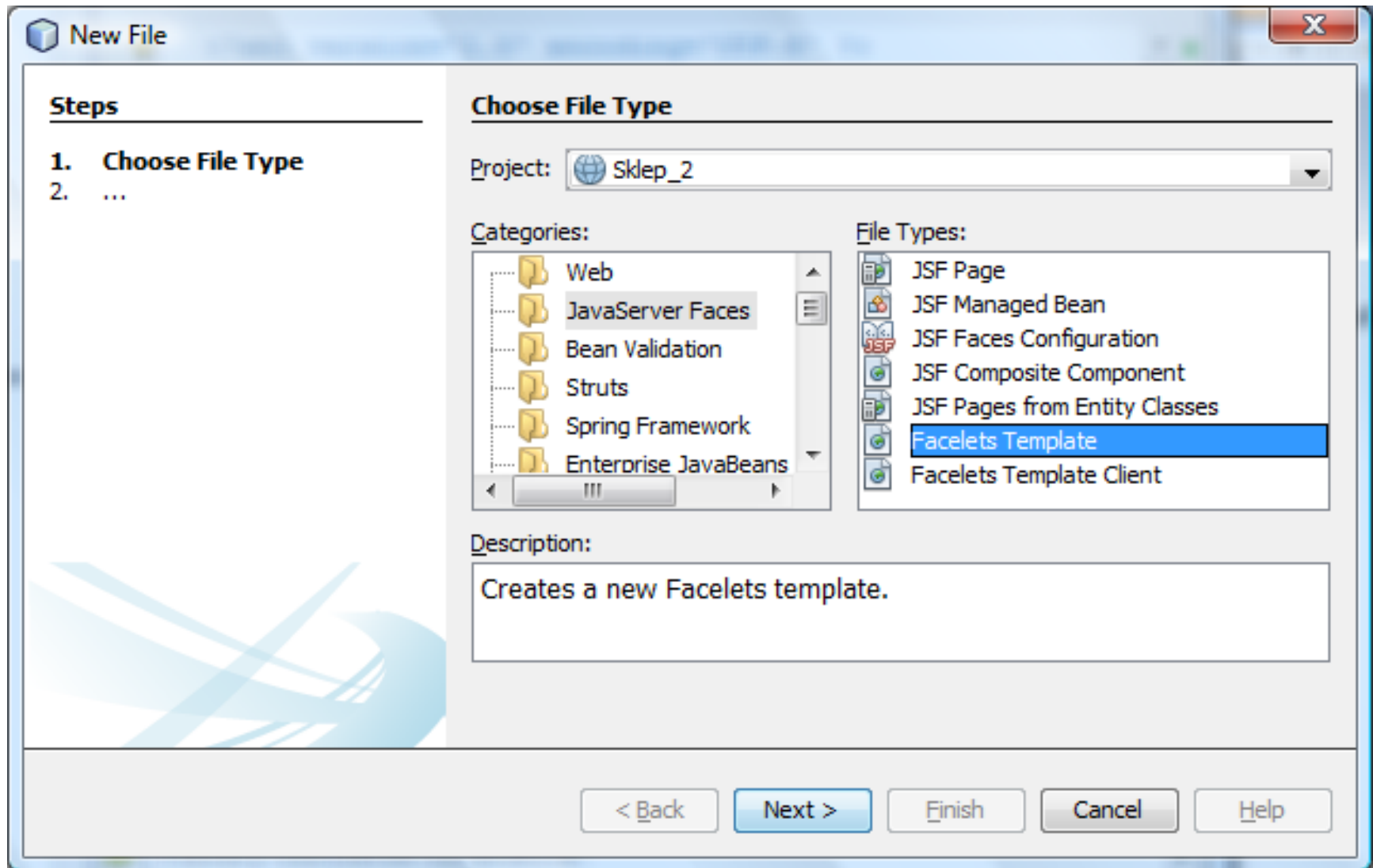
Tworzenie kopii projektu typu Web Application o nazwie Sklep_1 z lab1 (prawy klawisz myszy na nazwie projektu i wybór **Copy** – na formularzu kopiowania należy podać nową nazwę projektu **Sklep_2**. Projekt źródłowy Sklep_1 należy zamknąć, spakować do formatu zip lub rar i usunąć wersję niespakowaną.



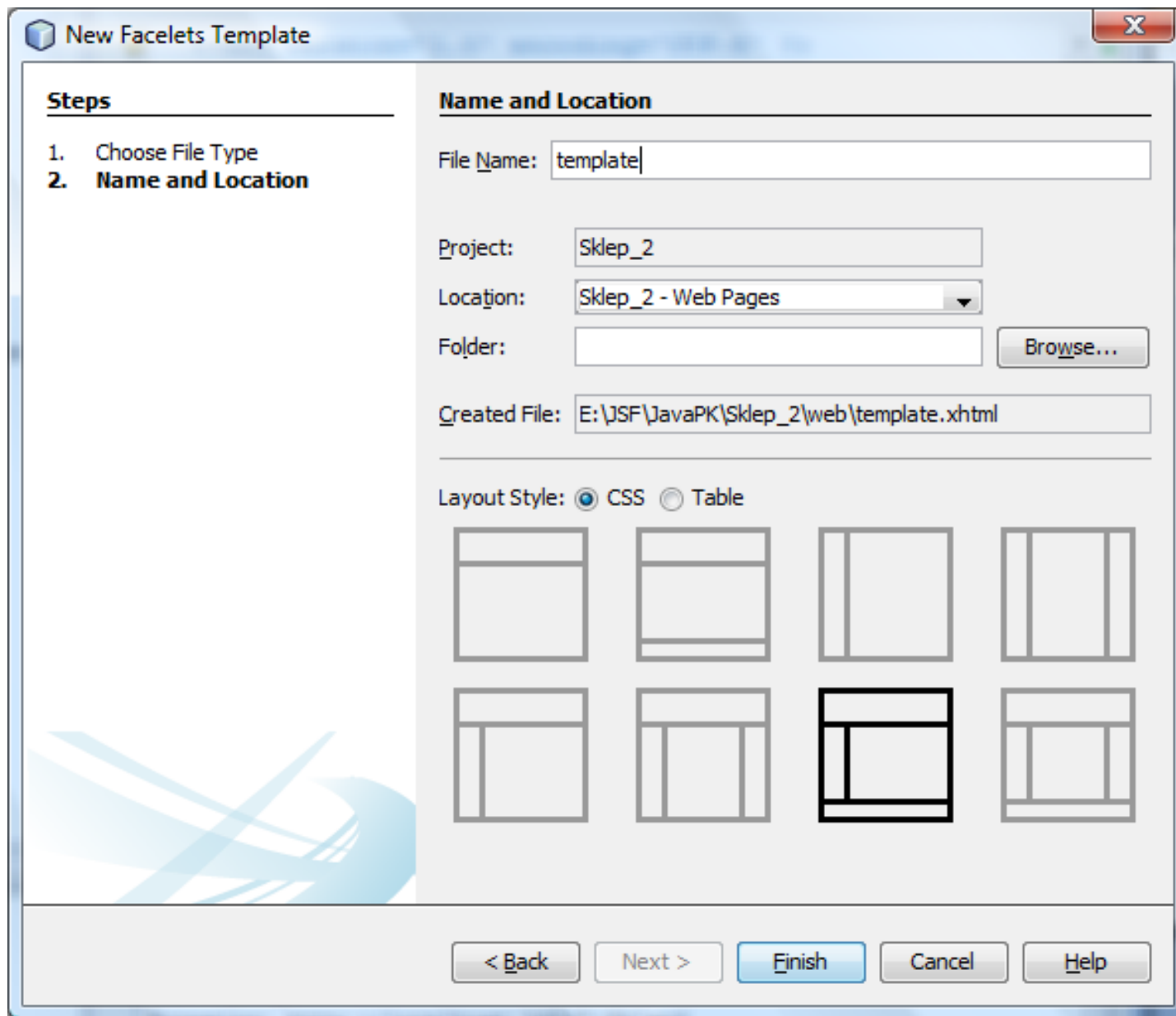
Do projektu Sklep_2 należy dodać plik **szablonu strony** - prawy klawisz myszy na nazwie projektu i wybór **New** oraz z kolejnej listy pozycji **Other**



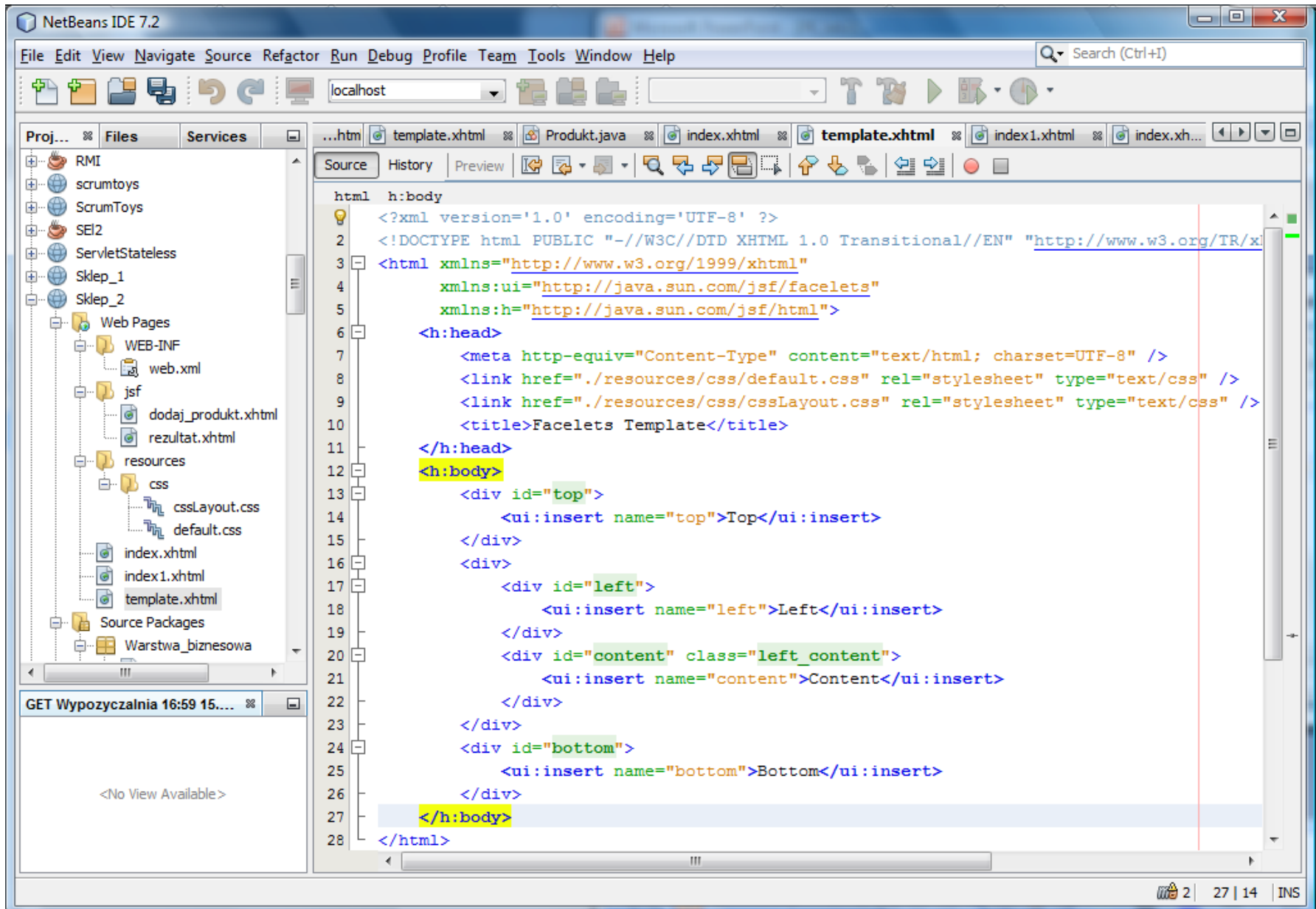
Wybór pliku **JavaServer Faces/Facelets Template** i **Next**



Nadać nazwę pliku template w polu **File Name** katalogu domyślnym **Web Pages** (puste pole Folder) - zaznaczyć wybrany Layout i **Finish**

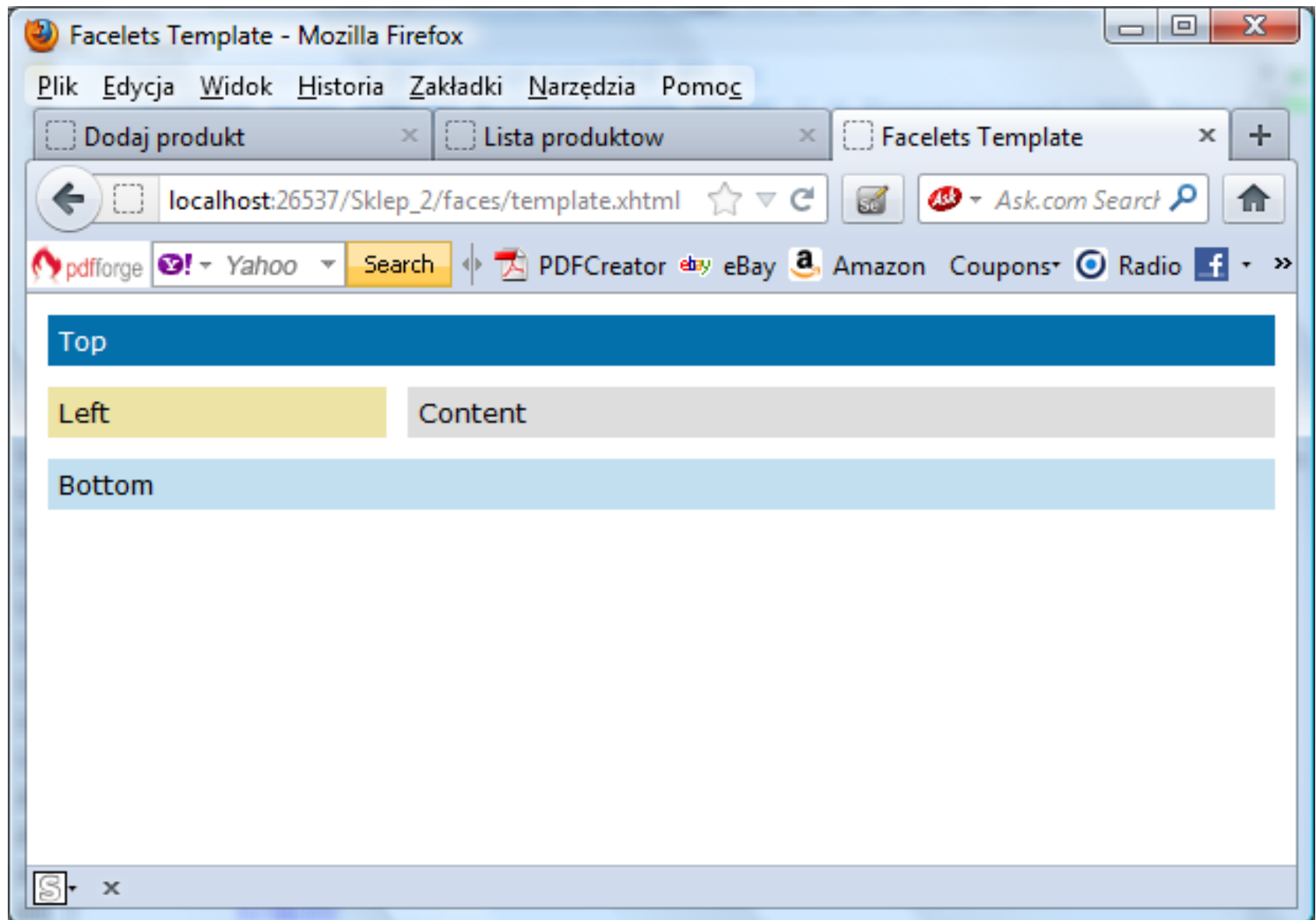


Otworzyć do edycji utworzony plik szablonu **template.xhtml** – znajduje się w katalogu domyślnym **Web Pages**. Plik zawiera cztery obszary **div** oraz 2 kaskadowe arkusze stylów w części **head**.



```
html h:body
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://java.sun.com/jsf/facelets"
5       xmlns:h="http://java.sun.com/jsf/html">
6   <h:head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8     <link href="./resources/css/default.css" rel="stylesheet" type="text/css" />
9     <link href="./resources/css/cssLayout.css" rel="stylesheet" type="text/css" />
10    <title>Facelets Template</title>
11  </h:head>
12  <h:body>
13    <div id="top">
14      <ui:insert name="top">Top</ui:insert>
15    </div>
16    <div>
17      <div id="left">
18        <ui:insert name="left">Left</ui:insert>
19      </div>
20      <div id="content" class="left_content">
21        <ui:insert name="content">Content</ui:insert>
22      </div>
23    </div>
24    <div id="bottom">
25      <ui:insert name="bottom">Bottom</ui:insert>
26    </div>
27  </h:body>
28 </html>
```

Widok strony szablonu po uruchomieniu aplikacji Sklep_2 i dopisaniu faces/template.xhtml do URL aplikacji i odświeżeniu strony



Modyfikacja zawartości pliku `template.xhtml` w zaznaczonych miejscach

The screenshot shows the NetBeans IDE 7.2 interface. The main editor displays the source code of `template.xhtml`. The code is as follows:

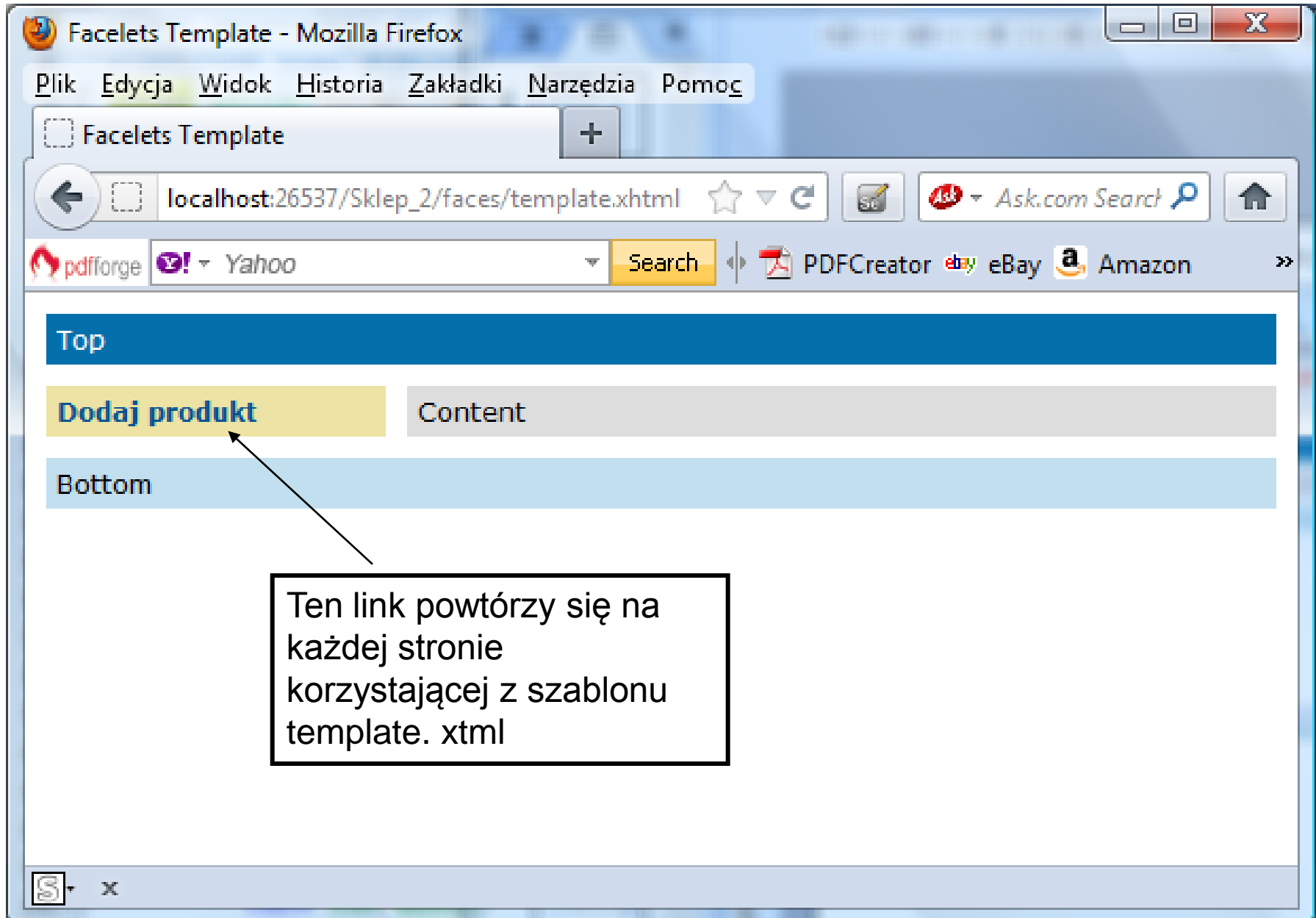
```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://java.sun.com/jsf/facelets"
5       xmlns:h="http://java.sun.com/jsf/html">
6   <h:head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8     <h:outputStylesheet name="css/default.css" />
9     <h:outputStylesheet name="css/cssLayout.css"/>
10    <title><ui:insert name="title">Facelets Template</ui:insert</title>
11  </h:head>
12  <h:body>
13    <div id="top">
14      <ui:insert name="top">Top</ui:insert>
15    </div>
16    <div id="left">
17      <h:link outcome="/faces/jsf/dodaj_produk" value="Dodaj produkt"/>
18    </div>
19    <div id="content" class="left_content">
20      <ui:insert name="content">Content</ui:insert>
21    </div>
22  </div>
23  <div id="bottom">
24    <ui:insert name="bottom">Bottom</ui:insert>
25  </div>
26 </h:body>
27 </html>
```

Annotations in the image:

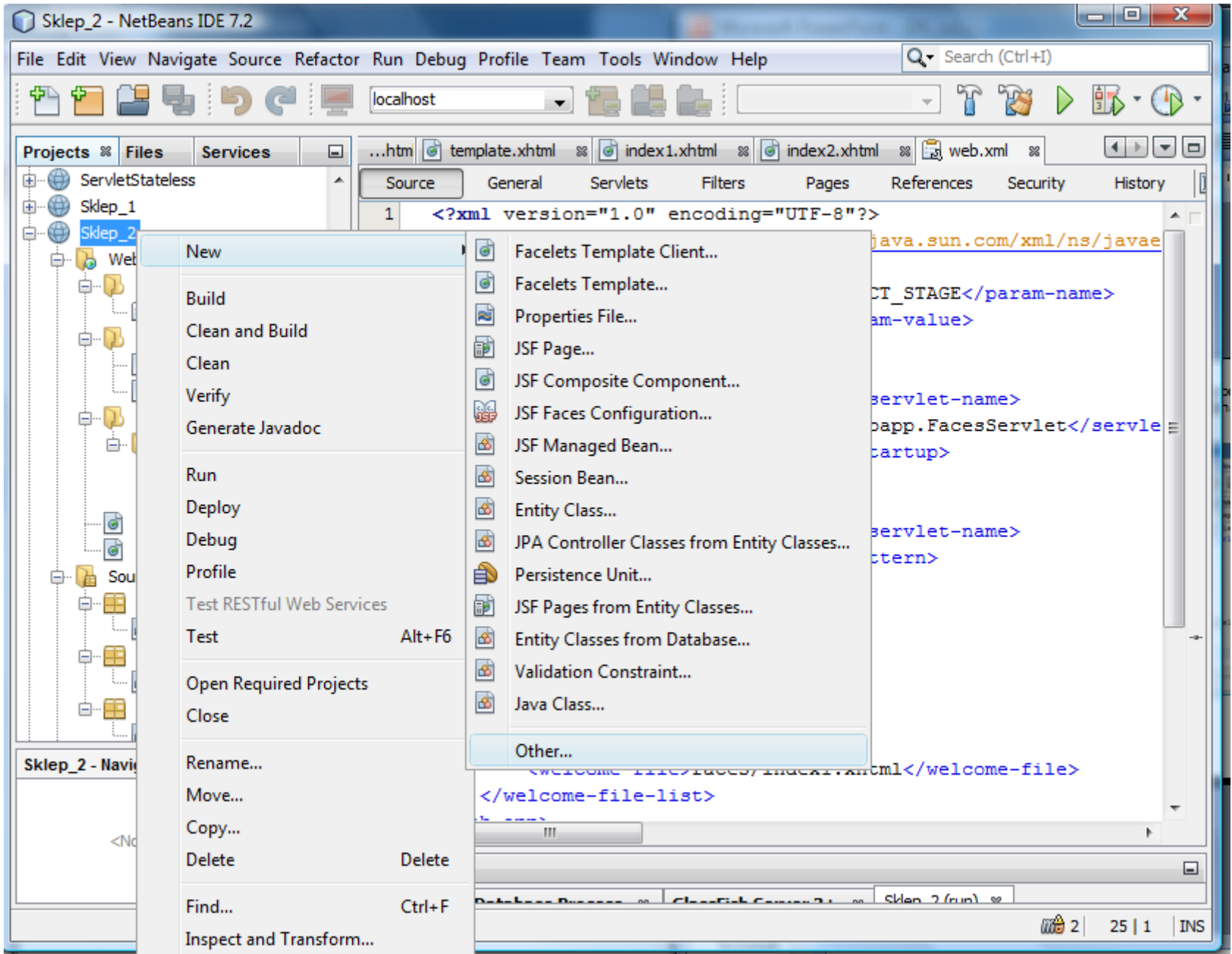
- Nagłówek strony**: Points to the `<h:head>` section (lines 6-11).
- Menu strony**: Points to the `<h:link>` element (line 17).
- Stopka strony**: Points to the `<div id="bottom">` section (lines 23-25).
- Robocza część strony**: Points to the `<div id="content">` section (lines 19-20).

The `template.xhtml` file is also visible in the Project Explorer on the left, under the `resources` folder.

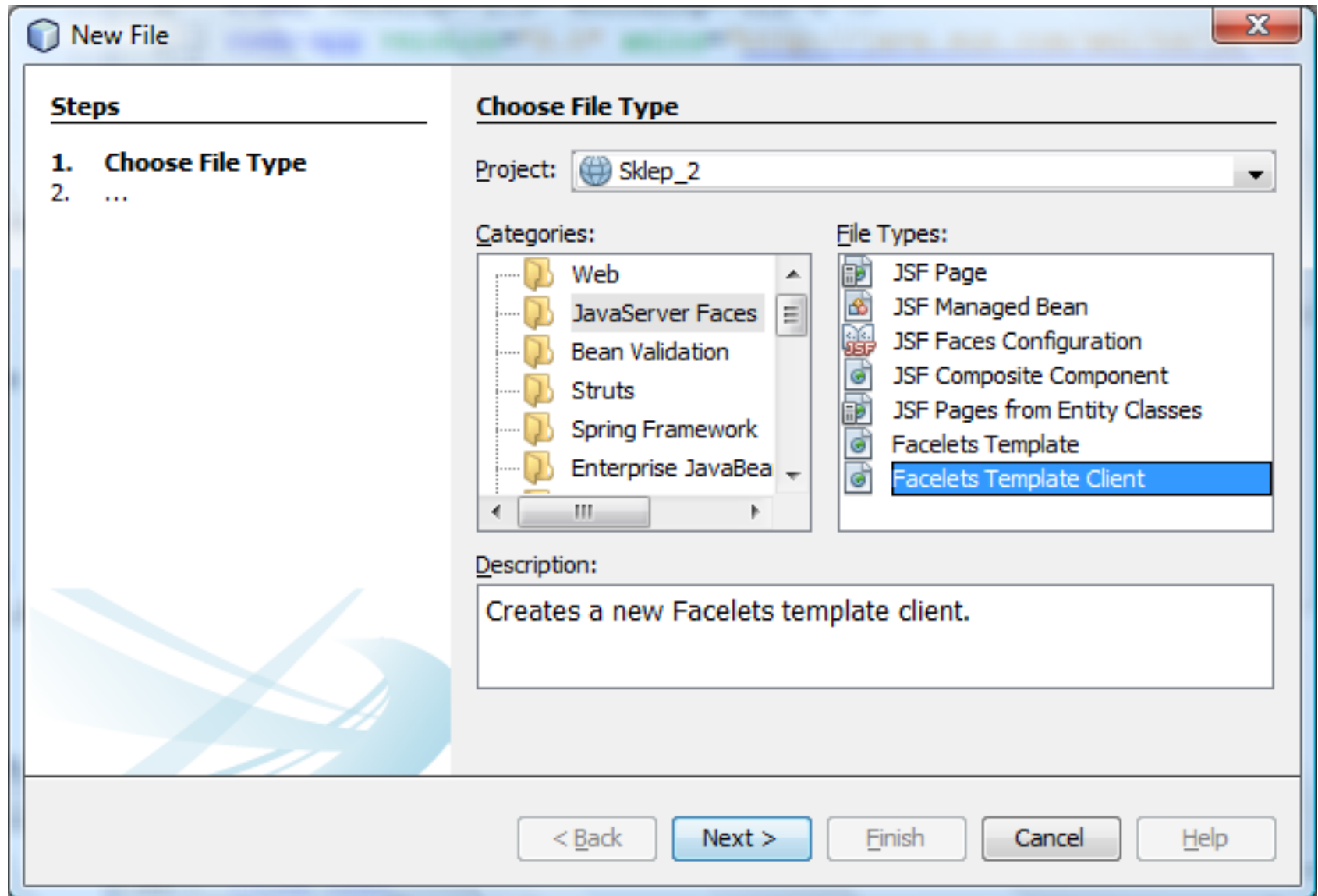
Widok strony szablonu po uruchomieniu aplikacji Sklep_2 i dopisaniu **faces/template.xhtml** do URL aplikacji i odświeżeniu strony



Do projektu Sklep_2 należy dodać plik strony głównej zbudowanej na podstawie **szablону strony** - prawy klawisz myszy na nazwie projektu i wybór **New** oraz z kolejnej listy pozycji **Other**



Wybór pliku **JavaServer Faces/Facelets Template Client** i **Next**



Podanie nazwy pliku **index1** w polu **File Name** w katalogu domyślnym (puste pole Folder) i wybór pliku **template.xhtml** w polu **Browse** pola **Template**. W nowym formularzu wybór w katalogu **Web Pages** pliku **template.xhtml**

New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

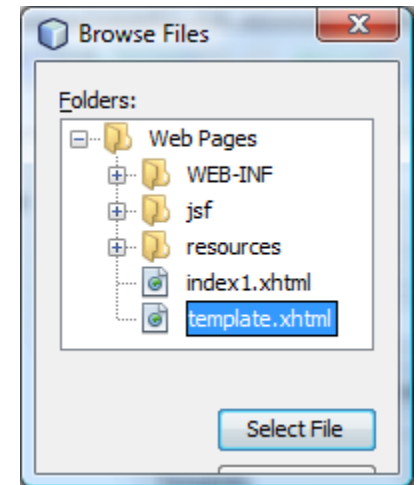
Folder:

Created File:

Template:

Generated Root Tag: <html>
 <ui:composition>

Select a template for which the client will be generated.



Utworzenie strony startowej **index1.xhtml** zbudowanej na szablonie strony **template.xhtml** - zatwierdzenie klawiszem **Finish**

New Facelets Template Client

Steps

1. Choose File Type
- 2. Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

Template:

Generated Root Tag: <html>
 <ui:composition>

Otworzyć w edytorze nową stronę startową **index1.xhtml**

The screenshot shows the NetBeans IDE interface. The main editor window displays the XML code for a new XHTML page named `index1.xhtml`. The code is as follows:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <body>
    <ui:composition template="./template.xhtml">
      <ui:define name="top">
        top
      </ui:define>
      <ui:define name="left">
        left
      </ui:define>
      <ui:define name="content">
        content
      </ui:define>
      <ui:define name="bottom">
        bottom
      </ui:define>
    </ui:composition>
  </body>
</html>
```

The left sidebar shows the project structure for `Sklep_2`, including `Web Pages`, `Source Packages`, and `Libraries`. The `index1.xhtml` file is highlighted in the `index1.xhtml - Navigator` pane. The bottom status bar shows the page number `22 | 1` and the text `INS`.

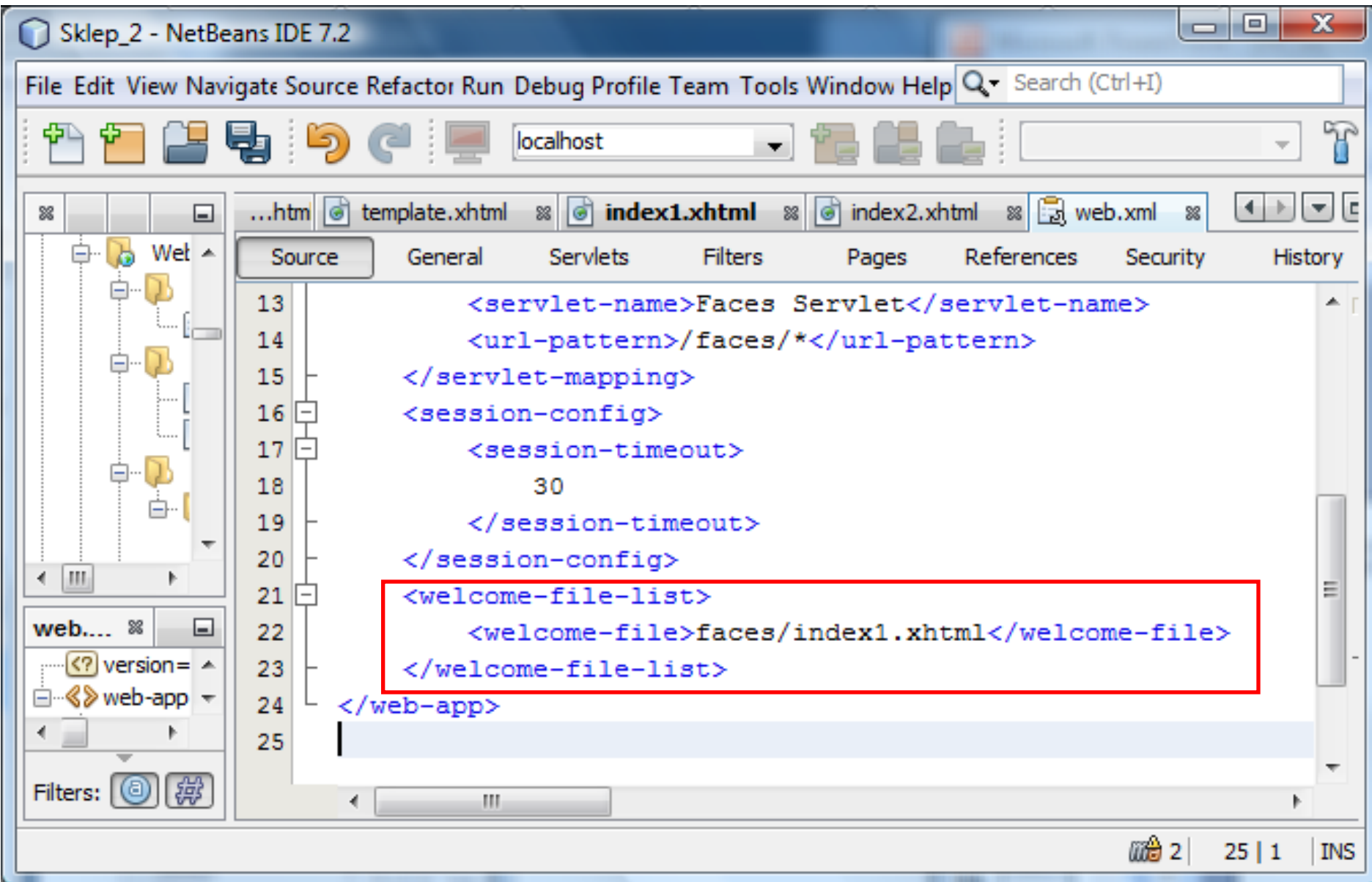
Usunąć wszystkie znaczniki w zakresie znacznika `<ui:composition>` i wstawić znacznik definiujący tytuł strony w zaznaczonym miejscu

The screenshot shows the NetBeans IDE 7.2 interface. The main editor window displays the source code of an XML file named `index1.xhtml`. The code is as follows:

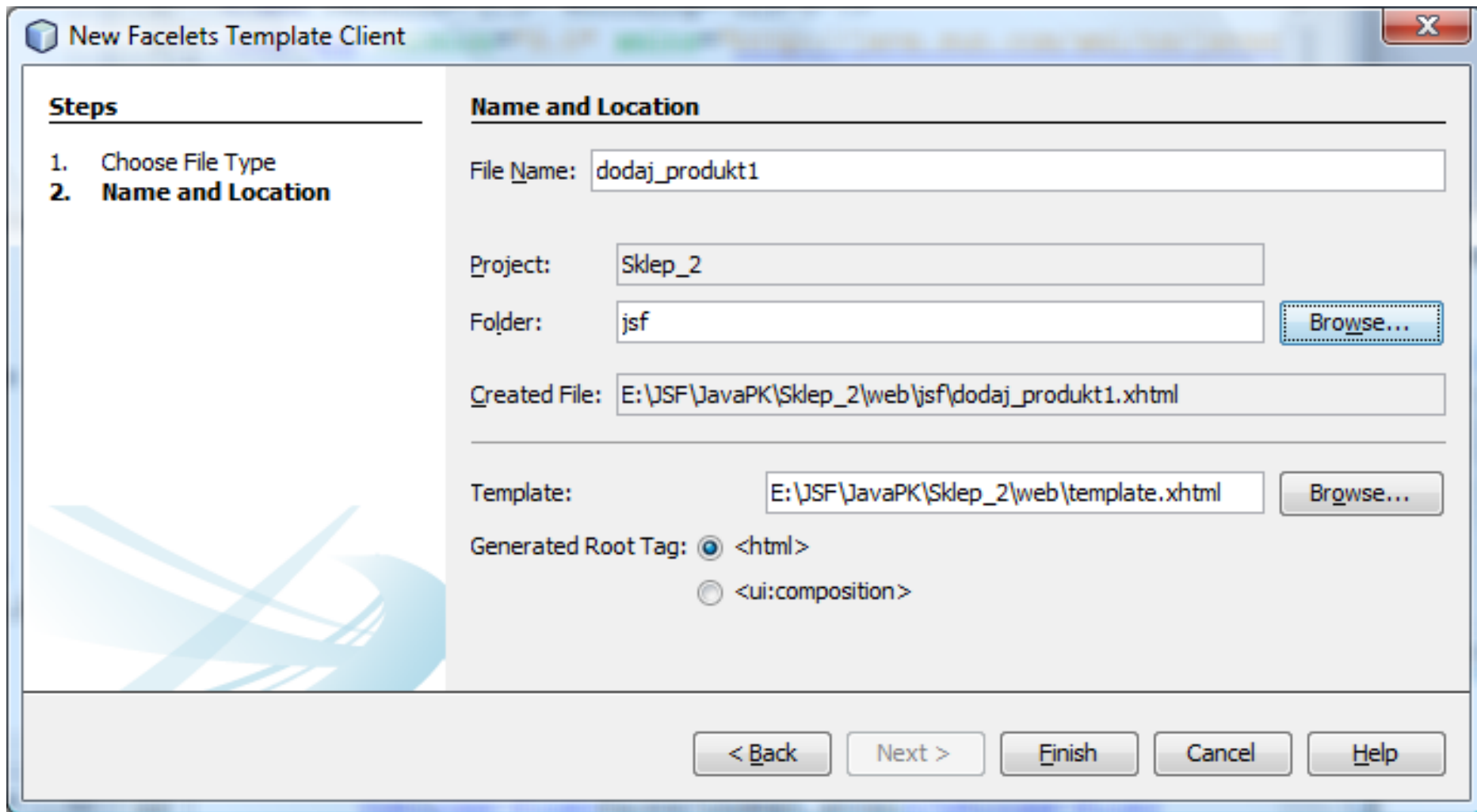
```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://java.sun.com/jsf/facelets">
5   <body>
6     <ui:composition template="./template.xhtml">
7       <ui:define name="title">
8         Sklep
9       </ui:define>
10    </ui:composition>
11  </body>
12 </html>
```

The `<ui:define name="title"> Sklep </ui:define>` block is highlighted with a red rectangular box. The left sidebar shows a project structure with folders like `jsf` and `resources`, and files like `index1.xhtml` and `template.xhtml`. The status bar at the bottom indicates "Sklep_2 deployed." and shows a page number of 12 out of 8.

Otworzyć plik deskryptora **web.xml** i zmienić nazwę strony startowej w zaznaczonym miejscu. Usunąć plik index.xhtml z projektu



Do projektu Sklep_2 należy dodać plik strony **dodaj_produkt1** zbudowanej na podstawie **szablonu strony** - prawy klawisz myszy na nazwie projektu i wybór **New** oraz z kolejnej listy pozycji **Other** i na kolejnej stronie wybrać **JavaServer Faces/Facelets Template Client...** W polu **File Name** kolejnego formularza wpisać **dodaj_produkt1**, w polu **Folder** wybrać katalog **jsf**, w polu **Browse** pola **Template** wybrać szablon **template.xhtml**.



New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

Template:

Generated Root Tag: `<html>`
 `<ui:composition>`

Zawartość strony **dodaj_produkt1** zawiera w znaczniku **content** zawartość znacznika **<h:form>** strony **dodaj_produkt z lab1**

```
<?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">
<body>
  <ui:composition template=" ../template.xhtml">
    <ui:define name="title">
      Dodaj produkt
    </ui:define>
    <ui:define name="content">
      <h:form>
        <h:panelGrid columns="2">
          <h:outputLabel value="Podaj nazwe produktu" for="nazwa" />
          <h:inputText
            id="nazwa"
            title="Podaj nazwe:"
            value="#{managed_produkt.nazwa}"
            required="true"
            requiredMessage="Blad: Podaj nazwe." >
          </h:inputText>
        </h:panelGrid>
      </h:form>
    </ui:define>
  </ui:composition>
</body>
</html>
```

```
<h:outputLabel value="Podaj cene netto produktu" for="cena" />
<h:inputText
    id="cena"
    title="Podaj cene:"
    value="#{managed_produkt.cena}"
    required="true"
    requiredMessage="Blad: Podaj cene." >
</h:inputText>
<h:outputLabel value="Podaj promocje produktu" for="promocja" />
<h:inputText
    id="promocja"
    title="Podaj promocje:"
    value="#{managed_produkt.promocja}"
    required="true"
    requiredMessage="Blad: Podaj promocje." >
</h:inputText>
</h:panelGrid>
<h:commandLink action="#{managed_produkt.dodaj_produkt}" value="OK" />
</h:form>
</ui:define>
</ui:composition>
</body>
</html>
```

Zmiana w pliku **template.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 name="css/default.css" />
    <h:outputStylesheet name="css/cssLayout.css"/>
    <title><ui:insert name="title">Facelets Template</ui:insert></title>
  </h:head>
  <h:body>
    <div id="top">
      <ui:insert name="top">Top</ui:insert>
    </div>
    <div>
      <div id="left">
        <h:link outcome="/faces/jsf/dodaj_produk1" value="Dodaj produkt"/>
      </div>
      <div id="content" class="left_content">
        <ui:insert name="content">Content</ui:insert>
      </div>
    </div>
    <div id="bottom">
      <ui:insert name="bottom">Bottom</ui:insert>
    </div>
  </h:body>
</html>
```

Należy zmodyfikować w pliku **template.xhtml** nazwę strony **dodaj_produk1** na **dodaj_produk1** w zaznaczonym znaczniku h:link

Do projektu Sklep_2 należy dodać plik strony **rezultat1** zbudowanej na podstawie **szablonu strony** - prawy klawisz myszy na nazwie projektu i wybór **New** oraz z kolejnej listy pozycji **Other** i na kolejnej stronie wybrać **JavaServer Faces/Facelets Template Client...** W polu **File Name** kolejnego formularza wpisać **rezultat1**, w polu **Folder** wybrać katalog **jsf**, w polu **Browse** pola **Template** wybrać szablon **template.xhtml**.

New Facelets Template Client

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: rezultat1

Project: Sklep_2

Folder: jsf [Browse...](#)

Created File: E:\JSF\JavaPK\Sklep_2\web\jsf\rezultat1.xhtml

Template: E:\JSF\JavaPK\Sklep_2\web\template.xhtml [Browse...](#)

Generated Root Tag: <html>
 <ui:composition>

< Back Next > Finish Cancel Help

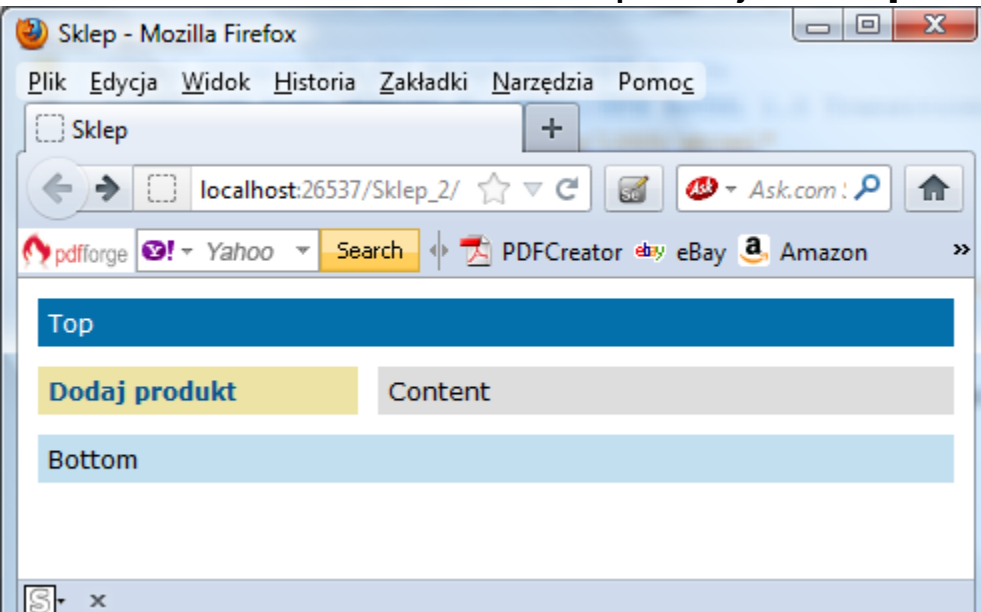
Zawartość strony **rezultat1** zawiera w znaczniku o nazwie **content** zawartość znacznika **<h:form>** strony **rezultat z lab1**

```
<?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">
<body>
  <ui:composition template=" ../template.xhtml">
    <ui:define name="title">
      Rezultat
    </ui:define>
    <ui:define name="content">
      <h:form>
        <h:panelGrid columns="2">
          <h:outputLabel value="Nazwa produktu" for="nazwa" />
          <h:outputText id="nazwa" value="#{managed_produkt.nazwa}"/>
          <h:outputLabel value="Cena produktu" for="cena" />
          <h:outputText id="cena" value="#{managed_produkt.cena}"/>
          <h:outputLabel value="Promocja produktu" for="promocja" />
          <h:outputText id="promocja" value="#{managed_produkt.promocja}"/>
          <h:outputLabel value="Cena brutto produktu" for="brutto" />
          <h:outputText id="brutto" value="#{managed_produkt.cena_brutto}" />
        </h:panelGrid>
        <h:commandButton id="powrot" value="Powrot" action="/faces/index1"/>
      </h:form>
    </ui:define>
  </ui:composition>
</body>
</html>
```

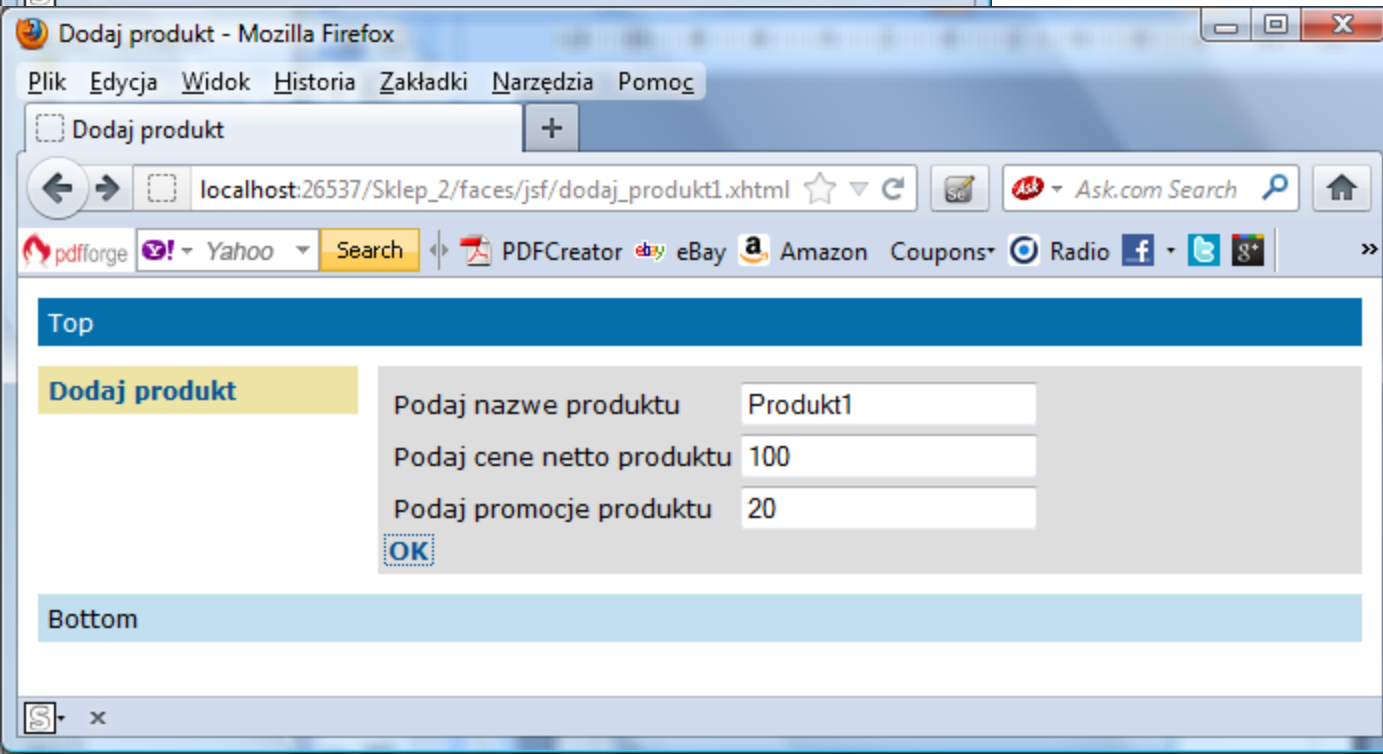
W klasie **Managed_produk**t należy zmienić nazwę wskazanej strony na **rezultat1**. Należy usunąć pliki **dodaj_produk**t i **rezultat**

```
public String dodaj_produk() {  
    String[] dane = {nazwa, cena, promocja};  
    fasada.utworz_produk(dane);  
    dane_produk();  
    return "rezultat1";  
}
```


Aplikacja Sklep_2 po uruchomieniu



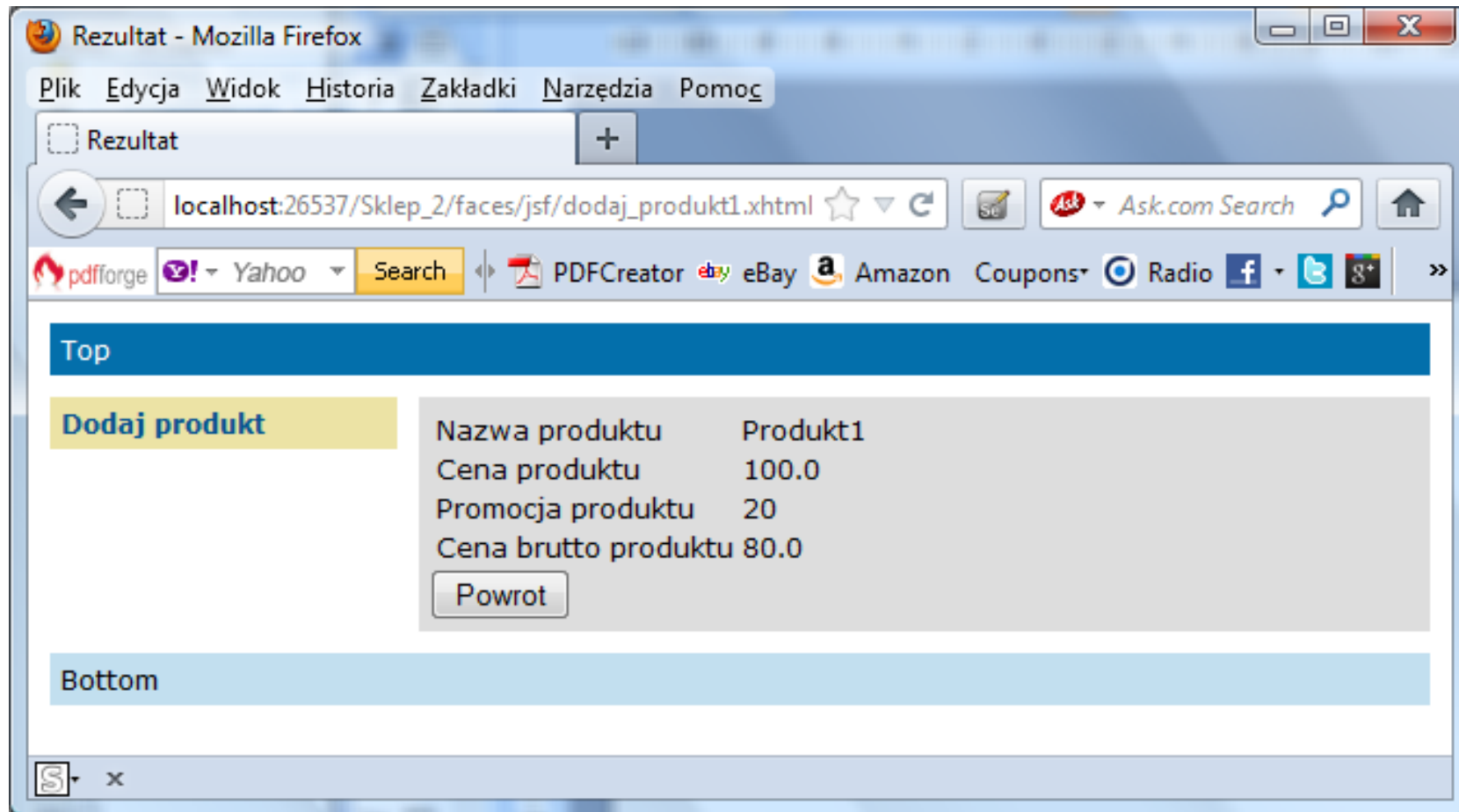
Widok strony głównej **index1**

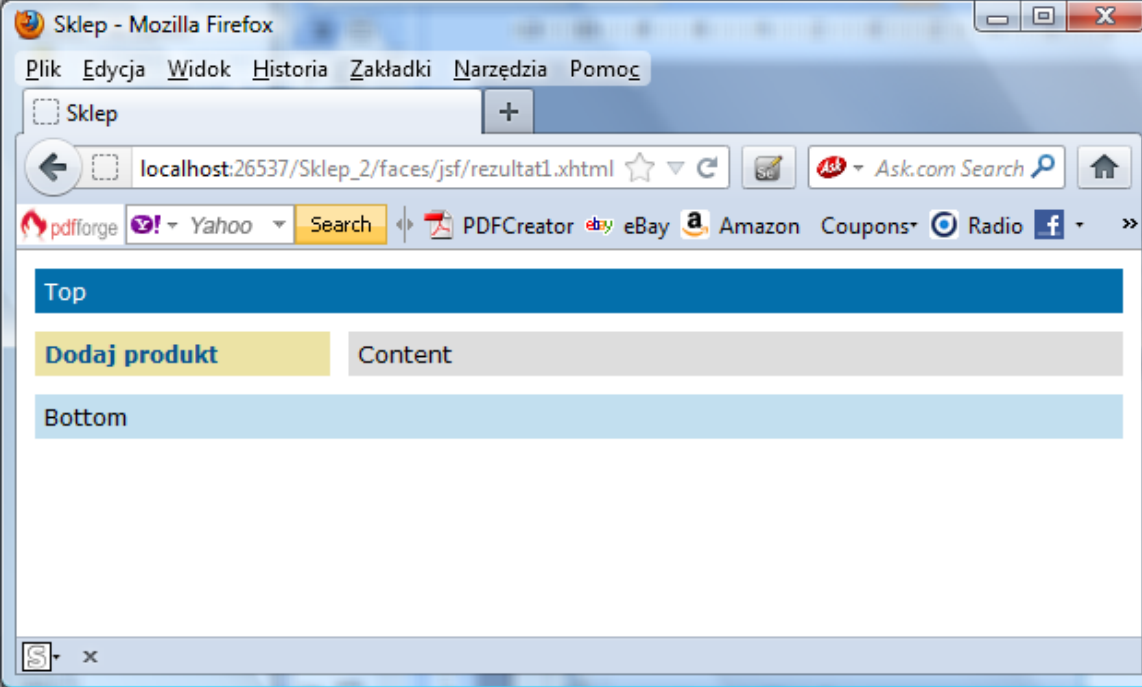


Widok strony
dodaj_produk1
po naciśnięciu

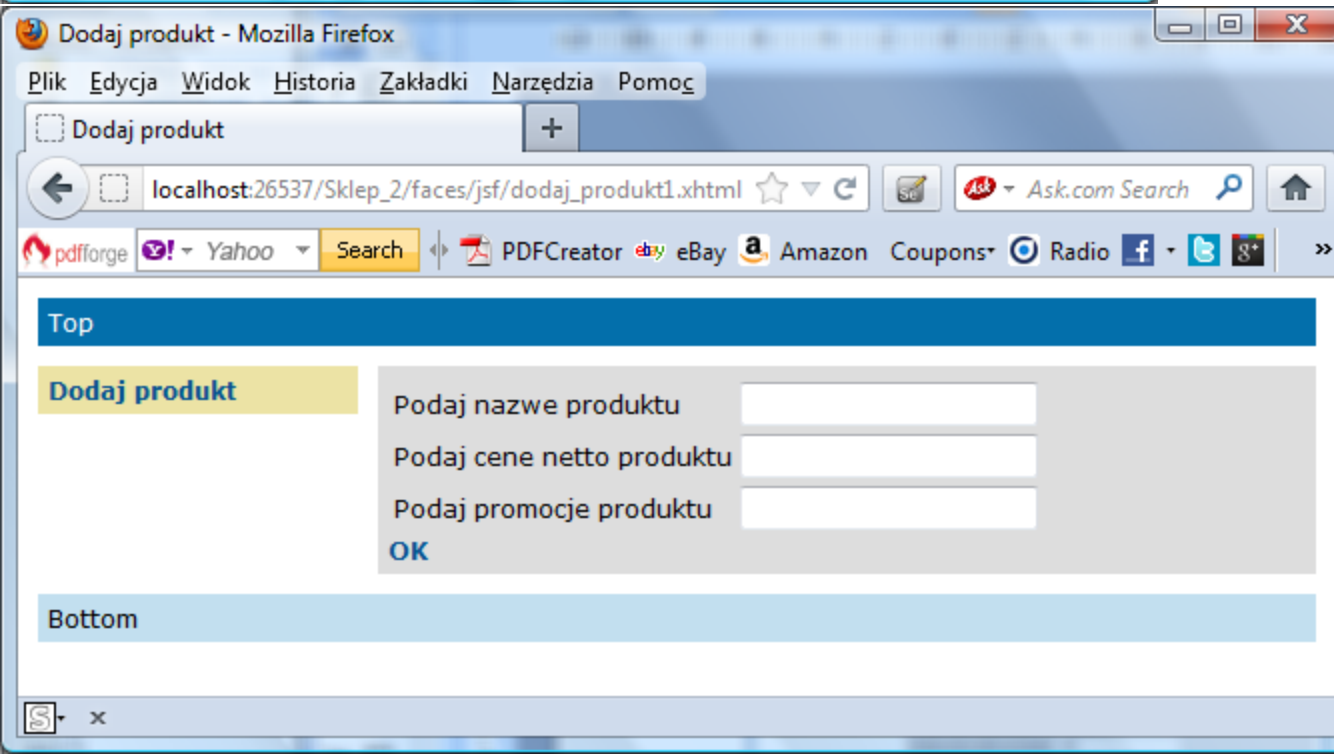
Dodaj produkt

Widok strony **rezultat1** po naciśnięciu **OK**



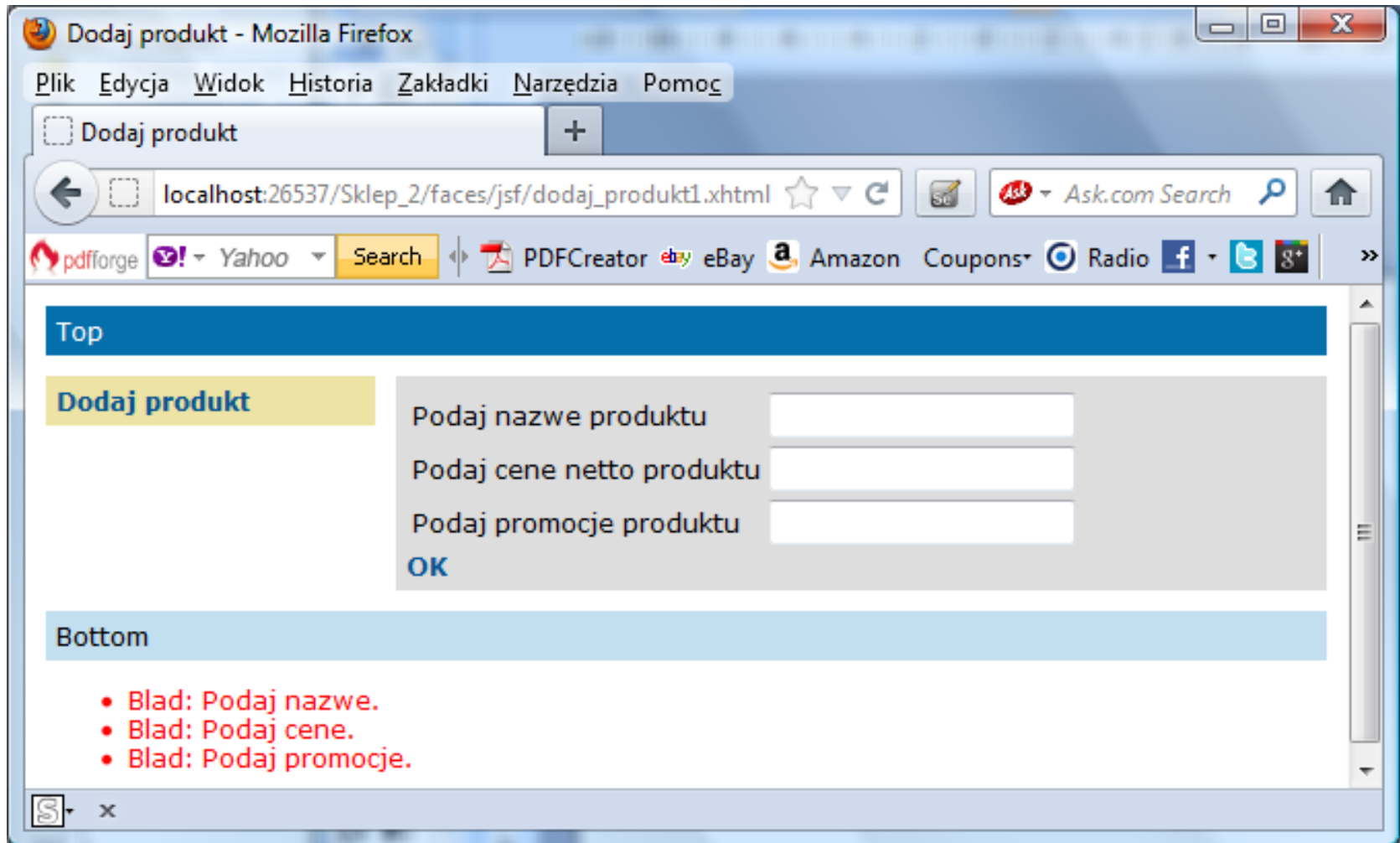


Widok strony **index1** po naciśnięciu **Powrot**



Widok strony **dodaj_produk1** po naciśnięciu **Dodaj produkt**

Widok strony **dodaj_produk1** po naciśnięciu **OK**, gdy pola formularzy nie zostały wypełnione



Przykład 2

Zastosowanie szablonu i znacznika
h:dataTable do prezentowania
zbioru produktów

Tworzenie kopii projektu typu Web Application o nazwie Sklep_3 z lab2-Przykład1 (prawy klawisz myszy na nazwie projektu i wybór **Copy** – na formularzu kopiowania należy podać nową nazwę projektu **Sklep_3**. Projekt źródłowy Sklep_2 należy zamknąć, spakować do formatu zip lub rar i usunąć wersję niespakowaną.

Należy zmodyfikować kod klasy Produkt – metodę equals

@Override

```
public boolean equals(Object object) {  
    if (!(object instanceof Produkt)) {  
        return false;  
    }  
    Produkt other = (Produkt) object;  
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id)))  
        if (!nazwa.equals(other.nazwa) || cena!=other.cena || promocja!=other.promocja){  
            return false;  
        }  
    return true;  
}
```

Zmodyfikowany kod klasy `Fasada_warstwy_biznesowej`

```
package Warstwa_biznesowa;
```

```
import java.util.ArrayList;  
import javax.ejb.Stateless;  
import jpa.Produkt;
```

```
@Stateless
```

```
public class Fasada_warstwy_biznesowej {
```

```
    static long klucz = 0;
```

```
    private ArrayList<Produkt> produkty = new ArrayList();
```

```
    boolean stan = false;
```

```
    public ArrayList<Produkt> getProdukty() {  
        return produkty;  
    }
```

```
    public void setProdukty(ArrayList<Produkt> produkty) {  
        this.produkty = produkty;  
    }
```

```
    public void utworz_produkt(String dane[]) {  
        Produkt produkt = new Produkt();  
        klucz++;  
        produkt.setId(new Long(klucz));  
        produkt.setNazwa(dane[0]);  
        produkt.setCena(Float.parseFloat(dane[1]));  
        produkt.setPromocja(Integer.parseInt(dane[2]));  
        dodaj_produkt(produkt);  
    }
```

Zmienna do nadawanie unikatowych wartości id dla obiektu typu Produkt

Przechowywanie listy produktów o unikatowych danych

Zmienna określająca, czy dodano nowy produkt: wartość false oznacza próbę wprowadzenia produktu o danych, które nie są unikatowe

Dodawanie nowego produktu


```
protected void dodaj_produkt(Produkt produkt) {  
    if (!produkty.contains(produkt)) {  
        produkty.add(produkt);  
        stan = true;  
    } else  
        stan = false;  
}
```

Dodawanie nowego produktu –
sprawdzenie, czy nowy obiekt jest
unikatowy. Wartość zmiennej stan równy
true oznacza wprowadzenie danej

```
public String[] dane_produktu() {  
    if (stan) {  
        Produkt produkt = produkty.get(produkty.size() - 1);  
        String nazwa = produkt.getNazwa();  
        String cena = "" + produkt.getCena();  
        String promocja = "" + produkt.getPromocja();  
        String cena_brutto = "" + produkt.cena_brutto();  
        String dane[] = {nazwa, cena, promocja, cena_brutto};  
        return dane; }  
    return null;  
}
```

Dane ostatnio wprowadzonego produktu
przenaczone do prezentacji, Wartość
null oznacza brak dodania nowego
produktu

```
public ArrayList<ArrayList<String>> items() {  
    ArrayList<ArrayList<String>> dane = new ArrayList();  
    for (Produkt p : produkty) {  
        ArrayList<String> wiersz = new ArrayList();  
        wiersz.add(p.getId().toString());  
        wiersz.add(p.getNazwa());  
        wiersz.add("" + p.getCena());  
        wiersz.add("" + p.getPromocja());  
        wiersz.add("" + p.cena_brutto());  
        dane.add(wiersz); }  
    return dane;  
}
```

Dane przechowywanych obiektów typu
Produkt przeznaczone do prezentacji w
komponencie dataTable – Jest to
kolekcja elementów, które są kolekcją
elementów typu String reprezentująca
atrybuty i wyliczoną cenę brutto obiektu
typu Produkt

Zmodyfikowana zawartość klasy Managed_produk

```
package jsf;

import Warstwa_biznesowa.Fasada_warstwy_biznesowej;
import javax.ejb.EJB;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;

@ManagedBean
@RequestScoped
public class Managed_produk {

    @EJB
    private Fasada_warstwy_biznesowej fasada;
    private String nazwa;
    private String cena;
    private String promocja;
    private String cena_brutto;
    private DataModel items;
    private int stan = 1;

    public Managed_produk() { }

    public Fasada_warstwy_biznesowej getFasada() { return fasada; }

    public void setFasada(Fasada_warstwy_biznesowej fasada) { this.fasada = fasada; }
```

DataModel – model danych komponentu dataTable

Stan – zmienna oznaczająca warunki renderowania

```
public String getNazwa() {  
    return nazwa;  
}  
public void setNazwa(String nazwa) {  
    this.nazwa = nazwa;  
}  
public String getCena() {  
    return cena;  
}  
public void setCena(String cena) {  
    this.cena = cena;  
}  
public String getPromocja() {  
    return promocja;  
}  
public void setPromocja(String promocja) {  
    this.promocja = promocja;  
}  
public String getCena_brutto() {  
    return cena_brutto;  
}  
public void setCena_brutto(String cena_brutto) {  
    this.cena_brutto = cena_brutto;  
}
```

```
public DataModel utworz_DataModel() {  
    return new ListDataModel(fasada.items());  
}
```

```
public DataModel getItems() {  
    if (items == null) {  
        items = utworz_DataModel();  
    }  
    return items;  
}
```

```
public void setItems(DataModel items) {  
    this.items = items;  
}
```

```
public int getStan() {  
    return stan;  
}
```

```
public void setStan(int stan) {  
    this.stan = stan;  
}
```

Utworzenie modelu komponentu **dataTable** na podstawie kolekcji zawierających elementy reprezentujące wiersz tabeli (kolekcja obiektów typu String reprezentująca atrybuty obiektu typu **Produkt** oraz cenę brutto)

Dodane metod do klasy **Managed_produk**t obsługujących dodawanie produktu (**dodaj_produk**t) po pobraniu danych z formularza za pomocą atrybutów: nazwa, cena, promocja i wywołaniu metody **utworz_produk**t ziarna EJB z obiektu fasada klasy typu Fasada_warstwy_biznesowej oraz wyświetlanie danych za pomocą metody **dane_produk**tu pobranych z warstwy biznesowej od obiektu typu EJB fasada za pomocą metody **dane_produk**tu

```
public String dodaj_produkt() {  
    String[] dane = {nazwa, cena, promocja};  
    fasada.utworz_produkt(dane);  
    dane_produkt();  
    return "rezultat2";  
}
```

```
public void dane_produktu() {  
    stan = 1;  
    String[] dane = fasada.dane_produktu();  
    if (dane == null) {  
        stan = 0;  
    } else {  
        nazwa = dane[0];  
        cena = dane[1];  
        promocja = dane[2];  
        cena_brutto = dane[3]; }  
}
```

Modyfikacja szablonu **template.xhtml** – dodanie w części przeznaczonej na menu (id=left) linku do strony **lista_produkow.xhtml**

```
<?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 name="css/default.css" />
  <h:outputStylesheet name="css/cssLayout.css"/>
  <title><ui:insert name="title">Facelets Template</ui:insert></title>
</h:head>

<h:body>
  <div id="top">
    <ui:insert name="top">Top</ui:insert>
  </div>
```

```
<div>
  <div id="left">
    <h:link outcome="/faces/jsf/dodaj_produkt2" value="Dodaj produkt"/><br/>
    <h:link outcome="/faces/jsf/lista_produktow" value="Lista produktow"/>
  </div>
  <div id="content" class="left_content">
    <ui:insert name="content">Content</ui:insert>
  </div>
</div>
<div id="bottom">
  <ui:insert name="bottom">Bottom</ui:insert>
</div>
</h:body>
</html>
```

Zmodyfikowana zawartość pliku **dodaj_produk2.xhtml** (zmiana nazwy z **dodaj_produk1**)

```
<?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">
  <body>

  <ui:composition template=" ../template.xhtml">
    <ui:define name="title">
      Dodaj produkt
    </ui:define>
```



```
<ui:define name="content">
```

```
<h:form><h:panelGrid columns="2">
```

```
<h:outputLabel value="Podaj nazwe produktu" for="nazwa" />
```

```
<h:inputText
```

```
id="nazwa"
```

```
title="Podaj nazwe:"
```

```
value="#{managed_produkt.nazwa}"
```

```
required="true"
```

```
requiredMessage="Blad: Podaj nazwe." >
```

```
</h:inputText>
```

```
<h:outputLabel value="Podaj cene netto produktu" for="cena" />
```

```
<h:inputText
```

```
id="cena"
```

```
title="Podaj cene:"
```

```
value="#{managed_produkt.cena}"
```

```
required="true"
```

```
requiredMessage="Blad: Podaj cene." >
```

```
</h:inputText>
```

```
<h:outputLabel value="Podaj promocje produktu" for="promocja" />
```

```
<h:inputText
```

```
id="promocja"
```

```
title="Podaj promocje:"
```

```
value="#{managed_produkt.promocja}"
```

```
required="true"
```

```
requiredMessage="Blad: Podaj promocje." >
```

```
</h:inputText>
```

```
</h:panelGrid>
```

Siatka **panelGrid** umożliwia wprowadzanie danych produktu do obiektu typu **Managed_produkt** w dwóch kolumnach za pomocą komponentów **outputLabel** oraz **inputText**.

Atrybuty **required** i **requiredMessage** obsługują błąd wynikający z braku wprowadzenia danych do komponentów typu **inputText**

```
<h:commandLink action="#{managed_produkt.dodaj_produkt}" value="OK" />

</h:form>
</ui:define>
</ui:composition>
</body>

</html>
```

Znacznik **<h:commandLink** pozwala powrócić do strony, której nazwę zwraca bezparametrowa metoda `dodaj_produkt` z obiektu klasy `Managed_produkt` (wartość atrybutu **action**) – jest to strona **rezultat.xhtml**:

```
public String dodaj_produkt() {
    String[] dane = {nazwa, cena, promocja};
    fasada.utworz_produkt(dane);
    dane_produktu();
    return "rezultat2";
}
```

Do strony **rezultat2.xhtml** dodano kod JSF do prezentacji danych produktu oraz ceny brutto, pobieranych z atrybutów obiektu **managed_produk**t typu **Managed_produk**t

```
<?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">
```

```
<body>
```

```
<ui:composition template="./../template.xhtml">
```

```
  <ui:define name="title">
```

```
    Rezultat
```

```
  </ui:define>
```

Dane ostatnio wprowadzonego produktu są
wyświetlane warunkowo



```
<ui:define name="content">
  <h:form>
    <h:outputText escape="false" value="#{bundle.Lista_produkow_niedodano}"
      rendered="#{managed_produkt.stan==0}"/>
    <h:panelGrid columns="2" rendered="#{managed_produkt.stan!=0}">
      <h:outputLabel value="Nazwa produktu" for="nazwa" />
      <h:outputText id="nazwa" value="#{managed_produkt.nazwa}"/>
      <h:outputLabel value="Cena produktu" for="cena" />
      <h:outputText id="cena" value="#{managed_produkt.cena}"/>
      <h:outputLabel value="Promocja produktu" for="promocja" />
      <h:outputText id="promocja" value="#{managed_produkt.promocja}"/>
      <h:outputLabel value="Cena brutto produktu" for="brutto" />
      <h:outputText id="brutto" value="#{managed_produkt.cena_brutto}" />

      <h:commandButton id="powrot" value="Powrot" action="/faces/index2"/>

    </h:panelGrid>
  </h:form>
</ui:define>

</ui:composition>
</body>
</html>
```

Zawartość strony **lista_produkow.xhtml** do wyświetlania listy produktów dodana do projektu- **New/Other/JavaServer Faces/ Facelets Template Client...**

```
<?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,"
xmlns:f="http://java.sun.com/jsf/core">

<body>
<ui:composition template=" ../template.xhtml">
<ui:define name="title">
<h:outputText value="#{bundle.Lista_produkow_tytul}"></h:outputText>
</ui:define>

<ui:define name="content">
<h:form styleClass="jsfcrud_list_form">

<h:outputText escape="false" value="#{bundle.Lista_produkow_pusta}"
rendered="#{managed_produk.items.rowCount == 0}"/>

<h:panelGroup rendered="#{managed_produk.items.rowCount > 0}">
```

Jeżeli brak danych pobranych z modelu **items** typu **DataModel** (`rowCount==0`), wtedy wyświetla się napis `bundle.Lista_produkow_pusta` (czyli Brak danych), w przeciwnym wypadku wyświetla się tabelę `<h:dataTable>` (następny slajd)

```
<h:dataTable value="#{managed_produkt.items}" var="item" border="0"
```

Komponent typu **dataTable** zbindowany z obiektem **items** typu **DataModel**

```
cellpadding="2" cellspacing="0"  
rowClasses="jsfcrud_odd_row,jsfcrud_even_row"  
rules="all" style="border:solid 1px">
```

Item – element kolekcji **items** (zawierający dane atrybutów obiektu typu **Produkt** oraz **cenę brutto**)

```
</h:dataTable>  
<h:column>
```

```
<f:facet name="header"> Nagłówek kolumny tabeli dataTable
```

```
<h:outputText value="#{bundle.Lista_produktow_id}"/>
```

```
</f:facet>
```

```
<h:outputText value="#{item.get(0)}"/>
```

Kolejny element kolekcji **item** (zawierający dane atrybutu **id** obiektu typu **Produkt**), która jest elementem kolekcji **items** typu **DataModel**

```
</h:column>
```

```
<h:column>
```

```
<f:facet name="header">
```

```
<h:outputText value="#{bundle.Lista_produktow_nazwa}"/>
```

```
</f:facet>
```

```
<h:outputText value="#{item.get(1)}"/>
```

Kolejny element kolekcji **item** (zawierający dane atrybutu **nazwa** obiektu typu **Produkt**), która jest elementem kolekcji **items** typu **DataModel**

```
</h:column>
```

```
<h:column>
```

```
<f:facet name="header">
```

```
<h:outputText value="#{bundle.Lista_produktow_cena}"/>
```

```
</f:facet>
```

```
<h:outputText value="#{item.get(2)}"/>
```

Kolejny element kolekcji **item** (zawierający dane atrybutu **cena** obiektu typu **Produkt**), która jest elementem kolekcji **items** typu **DataModel**

```
</h:column>
```

```
<h:column>
  <f:facet name="header">
    <h:outputText value="#{bundle.Lista_produkow_promocja}"/>
  </f:facet>
  <h:outputText value="#{item.get(3)}"/>
</h:column>
```

Kolejny element kolekcji **item** (zawierający dane atrybutu **promocja** obiektu typu **Produkt**), która jest elementem kolekcji **items** typu **DataModel**

```
<h:column>
  <f:facet name="header">
    <h:outputText value="#{bundle.Lista_produkow_cenabrutto}"/>
  </f:facet>
  <h:outputText value="#{item.get(4)}"/>
</h:column>
</h:dataTable>
```

Kolejny element kolekcji **item** (zawierający dane ceny brutto wyznaczonej przez metodę `cena_brutto` obiektu typu **Produkt**), która jest elementem kolekcji **items** typu **DataModel**

```
</h:panelGroup>
</h:form>
</ui:define>
```

```
</ui:composition>
```

```
</body>
</html>
```

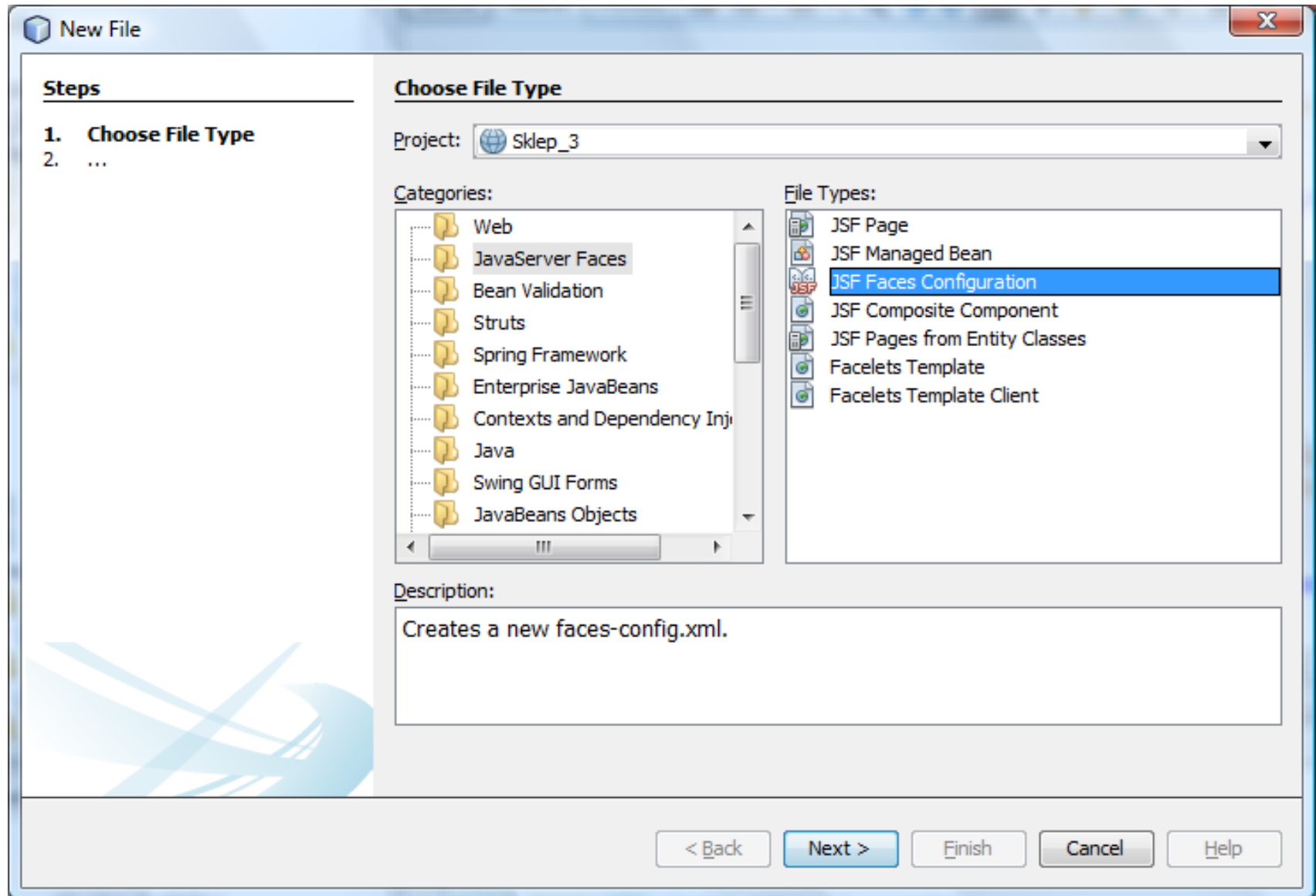
Kod klasy głównej **index2.xhtml**

```
<?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">
<body>
    <ui:composition template="./template.xhtml">

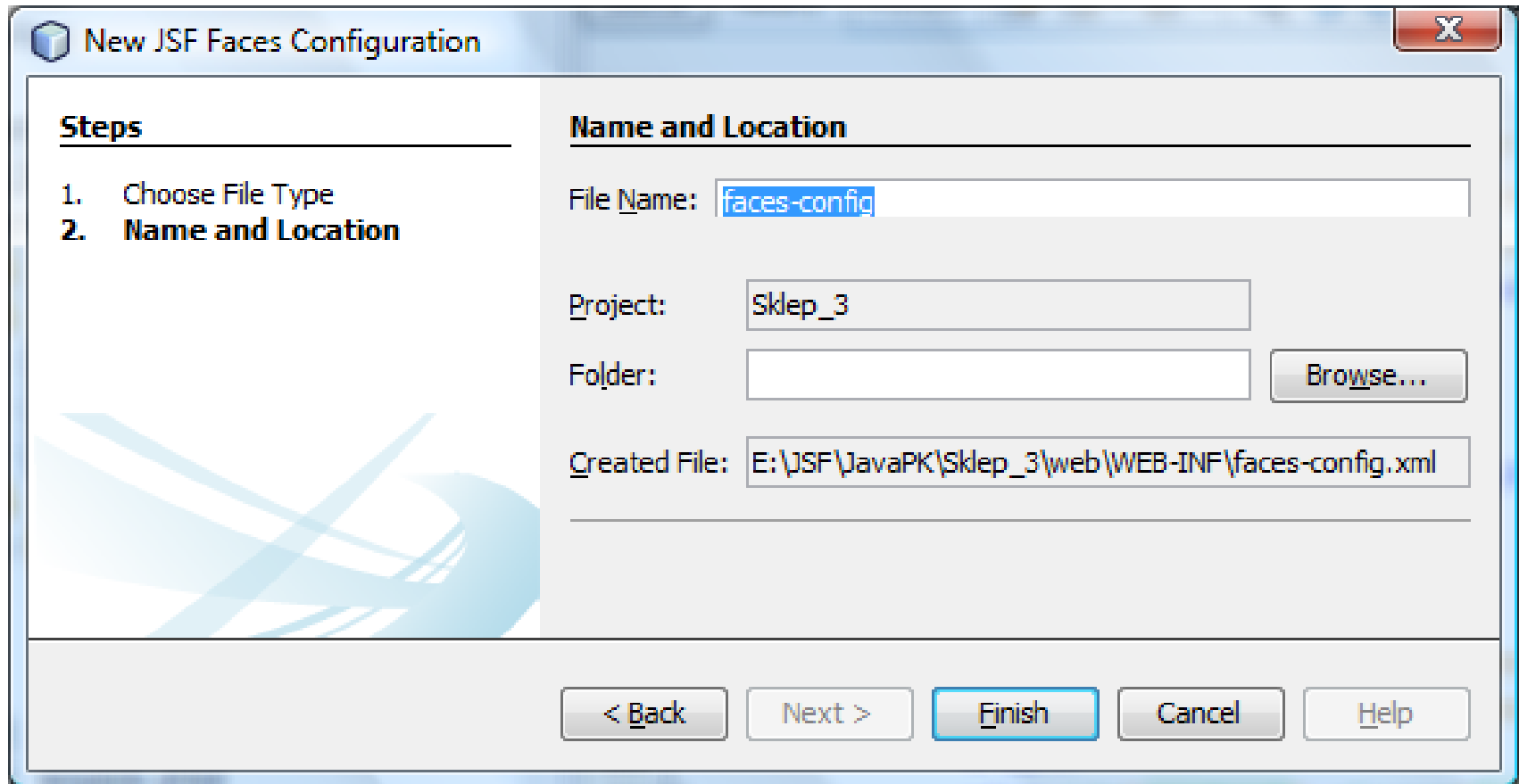
        <ui:define name="title">
            Sklep
        </ui:define>

    </ui:composition>
</body>
</html>
```


Dodanie pliku konfiguracji projektu **faces-config.xml** (New/Other/JavaServer Faces/ JSF Faces Configuration)



Dodanie pliku konfiguracji projektu **faces-config.xml**



New JSF Faces Configuration

Steps

1. Choose File Type
- 2. Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

Dodanie pliku typu **properties** do projektu: prawy klawisz na **Source Packages**, **New/Other/Other/Properties File**

The screenshot displays the NetBeans IDE 7.2 interface for a project named 'Sklep_3'. The 'Projects' view on the left shows the project structure, including a 'Source Packages' folder. A context menu is open over 'Source Packages', with 'New' selected, leading to a 'New File' dialog. The dialog shows the 'Choose File Type' step, where 'Properties File' is selected under the 'Other' category. The description of the Properties File type is visible at the bottom of the dialog.

Steps:

1. Choose File Type
2. ...

Choose File Type

Project: Sklep_3

Categories:

- AWT GUI Forms
- Unit Tests
- Selenium
- Persistence
- Groovy
- Hibernate
- Web Services
- XML
- GlassFish
- WebLogic
- Other

File Types:

- SQL file
- HTML File
- XHTML File
- SVG File
- JavaScript File
- JSON File
- JNLP File
- Properties File**
- Cascading Style Sheet
- Ant Build Script
- YAML File
- INI File

Description:

Creates a resource bundle (.properties) file suitable for internationalizing applications by separating out all human-visible text strings from your code. Resource bundle files can also be used to collect other types of strings, such as properties for Ant scripts. The created resource bundle contains only one locale,

< Back Next > Finish Cancel Help

Dodanie pliku typu **properties** do projektu

New Properties File

Steps

1. Choose File Type
- 2. Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

Zawartość pliku **Bundle.properties** zawierająca treść komunikatów. Należy dodatkowo wkleić plik **jsfcrud.css** do katalogu **resources/css** pobrany ze strony <http://zofia.kruczkiewicz.staff.iiar.pwr.wroc.pl/wyklady/javapk/jsfcrud.css>

The screenshot shows an IDE interface with the following components:

- Project Explorer:** Shows a project named 'Sklep_3' with a tree structure including 'Web Pages', 'resources/css', and 'Source Packages'. The 'Bundle.properties' file is highlighted in the 'Source Packages' folder.
- Bundle.properties Editor:** Contains the following content:

```
1 # To change this template, choose Tools | Templates
2 # and open the template in the editor.
3
4 Lista_produkow_tytul=Lista produktow
5 Lista_produkow_pusta=Brak danych
6 Lista_produkow_id=Id produktu
7 Lista_produkow_nazwa=Nazwa produktu
8 Lista_produkow_cena=Cena netto produktu
9 Lista_produkow_promocja=Promocja produktu
10 Lista_produkow_cenabrutto=Cena brutto
11 Lista_produkow_niedodano=Taki produkt juz istnieje
```
- Output Window:** Shows the output of a deployment process:

```
run-deploy:
Browsing: http://localhost:26537/Sklep2
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 16 seconds)
```
- Properties Panel:** Shows details for the selected 'Bundle.properties' file, including Name, Extension, All Files, File Size (92), and Modification (201...).

Zawartość pliku **Bundle.properties** zawierająca treść komunikatów używanych na stronie **lista_produkow.xhtml**. Należy w taki sam sposób zastąpić komunikaty w pozostałych plikach.xhtml

Lista_produkow_tytul=Lista produktow

Lista_produkow_pusta=Brak danych

Lista_produkow_id=Id produktu

Lista_produkow_nazwa=Nazwa produktu

Lista_produkow_cena=Cena netto produktu

Lista_produkow_promocja=Promocja produktu

Lista_produkow_cenabrutto=Cena brutto

Lista_produkow_niedodano=Taki produkt juz istnieje

Deklaracja pliku **Bundle.properties** w pliku konfiguracyjnym **faces-config.xml**.

The screenshot shows an IDE window with the following components:

- Projects View:** Shows a project named 'Sklep_3' with a tree structure including 'Web Pages', 'WEB-INF' (containing 'faces-config.xml'), 'jsf', 'resources', and 'Source Packages'.
- Editor:** Displays the 'faces-config.xml' file with the following XML content:

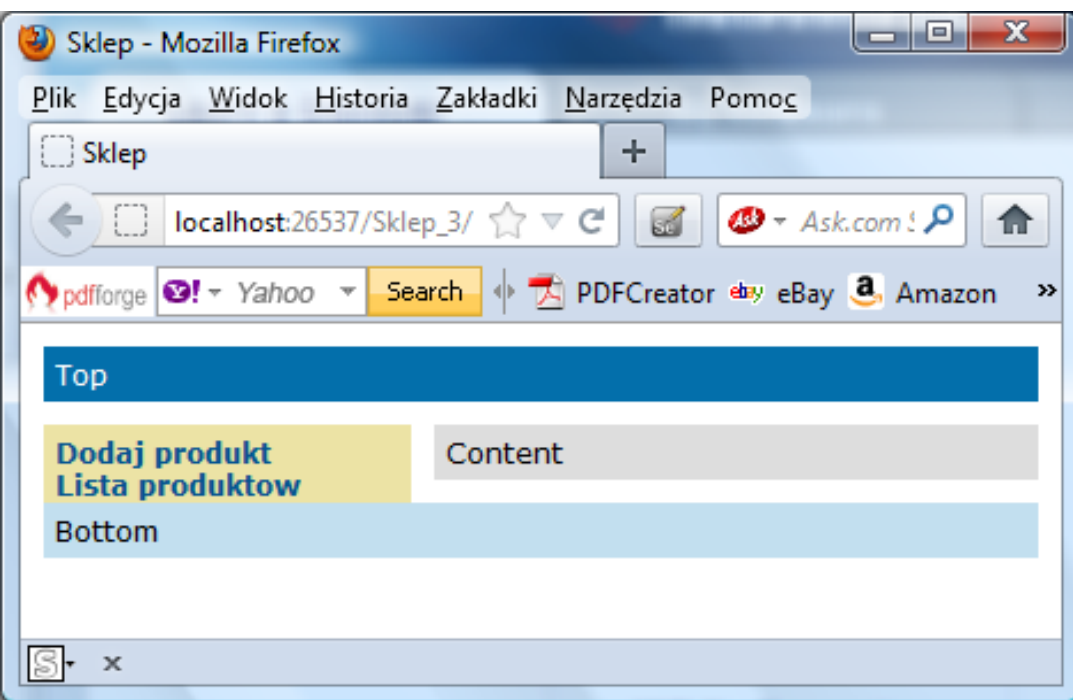
```
1 <?xml version='1.0' encoding='UTF-8'?>
2
3 <!-- ===== FULL CONFIGURATION FILE =====
4
5 <faces-config version="2.1"
6     xmlns="http://java.sun.com/xml/ns/javaee"
7     xmlns:xsi="http://www.w3.org/2001/XMLSchema-imp
8     xsi:schemaLocation="http://java.sun.com/xml/ns/
9
10 <application>
11     <resource-bundle>
12         <base-name>/Bundle</base-name>
13         <var>bundle</var>
14     </resource-bundle>
15 </application>
16 </faces-config>
17
```
- Properties View:** Shows details for the selected file: Name: face..., Extension: xml, All Files: E..., File Size: 400, Modification: 201...
- Output View:** Shows running processes: Java DB Database Process, GlassFish Server 3+, and Sklep2 (run).

Deklaracja strony startowej index2.xhtml w pliku deskryptora web.xml

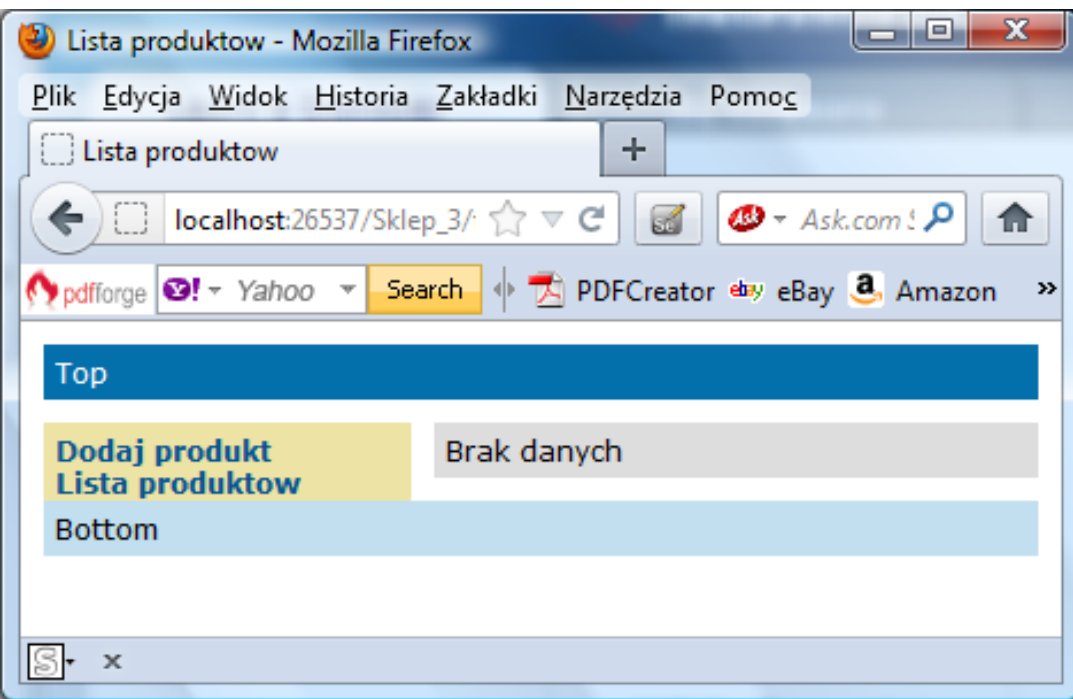
The screenshot displays an IDE interface with the following components:

- Projects View:** Shows a project named 'Skep_3' with a 'Web Pages' folder containing 'WEB-INF' (with 'faces-config.xml' and 'web.xml'), 'jsf' (with 'dodaj_produk2.xhtml', 'lista_produkow.xhtml', and 'rezultat2.xhtml'), and 'resources' (with 'css' folder containing 'cssLayout.css', 'default.css', and 'jscrud.css'). Other files include 'index2.xhtml', 'template.xhtml', and 'Source Packages'.
- Code Editor:** Shows the 'web.xml' file with the following XML content:

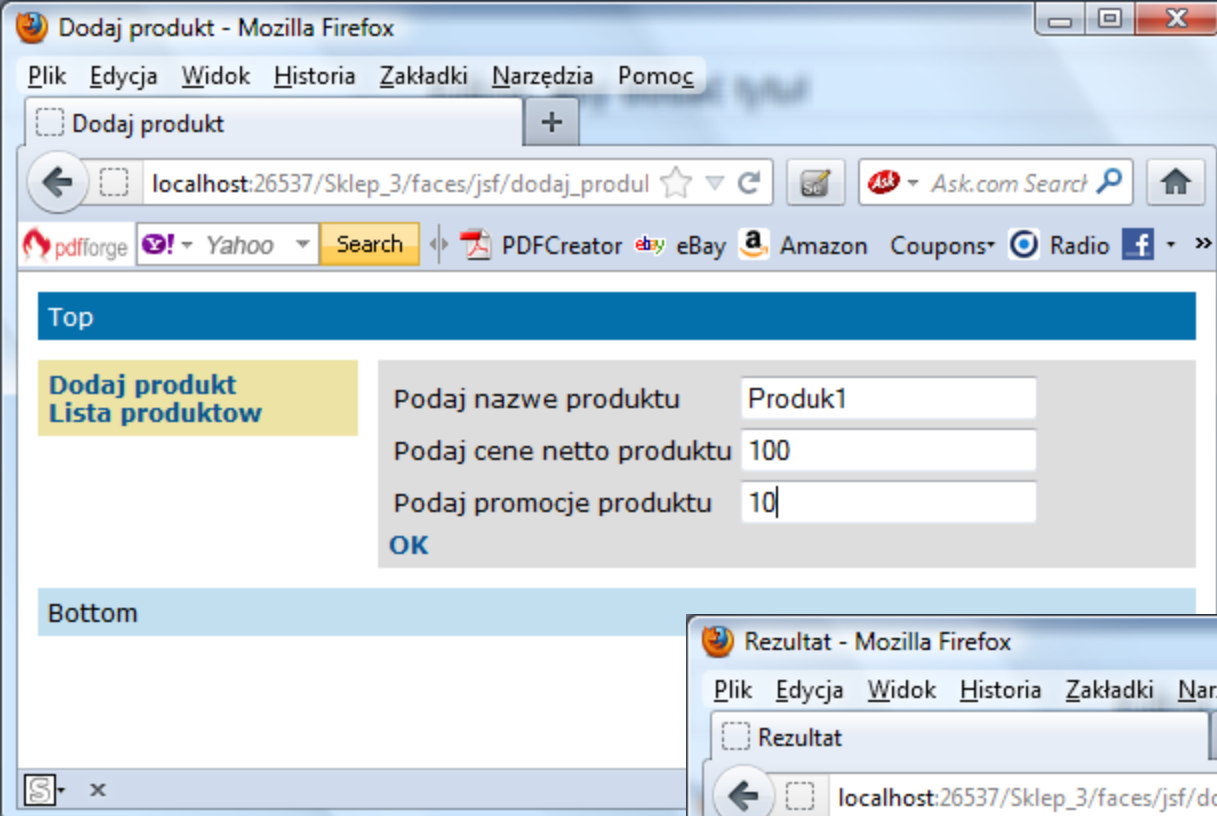
```
7 <servlet>
8     <servlet-name>Faces Servlet</servlet-name>
9     <servlet-class>javax.faces.webapp.FacesServlet</
10    <load-on-startup>1</load-on-startup>
11 </servlet>
12 <servlet-mapping>
13     <servlet-name>Faces Servlet</servlet-name>
14     <url-pattern>/faces/*</url-pattern>
15 </servlet-mapping>
16 <session-config>
17     <session-timeout>
18         30
19     </session-timeout>
20 </session-config>
21 <welcome-file-list>
22     <welcome-file>faces/index2.xhtml</welcome-file>
23 </welcome-file-list>
24 /web-app>
25
```
- Properties View:** Shows the 'web.xml' file properties: Name: web, Extension: xml, All Files: E..., File Size: 974, Modification: 201..., Servlet Spec: 3.0.
- Output View:** Shows 'HTTP Server Monitor' with running processes: 'Java DB Database Process', 'GlassFish Server 3+', and 'Skep2 (run)'.



Uruchomienie aplikacji

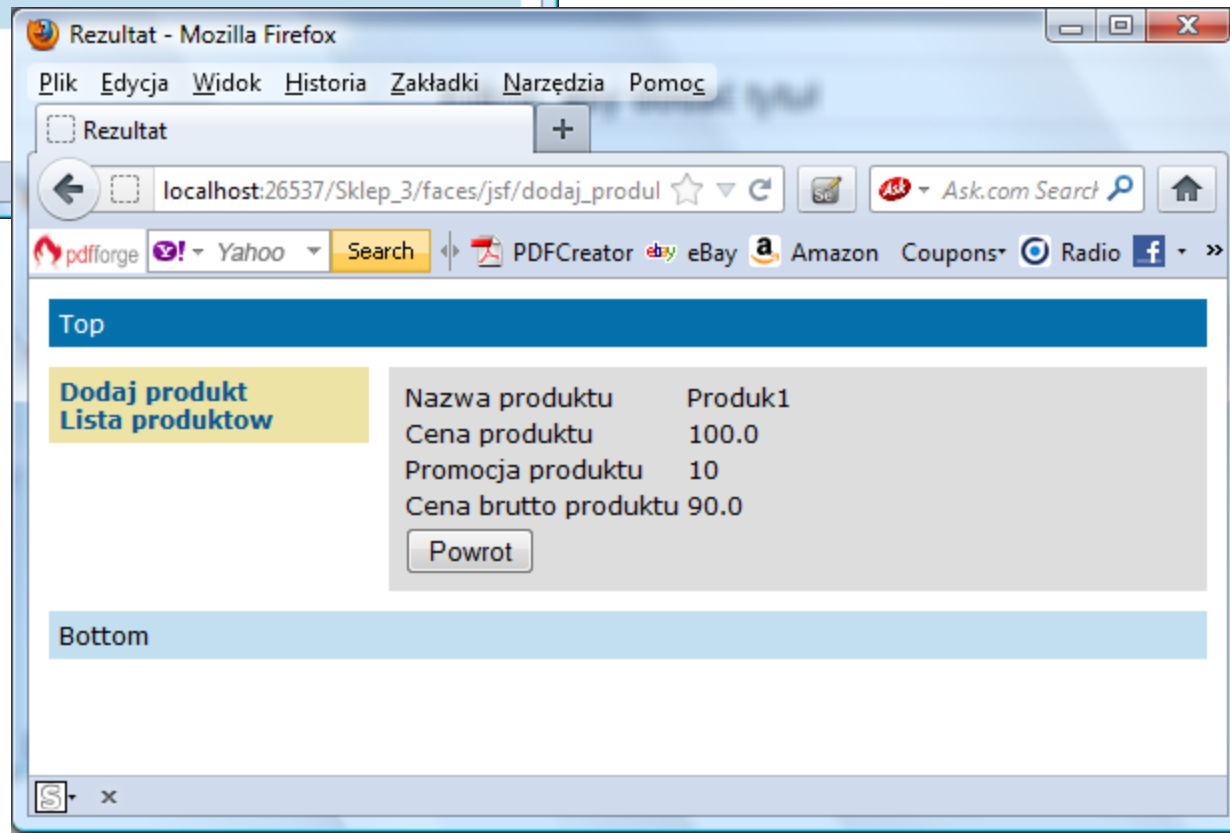


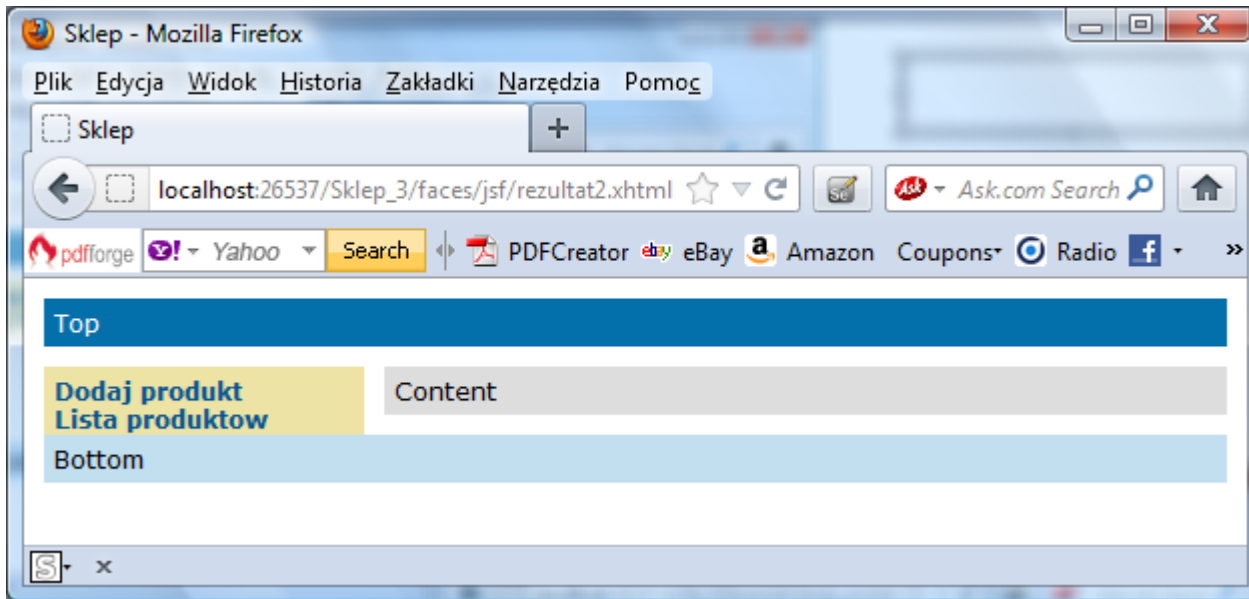
Widok po kliknięciu na
Lista produktów



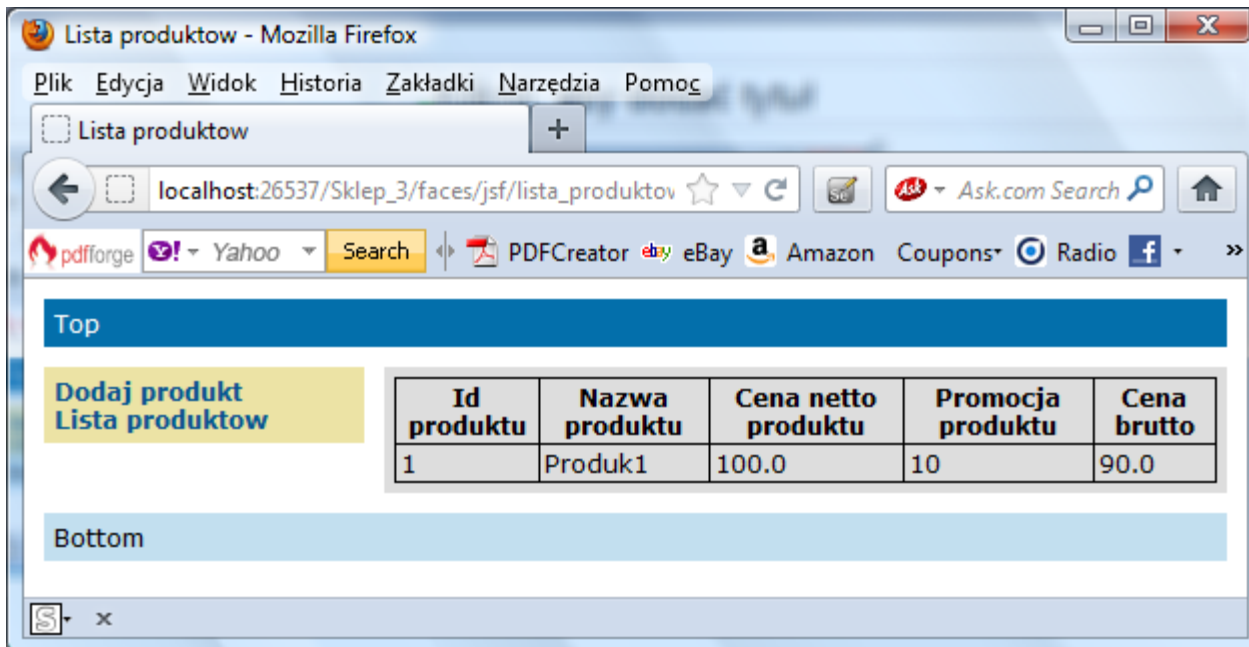
Widok po kliknięciu na
Dodaj produkt

Widok po kliknięciu na
Ok

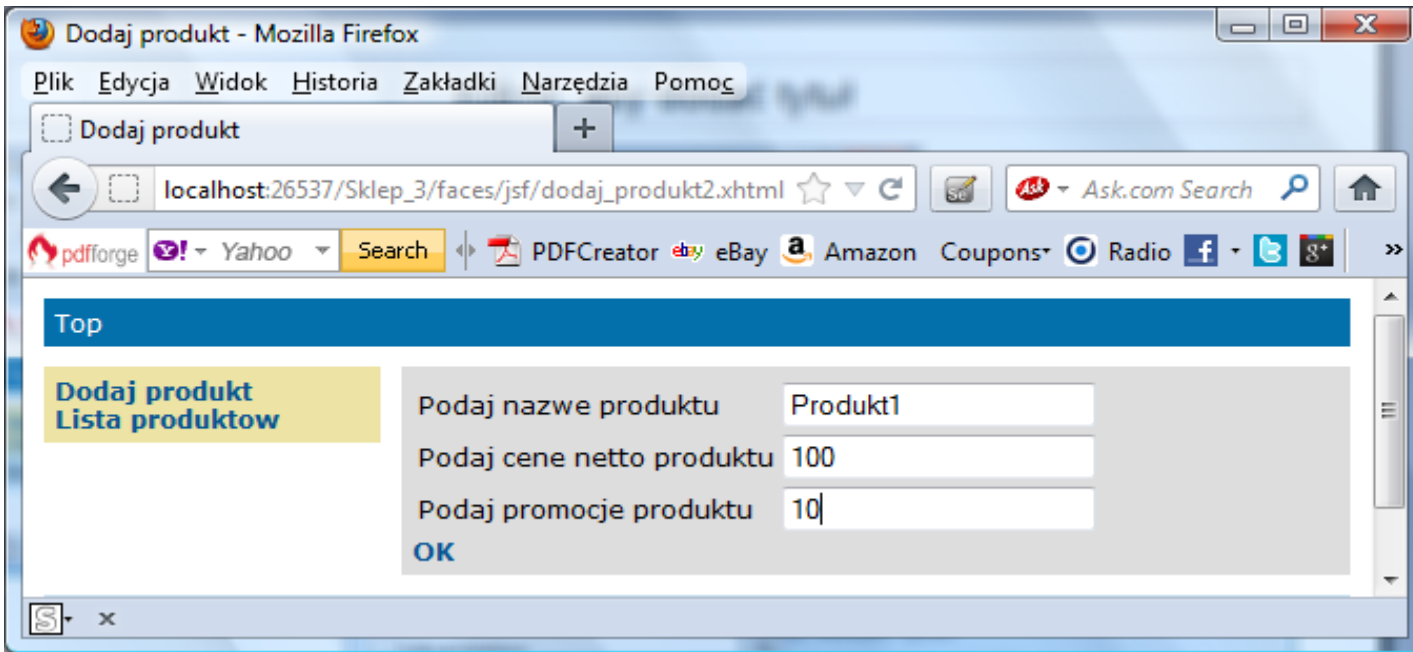




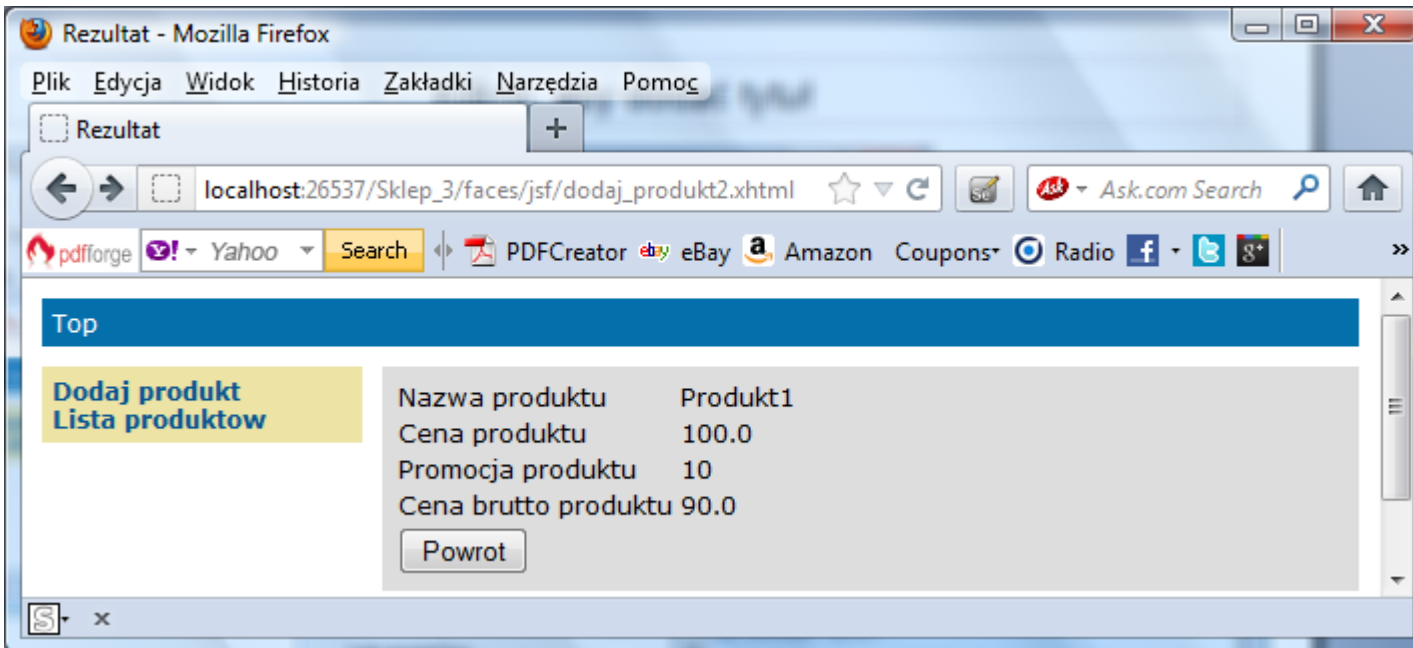
Widok po kliknięciu na
Powrot



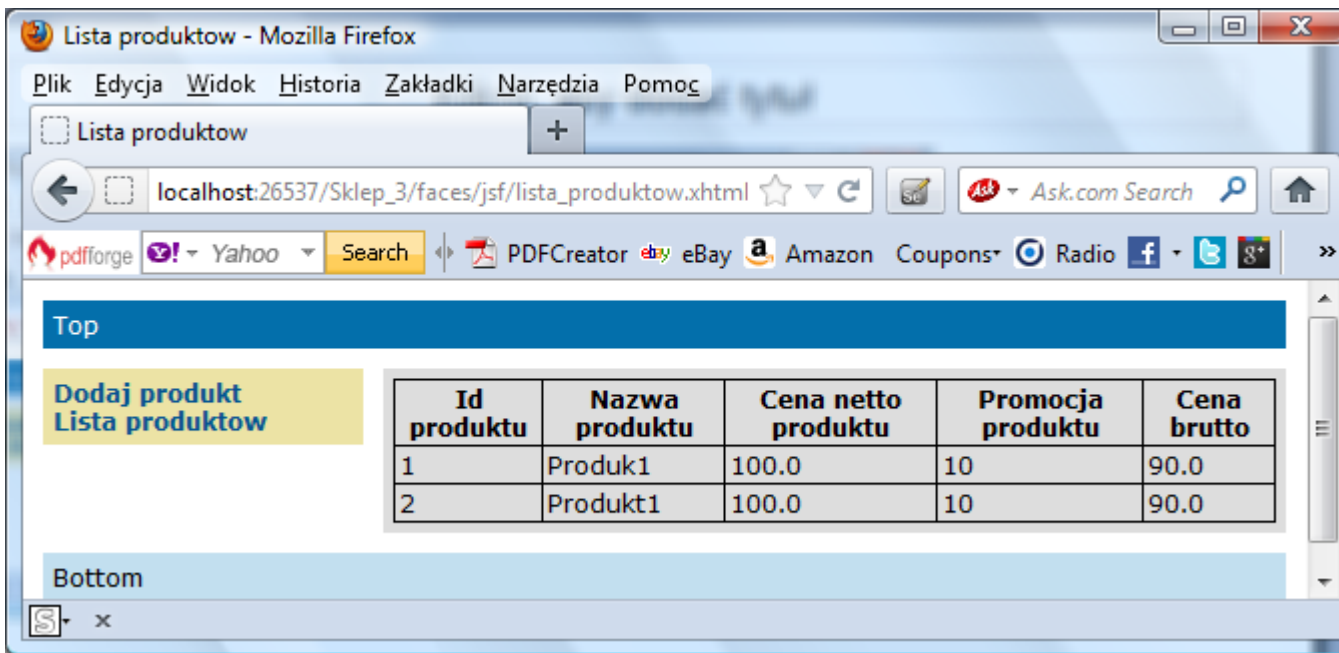
Widok po
kliknięciu na
Lista produktow



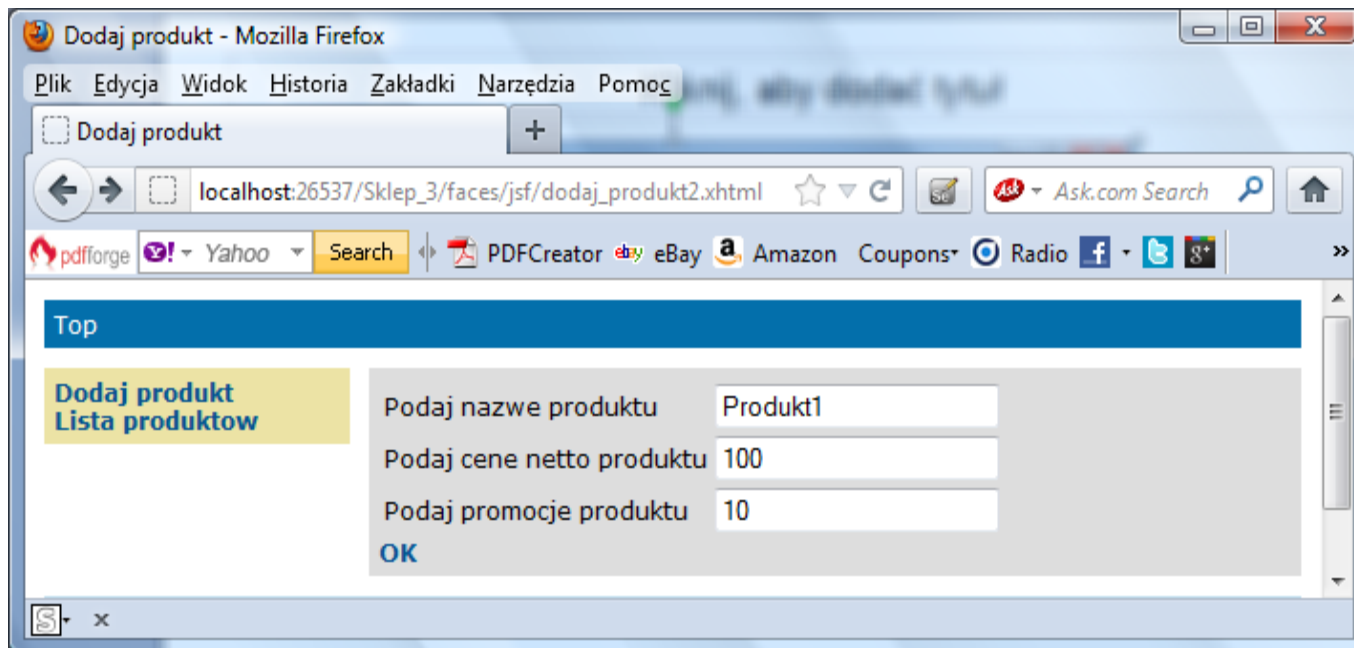
Widok po
kliknięciu na
Dodaj produkt



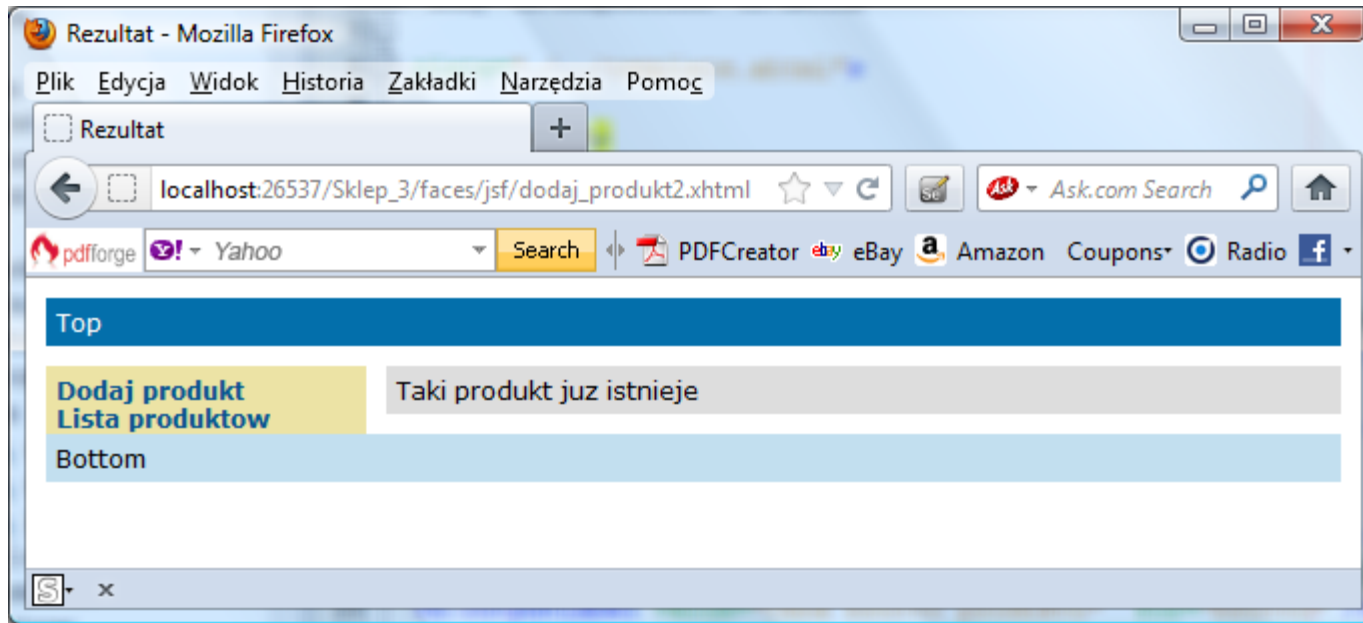
Widok po
kliknięciu na
Ok



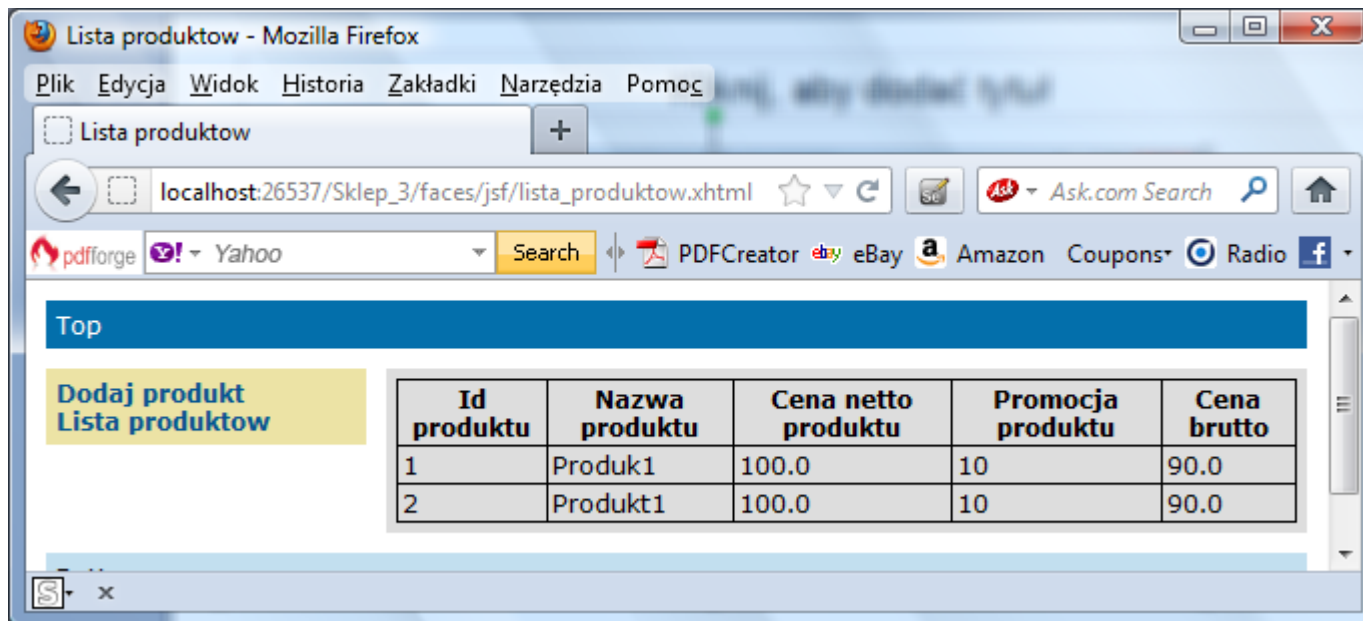
Widok po kliknięciu
na
Lista produktow



Widok po
kliknięciu na
Dodaj produkt



Widok po kliknięciu na
Ok



Widok po kliknięciu
na
Lista produktow