



Funded by the  
Erasmus+ Programme  
of the European Union



---

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

---



# KI204 - JAVA 7: JAVA ENTERPRISE EDITION

## JavaServer Faces

Lekcija 04

PRIRUČNIK ZA STUDENTE

# KI204 - JAVA 7: JAVA ENTERPRISE EDITION

## Lekcija 04

### *JAVASERVER FACES*

- ✓ JavaServer Faces
- ✓ Poglavlje 1: Uvod u JSF
- ✓ Poglavlje 2: Forme u JSF stranicama
- ✓ Poglavlje 3: Kreiranje CDI zrna
- ✓ Poglavlje 4: JSF stranica za potvrđivanje
- ✓ Poglavlje 5: JSF Validacija
- ✓ Poglavlje 6: Facelet šabloni
- ✓ Poglavlje 7: Biblioteka ugovora resursa
- ✓ Poglavlje 8: Primer 8
- ✓ Poglavlje 9: Domaći zadatak 3 - 1
- ✓ Poglavlje 10: PrimeFaces biblioteka komponentata
- ✓ Poglavlje 11: IceFaces biblioteka komponentata
- ✓ Poglavlje 12: RichFaces biblioteka komponentata
- ✓ Poglavlje 13: Domaći zadatak 3 - 2
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

# ▼ Uvod

## UVOD

*Cilj lekcije je analiza i demonstracija specifičnog Java okvira za razvoj veb aplikacija.*

JavaServer Faces (JSF) predstavlja specifičan Java okvir ( **framework** ) namenjen za razvoj veb aplikacija. Upravo, u ovoj lekciji će biti cilj pokazivanje načina na koji primena JSF pojednostavljuje razvoj veb aplikacija. U tu svrhu će biti obrađene značajne teme za analizu i demonstraciju ovog specifičnog Java okvira za razvoj veb aplikacija:

- Kreiranje JSF projekta primenom izabranog razvojnog okruženja;
- Postavljanje JSF tagova za korišćenje prednosti koju obezbeđuje primena taga JSF `<h:panelGrid>`;
- Korišćenje statičke i dinamičke navigacije za definisanje navigacije između stranica;
- Razvoj CDI ( **Contexts and Dependency Injection for the Java EE Platform** ) zrna za enkapsulaciju podataka i programske logike;
- Implementacija JSF validatora;
- Kreiranje JSF šablona;
- Rad sa JSF temama.
- Rad sa JSF bibliotekama komponenta.

Posebno će lekcija insistirati na analizi i demonstraciji primenom odabranih primera koji će na najbolji način istaći i pokazati prednosti primene JSF stranica u Java veb aplikacijama. Od posebnog značaja će biti fokusiranost primera na rad sa šablonima stranica i temama za pokazivanje veoma jednostavnog načina kreiranja JSF stranica sa predefinisanim atraktivnim izgledom. Konkretni primer će pokazati kako jedna šablon JSF datoteka na veoma jednostavan način upravlja nizom JSF datoteka klijenata šablona.

Savladavanjem ove lekcije studenti će biti osposobljeni da koriste JSF tehnologiju za razvoj Java veb aplikacija.

# ▼ Poglavlje 1

## Uvod u JSF

### UVODNA RAZMATRANJA

*JSF je standarsna komponenta Java EE platforme.*

Pre pojave JSF okvira, većina Java veb aplikacija je razvijana primenom nestandardnih razvojnih okvira za veb aplikacije (koji ne predstavljaju integralno deo Java EE platforme) kao što su: Apache Struts, Tapestry i Spring Web MVC, kao i mnogi drugi. Navedeni okviri su izgrađeni na servlet i JSP standardima i automatizuju brojne funkcionalnosti koje bi morale, inače, ručno da se kodiraju kada bi se navedeni API - ji direktno koristili.

Uvođenje JSF u Java EE specifikaciju dalo je za rezultat standardan, veoma moćan aplikativni razvojni okvir dostupan u bilo kojem Java EE aplikativnom serveru. Sa standarizovanjem JSF kao integracionog dela Java EE platforme, brojni programeri Java Enterprise aplikacija upravo koriste JSF za kreiranje korisničkog interfejsa u aplikaciji.

Kao što je postavljeno kao praksa, u ovim materijalima će svako izlaganje biti praćeno odgovarajućim primerom. Tako će i u ovom slučaju, uvodna razmatranja biti podržana razvojem jednostavne JSF aplikacije.

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

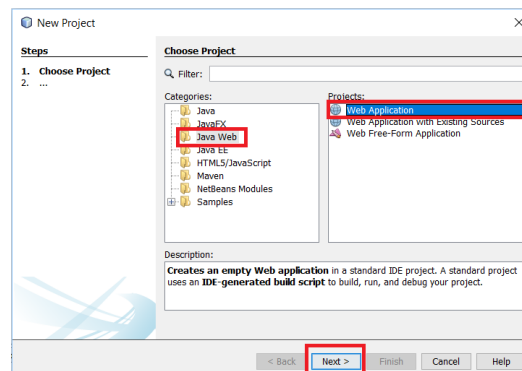
### PRIMER 1 - RAZVOJ JEDNOSTAVNE JSF APLIKACIJE

*Tekuća razmatranja će biti podržana razvojem jednostavne JSF aplikacije.*

Iz ugla aplikativnog programera JSF aplikacija se sastoji iz jedne ili više XHTML stranica koje sadrže JSF tagove, jedan ili više CDI zrna i opciono konfiguracione datoteke pod nazivom faces-config.xml.

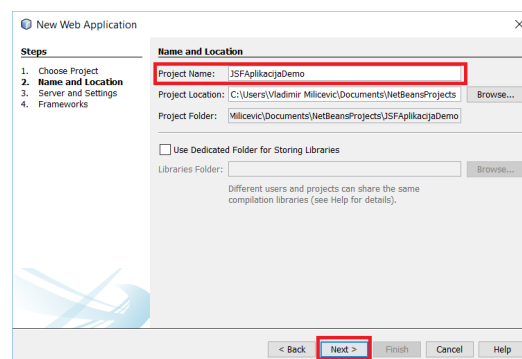
Starije verzije JSF su insistirale na primeni konfiguracija (JSF upravljanim zrnima) za implementaciju funkcionalnosti serverske strane. Zbog kompatibilnosti unazad, primena JSF konfiguracija je i dalje podržana ali je kao dobra praksa, i olakšanje, dodata podrška za anotacije (CDI zrna) koje se dominantno koriste u razvoju savremenih JSF veb aplikacija.

Za demonstraciju rada sa JSF projektima biće kreiran konkretan. U razvojnom okruženju (neka to bude NetBeans IDE) u meniju File i izborom opcije New Project, otvoriće se prozor u kome će biti izabrano da se razvija Java veb aplikacija (sledeća slika).



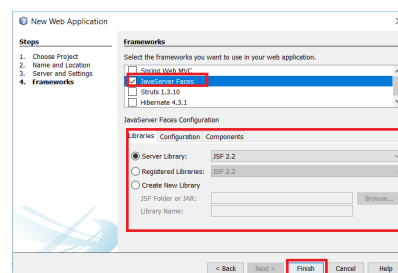
Slika 1.1 Započinjanje JSF projekta

Klikom na dugme **Next** otvara se nov prozor u kojem će biti određen naziv projekta, na primer **JSFAplikacijaDemo**.



Slika 1.2 Definisanje naziva projekta

Klikom na **Next**, otvoriće se prozor gde se bira verzija Java EE i tip aplikacionog servera. Klikom na Next se otvara poslednji prozor u kojem se bira JSF **framework** za razvoj kreiranog veb projekta.

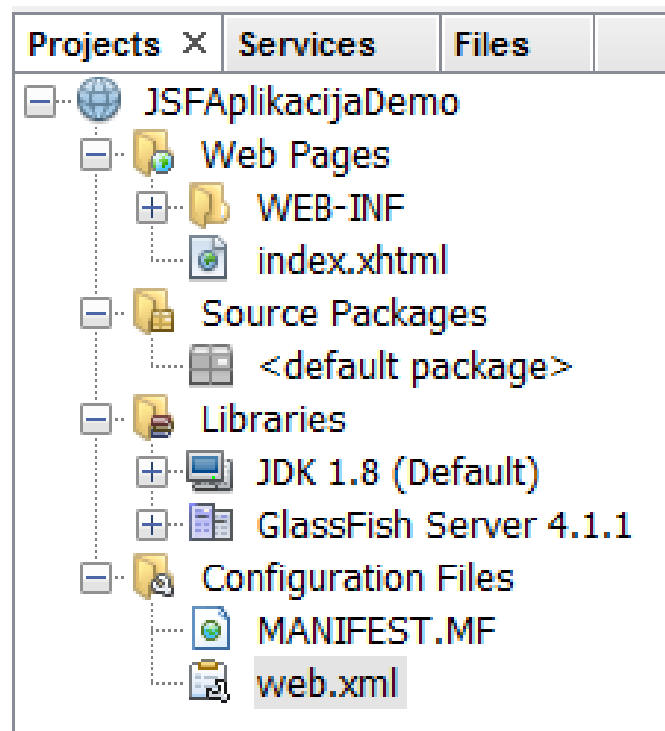


Slika 1.3 Izbor JSF razvojnog okvira

## STRUKTURA JSF PROJEKTA

*Klikom na **Finish**, nakon izbora JSF okvira, vrši se kreiranje projekta.*

Prethodne aktivnosti se završavaju klikom na dugme **Finish** prozora u okviru kojeg je izabran JSF okvir za korišćenje u tekućem primeru. Nakon toga razvojno okruženje je generisalo projekat sa strukturom koja je prikazana na sledećoj slici.



Slika 1.4 Struktura JSF projekta

U pokazanoj strukturi je moguće uočiti datoteku **web.xml** koja predstavlja standardni, opcioni konfiguracioni fajl kojeg koriste Java veb aplikacije. Fajl je postao opcionalan od verzije Servlet API 3.0 koji je uveden Java EE 6 platformom. U većini slučajeva, u savremenim Java veb aplikacijama, ovaj fajl više nije potreban budući da se podešavanja obavljaju putem anotacija. Međutim, za JSF aplikacije je dobra ideja da se kreira ova datoteka budući da se u okviru nje mogu specificirati faze razvoja aplikacije. Kreiranjem projekta, automatski je generisan sledeći kod za **web.xml**.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/
javaee/web-app_3_1.xsd">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
```

```

30
    </session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
</welcome-file-list>
</web-app>

```

## FAZE JSF PROJEKTA

*Razvojno okruženje je automatski podesilo JSF fazu projekta na **Development**.*

Kao što je moguće primetiti u prethodno priloženom kodu za deskriptor veb razvoja web.xml, razvojno okruženje je automatski podesilo JSF fazu projekta na vrednost **Development**. Na ovaj način razvojno okruženje je omogućilo primenu dopunskih alata za pomoć debugovanju koji nisu dostupni u ostalim fazama JSF projekta. Na primer, standardni problem kod razvoja JSF stranice može biti da, tokom razvoja stranice, proveru jednog ili više polja na stranici "pada" iz razloga što programer nije dodao neki od elemenata `<h:message>` ili `<h:messages>` u stranicu. Kada se ovo desi i odgovarajuća forma je potvrđena (**submit**) moguće je uočiti da se ništa ne dešava ili se navigacija između stranica ne obavlja na odgovarajući način. **Podešavanjem JSF faze projekta na vrednost Development, odgovarajuće greške validacije biće automatski dodate u stranicu, bez potrebe da ih programer eksplicitno dodaje u kod odgovarajućim tagovima.** Pre konačnog objavljivanja koda moguće je dodati odgovarajuće tagove u stranicu budući da korisnik, u suprotnom, neće biti u stanju da vidi automatski generisane informacije o greškama validacije.

U narednom izlaganju je neophodno navesti i obrazložiti dozvoljene vrednosti za `javax.faces.PROJECT_STAGE`:

- **Development**
- **Production**
- **SystemTest**
- **UnitTest.**

Kao što je već napomenuto u fazi **Development** se dodaju dopunske informacije koje pomažu prilikom debugovanja. Faza **Production** se bavi performansama. Ostale dve faze omogućavaju da se kroz testiranje uposli odgovarajuće ponašanje.

Klasa `javax.faces.application.Application` poseduje metodu `getProjectStage()` kojom je moguće dobiti informaciju o trenutnoj fazi JSF projekta. Navedeno je moguće ilustrovati sledećim kodom:

```

public void NekaMetoda() {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    Application application = facesContext.getApplication();
    ProjectStage projectStage = application.getProjectStage();
    if (projectStage.equals(ProjectStage.Development)) {

```



```
//faza razvoja
} else if (projectStage.equals(ProjectStage.Production)) {
//faza produkcije
} else if (projectStage.equals(ProjectStage.SystemTest)) {
// faza sistemskog testiranja
} else if (projectStage.equals(ProjectStage.UnitTest)) {
//faza jediničnog destiranja
}
}
```

Kao što je ilustrovano u prethodnom listingu, moguće je implementirati kod koji će se izvršiti u bilo kojoj od faza JSF projekta u zavisnosti od povratne vrednosti metode `getProjectStage()` klase `Application`.

## FACELET

*Facelet nije ništa drugo do XHTML datoteka sa specifičnim JSF tagovima.*

Sa kreiranjem projekta `facelet` je automatski generisan. Facelet nije ništa drugo do XHTML datoteka sa specifičnim JSF prostorom naziva ( `namespace`). Automatski generisan facelet ima sledeći oblik:

```
<?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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
  </h:body>
</html>
```

U ovom automatski generisanom kodu definicija prostora naziva koristi oznaku `"h"` (za HTML) JSF biblioteke komponenta. To je prikazano sledećim izolovanim kodom:

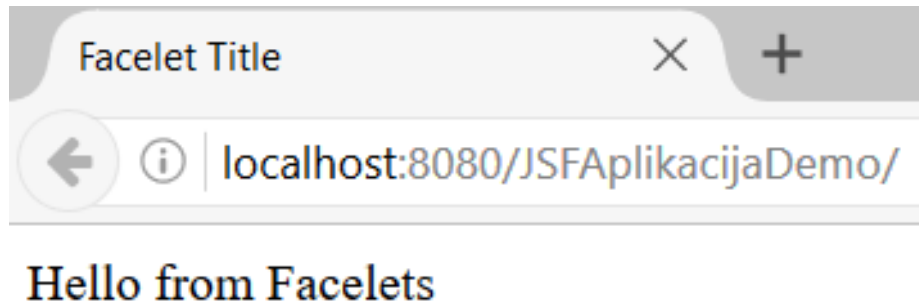
```
xmlns:h="http://xmlns.jcp.org/jsf/html"
```

Takođe, moguće je primetiti da su korišćeni i specijalni JSF tagovi `<h:head>` i `<h:body>` koji predstavljaju zamenu za standardne HTML/XHTML `<head>` i `<body>` tagove respektivno.

Još jedan prostor naziva koji se često koristi u JSF stranicama određen je prefiksom `"f"`. Ovom oznakom su određeni JSF tagovi koji se ne ugrađuju direktno u JSF stranice već specificiraju elemente drop - down listi, instance, `bind` akcije i tako dalje. Primer definisanja ovog prostora naziva je priložen sledećim izolovanim kodom:

```
xmlns:f= "http://xmlns.jcp.org/jsf/core"
```

Sada je moguće pokrenuti ovu najjednostavniju JSF veb aplikaciju. U razvojnom okruženju NetBeans IDE je dovoljno kliknuti na opciju **Run Project (ili F6)**, aplikacioni server se pokreće i aplikacija se angažuje. U veb pregledaču je moguće pogledati rezultate izvršavanja kreirane aplikacije. To je prikazano sledećom slikom.



Slika 1.5 Prva JSF aplikacija

## ZADATAK 1

### *Prvi facelet*

Na osnovu prethodnog primera, kreirajte prvi facelet.

## ▼ Poglavlje 2

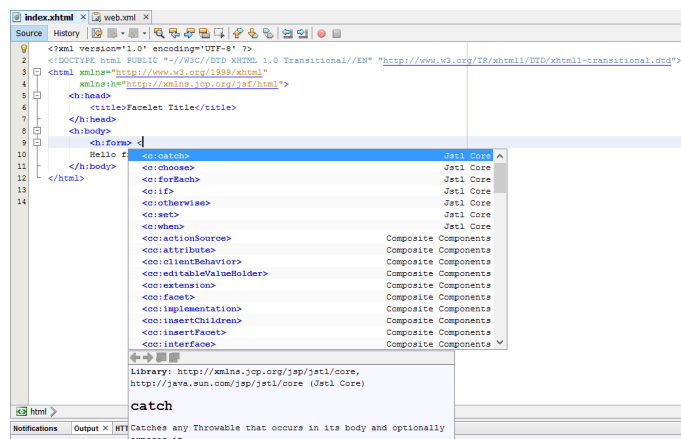
# Forme u JSF stranicama

## MODIFIKOVANJE AUTOMATSKI GENERISANE JSF STRANICE

*Cilj je konkretizovanje zadataka koje bi JSF stranica trebalo da obavi.*

Kada se kreira JSF projekat dobija se veoma prosta aplikacija koja u veb pregledaču prikazuje jedan string: "Hello from Facelets" i pored toga ne radi ništa konkretno. Za dobijanje više informacija o načinima funkcionisanja i elementima JSF stranica, moguće je izvršiti određene modifikacije na stranici [index.xhtml](#). Na primer, na ovoj stranici je moguće kreirati formu koja može da pokupi izvesne korisničke podatke.

Svaka JSF form se realizuje tagom `<h:form>` koji je ekvivalentan tagu `<form>` standardne HTML stranice. Kucanjem prvih nekoliko karaktera u kodu stranice za `<h:form>` razvojno okruženje, konkretno NetBeans, zapoćeće davanje sugestija o upotrebi mogućih tagova. Navedeno je pokazano sledećom slikom:



Slika 2.1 Početak kreiranja JSF forme

## PRIMER 2 - DODAVANJE FORME

*Sledi dodavanje forme na generisanu JSF stranicu*

Odmah po obeležavanju odgovarajućeg JSF taga razvojno okruženje prikazuje njegovu dokumentaciju (kao na slici). Sledećim kodom je moguće realizovati konkretnu formu u okviru JSF stranice:

```
<?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:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core">

  <h:head>

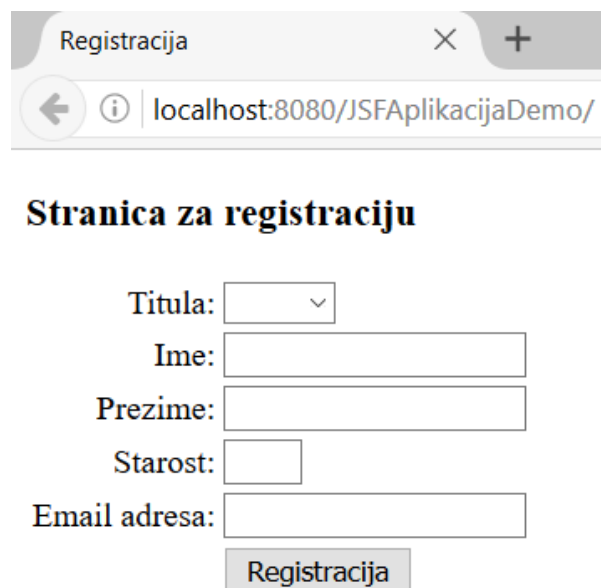
    <title>Registracija</title>
    <h:outputStylesheet name="styles.css"/>
  </h:head>
  <h:body>
    <h3>Stranica za registraciju</h3>
    <h:form>
      <h:panelGrid columns="3"
                  columnClasses="rightalign, leftalign, leftali
                                gn">
        <h:outputLabel value="Titula: " for="titula"/>
        <h:selectOneMenu id="titula" label="Titula"
                        value="#{registrationBean.
                                titula}" >
          <f:selectItem itemLabel="" itemValue=""/>
          <f:selectItem itemLabel="St." itemValue="ST"/>
          <f:selectItem itemLabel="Ma." itemValue="MA"/>
          <f:selectItem itemLabel="MSci" itemValue="MSCI"/>
          <f:selectItem itemLabel="Dr" itemValue="DR"/>
        </h:selectOneMenu>
        <h:message for="titula"/>
        <h:outputLabel value="Ime:" for="ime"/>
        <h:inputText id="ime" label="Ime"
                    required="true"
                    value="#{registrationBean.ime}" />
        <h:message for="ime" />
        <h:outputLabel value="Prezime:" for="prezime"/>
        <h:inputText id="prezime" label="Prezime"
                    required="true"
                    value="#{registrationBean.prezime}" />
        <h:message for="prezime" />
        <h:outputLabel value="starost" value="Starost:"/>
        <h:inputText id="starost" label="Starost" size="2"
                    value="#{registrationBean.starost}"/>
        <h:message for="starost"/>
        <h:outputLabel value="Email adresa:" for="email"/>
        <h:inputText id="email" label="Email adresa"
                    required="true"
                    value="#{registrationBean.email}"/>
        </h:inputText>
        <h:message for="email" />
        <h:panelGroup/>
        <h:commandButton id="registracija" value="Registracija"
                        action="potvrda" />
      </h:panelGrid>
    </h:form>
  </h:body>
</html>
```

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

## MENADŽER RASPOREDA I CSS

*Nophodno je u narednom izlaganju objasniti priloženi JSF kod.*

JSF kod, koji je priložen u prethodnoj sekciji, tokom vremena izvršavanja će generisati sledeću stranicu.



Slika 2.2 Kreirana JSF forma

Sva polja za unos podataka morala su da budu definisana u okviru JSF taga `<h:form>`. Takođe, stranica je veoma jednostavno uređena primenom menadžera rasporeda `<h:panelGrid>` o formi matrice (rešetke) koja je izdvojena na ćelije. Budući da se radi o dobro poznatom menadžeru rasporeda, o njemu ovde neće biti detaljno govora. Dovoljno je napomenuti da se svaka JSF komponenta pakuje u posebnu ćeliju, a da je atributom **columns**, taga `<h:panelGrid>`, određeno koliko će kolona matrica imati - u konkretnom slučaju 3. Kada broj komponenta u vrsti matrice dostigne broj definisanih kolona, nova vrsta za pakovanje novih elemenata automatski se kreira. Svaki red u matrici, koja uređuje redosled komponenta na kreiranoj JSF stranici, sastoji se od: `<h:outputLabel>` taga, polja za unos i `<h:message>` taga.

Atribut **columnClasses**, taga `<h:panelGrid>`, omogućava primenu CSS stilova na svaku kolonu iz matrice određene navedenim menadžerom rasporeda.

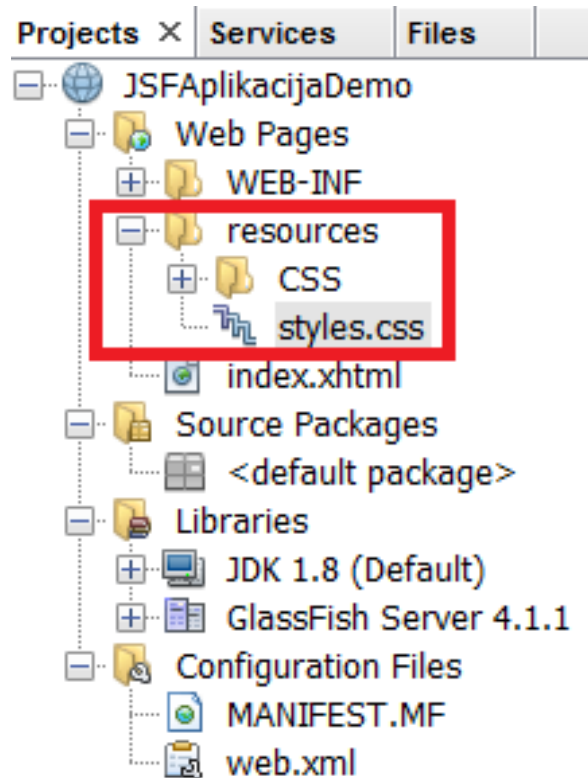
Vrednosti ovog atributa su razdvojene zarezima i definisane su u datoteci **styles.css** koja je u JSF stranicu uključena elementom `<h:outputStylesheet name="styles.css"/>`.

Pošto matrica ima tri kolone (pogledati kod) prvi stil se primenjuje na prvu, drugi na drugu, a treći na treću kolonu. Za slučaj postojanja četvrte kolone na nju bi bio primenjen prvi, na petu drugi stil i tako dalje. Stilove je moguće primeniti i na vrste, ali bi u tom slučaju bio angažovan atribut **rowClasses**.

## CSS DOPUNSKA RAZMATRANJA

*Tag za uključivanje CSS datoteke u JSF stranicu je dostupan od JSF verzije JSF 2.0.*

Obavezan položaj CSS datoteke, u Web Pages folderu projekte, prikazan je sledećom slikom.



Slika 2.3 CSS datoteka u JSF projektu

Tag za uključivanje CSS datoteke u JSF stranicu je dostupan od JSF verzije JSF 2.0. Resursi, kao što su **CSS, JavaScript datoteke i slike**, mogu biti čuvani u **top - level** direktorijumu pod nazivom **resources**. Ovim resursima JSF tagovi mogu automatski da pristupaju. Da bi kreirana JSF stranica mogla da pristupa datoteci **styles.css**, kreiran je prvo pomenuti folder **resources**, a onda u okviru njega odgovarajući podfolder **CSS**. **Datoteka sa stilovima je smeštena u folder CSS**.

Tag **<h:outputStylesheet>** može da sadrži i atribut **library** čija vrednost mora da se podudara sa lokacijom na kojoj se CSS fajl čuva. Vrednost atributa **name** mora da odgovara nazivu CSS datoteke.

Po analogiji, ukoliko bi u projektu bile korišćene JavaScript datoteke i slike, datoteke bi bile čuvane u podfolderima foldera **resources**, **javascript** i **images**, respektivno.

# JSF ELEMENTI

*U narednom izlaganju će biti opisani JSF elementi koji su korišćeni u primeru.*

Kada se pogleda priloženi kod moguće je uočiti brojne JSF elemente kojima je realizovana forma, odnosno odgovarajući korisnički interfejs koji je dostavljen korisniku učitavanjem kreirane JSF stranice u veb pregledač.

Tagom `<h:outputLabel>` su određene labele koje označavaju polja za unos teksta koja postoje na formi stranice. Vrednost za atribut **for** ovog taga, odgovara vrednosti **id** odgovarajućeg polja za unos teksta.

```
<h:outputLabel value="Titula: " for="titula"/>
```

Tagom `<h:message>` je generisana greška unosa za tekst polje. Vrednost za atribut **for** ovog taga, odgovara vrednosti **id** odgovarajućeg polja za unos teksta.

```
<h:message for="titula"/>
```

Na formi je moguće uočiti i jednu kontrolu koja je u brojnoj literaturi i implementacijama poznata kao **padajuća lista (padajući meni)** ili **ComboBox**. Ovom kontrolom se bira "titula" osobe čiji se podaci unose u formu. Kontrola je realizovana JSF tagom `<h:selectOneMenu>` koji je ekvivalentan standardnom HTML tagu `<select>`.

Dakle, moguće je primetiti da brojni JSF tagovi sadrže atribut **id**. Ovaj atribut je predstavljen stringom koji jednoznačno označava tag pomoću identifikatora. **Ukoliko programer nije specificirao vrednost za atribut id, ona će automatski biti generisana.**

Prilikom korišćenja taga `<h:label>` za definisanje polja za unos teksta ili taga `<h:message>` za definisanje greške validacije, neophodno je eksplicitno definisati **id** atribut polja za unos koji mora da se podudara sa atributima **for** navedenih tagova.

Svaki JSF tag za unos sadrži atribut **label**. Ovaj atribut se koristi za generisanje greške provere na kreiranoj stranici. Ukoliko ovaj atribut nije specificiran za polje, ono će automatski biti prepoznato, u poruci greške, na osnovu njegovog **id** atributa.

Takođe, Svaki JSF tag za unos sadrži atribut **value**. U slučaju taga `<h:selectOneMenu>` ovaj atribut ukazuje na to koja će vrednost iz generisanog HTML taga `<select>` biti izabrana. Vrednost ovog atributa mora da se podudara sa vrednošću **itemValue** atributa nekog od ugnježenih `<f:selectItem>` tagova, unutra taga `<h:selectOneMenu>`.

Vrednost **itemLabel** atributa taga `<f:selectItem>` predstavlja vrednost koju će korisnik moći da vidi u listi (meniju) iz koje vrši izbor. Kada korisnik izabere tu vrednost, vrednost određena atributom **itemValue** se šalje na server kada se forma popuni i potvrdi.

## JSF ELEMENTI - NASTAVAK

*Neki od*

*&amp;&amp;&amp;&amp;&amp;&amp;&lt;h:inputText&amp;&amp;&amp;&amp;&*

Neki od `<h:inputText>` tagova zahtevaju vrednost. Ovi tagovi poseduju attribute **required** postavljene na vrednost **true**. To znači da je obaveza korisnika da unese vrednost u polje koje odgovara ovom JSF elementu. Ovaj atribut je opcionalan i ukoliko nije obavezan unos u polje on može da se izostavi ili da bude podešen na **false** (ovo je i podrazumevana vrednost atributa). Navedeno je pokazano sledećim izolovanim listingom:

```
<h:inputText id="email" label="Email adresa"
              required="true"
              value="#{registrationBean.email}">
</h:inputText>
```

Ako se dalje pogleda priloženi kod moguće je primetiti da je tag `<h:panelGroup/>` dodat u poslednji red matrice koja određuje raspored GUI elemenata. Svha navedenog je dodavanje više JSF tagova u jednu ćeliju `<h:panelGrid>` rasporeda. U konkretnom slučaju na ovaj način je omogućeno preskakanje jedne ćelije (dodavanjem prazne ćelije), dodavanje kontrole `<h:commandButton>` i njeno poravnanje sa poljima za unos teksta na kreiranoj formi. Zadatak dugmeta je da potvrdi popunjenu formu i omogući slanje podataka na server. Atribut dugmeta **value** ima zadatak da pokaže korisniku tekst na kreiranom dugmetu, a atribut **action** je zadužen za određivanje stranice koja će biti prikazana nakon klika na dugme.

U konkretnom slučaju, radi se o statičkoj navigaciji. Klikom na dugme "Registracija" ka datoteci pod nazivom `potvrda.xhtml`.

```
<h:commandButton id="registracija" value="Registracija"
                  action="potvrda" />
```

```
<h:commandButton id="registracija" value="Registracija"
                  action="potvrda" />
```

Alternativno, moguće je koristiti dinamičku navigaciju. U tom slučaju, vrednost atributa **action**, dugmeta za potvrdu forme, odgovara vrednosti izraza koji odgovara metodi i povratnom stringu CDI zrna. To znači da povratna vrednost navedene i izvršene metode mora da odgovara nazivu stranice ka kojoj se vrši navigacija.

## ZADATAK 2

*Kreiranje forme*

Kreirajte vlastitu JSF stranicu sa formom po slobodnom izboru.



## ▼ Poglavlje 3

# Kreiranje CDI zrna

## DODAVANJE KLASA ZRNA U JSF APLIKACIJU

*CDI zrna prihvataju korisničke podatke.*

CDI (Context and Dependency Injection) zrna su klasična Java zrna čiji je zadatak prihvatanje podataka koje je korisnik uneo u JSF aplikaciji. Budući da se radi Java klasi, njeno kreiranje se odvija po istom scenariju kao i za ostale Java klase.

U razvojnom okruženju, desnim klikom na **Source Package** aktuelnog projekta, se bira opcija **New** i nakon toga se bira opcija **Java Class**. Otvara se prozor u kojem se unosi naziv klase i paketa: **RegistrationBaan** i **com.metropolitan**, respektivno. Generisana je prazna klasa priložena sledećim listingom:

```
package com.metropolitan;

/**
 *
 * @author Vladimir Milicevic
 */
public class RegistrationBean {

}
```

Da bi kreirana Java klasa postala šablon za CDI (obeleženo, imenovano) zrno neophodno je izvršiti obeležavanje klase anotacijom **@Named**. Po osnovnim podešavanjima naziv CDI zrna u potpunosti odgovara nazivu klase sa razlikom što počinje mail slovom. Dakle, ovoj klasi odgovara CDI zrno **registrationBean** kojeg je moguće uočiti u priloženom kodu datoteke **index.xhtml**. Podrazumevano ime može biti i redefinisano kada se u anotaciji **@Named** podesi atribut **value** na željenu vrednost.

CDI zrna mogu pokrивati različite oblasti (sledeća tabela):

Oblast	Anotacija
Request	@RequestScoped
Session	@SessionScoped
Conversation	@ConversationScoped
Application	@SessionScoped
Dependent	@Dependent
Flow	@FlowScoped

Slika 3.1 Oblasti CDI zrna

Sledi objašnjenje za pomenute oblasti i odgovarajuće anotacije:

- **Request** - znači da je zrna dostupno isključivo u jednom HTTP zahtevu;
- **Session** - zrna dostupno isključivo u jednoj korisničkoj HTTP sesiji;
- **Conversation** - zrna dostupno preko više HTTP zahteva;
- **Application** - zrna je dostupno svim korisnicima aplikacije preko njihovih sesija;
- **Dependent** - zrna može imati zavisan pseudo - kod, a to podrazumeva umetanje nove instance svaki puta kada se zahteva;
- **Flow** - zrna je dostupno kroz specifične JSF tokove.

## PRIMER 3 - KODIRANJE KLASA CDI ZRNA

*Na osnovu prethodnih izlaganja neophodno je konkretizovati definiciju kreirane klase.*

Ako se vrati fokus na prethodno izložene anotacije trebalo bi napomenuti i da su sve anotacije, izuzev **@FlowScoped**, definisane u okviru paketa **javax.enterprise.context**. Anotacija **@FlowScoped** je definisana unutar paketa pod nazivom **javax.faces.flow**. Dakle, neophodno je kreiranu klasu obeležiti odgovarajućim anotacijama i obezbediti odgovarajuće **import** instrukcije za podršku. Kada se dodaju anotacije, trebalo bi imati na umu da klasa implementira CDI zrna (**@Named**) i da osluškuje zahtev korisnika koji popunjava formu (**@RequestScope**).

Privremeni oblik klase **RegistrationBean** priložen je sledećim listingom:

```
package com.metropolitan
;
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

@Named
@RequestScoped
public class RegistrationBean {
}
```

Ukoliko razvojno okruženje ne može da pronađe anotaciju `@RequestScope`, neophodno je, desnim klikom na Libraries, izborom opcije Add JAR /Folder, izabrati `cdi-api.jar` iz foldera modules GlassFish instalacije.

Konačno, moguće je konkretizovati i finiširati definiciju posmatrane klase dodavanjem osobina koje će čuvati vrednosti koje je korisnik uneo.

Sledećim listingom je priložen konačan kod klase koja odgovara CDI zrnu:

```
package com.metropolitan;

/**
 *
 * @author Vladimir Milicevic
 */
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

@Named
@RequestScoped
public class RegistrationBean {
    private String titula;
    private String ime;
    private String prezime;
    private Integer staros;
    private String email;
    //geteri i seteri su izostavljeni zbog preglednosti
    //bice prilozeni u materijalima za vezbe
}
```

Moguće je primetiti da nazivi osobina zrna u potpunosti odgovaraju nazivima korišćenim u JSF stranici `index.xhtml` za podatke koje unosi korisnik. Ovi nazivi moraju da se podudaraju da bi JSF stranica znala koje vrednosti može da mapira.

## ZADATAK 3

### *Kreiranje vlastitog CDI zrna*

Kao nastavak vlastitog primera, započetog u Zadatak 2, kreirajte vlastito CDI zrno za upravljanje podacima koje unosite na kreiranoj JSF stranici.

## ▼ Poglavlje 4

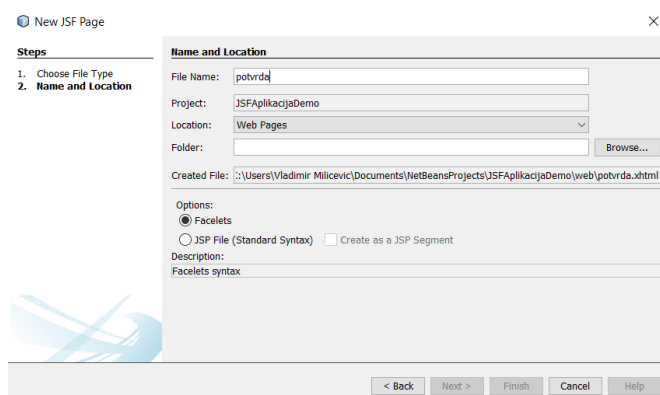
# JSF stranica za potvrđivanje

## KREIRANJE STRANICE ZA POTVRĐIVANJE

### *Kreiranje stranice za prikaz unetih korisničkih podataka.*

Kada korisnik unese sve potrebne podatke na početnoj stranici i potvrdi unos na formi stranice, neophodno je na odgovarajućoj stranici za potvrđivanje prikazati podatke koje je korisnik uneo. Vrednosti koje odgovaraju atributima **values** svakog polja za unos teksta forme, biće pridruženi odgovarajućim osobinama kreiranog CDI zrna. U nastavku, zadatak je jednostavan. Stranica za potvrđivanje će biti generisana kao serija `<h:outputText>` JSF tagova koji prikazuju vrednosti koje kreirano CDI zrno čuva.

U razvojnom okruženju, u konkretnom slučaju NetBeans, klikom na **File**, a zatim izborom opcije **New File**, pa kategorije **JavaServer Faces** i tipom datoteke **JSF Page**, otvara se prozor za kreiranje nove JSF stranice kojoj će biti dodeljen naziv **potvrda.xhtml** (sledeća slika).



Slika 4.1 Nova JSF stranica

Još jednom je neophodno proveriti da li dodeljeni naziv ovoj datoteci odgovara vrednosti atributa **action** dugmeta forme koja se nalazi na stranici **index.xhtml**. U nastavku, automatski generisana stranica je modifikovana u skladu sa istaknutim zahtevima u prethodnom izlaganju. Nakon modifikacije stranicu sačinjava kod koji je priložen sledećim listingom:

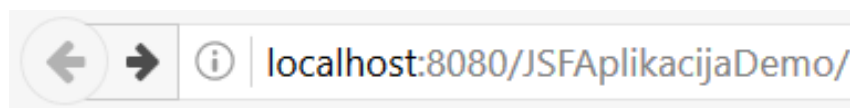
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Stranica za potvrdu</title>
    <h:outputStylesheet name="styles.css"/>
```

```
</h:head>
<h:body>
  <h2>Stranica za potvrdu</h2>
  <h:panelGrid columns="2"
    columnClasses="rightalign-bold,leftalign">
    <h:outputText value="Titula:"/>
    ${registrationBean.titula}
    <h:outputText value="Ime:"/>
    ${registrationBean.ime}
    <h:outputText value="Prezime:"/>
    ${registrationBean.prezime}
    <h:outputText value="Starost:"/>
    ${registrationBean.starost}
    <h:outputText value="Email adresa:"/>
    ${registrationBean.email}
  </h:panelGrid>
</h:body>
</html>
```

## PRIMER 4 - PRIKAZIVANJE UNETIH PODATAKA

*Na kraju je neophodno proveriti funkcionalnost kreirane JSF aplikacije.*

Sa stranicom za potvrđivanje unetih podataka zaokružena je konstrukcija kreirane veb aplikacije. Najlakši način da se aplikacija testira je klik na **Run** opciju u razvojnom okruženju. Aplikacija se pokreće, učitava se stranica **index.xhtml** sa odgovarajućom formom koju je neophodno popuniti kao na sledećoj slici.



### Stranica za registraciju

Titula:

Ime:

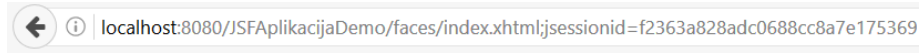
Prezime:

Starost:

Email adresa:

Slika 4.2 Popunjena forma sa početne stranice

Klikom na dugme pod nazivom "Registracija", svi uneti podaci se poveravaju na čuvanje CDI zrnju čija klasa predstavlja sastavni deo aplikacije. Zrno se ponaša kao posrednik između dve stranice, za registraciju i potvrđivanje. Stranica za potvrđivanje, određena je vrednošću `action`, dugmeta "Registracija" dobija od zrna odgovarajuće podatke i prikazuje ih na ekranu na način prikazan sledećom slikom.



### Stranica za potvrdu

**Titula:** DR  
**Ime:** Vladimir  
**Prezime:** Milicevic  
**Starost:** 42  
**Email adresa:** vladimir.milicevic@metropolitan.ac.rs

Slika 4.3 Prikazivanje podataka na stranici za potvrđivanje

Ukoliko se u veb pregledaču pojave podaci na način prikazan prethodnom slikom, posao je uspešno obavljen i može se preći na izučavanje naprednijih JSF tema.

## ZADATAK 4

### *Kreirajte vlastitu stranicu za potvrdu*

Nastavite vaš primer i kreirajte novu JSF stranicu koja prikazuje podatke koje ste uneli na početnoj formi.

## ▼ Poglavlje 5

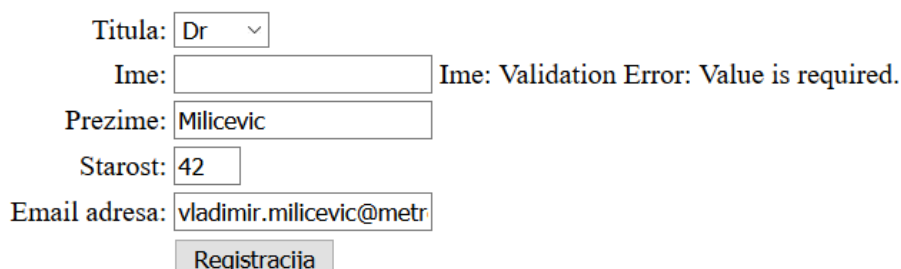
# JSF Validacija

## VALIDACIJA U JSF STRANICAMA

*Greške validacije, za obavezna polja unosa, automatski se generišu.*

U prethodnim izlaganjima je bilo govora o jednom posebnom atributu pod nazivom **required**. Ovaj atribut je od posebnog značaja kada su u pitanju polja za unos teksta kreiranih formi neke JSF stranice. Atribut označava polja koja su obavezna za unos i ukoliko neko od njih nije popunjeno javiće se greška kao na sledećoj slici.

### Stranica za registraciju

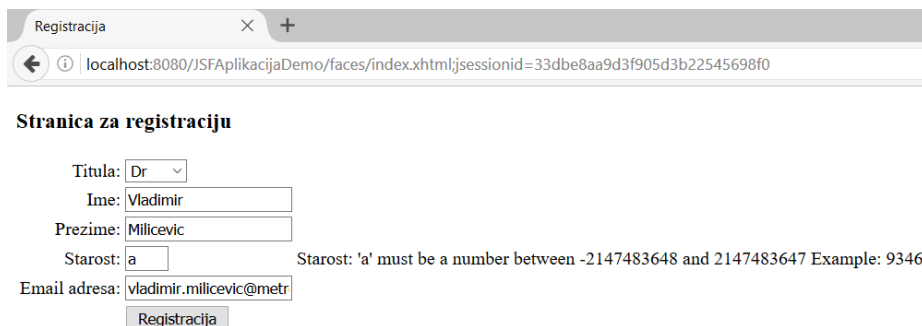


Titula:    
Ime:  Ime: Validation Error: Value is required.  
Prezime:   
Starost:   
Email adresa:

Slika 5.1 Obavezno polje nije popunjeno

Za generisanje poruke sa informacijom o grešci zadužen je JSF tag `<h:message>` odgovarajućeg polja koje je obavezno, a nije popunjeno prilikom potvrđivanja forme.

Dalje, ako se fokusiramo na polje forme, zaduženo za unos podatka kojim je predstavljena starost korisnika, i u njega unesemo podatak koji nije tipa **Integer**, javiće se sledeća greška validacije koja je, takođe, automatski generisana.



Registracija

localhost:8080/JSFAplikacijaDemo/faces/index.xhtml;jsessionid=33dbe8aa9d3f905d3b22545698f0

### Stranica za registraciju

Titula:    
Ime:   
Prezime:   
Starost:  Starost: 'a' must be a number between -2147483648 and 2147483647 Example: 9346  
Email adresa:

Slika 5.2 Unos podatka neodgovarajućeg tipa podataka

Ovde je moguće uočiti jedan nedostatak kreirane aplikacije. Ako se za godine unese negativan broj, to će za aplikaciju biti u redu, jer se uklapa u predviđeni tip podataka, ali u realnim okolnostima to nema nikakvog smisla.

Takođe, kao vrednost za email adresu će proći bilo koji **String** objekat bez ikakve provere da li je on korektno unet. Ne postoji ugrađen mehanizam validacije za proveru korektnosti unosa email adrese.

Upravo u nastavku, neophodno je pokazati mehanizme i načine koji obezbeđuju da forma pokupi pravilno unete i formatirane podatke i kao takve da ih nakon potvrđivanja pošalje na server.

## PRIMER 5 - KLASA VALIDATOR

*Klasa validator mora da implementira interfejs `javax.faces.validator.Validator`.*

Da bi forma pokupila podatke koji su pravilno formatirani i odgovarajućeg tipa, neophodno je u projekat dodati `validator` klasu. Ova klasa mora da implementira interfejs `javax.faces.validator.Validator`. Navedeni interfejs sadrži jednu metodu `validate()` koja preuzima sledeća tri parametra, instance klase: `javax.faces.context.FacesContext`, `javax.faces.component.UIComponent` (sadrži JSF komponentu koja se proverava) i `java.lang.Object` (sadrži vrednost koju je korisnik uneo za komponentu), respektivno.

Sledećim listingom je moguće ilustrovati email validator:

```
package com.metropolitan;

/**
 *
 * @author Vladimir Milicevic
 */
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.component.html.HtmlInputText;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator(value = "emailValidator")
public class EmailValidator implements Validator {

    @Override
    public void validate(FacesContext facesContext,
                        UIComponent uiComponent, Object value) throws
        ValidatorException {
        Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");
```



```

    Matcher matcher = pattern.matcher(
        (CharSequence) value);
    HtmlInputText htmlInputText
        = (HtmlInputText) uiComponent;
    String label;
    if (htmlInputText.getLabel() == null
        || htmlInputText.getLabel().trim().equals("")) {
        label = htmlInputText.getId();
    } else {
        label = htmlInputText.getLabel();
    }
    if (!matcher.matches()) {
        FacesMessage facesMessage
            = new FacesMessage(label
                + ": email adresa nije ispravna");
        throw new ValidatorException(facesMessage);
    }
}
}

```

U ovom primeru metoda `validate()` vrši upoređivanje regularnog izraza sa vrednošću JSF komponente koja se proverava. Ako je prisutno poklapanje, validacija je uspešna. U suprotnom, biće izbačena instanca izuzetka `javax.faces.validator.ValidatorException`.

Takođe, važno je napomenuti da klasa validator mora da bude obeležena anotacijom `@FacesValidator`. Vrednost anotacije predstavlja *id* koji će biti upotrebljen da referencira na validator - a u JSF stranici.

Da bi ova klasa obavila svoju ulogu, neophodno je vratiti se u datoteku `index.xhtml`, i za polje za unos teksta email, obaviti modifikaciju koja je priložena sledećim listingom. Ukoliko email adresa nije korektno uneta, JSF stranica će reagovati kao na slici broj 4.

```

<h:inputText id="email" label="Email Address"
    required="true" value="#{registrationBean.email}">
<f:validator validatorId="emailValidator"/>
</h:inputText>

```

The screenshot shows a web browser window with the title 'Registracija'. The address bar shows 'localhost:8080/JSFAplikacijaDemo/faces/index.xhtml'. The page content is titled 'Stranica za registraciju'. It contains a form with the following fields: 'Titula:' with a dropdown menu showing 'Dr', 'Ime:' with a text input containing 'Vladimir', 'Prezime:' with a text input containing 'Milicevic', and 'Starost:' with a text input containing '42'. Below these is the 'Email adresa:' field with a text input containing 'vladimir.milicevic'. To the right of this field is a red error message: 'Email adresa: email adresa nije ispravna'. At the bottom of the form is a 'Registracija' button.

Slika 5.3 Angažovanje email validatora

## ZADATAK 5

### Kreiranje validatora

Kreirajte vlastiti validator koji proverava polja forme iz Zadatak 2.

## ▼ Poglavlje 6

# Facelet šabloni

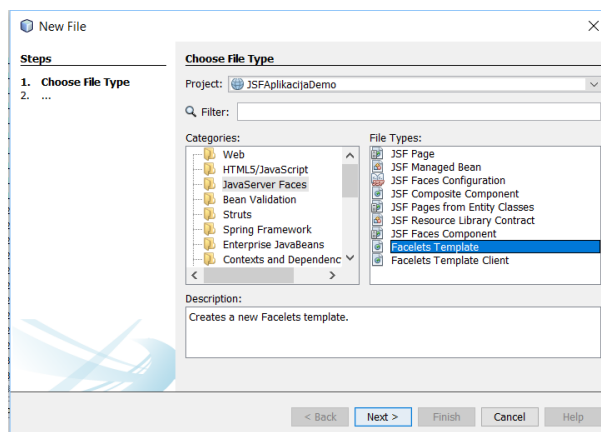
## MEHANIZAM ŠABLONA

*Prednost Facelet - u odnosu na JSP ogleda se u primeni mehanizma šablona.*

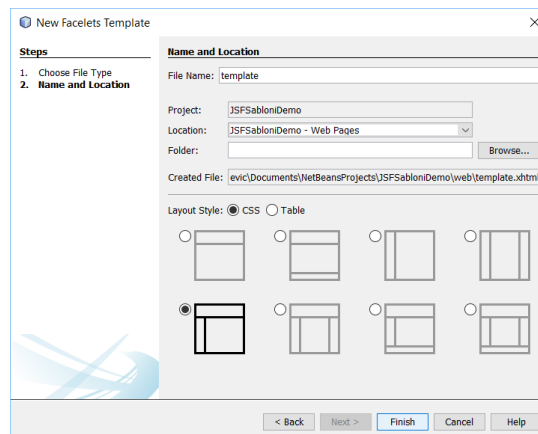
Prednost Facelet - u odnosu na JSP ogleda se u primeni mehanizma šablona. Šabloni dozvoljavaju da se izgled i raspored JSF stranice specificira na jednom mestu. Takođe, primenom šablona postiže se veći stepen održivosti kreirane veb aplikacije budući da se sve izmene u rasporedu komponenata stranice dešavaju na jednom mestu. Ukoliko se u jednoj tački stranice traži sprovođenje neke izmene, kao što je na primer dodavanje futera ili pomeranje kolone sa leve na desnu stranu stranice, neophodno je izvršiti samo izmenu u šablonu koja će se reflektovati na sve klijente šablona.

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

Neophodno je, u nastavku, demonstrirati kako se dodaje facelet šablon u JSF projekat. Klikom na opciju **File**, u razvojnom okruženju NetBeans, otvara se meni u okviru kojeg se bira opcija **New File**. Otvara se novi prozor pod nazivom **New File** gde se bira kategorija **JavaServer Faces** i konačno, u istom prozoru bira se za tip datoteke Facelets Templates. Navedeni prozor i izbor iz njega je prikazan slikom broj 1. Dalje, klikom na **Next** otvara se prozor u okviru kojeg se definiše naziv šablona i bira neki od predefinisanih šablona (Slika 2)



Slika 6.1 Dodavanje Facelet šablona u projekat



Slika 6.2 Prozor New Facelets Template

## DATOTEKA ŠABLONA

*Kreiranu datoteku šablona je nophodno konkretizovati odgovarajućim modifikacijama koda.*

Na prethodno priloženoj slici moguće je bilo primetiti da razvojno okruženje **NetBeans IDE** nudi mogućnost izbora između HTML tabela i CSS - a za određivanje rasporeda. Većina savremenih veb aplikacija podrazumeva CSS kao prirodan izbor. Neka je, u ovom slučaju, izbor pao na raspored koji uključuje polje zaglavlja (**header**), kolonu na levoj strani i glavno polje (izbor je prikazan Slikom 2). Konačno, u prozoru **New Facelets Template**, klikom na **Finish** se automatski generiše kod šablona koji je priložen sledećim listingom:

```
<?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://xmlns.jcp.org/jsf/facelets"
      xmlns:h="http://xmlns.jcp.org/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>Facelets Template</title>
  </h:head>

  <h:body>

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

```
</div>
<div id="content" class="left_content">
  <ui:insert name="content">Content</ui:insert>
</div>
</div>
</h:body>

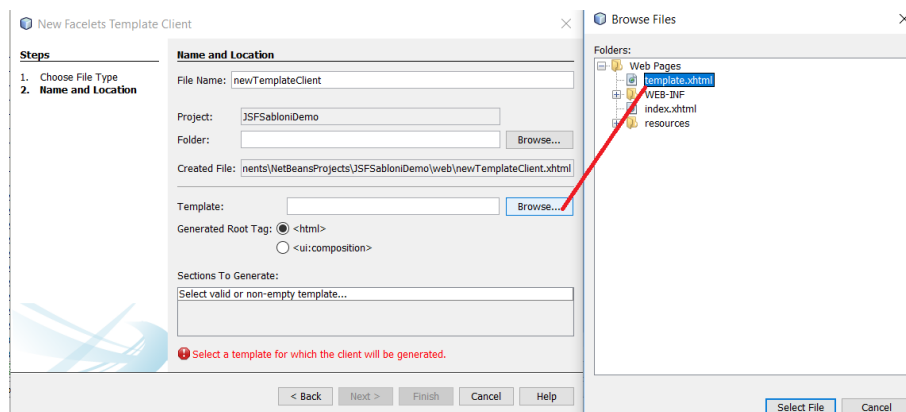
</html>
```

Na prvi pogled je moguće primetiti da se šablon ne razlikuje mnogo od standardne JSF stranice. Takođe, moguće je primetiti da koristi prostor naziva `xmlns:ui=http://xmlns.jcp.org/jsf/facelets` koji omogućava primenu JSF taga `<ui:insert>`. Sadržaj ovog taga biće zamenjen sadržajima odgovarajućih `<ui:define>` tagova klijenata šablona.

## PRIMER 6 - UPOTREBA ŠABLONA

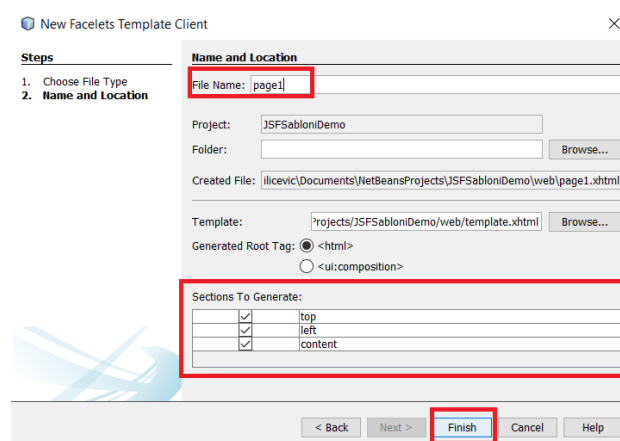
*Za upotrebu šablona neophodna je implementacija klijenata šablona.*

Da bi kreirani šablon mogao da bude upotrebljen neophodno je kreirati odgovarajući **Facelets template client**. Postupak je veoma sličan prethodnom. Klikom na opciju **File**, u razvojnom okruženju NetBeans, otvara se meni u okviru kojeg se bira opcija **New File**. Otvara se novi prozor pod nazivom **New File** gde se bira kategorija **JavaServer Faces** i konačno, u istom prozoru bira se za tip datoteke **Facelets Template Clients**. Klikom na **Next** otvara se prozor u okviru kojeg se definiše naziv datoteke klijenta šablona, a takođe se vrši dodeljivanje kreiranog šablona ovoj datoteci (sledeća slika).



Slika 6.3 Dodela kreiranog šablona klijentu šablona

U nastavku je neophodno završiti definiciju ove datoteke. Poslednji koraci su zadavanje njenog naziva, na primer **page1**, i klik na dugme **Finish**. Navedeni elementi su obeleženi na sledećoj slici, zajedno sa učitanim šablonom.



Slika 6.4 Završetak definicije klijenta šablona.

Kao posledica ovih akcija javlja se sledeći automatski generisan kod:

```
<?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://xmlns.jcp.org/jsf/facelets">

  <body>

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

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

    </ui:composition>

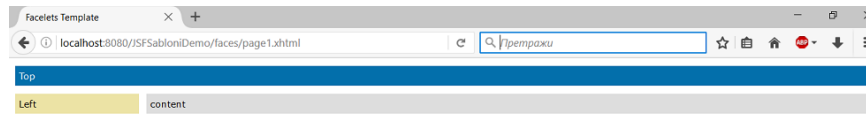
  </body>
</html>
```

## UPOTREBA ŠABLONA - DODATNA RAZMATRANJA

*Aplikacija se testira i uočava se ugrađeni šablon rasporeda stranice.*

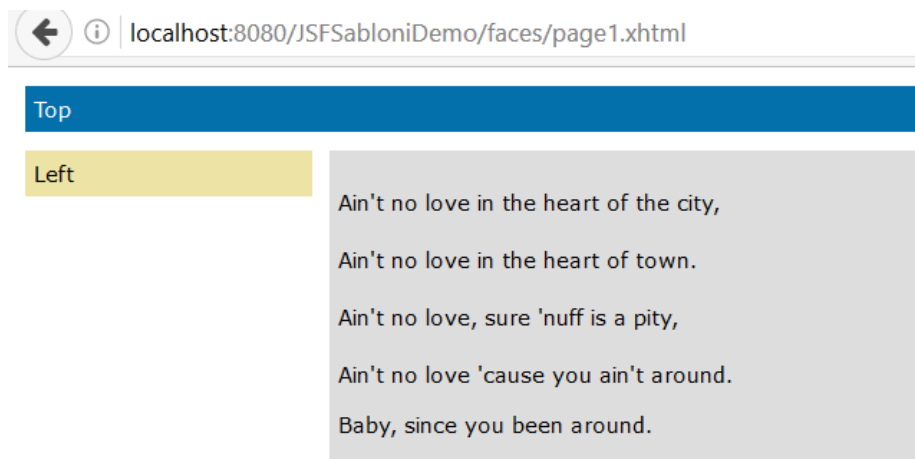
Kao što je moguće primetiti iz prethodno priloženog koda, kreirana datoteka klijenta šablona takođe uključuje prostor naziva `xmlns:ui="http://xmlns.jcp.org/jsf/facelets"`. Dalje, neophodno je napomenuti da u ovoj datoteci tag `<ui:composition>` mora da bude roditeljski tag za bilo koji drugi tag koji pripada ovom prostoru naziva. Sve što se nalazi izvan ovog taga neće biti postavljeno u stranicu i umesto toga će se pojaviti odgovarajući sadržaj šablona. Tagom `<ui:define>`, klijenta šablona, se umeće sadržaj koji odgovara sadržaju taga `<ui:insert>` šablona. Vrednost atributa **name** u tagu `<ui:define>` mora da se podudara sa odgovarajućom vrednošću u `<ui:insert>` tagu šablona.

Sada je moguće testirati kreirane datoteke projekta. Nakon prevođenja, primenom razvojnog okruženja, u veb pregledaču, u polju za unos, poziva se sledeći link <http://localhost:8080/JSFSablioniDemo/faces/page1.xhtml> i rezultat poziva je priložen sledećom slikom.



Slika 6.5 Primena šablona

Sada je moguće uneti izmene u klijent šablona i posmatrati kako će se ponašati kreirana stranica page1. Između tagova `<ui:define name = "content">` i `</ui:define>` ubačen je deo teksta jedne pesme i rezultat se vidi na sledećoj slici i u kodu koji sledi iza slike.



Slika 6.6 Zamena vrednosti content konkretnim sadržajem

```
<?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://xmlns.jcp.org/jsf/facelets">

  <body>

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

      <ui:define name="content">
        <br>Ain't no love in the heart of the city,</br>
        <br>Ain't no love in the heart of town.</br>
        <br>Ain't no love, sure 'nuff is a pity,</br>
        <br>Ain't no love 'cause you ain't around.</br>
        <p>Baby, since you been around.</p>
      </ui:define>
    </ui:composition>
  </body>
</html>
```

```

        </ui:define>

    </ui:composition>

</body>
</html>

```

## KOREKCIJE U DATOTECI ŠABLONA

*Svaka korekcija u šablonu reflektovaće se na klijente šablona.*

U prethodnoj aktivnosti izvršeno je korigovanje datoteke `page.xhtml` na način da ona prikazuje određeni tekst. Sada je moguće uraditi i određene korekcije na samom šablonu `template.xhtml` i posmatrati kako će se to reflektovati na klijent šablona. Budući da klijent šablona ne poseduje elemente koje odgovaraju `top` i `left` sekcijama šablona, njihovo redefinisanje je neophodno izvršiti u samom šablonu.

Top sekcija može biti redefinisana na sledeći način:

```

<div id="top" class="top">
    <ui:insert name="top">
        <h2>Malo RnR lirike</h2>
    </ui:insert>
</div>

```

Left sekcija može biti redefinisana na malo složeniji način:

```

<div id="left">
    <ui:insert name="left"><h3>Links</h3>
        <ul>
            <li>
                <h:outputLink value="http://www.whitesnake.com/">
                    <h:outputText value="Whitesnake Official Web Site"/>
                </h:outputLink>
            </li>
            <li>
                <h:outputLink value="http://www.azlyrics.com/w/
whitesnake.html">
                    <h:outputText value="Whitesnake Lyrics"/>
                </h:outputLink>
            </li>
            <li>
                <h:outputLink value="https://www.youtube.com/
watch?v=MA3DNlBMNL0">
                    <h:outputText value="Jedna lepa pesma"/>
                </h:outputLink>
            </li>
        </ul>
    </ui:insert>
</div>

```



```
</ul></ui:insert>
</div>
```

Budući da klijent šablone ne redefiniše sekcije top i left, naslediće ih direktno od šablona. Nakon navedenih korekcija u šablonu i ponovnog prevođenja kreirane veb aplikacije, pozivom linka stranice **page1.xhtml**: <http://localhost:8080/JSFSablonaDemo/faces/page1.xhtml>, moguće je primetiti sasvim nov sadržaj stranice **page1.xhtml**. Navedeno je ilustrovano sledećom slikom.



Slika 6.7 Klijent šablona nakon korekcija šablona

## ZADATAK 6

### *Kreirajte vašu JSF stranicu iz šablona*

Nastavite vaš primer tako što ćete kreirati novu ili redefinisati postojeću na osnovu obrađenih šablona.

## ▼ Poglavlje 7

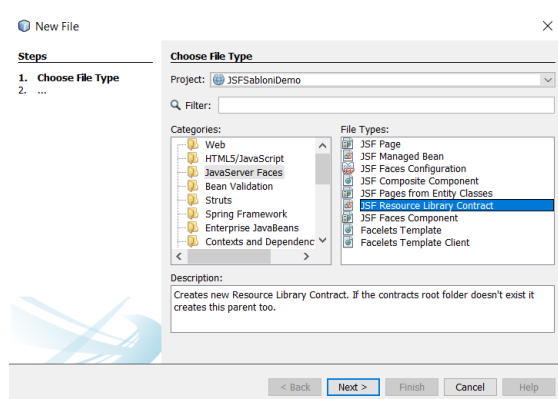
# Biblioteka ugovora resursa

## JSF 2.2 - NOVITET

*Biblioteka ugovora resursa je Java EE novitet.*

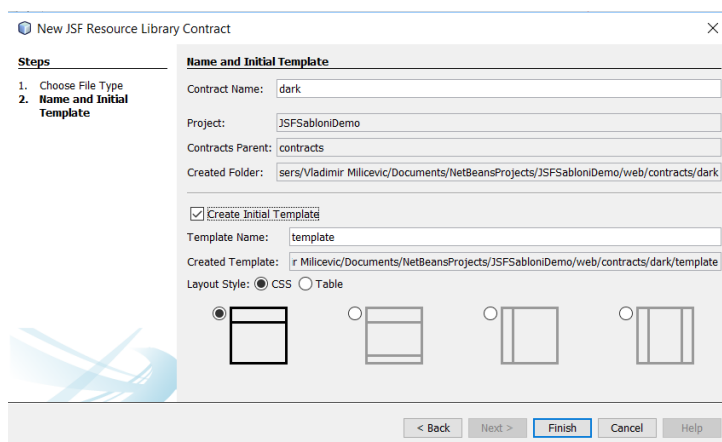
Biblioteka ugovora resursa (Resource library contracts) predstavlja potpuno nov JSF 2.2 alat. Osmišljena je kao podrška za kreiranje Facelet šablona za kreiranje "tematskih" veb aplikacija. Na primer, moguće je da postoji aplikacija koja servisira više potrošača ili je moguće kreirati korisnički interfejs tako da svaki korisnik vidi logo svoje kompanije posle logovanja na sistem. Kao alternativa, moguće je izabrati korisnika iz predefinisanoj skupa tema, a to je upravo zadatak kojim će se baviti ovaj deo lekcije.

Za početak, biće kreirana biblioteka resursa, u aktuelnom projektu, izborom opcije **File**, a zatim **New File**, otvara se prozor pod nazivom **New File**. U ovom prozoru, bira se kategorija **JavaServer Faces** i tip datoteke **JSF Resource Library Contracts** (pogledati sledeću sliku).



Slika 7.1 Kreiranje datoteke JSF Resource Library Contracts

Klikom na dugme **Next**, otvara se prozor u kojem se kreira inicijalni šablon teme zadaje naziv datoteke.



Slika 7.2 Izbor šablona i naziva JSF Resource Library Contracts fajla

Ideja je da razvojno okruženje generiše inicijalni šablon koji će kasnije biti modifikovan na način da će rezultujuća stranica imati tamnu pozadinu sa svetlim slovima. Upravo će to predstavljati tamnu temu.

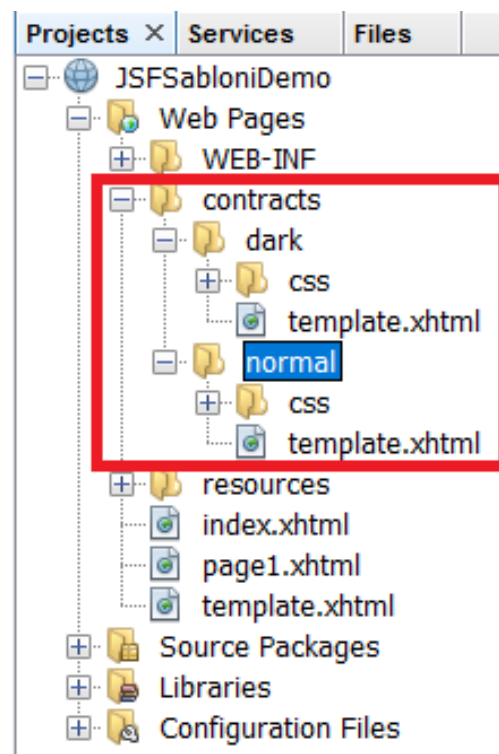
Na isti način je moguće kreirati temu koja alternativno biti korišćena u odnosu na temu **dark**, neka se ova tema naziva **nomal**.

U narednom izlaganju biće više reči o radu sa kreiranim temama i novoj hijerarhiji aktuelnog projekta koji ilustruje tekuću problematiku.

## PRIMER 7- KORIŠĆENJE UGOVORA

*Razvojno okruženje je kreiralo dva fajla u zajedničkom folderu contracts.*

Nakon kreiranja pomenutih datoteka tema, razvojno okruženje je kreiralo dva fajla u zajedničkom folderu contracts i odgovarajućim podfolderima (sledeća slika).



Slika 7.3 Nova hijerarhija projekta

Sada je neophodno prikazati kako je moguće primeniti temu na kreiranu datoteku `page1.xhtml`. Za početak je u okviru taga `<html>` neophodno dodati sledeće: `xmlns:f="http://xmlns.jcp.org/jsf/core"`. Zatim, u okviru bloka `<body>` vrše se korekcije priložene sledećim kodom:

```
<body>
<!--
    <ui:composition template="./template.xhtml">
-->
    <f:view contracts="normal">
    <ui:composition template="/template.xhtml">
        <ui:define name="content">

            ovde ide sadržaj za prikazivanje

        </ui:define>
    </ui:composition>

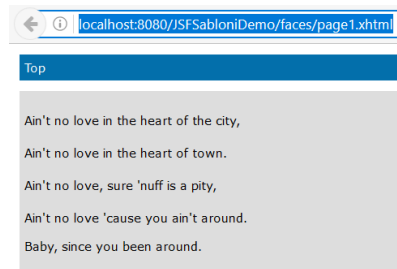
    </f:view>

</body>
```

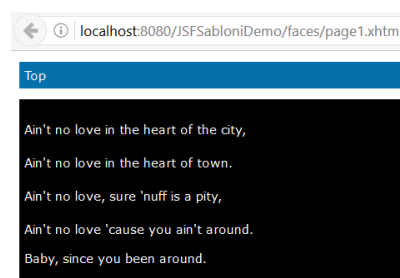
Za korišćenje ugovora, umesto prethodno kreiranih Facelet šablona, neophodno je enkapsulirati `<ui:composition>` u `<f:view>` tag.

Atribut **contract** taga `<f:view>` poseduje vrednost koja se podudara sa nazivom biblioteke ugovora resursa koji će biti upotrebljen u stranici.

Prevođenjem aplikacije i aktiviranjem linka <http://localhost:8080/JFSabloniDemo/faces/page1.xhtml>, u veb pregledaču se pojavljuje sledeća stranica.



Slika 7.4 Korišćenje "normal" teme



Slika 7.5 Korišćenje "dark" teme

## UPRAVLJANJE TEMAMA

*Cilj je kreiranje stranice na kojoj je moguće izabrati željenu temu za prikazivanje sadržaja.*

Da bi **dark** tema, sa prethodne slike, bila prikazana, napravljen su sledeće korekcije u izvornom **cssLayout.css** fajlu iz foldera **dark**.

```
#top {
    position: relative;
    background-color: #036fab;
    color: white;
    padding: 5px;
    margin: 0px 0px 10px 0px;
}

#bottom {
    position: relative;
    background-color: #c2dfef;
    padding: 5px;
    margin: 10px 0px 0px 0px;
}

#left {
    float: left;
    background-color: #ece3a5;
    padding: 5px;
```

```
        width: 150px;
    }

    #right {
        float: right;
        background-color: #ece3a5;
        padding: 5px;
        width: 150px;
    }

    .center_content {
        position: relative;
        background-color: black;
        color: white;
        padding: 5px;
    }

    .left_content {
        background-color: black;
        padding: 5px;
        margin-left: 170px;
    }

    .right_content {
        background-color: black;
        padding: 5px;
        margin: 0px 170px 0px 170px;
    }

    #top a:link, #top a:visited {
        color: white;
        font-weight : bold;
        text-decoration: none;
    }

    #top a:link:hover, #top a:visited:hover {
        color: black;
        font-weight : bold;
        text-decoration : underline;
    }
}
```

Sada je cilj da se kreira, za aktuelnu aplikaciju, CDI zrna koje će omogućiti dinamičko biranje teme na JSF stranici koja je zadužena za prikazivanje konkretnog sadržaja. Neka se ova klasa naziva `SelektorTema.java` i priložena je sledećim listingom.

```
package com.metropolitan.teme;

import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

/**
 *
 */
```

```
* @author Vladimir Milicevic
*/
@Named
@RequestScoped
public class SelektorTema {
    private String themeName = "normal";
    public String getThemeName() {
        return themeName;
    }
    public void setThemeName(String themeName) {
        this.themeName = themeName;
    }
}
```

## UPRAVLJANJE TEMAMA - NASTAVAK

*Neophodne su modifikacije klijenta šablona za korišćenje selektora tema.*

Da bi bilo moguće korišćenje selektora tema, u JSF stranici `page1.xhtml`, neophodno je u kodu ove stranice ukazati na kreirano CDI zrno, te izvršiti odgovarajuće korekcije. Navedeno je priloženo sledećim listingom.

```
<?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://xmlns.jcp.org/jsf/facelets"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:h="http://xmlns.jcp.org/jsf/html">

    <body>
<!--
        <ui:composition template="./template.xhtml">
-->
<!--
        <f:view contracts="dark">
-->
            <f:view contracts="#{selektorTema.themeName}">
                <ui:composition template="/template.xhtml">
                    <ui:define name="top">
                        <h:form>
                            <h:outputLabel for="themeSelector"
                                value="Izaberite temu"/>
                            <h:selectOneMenu id="selektorTema"
                                value="#{selektorTema.themeName}"
                                <f:selectItem itemLabel="normal"
                                    itemValue="normal"/>
                                <f:selectItem itemLabel="dark"
                                    itemValue="dark"/>

```

```

        </h:selectOneMenu>
        <h:commandButton value="Potvrđi"
                        action="page1"/>

    </h:form>
</ui:define>
<ui:define name="content">

    Ovde ide sadržaj za prikazivanje

</ui:define>

</ui:composition>

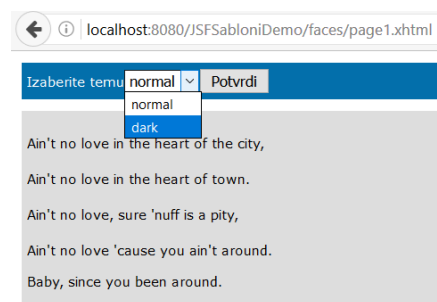
</f:view>

</body>
</html>

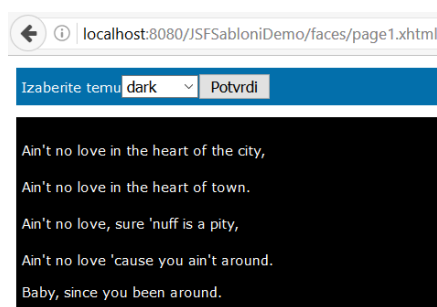
```

U priloženom kodu je moguće primetiti da je prvo uključena podrška za JSF `<h:***>` tagove instrukcijom `xmlns:h="http://xmlns.jcp.org/jsf/html"` taga `<html>`. Dalje je u `top` sekciju uključeno: labela sa informacijom za izbor teme, tag `<h:selectOneMenu>` za listu opcija koje odgovaraju temama i dugme za potvrdu izbora teme.

Kada je prethodna procedura kreiranja koda za upravljanje temama opisana, neophodno je izvršiti i demonstriranje funkcionisanja izmenjene stranice `page1.xhtml`. Prvo je neophodno izvršiti ponovno kompajliranje kreirane veb aplikacije. Nakon toga, aktivira se link `http://localhost:8080/JSFSablioniDemo/faces/page1.xhtml` u veb pregledaču. Sledećim slikama je prikazan početni izgled stranice sa vršenjem izbora i izgled stranice nakon što je tema promenjena u `dark`.



Slika 7.6 Početni izgled i izbor





Slika 7.7 Promenjena tema

## ZADATAK 7

*Koristite teme u vašoj aplikaciji*

Na način opisan u prethodnom izlaganju, omogućite upotrebu tema u vašim JSF stranicama.

## ▼ Poglavlje 8

### Primer 8

#### PRIMER 8 -1

*Kreira se veb aplikacija koja koristi JSF stranice i šablone.*

1. Otvoriti razvojno okruženje i kreirati novi Java veb projekat
2. Kreirati veb aplikaciju koja koristi izabrani layout šablon (CSS) ili temu;
3. Kreirati klijent šablon stranicu za prikazivanje sadržaja raspoređenog na osnovu izabranog rasporeda;

Sledećim listingom je priložena datoteka kojom je definisan šablon:

```
<?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://xmlns.jcp.org/jsf/facelets"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:h="http://xmlns.jcp.org/jsf/html">

    <body>
<!--
        <ui:composition template="./template.xhtml">
-->
<!--
        <f:view contracts="dark">
-->
            <f:view contracts="#{selektorTema.themeName}">
                <ui:composition template="/template.xhtml">
                    <ui:define name="top">
                        <h:form>
                            <h:outputLabel for="themeSelector"
                                value="Izaberite temu"/>
                            <h:selectOneMenu id="selektorTema"
                                value="#{selektorTema.themeName}">
                                <f:selectItem itemLabel="normal"
                                    itemValue="normal"/>
                                <f:selectItem itemLabel="dark"
                                    itemValue="dark"/>
                            </h:selectOneMenu>
                            <h:commandButton value="Potvrđi"
                                action="page1"/>
                        </h:form>
```

```

        </ui:define>
        <ui:define name="content">
            <br>Ain't no love in the heart of the city,</br>
            <br>Ain't no love in the heart of town.</br>
            <br>Ain't no love, sure 'nuff is a pity,</br>
            <br>Ain't no love 'cause you ain't around.</br>
            <p>Baby, since you been around.</p>

        </ui:define>

    </ui:composition>

</f:view>

</body>
</html>

```

Upravljanje temama je u domenu CDI zrna i u narednom izlaganju se prilaže njegov kod:

```

package com.metropolitan.teme;

import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

/**
 *
 * @author Vladimir Milicevic
 */
@Named
@RequestScoped
public class SelektorTema {
    private String themeName = "normal";
    public String getThemeName() {
        return themeName;
    }
    public void setThemeName(String themeName) {
        this.themeName = themeName;
    }
}

```

U nastavku je neophodno priložiti kod stranice koja će prikazivati sadržaj. Stranica po potrebi može da koristi i šablone i teme, a to je priloženo sledećim kodom.

```

<?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://xmlns.jcp.org/jsf/facelets"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:h="http://xmlns.jcp.org/jsf/html">

```

```

<body>
<!--
    <ui:composition template="./template.xhtml">
-->
<!--
    <f:view contracts="dark">
-->
        <f:view contracts="#{selektorTema.themeName}">
            <ui:composition template="/template.xhtml">
                <ui:define name="top">
                    <h:form>
                        <h:outputLabel for="themeSelector"
                            value="Izaberite temu"/>
                        <h:selectOneMenu id="selektorTema"
                            value="#{selektorTema.themeName}">
                            <f:selectItem itemLabel="normal"
                                itemValue="normal"/>
                            <f:selectItem itemLabel="dark"
                                itemValue="dark"/>
                        </h:selectOneMenu>
                        <h:commandButton value="Potvrdi"
                            action="page1"/>
                    </h:form>
                </ui:define>
                <ui:define name="content">
                    <br>Ain't no love in the heart of the city,</br>
                    <br>Ain't no love in the heart of town.</br>
                    <br>Ain't no love, sure 'nuff is a pity,</br>
                    <br>Ain't no love 'cause you ain't around.</br>
                    <p>Baby, since you been around.</p>

                </ui:define>
            </ui:composition>

        </f:view>
    </body>
</html>

```

Za rad sa temama koriste se opisane biblioteke resursa. Neophodno je modifikovati odgovarajuće `cssLayout.css` fajlove u zavisnosti od toga kako želite da vam tema izgleda (pogledati u predavanjima).

Urađen i testiran zadatak je priložen u dodatnim aktivnostima materijala.

## PRIMER 8 - 2

*Vežbanje korišćenja formi na JSF stranicama*

1. Otvoriti razvojno okruženje i kreirati nov Java veb projekat;
2. Kreirati JSF stranicu na kojoj se nalazi forma za unos korisničkih podataka;
3. Kreirati JSF stranicu na kojoj će biti prikazani uneti podaci;
4. Kreirati CDI klasu zrna za obezbeđivanje setter i getter metoda za upravljanje podacima sa forme;
5. Kreirati jednu validator klasu za proveru proizvoljnog polja sa forme.

Forma je realizovana sledećim programskim kodom početne stranice:

```
<?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:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core">

  <h:head>

    <title>Registracija</title>
    <h:outputStylesheet name="styles.css"/>
  </h:head>
  <h:body>
    <h3>Stranica za registraciju</h3>
    <h:form>
      <h:panelGrid columns="3"
                  columnClasses="rightalign, leftalign, leftali
gn">
        <h:outputLabel value="Titula: " for="titula"/>
        <h:selectOneMenu id="titula" label="Titula"
                        value="#{registrationBean.
                            titula}" >
          <f:selectItem itemLabel="" itemValue=""/>
          <f:selectItem itemLabel="St." itemValue="ST"/>
          <f:selectItem itemLabel="Ma." itemValue="MA"/>
          <f:selectItem itemLabel="MSci" itemValue="MSCI"/>
          <f:selectItem itemLabel="Dr" itemValue="DR"/>
        </h:selectOneMenu>
        <h:message for="titula"/>
        <h:outputLabel value="Ime:" for="ime"/>
        <h:inputText id="ime" label="Ime"
                    required="true"
                    value="#{registrationBean.ime}" />
        <h:message for="ime" />
        <h:outputLabel value="Prezime:" for="prezime"/>
        <h:inputText id="prezime" label="Prezime"
                    required="true"
                    value="#{registrationBean.prezime}" />
        <h:message for="prezime" />
        <h:outputLabel for="starost" value="Starost:"/>
        <h:inputText id="starost" label="Starost" size="2"
                    value="#{registrationBean.starost}"/>
        <h:message for="starost"/>
      </h:panelGrid>
    </h:form>
  </h:body>
</html>
```

```

        <h:outputLabel value="Email adresa:" for="email"/>

        <h:inputText id="email" label="Email adresa"
                    required="true"
                    value="#{registrationBean.email}">
            <f:validator validatorId="emailValidator"/>
        </h:inputText>
        <h:message for="email" />
        <h:panelGroup/>
        <h:commandButton id="registracija" value="Registracija"
                        action="potvrda" />

    </h:panelGrid>
</h:form>
</h:body>
</html>

```

Stranica za prikazivanje rezultata je realizovana sledećim JSF kodom:

```

<?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:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>Stranica za potvrdu</title>
        <h:outputStylesheet name="styles.css"/>
    </h:head>
    <h:body>
        <h2>Stranica za potvrdu</h2>
        <h:panelGrid columns="2"
                    columnClasses="rightalign-bold,leftalign">
            <h:outputText value="Titula:"/>
            ${registrationBean.titula}
            <h:outputText value="Ime:"/>
            ${registrationBean.ime}
            <h:outputText value="Prezime:"/>
            ${registrationBean.prezime}
            <h:outputText value="Starost:"/>
            ${registrationBean.starost}
            <h:outputText value="Email adresa:"/>
            ${registrationBean.email}
        </h:panelGrid>
    </h:body>
</html>

```

## PRIMER 8 - 2 - JAVA KLASA

*Neophodno je kreirati CDI klasu i klasu validator.*

Klasa koja obezbeđuje CDI zrno za podršku podacima forme data je sledećim listingom:

```
package com.metropolitan;

/**
 *
 * @author Vladimir Milicevic
 */
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

@Named
@RequestScoped
public class RegistrationBean {
    private String titula;
    private String ime;
    private String prezime;
    private Integer starost;
    private String email;
    public String getTitula() {
        return titula;
    }

    public void setTitula(String titula) {
        this.titula = titula;
    }

    public String getIme() {
        return ime;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

    public String getPrezime() {
        return prezime;
    }

    public void setPrezime(String prezime) {
        this.prezime = prezime;
    }

    public Integer getStarost() {
        return starost;
    }

    public void setStarost(Integer starost) {
        this.starost = starost;
    }

    public String getEmail() {
        return email;
    }
}
```

```
public void setEmail(String email) {  
    this.email = email;  
}  
  
}
```

Konačno, klasa validator zaključuje kreiranje ove veb aplikacije.

```
package com.metropolitan;  
  
/**  
 *  
 * @author Vladimir Milicevic  
 */  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
import javax.faces.application.FacesMessage;  
import javax.faces.component.UIComponent;  
import javax.faces.component.html.HtmlInputText;  
import javax.faces.context.FacesContext;  
import javax.faces.validator.FacesValidator;  
import javax.faces.validator.Validator;  
import javax.faces.validator.ValidatorException;  
  
@FacesValidator(value = "emailValidator")  
public class EmailValidator implements Validator {  
  
    @Override  
    public void validate(FacesContext facesContext,  
        UIComponent uiComponent, Object value) throws  
        ValidatorException {  
        Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");  
        Matcher matcher = pattern.matcher(  
            (CharSequence) value);  
        HtmlInputText htmlInputText  
            = (HtmlInputText) uiComponent;  
        String label;  
        if (htmlInputText.getLabel() == null  
            || htmlInputText.getLabel().trim().equals("")) {  
            label = htmlInputText.getId();  
        } else {  
            label = htmlInputText.getLabel();  
        }  
        if (!matcher.matches()) {  
            FacesMessage facesMessage  
                = new FacesMessage(label  
                    + ": email adresa nije ispravna");  
            throw new ValidatorException(facesMessage);  
        }  
    }  
}
```



Urađen i testiran zadatak je priložen u dodatnim aktivnostima lekcije.

## ZADATAK 8

### *Pokušajte sami*

1. Unesite modifikaciju u Zadatak 2, tako da forma preuzima dodatne podatke o korisniku kao što je naziv fakulteta koji studira, skraćenica fakulteta i studijski program.
2. Obezbediti da ove informacije budu obrađene i prikazane.

## ▼ Poglavlje 9

### Domaći zadatak 3 - 1

#### ZADATAK 1

*Unapređenje i proširenje primera urađenog na vežbama.*

1. Kao osnovu za domaći zadatak izabrati Zadatak 2 sa vežbi;
2. Umesto kategorije titula, dodati kategoriju "Vrsta studiranja" sa opcijama: tradicionalno i internet;
3. Polja za unos teksta su za podatke o studentu;
4. Dodati još jednu padajuću listu čiji su elementi fakulteti našeg Univerziteta;
5. Obezbediti mehanizme validacije za polja;
6. Realizovati CDI zrno za pomoć prilikom prihvatanja unetih podataka forme;
7. Na stranci za prikazivanje podataka, prikazati sve podatke o studentu, uključujući vrstu studiranja i fakultet koji studira.
8. Za realizovanje aplikacije poželjno je korišćenje nekog css šablona ili teme.

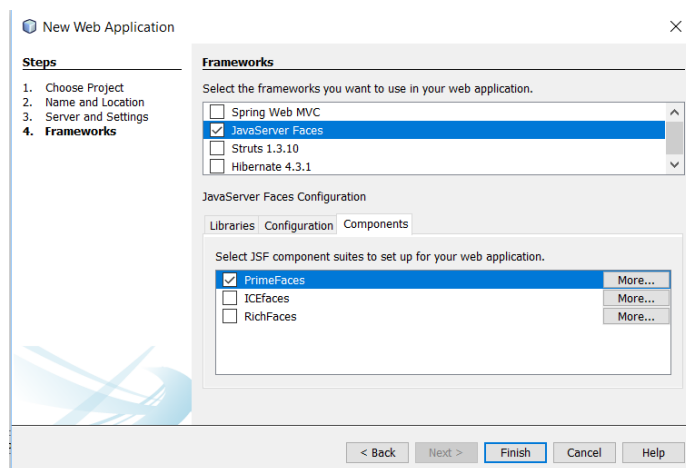
## ▼ Poglavlje 10

# PrimeFaces biblioteka komponentata

## UKLJUČIVANJE PRIMEFACES BIBLIOTEKE U PROJEKAT

*PrimeFaces je veoma popularna JSF biblioteka komponentata.*

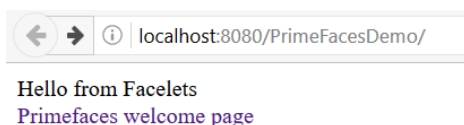
PrimeFaces je veoma popularna JSF biblioteka komponentata koja omogućava programerima razvoj elegantnih veb aplikacija, modernog izgleda, a sa relativno malo uloženog truda. Da bi PrimeFaces biblioteka bila korišćena neophodno je, na dosta puta obrađen i veoma poznat način, kreirati Java veb aplikaciju u okviru nekog razvojnog okruženja, ovde će to biti NetBeans IDE. Prilikom kreiranja projekta neophodno je izabrati JavaServer Faces okvir (sledeća slika).



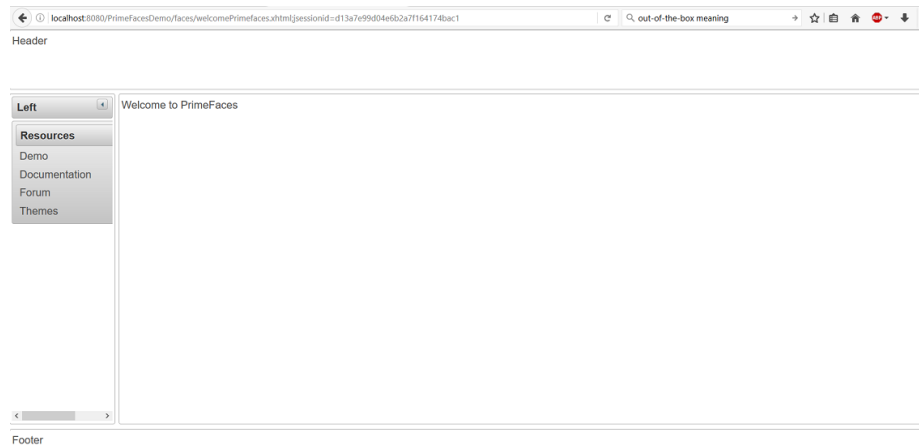
Slika 10.1 Izbor JSF i biblioteke komponentata

Na slici je moguće primetiti da je klikom na tab Components moguće izabrati neku od ponuđenih JSF biblioteka komponentata. Za ovaj deo lekcije neophodno je izabrati opciju PrimeFaces selektovanjem odgovarajuće CheckBox kontole.

Klikom na Finish razvojno okruženje generiše inicijalnu JSF aplikaciju uz uključivanje PrimeFaces biblioteke komponentata. Prevođenjem i pokretanjem ove aplikacije, izborim opcije Run u razvojnom okruženju, dostupne su JSF stranice prikazane sledećim slikama.



Slika 10.2 Početna strana inicijalne JSF aplikacije



Slika 10.3 Stranica dobrodošlice kreirane aplikacije

## GENERISANE JSF STRANICE

*Kreiranjem projekta, automatski su kreirane dve JSF stranice.*

Kada se kreira JSF projekat, na prethodno opisani način, koji uključuje **PrimeFaces** biblioteku komponenta, dolazi do kreiranja i dve JSF datoteke u podfolderu **WEB-INF** projekta veb aplikacije: **index.xhtml** i **welcomePrimefaces.xhtml**.

Ove datoteke su realizovale jednostavnu veb aplikaciju, opisanu slikama 2 i 3, koja je mnogo elegantnija nego što je bio slučaj sa aplikacijama kojima se bavila prethodna lekcija. Posebno je bitno što je ova aplikacija kreirana, a da nije uloženi dodatni napor (po pitanju angažovanja CSS) iz razloga što su **JavaScript** i **CSS** obezbeđeni uključivanjem **PrimeFaces** biblioteke **komponenta**.

U sledećem izlaganju slede listinzi navedenih datoteka. Prvo **index.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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
    <br/>
    <h:link outcome="welcomePrimefaces" value="Primefaces welcome page" />
  </h:body>
</html>
```

Automatski generisana JSF stranica **welcomePrimefaces.xhtml** je realizovana sledećim kodom:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

  <f:view contentType="text/html">
    <h:head>
      <f:facet name="first">
        <meta content='text/html; charset=UTF-8' http-equiv="Content-Type"/>
        <title>PrimeFaces</title>
      </f:facet>
    </h:head>

    <h:body>

      <p:layout fullPage="true">

        <p:layoutUnit position="north" size="100" resizable="true"
closable="true" collapsible="true">
          Header
        </p:layoutUnit>

        <p:layoutUnit position="south" size="100" closable="true"
collapsible="true">
          Footer
        </p:layoutUnit>

        <p:layoutUnit position="west" size="175" header="Left"
collapsible="true">
          <p:menu>
            <p:submenu label="Resources">
              <p:menuitem value="Demo" url="http://www.primefaces.org/
showcase-labs/ui/home.jsf" />
              <p:menuitem value="Documentation"
url="http://www.primefaces.org/documentation.html" />
              <p:menuitem value="Forum"
url="http://forum.primefaces.org/" />
              <p:menuitem value="Themes"
url="http://www.primefaces.org/themes.html" />
            </p:submenu>
          </p:menu>
        </p:layoutUnit>

        <p:layoutUnit position="center">
          Welcome to PrimeFaces
        </p:layoutUnit>

      </p:layout>

    </h:body>
```

```
</f:view>  
</html>
```

## PRIMEFACES KOMPONENTE RASPOREDA

*Neophodno je uključivanje odgovarajućeg prostora naziva.*

Da bi bilo moguće korišćenje PrimeFaces komponentata u JSF Java veb aplikaciji, neophodno je koristiti prostor naziva:

```
xmlns:p="http://primefaces.org/ui" namespace
```

Razvojno okruženje NetBeans automatski dodaje prostor naziva xmlns:p="http://primefaces.org/ui" u generisanu JSF stranicu.

Kada se pogleda Slika 2, koja reprezentuje `welcomePrimefaces.xhtml` stranicu, moguće je primetiti da je stranica podeljena na sekcije: zaglavlje (**header**), podnožje (**footer**), leva sekcija i sekcija sa glavnim sadržajem aplikacije. Važno je napomenuti da ovde nema potrebe za korišćenjem HTML `<div>` tagova i CSS - a da bi ovakav raspored bio realizovan. PrimeFaces koristi `<p:layout>` komponentu koja vodi računa o ovako složenim zadacima.

Unutar `<p:layout>` komponente neophodno je ugnezditi nekoliko `<p:layoutUnit>` komponentata kojima su kreirane odgovarajuće sekcije stranice. posebnu pažnju je neophodno obratiti na atribut **position** komponente `<p:layoutUnit>` kojom se, zapravo i vrši pozicioniranje sekcija. Postavljanje sekcija na posmatranoj stranici je obavljeno kada su odgovarajućim atributima **position** dodeljivane sledeće vrednosti:

- **north** - postavlja sekciju na vrh stranice. Širina sekcije će automatski prekriti sav dostupan horizontalni prostor u prozoru veb pregledača. **Ovako je generisana Header sekcija.**
- **west** - postavlja sekciju na levu stranu stranice. Visina sekcije će automatski prekriti sav dostupan vertikalni prostor u prozoru veb pregledača. **Ovako je generisana Left sekcija.**
- **east** - postavlja sekciju na desnu stranu stranice. Visina sekcije će automatski prekriti sav dostupan vertikalni prostor u prozoru veb pregledača. **Ova vrednost nije korišćena u kreiranoj stranici.**
- **south** - postavlja sekciju na dno stranice. Širina sekcije će automatski prekriti sav dostupan horizontalni prostor u prozoru veb pregledača. **Ovako je generisana Footer sekcija.**
- **center** - postavlja sekciju na sredinu stranu stranice. Visina i širina sekcije će automatski prekriti sav dostupan vertikalni i horizontalni prostor u prozoru veb pregledača, respektivno. **Ovako je generisana sekcija sa glavnim sadržajem aplikacije (trenutno je to String "Welcome to PrimeFaces").**

Komponenta `<p:layoutUnit>` poseduje atribut **size** kojim se podešava širina (za vrednosti **west** i **east**) ili visina (za vrednosti **north** i **south**) odgovarajuće sekcije na stranici. Za ovu

komponentu moguće je koristiti i attribute **resizable**, **closeable** i **collapsible**. Podešavanjem ovih atributa na **true** omogućena je promena veličine, zatvaranje i sakrivanje nekog sadržaja na stranici.

## OSTALE PRIMEFACES KOMPONENTE

*Na veoma jednostavan način je moguće realizovati meni i unos podataka.*

Primenom PrimeFaces biblioteke komponenta, omogućeno je i veoma jednostavno kreiranje menija u JSF stranicama. Kreiranje menija se obavlja angažovanjem komponente PrimeFaces `<p:menu>` nakon čega je korisniku moguće omogućiti veoma laku navigaciju kroz kreiranu veb aplikaciju.

Takođe, za ovu komponentu je moguće primeniti više ugneženih `<p:submenu>` komponenta kojima se realizuju grupe srodnih stavki menija (u formi podmenija). Pomoću njenog atributa **label** obeležavaju se stavke menija u formi u kojoj ih korisnik vidi. U priloženom primeru, priložena je `<p:submenu>` komponenta sa labelom **Resources**. Na komponentu `<p:submenu>` moguće je primeniti više `<p:menuitem>` tagova za svaku od stavki menija posebno. Ove komponente poseduju atribut **value** čija vrednost odgovara URL stringu pomoću kojeg se vrši navigacija na neku od željenih veb stranica.

Kada su u pitanju kontrole koje se nalaze na JSF formama, PrimeFaces biblioteka komponenta poseduje zamene za sve standardne JSF kontrole, uglavnom tako što se prefiks **h** menja prefiksom **p**. Na primer, polje za unos teksta se korišćenjem PrimeFaces biblioteke komponenta realizuje tagom `<p:inputText>` umesto sa `<h:inputText>`.

U nastavku izlaganja, uvodi se primer koji će poslužiti za demonstraciju primene PrimeFaces komponenta u JSF aplikacijama. Idealno bi bilo uraditi primer koji će biti ekvivalentan primeru iz prošle lekcije koji poseduje formu za unos korisničkih podataka, stranicu za prikazivanje unetih podataka, mehanizme za validaciju i odgovarajuću klasu CDI zrna. Na ovaj način najbolje može da se sagledaju sve prednosti primene PrimeFaces biblioteke komponenta.

## PRIMER 9 - PRIMEFACES KOMPONENTE - PRIMER

*Uvodi se primer za demonstraciju primene PrimeFaces komponenta.*

Primenom PrimeFaces biblioteke komponenta biće razvijena aplikacija koja u potpunosti odgovara aplikaciji koja koristi formu za prikupljanje korisničkih podataka iz prethodne lekcije. Sledećom slikom je prikazan izgled forme originalne aplikacije (Slika 4 - a), a nakon toga je prikazana i forma koja je kreirana PrimeFaces komponentama (Slika 4 - b).

**Stranica za registraciju**

a)

Titula:

Ime:

Prezime:

Starost:

Email adresa:

**Registracija**

b)

Titula:	<input type="text"/>
Ime: *	<input type="text"/>
Prezime: *	<input type="text"/>
Starost:	<input type="text"/>
Email adresa: *	<input type="text"/>
<input type="button" value="Registracija"/>	

Slika 10.4 Poređenje izgleda formi

Kao što je moguće primetiti, forma kreirana primenom **PrimeFaces** je veoma lepa, a njena obavezna polja su istaknuta zvezdicama.

Za razliku od prethodne aplikacije, ovde je iskorišćena **index.html** datoteka koja će poželeti dobrodošlicu korisniku i obezbediti link koji vrši navigaciju ka stranici sa formom za prikupljanje podataka korisnika.

Izvršavanje datoteke **index.html** prikazuje korisniku sledeću stranicu:



Slika 10.5 Početna stranica aplikacije

Početna stranica je realizovana kodom koji je priložen sledećim listingom:

```
<?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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Pocetna strana</title>
  </h:head>
  <h:body>
    Dobrodosli!!!
    Klikom na link predjite na stranicu za unos podataka.
    <br/>
    <h:link outcome="welcomePrimefaces" value="Stranica za unos podataka" />
  </h:body>
</html>
```

Klikom na link **"Stranica za unos podataka"** vrši se navigacija do stranice na kojoj je postavljena forma sa slike 4 - b. O ovoj stranici biće više govora u narednom izlaganju.



# PRIMEFACES KOMPONENTE - PRIMER - JSF DATOTEKA SA FORMOM

*U narednom izlaganju cilj je prikazivanje JSF datoteka razvijenih primenom PrimeFaces*

Kao što je istaknuto u prethodnom izlaganju, klikom na link "[Stranica za unos podataka](#)" vrši se navigacija do stranice na kojoj je postavljena forma. Ova stranica, kada je učitana u veb pregledač, ima izgled kao na sledećoj slici:

The image is a screenshot of a web browser displaying a registration page. The browser's address bar at the top shows the URL: localhost:8080/PrimeFacesDemo/faces/welcomePrimeFaces.xhtml?sessionId=f804bc62b2fad8bea6db9588eea9. The page title, located below the address bar, is "Stranica za unos korisnickih podataka". The page layout includes a sidebar on the left and a main content area. The sidebar contains a "Dokumentacija" button and a "Resursi" menu with links to "Demo", "Dokumentacija", "Forum", and "Terme". The main content area is titled "Registracija" and features a registration form. The form consists of several input fields: "Titula:" with a dropdown menu, "Ime: \*" and "Prezime: \*" as text inputs, "Starost:" with a small text input, and "Email adresa: \*" as a text input. At the bottom of the form is a "Registracija" button. The browser's status bar at the very bottom shows navigation icons and a scrollbar.

## CS230 - Lekcija 06

Slika 10.6 Stranica sa formom

Ova stranica je realizovana programskim kodom datoteke `welcomePrimeFaces.xhtml` koji je priložen sledećim listingom:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

  <f:view contentType="text/html">
    <h:head>
      <f:facet name="first">
        <meta content='text/html; charset=UTF-8' http-equiv="Content-Type"/>
        <title>PrimeFaces - primer</title>
```

```

        </f:facet>
    </h:head>

    <h:body>

        <p:layout fullPage="true">

            <p:layoutUnit position="north" size="100" resizable="true"
closable="true" collapsible="true">
                <h2>Stranica za unos korisnickih podataka</h2>
            </p:layoutUnit>

            <p:layoutUnit position="south" size="100" closable="true"
collapsible="true">
                <h2>KI204 - Lekcija 06</h2>
            </p:layoutUnit>

            <p:layoutUnit position="west" size="175" header="Dokumentacija"
collapsible="true">
                <p:menu>
                    <p:submenu label="Resursi">
                        <p:menuitem value="Demo" url="http://www.primefaces.org/
showcase-labs/ui/home.jsf" />
                        <p:menuitem value="Dokumentacija"
url="http://www.primefaces.org/documentation.html" />
                        <p:menuitem value="Forum"
url="http://forum.primefaces.org/" />
                        <p:menuitem value="Teme" url="http://www.primefaces.org/
themes.html" />
                    </p:submenu>
                </p:menu>
            </p:layoutUnit>

            <p:layoutUnit position="center">
                <h3>Registracija</h3>
                <p:messages/>
                <h:form>
                    <p:panelGrid columns="2"
                        columnClasses="rightalign,leftalign">
                        <p:outputLabel value="Titula: " for="titula"/>
                        <p:selectOneMenu id="titula" label="titula"
                            value="#{registrationBean.titula}" >
                            <f:selectItem itemLabel="" itemValue="" />
                            <f:selectItem itemLabel="St." itemValue="ST"/>
                            <f:selectItem itemLabel="MA" itemValue="MA"/>
                            <f:selectItem itemLabel="MSc" itemValue="MSC"/>
                            <f:selectItem itemLabel="Dr." itemValue="DR"/>
                        </p:selectOneMenu>

                        <p:outputLabel value="Ime:" for="ime"/>
                        <p:inputText id="ime" label="Ime"

```

```

        required="true"
        value="#{registrationBean.ime}" />
<p:outputLabel value="Prezime:" for="prezime"/>
<p:inputText id="prezime" label="Prezime"
        required="true"
        value="#{registrationBean.prezime}" />

<p:outputLabel for="starost" value="Starost:"/>
<p:inputText id="starost" label="Starost" size="2"
        value="#{registrationBean.starost}"/>

<p:outputLabel value="Email adresa:" for="email"/>
<p:inputText id="email" label="Email dresa"

required="true"

        value="#{registrationBean.email}">
        <f:validator validatorId="emailValidator"/>
</p:inputText>

<h:panelGroup/>
<p:commandButton id="register" value="Registracija"
        action="confirmation" ajax="false"/>

</p:panelGrid>
</h:form>

</p:layoutUnit>

</p:layout>

</h:body>

</f:view>
</html>

```

## JSF DATOTEKA SA FORMOM - DODATNA RAZMATRANJA

*Većina specifičnih JSF tagova poseduje odgovarajuću PrimeFaces alternativu.*

Kao što je moguće primetiti iz priloženog koda, većina specifičnih JSF tagova poseduje odgovarajuću PrimeFaces alternativu. Ono što je moguće primetiti jeste da je serija originalnih JSF tagova `<h:message>` zamenjena jednim **top - level** `<p:messages>` tagom za prikazivanje poruka koje su u vezi sa nekorektnim unosom u formu.

Kada se koristi PrimeFaces, komandna dugmad na stranicama imaju podrazumevanu vrednost atributa **ajax** podešenu na **true**. Ovde je ta vrednost promenjena na **false**.

Sada je moguće popuniti formu i neko od obaveznih polja, na primer polje za unos adrese elektronske pošte, bi trebalo ostaviti nepopunjenim. Klikom na dugme za potvrdu, javiće se greška. Navedeno je prikazano sledećom slikom.

**Registracija**

✖ Email adresa: Validation Error: Value is required.

Titula:	Dr. ▾
Ime: *	Vladimir
Prezime: *	Milicevic
Starost:	42
Email adresa: *	
<b>Registracija</b>	

Slika 10.7 Obavezno polje nije uneto

Dalje, ukoliko je email adresa uneta, ali u neispravnom formatu, javiće se poruka da email adresa nije korektna. Navedeno je realizovano kreiranjem i angažovanjem klase EmailValidator koja je u potpunosti identična sa klasom iz prethodnog primera i zbog toga ovde neće biti priložena niti analizirana.

Sledećom slikom je prikazana stranica sa formom koja javlja poruku u slučaju loše unete email adrese.

## Registracija

ℹ Email adresa: Email adresa nije korektna

Titula:	Dr. ▾
Ime: *	Vladimir
Prezime: *	Milicevic
Starost:	42
Email adresa: *	vladimir.milicevic
<b>Registracija</b>	

Slika 10.8 Nekorektan unos email adrese

## JSF DATOTEKA ZA PRIKAZIVANJE PODATAKA SA POPUNJENE FORME

*Stranica za prikazivanje podataka, posredstvom CDI zrna, prikazuje unete podatke forme.*

Kada su sva obavezna polja popunjena i kada je zadovoljen format podataka koji su uneti u formu, klikom na dugme "Registracija" podaci će biti sačuvani u objektu (CDI zrnu) klase **RegistrationBean**. Klasa je, kao prvo, obeležena obaveznim anotacijama **@Named** i **@RequestScoped** (obrađeno u prethodnoj lekciji) i poseduje sve **setter** i **getter** metode za manipulaciju sa poljima forme, odnosno odgovarajućim podacima unetim u ta polja. Klasa CDI zrna je u potpunosti identična sa odgovarajućom klasom iz prethodnog primera i zbog toga ovde neće biti priložena niti analizirana.

Osobine ove klase čuvaju korektno unete podatke prikupljene sa forme prethodno opisane stranice koji se dalje prosleđuju stranici pod nazivom **confirmation.xhtml** na prikazivanje.

Slika 10.9 Stranica za prikazivanje korektno unetih podataka korisnika

Pomenuta datoteka **confirmation.xhtml** je nova JSF datoteka sa **PrimeFaces** komponentama i njen kod je priložen sledećim listingom.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

  <f:view contentType="text/html">
    <h:head>
      <f:facet name="first">
        <meta content='text/html; charset=UTF-8' http-equiv="Content-Type"/>
        <title>PrimeFaces - primer</title>
      </f:facet>
    </h:head>

    <h:body>

      <p:layout fullPage="true">

        <p:layoutUnit position="north" size="100" resizable="true"
          closable="true" collapsible="true">
          <h2>Podaci o korisniku</h2>
        </p:layoutUnit>
```

```

        <p:layoutUnit position="south" size="100" closable="true"
collapsible="true">
            <h2>KI204 - Lekcija 06</h2>
        </p:layoutUnit>

        <p:layoutUnit position="west" size="175" header="Dokumentacija"
collapsible="true">
            <p:menu>
                <p:submenu label="Resursi">
                    <p:menuitem value="Demo" url="http://www.primefaces.org/
showcase-labs/ui/home.jsf" />
                    <p:menuitem value="Dokumentacija"
url="http://www.primefaces.org/documentation.html" />
                    <p:menuitem value="Forum"
url="http://forum.primefaces.org/" />
                    <p:menuitem value="Teme" url="http://www.primefaces.org/
themes.html" />
                </p:submenu>
            </p:menu>
        </p:layoutUnit>

        <p:layoutUnit position="center">
            <h2>Podaci koje je korisnik uneo:</h2>
            <p:panelGrid columns="2"
                columnClasses="rightalign-bold,leftalign">
                <h:outputText value="Titula:"/>
                ${registrationBean.titula}
                <h:outputText value="Ime:"/>
                ${registrationBean.ime}
                <h:outputText value="Prezime:"/>
                ${registrationBean.prezime}
                <h:outputText value="Starost:"/>
                ${registrationBean.starost}
                <h:outputText value="Email adresa:"/>
                ${registrationBean.email}
            </p:panelGrid>
        </p:layoutUnit>

    </p:layout>

</h:body>

</f:view>
</html>

```

## POSEBNE NAPOMENE

*Posebne napomene se odnose na fazu JSF projekta*

Za početak je neophodno pogledati datoteku **web.xml** koja sadrži informaciju o trenutnoj fazi razvoja aktuelne veb aplikacije. Deskriptor veb razvoja je priložen sledećim XML listingom:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/
javaee/web-app_3_1.xsd">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```

Ako se pažljivo pogleda priloženi kod, moguće je primetiti da je aplikacija u **Development** fazi razvoja, a to je istaknuto sledećim izolovanim kodom.

```
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>
```

Kada se testira, ovako podešena aplikacija, prilikom navigacije na stranicu sa formom javiće se informacije od značaja za programera, a koje ne bi trebalo da se vide kada se pusti konačna verzija aplikacije za upotrebu od strane korisnika. Navedeno je prikazano sledećom slikom:

#### Registracija

⚠ The form component needs to have a UIForm in its ancestry. Suggestion: enclose the necessary components within <h:form>  
The form component needs to have a UIForm in its ancestry. Suggestion: enclose the necessary components within <h:form>

Titula:	<input type="text"/>
Ime: *	<input type="text"/>
Prezime: *	<input type="text"/>
Starost:	<input type="text"/>
Email adresa: *	<input type="text"/>
<input type="button" value="Registracija"/>	

Da bi ove informacije bile uklonjene, neophodno je u datoteci, **web.xml** aktuelnog projekta, promeniti fazu razvoja iz **Development** u **Production**, na način urađen sledećim izolovanim kodom iz ove datoteke:

```
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Production</param-value>
</context-param>
```

Nakon toga, stranica će imati željeni izgled, upravo kao što je prikazan na Slici 6.

## PRIMEFACES - VIDEO MATERIJAL

*Izlaganje u vezi sa primenom PrimeFaces biblioteke komponenta je zaokruženo video materijalom.*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ZADATAK 9

*Kreirajte sami PrimeFaces stranice*

Pokušajte da realizujete JSF stranice iz primera 8 primenom PrimeFaces biblioteke komponenta.



## ▼ Poglavlje 11

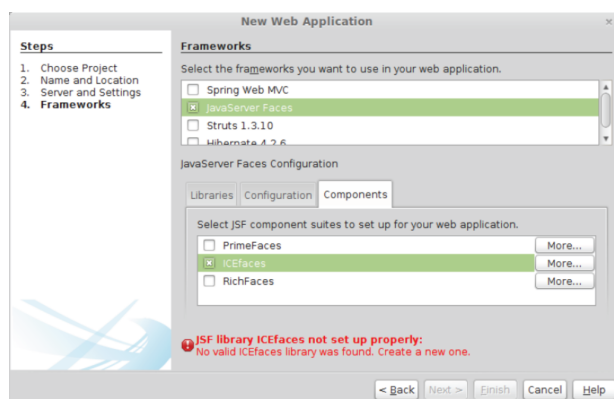
# IceFaces biblioteka komponentata

## UKLJUČIVANJE ICEFACES BIBLIOTEKE U PROJEKAT

### *Još jedna biblioteka za pojednostavljivanje razvoja JSF aplikacija*

IceFaces je nova biblioteka koja će biti izučavana, takođe veoma popularna i u velikoj meri olakšava razvoj veb aplikacija koje koriste JSF stranice. Pre bilo kakvog rada, neophodno je preuzeti biblioteku, priložena je dodatnim materijalima **Shared Resources** ili sa linka <http://www.icesoft.org/java/downloads/icefaces-downloads.jsf>, i odgovarajuće JAR fajlove angažovati prilikom izrade novog Java veb projekta.

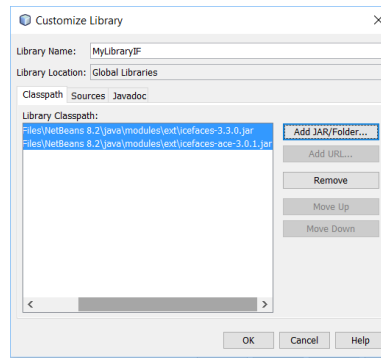
Dakle, na dobro poznat način će biti kreiran novi Java veb projekat, dobiće naziv **IceFacesDemo**, U prozoru u kojem se bira korišćenje JavaServer Faces okvira, klikom na tab **Components**, moguće je selektovati **CheckBox** kontrolu koja odgovara izboru IceFaces biblioteke. Navedeno je prikazano sledećom slikom.



Slika 11.1.1 Izbor IceFaces biblioteke

Kao što je moguće primetiti, biblioteka **IceFaces**, za razliku od **PrimeFaces**, nije uključena u **razvojno okruženje NetBeans** pa je neophodno izvesti dopunske korake da bi ovaj problem bio prevaziđen. Neophodno je preuzeti napomenute fajlove i kreirati novu biblioteku. Poslednja verzija IceFace biblioteke nosi oznaku 4.2. Biblioteka je zapakovana u ZIP formatu i neophodno je izvršiti njeno raspakivanje. Nakon toga u folderu **lib**, od posebnog su značaja datoteke **icefaces.jar** i **icefaces-ace.jar** koje je neophodno uključiti u biblioteku koja se kreira u **razvojnom okruženju**.

Na slici 1, pored opcije IceFaces moguće je uočiti dugme "More". Klikom na ovo dugme započinje kreiranje IceFace biblioteke za NetBeans IDE razvojno okruženje. Dalje, u novom prozoru, klikom na dugme IceFace Library dobija se mogućnost unosa naziva biblioteke, na primer IFLibrary, kao i dodavanja pomenuti JAR fajlova u biblioteku (sledeća slika).

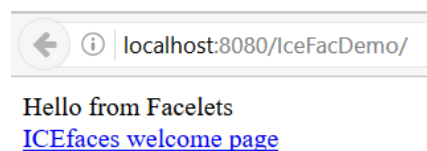


Slika 11.1.2 Kreiranje IceFace biblioteke u NetBeans IDE

## INICIJALNI ICEFACES PROJEKAT

*Kreiranjem IceFace biblioteke razvojno okruženje je spremno za rad na novom projektu.*

U prethodnom izlaganju je pokazano kako se neophodne JAR datoteke ugrađuju u biblioteku koja je neophodna za razvoj Java veb aplikacija u kojima se koriste IceFaces stranice. Kada je biblioteka formirana, obezbeđena je IceFace podrška i na stranici, koja je prikazana Slikom 1, klikom na dugme Finish, kreira se inicijalni Java veb projekat sa IceFaces JSF stranicama. Projekat, veoma slično slučaju sa PrimeFaces, sadrži dve stranice. Osnovna stranica nudi pozdravnu poruku (Slika 3) i link za prelazak na drugu stranicu (Slika 4).

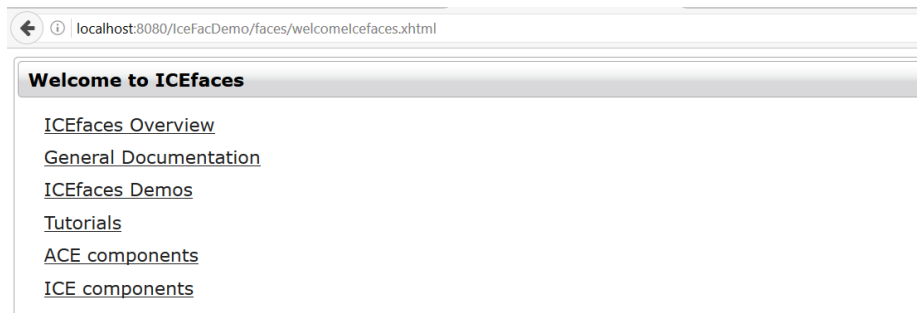


Slika 11.1.3 Početna stranica automatski kreirane veb aplikacije

Ovoj stranici odgovara sledeći kod:

```
<?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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
    <br/>
    <h:link outcome="welcomeIcefaces" value="ICEfaces welcome page" />
  </h:body>
</html>
```

Klikom na link "ICEFaces welcome page" vrši se navigacija ka sledećoj stranici prikazanoj narednom slikom.



Slika 11.1.4 JSF stranica sa IceFaces komponentama

Ova datoteka nosi inicijalni naziv **welcomelcefaces.xhtml** i njen kod, sa odgovarajućim komponentama, biće detaljno razmatran u izlaganju koje sledi.

## ICEFACES KOMPONETE

*Datoteka **welcomelcefaces.xhtml** poseduje ugrađene IceFace komponente koje prikazuje.*

Prethodnom slikom je prikazana JSF stranica **welcomelcefaces.xhtml** koja pokazuje veoma lepo organizovan sadržaj, bez preterano velikog utrošenog truda, pomoću IceFaces komponentata. Sledećim listingom je priložen inicijalni kod ove datoteke:

```
<?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:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:icecore="http://www.icefaces.org/icefaces/core"
      xmlns:ace="http://www.icefaces.org/icefaces/components">
  <h:head>
    <title>ICEfaces Welcome Page</title>
    <!-- This line is only for ICE component, remove it if no ice component is
used in this page.-->
    <link rel="stylesheet" type="text/css" href="./xmlhttp/css/rime/rime.css"/>
```

```

</h:head>
<h:body>
  <h:form>

    <ace:panel header="Welcome to ICEfaces">
      <h:panelGrid columns="1">
        <!-- NOTICE -To run this page you must have also ICEfaces ACE
components library on your classpath (project dependencies). -->
        <ace:linkButton id="linkButton1" value="ICEfaces Overview"
href="http://wiki.icesoft.org/display/ICE/ICEfaces+Overview"></ace:linkButton>
        <ace:linkButton id="linkButton2" value="General Documentation"
href="http://wiki.icesoft.org/display/ICE/ICEfaces+Documentation"></ace:linkButton>
        <ace:linkButton id="linkButton3" value="ICEfaces Demos"
href="http://www.icesoft.org/demos/icefaces-demos.jsf"></ace:linkButton>
        <ace:linkButton id="linkButton4" value="Tutorials"
href="http://www.icesoft.org/community/tutorials-samples.jsf"></ace:linkButton>
        <ace:linkButton id="linkButton5" value="ACE components"
href="http://wiki.icesoft.org/display/ICE/ACE+Components"></ace:linkButton>
        <ace:linkButton id="linkButton6" value="ICE components"
href="http://wiki.icesoft.org/display/ICE/ICE+Components"></ace:linkButton>
        <!-- You can also use ICE components. Adds ICE namespace in
that case: xmlns:ice="http://www.icesoft.com/icefaces/component" -->
        <!-- <ice:outputLink id="aceLink"
value="http://wiki.icesoft.org/display/ICE/ACE+Components" target="_blank">ACE
components</ice:outputLink> -->
        <!-- <ice:outputLink id="iceLink"
value="http://wiki.icesoft.org/display/ICE/ICE+Components" target="_blank">ICE
components</ice:outputLink> -->
      </h:panelGrid>
    </ace:panel>

  </h:form>
</h:body>
</html>

```

Kada je u pitanju rad sa IceFaces komponentama, postoje dva skupa komponentata: ICE komponente i ACE komponente. ICE komponente poseduju funkcionalnost koja je primarno implementirana kao **server - side** kod sa ograničenom primenom JavaScript -a.

Druga komponenta, ACE, predstavlja noviju implementaciju u formi kombinacije **server - side** i **client - side** koda. Prva komponenta se koristi u situacijama kada je neophodno obezbediti kompatibilnost sa starijim verzijama veb čitača, dok je novi skup komponentata u potpunosti okrenuta ka savremenim verzijama veb pregledača i u ovim predavanjima će fokus biti usmeren upravo ka njoj. Iz navedenog razloga, **poslednja verzija NetBeans razvojnog okruženja generiše ICEFaces aplikacije koje se oslanjaju isključivo na primenu ACEkomponentata u kombinaciji sa standardnim JSF komponentama.**

Kada se pogleda priloženi kod uočavaju se komponente stranice o kojima do sada nije bilo reči. Prvo što se primećuje jeste tag **<ace:panel>** čiji je zadatak generisanje panela koji sadrži niz linkova na kreiranoj JSF stranici. Navedeni tag poseduje **header**atribut čiji je zadatak, upravo, kreiranje zaglavlja stranice koja se konstruiše.

Dalje, unutar `<ace:panel>` taga, ugrađeno je nekoliko `<ace:linkButton>` tagova čiji je zadatak generisanje linkova na stranici, slično kao što to čine obrađeni standardni JSF tagovi `<h:outputLink>` i `<h:commandLink>`. U okviru ovih tagova definisani su **String** koji označava stranicu na koju se vrši navigacija (atribut **value**) i sam link koji obezbeđuje navigaciju (atribut **href**).

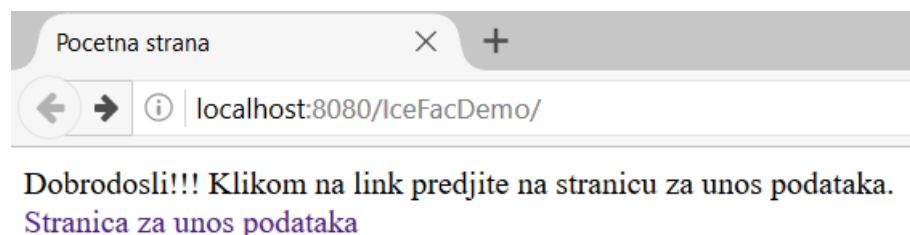
U nastavku izlaganja, inicijalni kod će biti zamenjen kodom koji realizuje aplikaciju sa formom za preuzimanje korisničkih podataka zbog lakšeg poređenja biblioteka komponenta.

## PRIMER 10 - ICEFACES KOMPONENTE

*Uvodi se primer za demonstraciju primene IceFaces komponenta.*

U nastavku će biti kreiran primer sa potpuno identičnim funkcionalnostima kao prethodna dva primera (primena standardnih JSF i PrimeFaces komponenta respektivno) sa idejom da se direktnim poređenjem uoče razlike i prednosti različitih korišćenih pristupa.

Kao i u prethodna dva slučaja, postoji uvodna stranica, označena je nazivom `index.xhtml` koja sadrži pozdravnu poruku i link ka stranici na kojoj se nalazi forma u koju je neophodno uneti odgovarajuće korisničke podatke. Izgled uvodne stranice prikazan je sledećom slikom.



Slika 11.1.5 Početna stranica ICEFace projekta

Kao što je napomenuto klikom na link, kojeg korisnik vidi kao **String** "Stranica za unos podataka" vrši se navigacija ka JSF stranici na kojoj je realizovana dobro poznata forma, ali ovaj put primenom ICEFace komponenta. Ovako, ova forma dobija jedan novi i veoma atraktivan izgled o kojem će biti dosta reči u izlaganju koje neposredno sledi iza ovoga.

U nastavku je neophodno priložiti izmenjeni kod `index.xhtml` stanice na osnovu kojeg je prikazan sadržaj sa Slike 5. Sledi listing ove datoteke:

```
<?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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Početna strana</title>
  </h:head>
  <h:body>
    Dobrodosli!!!
```

Klikom na link predjite na stranicu za unos podataka.

```
<br/>
<h:link outcome="welcomeIcefaces" value="Stranica za unos podataka" />
</h:body>
</html>
```

Kao i u prethodna dva primera, ključni deo aplikacije predstavlja stranica sa formom pa će, u narednom izlaganju, diskusija sa početne stranice biti preusmerena sa stranicu sa formom za unos korisničkih podataka.

## ICEFACES KOMPONENTE - PRIMER - JSF DATOTEKA SA FORMOM

*Forma u ICEFaces reprezentaciji poseduje veoma atraktivan izgled.*

Datoteku `welcomeIcefaces.xhtml`, koja je automatski generisana kreiranjem ICEFaces projekta, moguće je iskoristiti i redefinisati tako da umesto skupa prikazanih linkova sadrži dobro poznatu formu za unos korisničkih podataka. Forma će imati izgled prikazan sledećom slikom.

The screenshot shows a web browser window titled 'ICEfaces Welcome Page'. The address bar shows 'localhost:8080/IceFacDemo/faces/welcomeIcefaces.xhtml'. The main content area displays a form titled 'Registracija'. The form contains the following elements:

- Titula:** A dropdown menu with a downward arrow.
- Ime:** A text input field.
- Prezime:** A text input field.
- Starost:** A range slider with values 19 and 65.
- Email adresa:** A text input field.
- Registracija:** A button at the bottom of the form.

Slika 11.1.6 Forma sa ICEFace izgledom

Stranica sa slike je realizovana kodom kojem odgovara sledeći listing:

```
<?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:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:icecore="http://www.icefaces.org/icefaces/core"
xmlns:ace="http://www.icefaces.org/icefaces/components">
<h:head>
  <title>ICEfaces Welcome Page</title>
  <!-- This line is only for ICE component, remove it if no ice component is
used in this page.-->
  <link rel="stylesheet" type="text/css" href="./xmlhttp/css/rime/rime.css"/>
</h:head>
<h:body>
  <h:form>
    <ace:panel header="Registracija">
      <ace:messages/>
      <h:panelGrid columns="2"
        columnClasses="rightalign,leftalign">
        <h:outputLabel value="Titula: "
          for="titula"/>
        <ace:selectMenu id="titula" label="Titula"
          value="#{registrationBean.titula}" >
          <f:selectItem itemLabel="" itemValue=""/>
          <f:selectItem itemLabel="St." itemValue="ST"/>
          <f:selectItem itemLabel="MA"
            itemValue="MA"/>
          <f:selectItem itemLabel="MSc"
            itemValue="MSC"/>
          <f:selectItem itemLabel="Dr" itemValue="DR"/>
        </ace:selectMenu>
        <h:outputLabel value="Ime:"
          for="ime"/>
        <h:inputText id="ime" label="Ime"
          required="true"
          value="#{registrationBean.ime}"
          />
        <h:outputLabel value="Prezime:" for="prezime"/>
        <h:inputText id="prezime" label="Prezime"
          required="true"
          value="#{registrationBean.prezime}"
          />
        <h:outputLabel for="starost" value="Starost:"/>
        <ace:sliderEntry id="starost"
          value="#{registrationBean.starost}"
          min="19" max="65" showLabels="true"
          />
        <h:outputLabel value="Email adresa:"
          for="email"/>
        <h:inputText id="email" label="Email adresa"
          required="true"
          value="#{registrationBean.email}">
          <f:validator validatorId="emailValidator"/>
        </h:inputText>
      </h:panelGrid>
    </ace:panel>
  </h:form>
</h:body>

```

```

        <h:panelGroup/>
        <ace:pushButton id="register" value="Registracija"
                        action="confirmation" />
    </h:panelGrid>
</ace:panel>
</h:form>
</h:body>
</html>

```

## JSF ICEFACES DATOTEKA SA FORMOM - DODATNA RAZMATRANJA

### *Analiza priloženog koda sa ICEFaces komponentama.*

Prva komponenta koju je moguće uočiti jeste panel kojim je istaknuta informacija da se na stranici obavlja "Registracija" korisnika. Ova komponenta je realizovana sledećom linijom koda koji je izolovan iz prethodno priloženog koda u celini:

```
<ace:panel header="Registracija">
```

Nakon toga moguće je uočiti sledeću liniju koda:

```
<ace:messages/>
```

Ovaj kod omogućava prikazivanje poruka sa greškama, odmah ispod kreiranog panela. Poruke sa greškama se odnose na greške koje su rezultat validacije korisničkih podataka (više o tome kasnije) ili su rezultat izostanka unosa u obaveznim poljima forme.

U nastavku koda nalazi se dobro poznata instrukcija za kreiranje matričnog **layout** - a u formi JSF taga `<h:panelGrid>`. Atributom **columns** ovog taga određeno je da za kreiranje ovako lepe forme budu rezervisane dve kolone. Atributom **columnClasses** obavljena su izvesna poravnanja teksta u odnosu na kolone. Ovo je realizovano sledećim kodom koji je izolovan iz prethodno priloženog koda u celini:

```
<h:panelGrid columns="2" columnClasses = "rightalign,leftalign">
```

Poznatim komponentama `<h:outputLabel>`, u stranici su realizovane labele koje posebno ističu kontrole na koje se direktno odnose.

ICEFaces komponentom `<ace:selectMenu>` realizovan je padajući meni za izbor titule koju korisnik čiji se podaci unose poseduje. Svaka od stavki padajućeg menija realizovana je vlastitim `<f:selectItem>` i atributima **itemLabel** i **itemValue** koji se odnose na naziv stavke koju vidi korisnik i na njenu vrednost.

Polja za unos teksta su kreirana primenom poznatih tagova `<h:inputText>` sa poznatim atributima: **id**, **label**, **required** i **value**. Sva polja za unos teksta koja poseduju atribut **required** podešen na vrednost **true**, obavezna su za unos. Vrednost **value** ukazuje na odgovarajuću osobinu CDI zrna kojom se čuva vrednost koja je uneta u odgovarajuće polje za unos teksta.



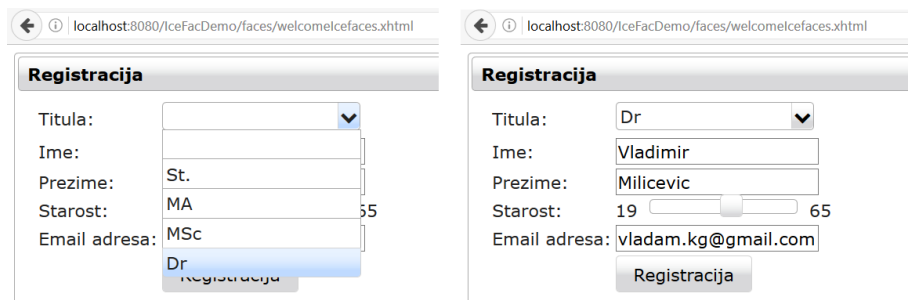
Kao posebno atraktivan ICEFaces komponenta javlja se klizač (**slider**) komponenta kreirana ICEFaces tagom `<ace:sliderEntry>`. Ovom komponentom se zadaju naziv osobine (**id** atribut) čija će vrednost biti određena položajem klizača, zatim atribut **value** čija vrednost ukazuje na odgovarajuću osobinu CDI zrna kojom se čuva vrednost određena položajem klizača, minimalna i maksimalna vrednost položaja klizača (**min** = 19 - student brucoš i **max** = 65 - maksimum godina pred odlazak u penziju) i, konačno, atribut **value** koji ukazuje na odgovarajuću osobinu CDI zrna kojom se čuva vrednost koja odgovara starosti korisnika čiji se podaci unose u formu.

Potvrđivanje forme je realizovano dugmetom koje je kreirano ICEFaces komponentom `<ace:pushButton>`.

## UNOS PODATAKA U FORMU I STRANICA ZA PRIKAZIVANJE REZULTATA

### *Demostracija korišćenja kontrola forme i prikazivanje unetih rezultata.*

Za početak je neophodno izabrati jednu od ponuđenih vrednosti: St. (student), Ma (master), MSc (magistar nauka) ili dr (doktor nauka) kao titulu iz kreiranog ICEFaces padajućeg menija (prikazano sledećom slikom).



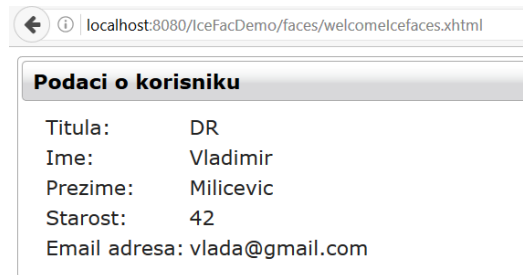
Slika 11.1.7 Izbor iz ICEFace padajućeg menija i popunjena forma

Nakon izbora titule, u polja za unos teksta je neophodno uneti tražene podatke, a klizač postaviti na vrednost koja odgovara broju godina korisnika.

Kada se popune odgovarajući podaci forme, klikom na dugme "Registracija", podaci se šalju na server, i za njihovo čuvanje je od sada zaduženo CDI zrno `registrationBean` koje je instanca klase `RegistrationBean` (više reči u narednom izlaganju).

Stranica za prikazivanje podataka dobija vrednosti koje su unete na formi od pomenutog CDI zrna i prikazuje ih korisniku u veb pregledaču.

Navedeno je prikazano sledećom slikom:



Slika 11.1.8 Stranica za prikazivanje podataka o korisniku

Datoteka koja odgovara ovoj JSF stranici naziva se **confirmation.xhtml**, sadrži već pomenute i opisane ICEFaces komponente i priložena je sledećim listingom.

```
<?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:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ace="http://www.icefaces.org/icefaces/components">
  <h:head>
    <title>Stranica sa korisnickim podacima</title>

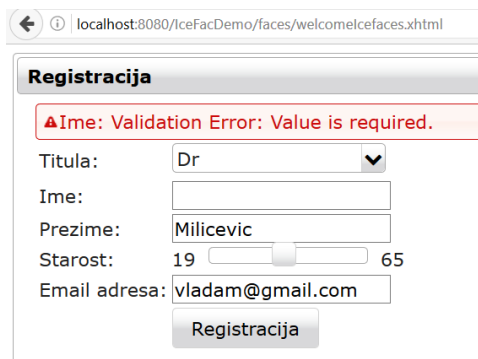
  </h:head>
  <h:body>
    <ace:panel header="Podaci o korisniku">
      <h:panelGrid columns="2"
        columnClasses="rightalign-bold,leftalign">
        <h:outputText value="Titula:"/>
        ${registrationBean.titula}
        <h:outputText value="Ime:"/>
        ${registrationBean.ime}
        <h:outputText value="Prezime:"/>
        ${registrationBean.prezime}
        <h:outputText value="Starost:"/>
        ${registrationBean.starost}
        <h:outputText value="Email adresa:"/>
        ${registrationBean.email}
      </h:panelGrid>
    </ace:panel>
  </h:body>
</html>
```

## VALIDACIJA PODATAKA U ICEFACES STRANICI SA FORMOM

*Sva obavezna polja moraju da budu popunjena, email adresa mora da ima standardni format.*

Validacija unetih podataka u formu je veoma važan zadatak. Ozbiljna veb aplikacija ne bi trebalo da dozvoli unos nekorektnih korisničkih podataka, u smislu ne poklapanja tipova podataka, unosa email adrese koja ne odgovara standardnom email formatu adrese ili unos broja godina koji ne može biti realan.

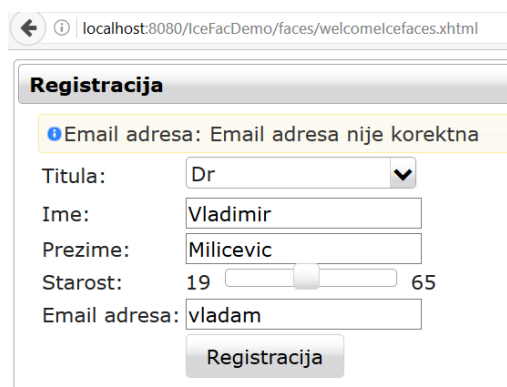
Ukoliko je polje za unos teksta forme ima atribut **required** podešen na vrednost **true**, polje je obavezno za unos i ukoliko se ne unese odgovarajući podatak u polje u prostoru predviđenom za poruke biće prikazan odgovarajuća poruka sa greškom. Navedeno je prikazano sledećom slikom:



Slika 11.1.9 Poruka sa greškom

Postoji u drugi scenario. U polje je unet podatak koji nema odgovarajući format. Na primer unet je neki **String** koji nikako ne **odgovara formatu u kojem se inače predstavljaju email adrese**.

Tada klasa **Validator** (priložen je listing koda klase) stupa na scenu i **šalje predefinisanu informaciju da je unos email adrese nekorektan**. Navedeno je prikazano sledećom slikom.



Slika 11.1.10 0 0 Validacija email adrese

```
package com.metropolitan.IceFaces;

/**
 *
 * @author Vladimir Milicevic
 */
import java.util.regex.Matcher;
import java.util.regex.Pattern;
```

```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.component.html.HtmlInputText;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator(value = "emailValidator")
public class EmailValidator implements Validator {

    @Override
    public void validate(FacesContext context, UIComponent uiComponent, Object
value) throws ValidatorException {
        Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");
        Matcher matcher = pattern.matcher(
            (CharSequence) value);
        HtmlInputText htmlInputText
            = (HtmlInputText) uiComponent;
        String label;

        if (htmlInputText.getLabel() == null
            || htmlInputText.getLabel().trim().equals("")) {
            label = htmlInputText.getId();
        } else {
            label = htmlInputText.getLabel();
        }

        if (!matcher.matches()) {
            FacesMessage facesMessage;
            facesMessage = new FacesMessage(label
                + ": Email adresa nije korektna");

            throw new ValidatorException(facesMessage);
        }
    }
}
```

## KLASA CDI ZRNA

*Medijator između forme i stranice za prikazivanje rezultata je CDI zrno.*

U nastavku je neophodno skrenuti pažnju na već izučavana *CDI zrno*, njihov zadatak i ponašanje. Kao instanca klase koja sadrži sve *setter* i *getter* metode, neophodne za upravljanje podacima koje korisnik zadaje putem neke forme, *CDI zrno*, pomoću svojih osobina, čuva podatke unete putem forme i prosleđuje ih na dalju obradu ili prikazivanje korisnicima. Na ovaj način *CDI zrno* se ponaša kao posrednik između stranica koje su izvori podataka i stranica koje su potrošači pomenutih podataka.

*CDI zrna*, koje je u kodu stranice sa formom dobilo na čuvanje podatke unete putem forme, predstavlja instancu klase koja je obeležena obaveznim anotacijama **@Named** i **@RequestScoped** i priložena je sledećim listingom.

```
package com.metropolitan.IceFaces;

/**
 *
 * @author Vladimir Milicevic
 */
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

@Named
@RequestScoped
public class RegistrationBean {

    private String titula;
    private String ime;
    private String prezime;
    private Integer starost;
    private String email;

    public String getTitula() {
        return titula;
    }

    public void setTitula(String titula) {
        this.titula = titula;
    }

    public String getIme() {
        return ime;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

    public String getPrezime() {
        return prezime;
    }

    public void setPrezime(String prezime) {
        this.prezime = prezime;
    }

    public Integer getStarost() {
        return starost;
    }

    public void setStarost(Integer starost) {
        this.starost = starost;
    }
}
```

```
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

}
```

Na samom kraju izlaganja, u vezi sa primenom ICEFaces biblioteke komponenta u veb aplikacijama koje koriste JSF stranice, moguće je dodati prigodne video materijale kojima će ovaj deo lekcije biti zaokružen. Materijali daju odličan ICEFaces tutorijal koji je priložen iz dva dela.

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ZADATAK 9

### *Kreirajte sami IceFaces stranice*

Pokušajte da realizujete JSF stranice iz primera 8 primenom IceFaces biblioteke komponenta.

## ✓ 11.1 JSF biblioteke komponenta - pokazni primeri 2

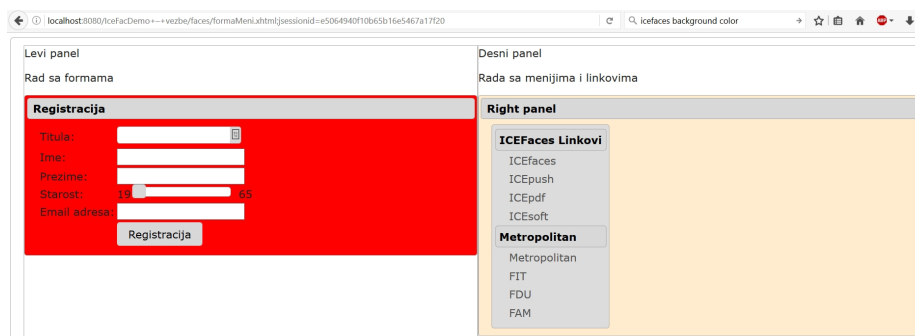
### PRIMER 11

*Vežbanje na računaru kreiranja veb aplikacija koje koriste neku od biblioteka komponenta.*

1. Otvoriti NetBeans razvojno okruženje i kreirati nov projekat pod nazivom IceFaceVezba6;
2. Projekat mora da ima uključen JSF okvir i ICEFaces biblioteku komponenta;
3. Projekat sadrži tri ICEFaces stranice: index.xhtml, formaMeni.xhtml, potvrda.xhtml.
4. index.xhtml (sadrži poruku dobrodošlice i link za navigaciju ka stranici formaMeni.xhtml);
5. Stranica formaMeni.xhtml sadrži glavni panel podeljen na dva panela: levi i desni;

6. Levi panel sadrži formu za prikupljanje korisničkih podataka- može biti i forma koja je korišćena na predavanjima;
7. Desni panel sadrži dva menija sa likovima - prvu grupu čine ICEFaces korisni linkovi, a drugu linkovi značajni za odlazak na prezentacije našeg Univerziteta i njegovih fakulteta;
8. Stranica potvrda.xhtml prikazuje podatke koje je korisnik uneo preko forme.
9. Kreirati klasu odgovarajućeg CDI zrna;
10. Kreirati klasu za validaciju email formata.

Prvo je neophodno prikazati kako bi trebalo da izgleda stranica formaMeni.xhtml. Stranica je prikazana sledećom slikom:



Slika 11.2.1 Izgled stranice formaMeni.xhtml

Prilaže se kod JSF stranice index.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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Pocetna strana</title>
  </h:head>
  <h:body>
    Dobrodošli!!!
    <p>
      Klikom na link predjite na stranicu za unos podataka.
    </p>
    <br/>
    <h:link outcome="formaMeni" value="Stranica za unos podataka i izbor linka"
  />
  </h:body>
</html>
```

## PRIME3 11 - JSF STRANICE FORMAMENI.XHTML I POTVRDA.XHTML.

*Slede JSF stranice kreirane pomoću ICEFaces biblioteke komponenta*

Sledi kod JSF datoteke formaMeni.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:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ace="http://www.icefaces.org/icefaces/components">
  <h:head>
    <title>JSF - Vezba broj 6</title>
    <!-- This line is only for ICE component, remove it if no ice component is
used in this page.-->
    <link rel="stylesheet" type="text/css" href="http://www.icefaces.org/icefaces/css/rime/rime.css"/>
  </h:head>
  <h:body>
    <h:form id="form">

      <ace:panel id="panel">

        <ace:splitPane id="splitPane" scrollable="true"
          columnDivider="50">

          <f:facet name="left" id="leftFacet">
            <h:outputText id="out1" value="Levi panel" />
            <h:panelGroup id="pg1">
              <p>Rad sa formama</p>
            </h:panelGroup>
            <h:form>
              <ace:panel header="Registracija"
style="background-color:red">
                <ace:messages/>
                <h:panelGrid columns="2"
                  columnClasses="rightalign,leftalign">
                  <h:outputLabel value="Titula: "
                    for="titula"/>
                  <ace:selectMenu id="titula" label="Titula"
value="#{registrationBean.titula}" >
                    <f:selectItem itemLabel="" itemValue=""/>
                    <f:selectItem itemLabel="St."
itemValue="ST"/>
                    <f:selectItem itemLabel="MA"
                      itemValue="MA"/>
                    <f:selectItem itemLabel="MSc"
                      itemValue="MSC"/>
                    <f:selectItem itemLabel="Dr"
itemValue="DR"/>
                  </ace:selectMenu>
                  <h:outputLabel value="Ime: "
                    for="ime"/>
                  <h:inputText id="ime" label="Ime"
                    required="true"
                    value="#{registrationBean.ime}"
                    />
                </h:form>
              </ace:panel>
            </h:form>
          </ace:splitPane>
        </ace:panel>
      </h:form>
    </h:body>
  </html>
```



```

        <h:outputLabel value="Prezime:" for="prezime"/>
        <h:inputText id="prezime" label="Prezime"
                    required="true"
                    value="#{registrationBean.prezime}"
                    />
        <h:outputLabel for="starost" value="Starost:"/>
        <ace:sliderEntry id="starost"

value="#{registrationBean.starost}"
                                min="19" max="65"
showLabels="true"
                                />
        <h:outputLabel value="Email adresa:"
                    for="email"/>
        <h:inputText id="email" label="Email adresa"
                    required="true"
                    value="#{registrationBean.email}">
        <f:validator validatorId="emailValidator"/>

        </h:inputText>
        <h:panelGroup/>
        <ace:pushButton id="register"

value="Registracija"
                                action="potvrda" style=""/>
        </h:panelGrid>
    </ace:panel>
</h:form>

</f:facet>
<f:facet name="right" id="rightFacet">
    <h:outputText id="out2" value="Desni panel" />
    <h:panelGroup id="pg2">
        <p>Rada sa menijima i linkovima</p>
    </h:panelGroup>
    <ace:panel id="rightPanel" header="Right panel"
style="background-color:blanchedalmond">
        <ace:menu style="background-color: gainsboro">
            <ace:submenu label="ICEFaces Linkovi">
                <ace:menuItem value="ICEfaces"
url="http://www.icefaces.org" target="_blank" icon="ui-icon ui-icon-home"/>
                <ace:menuItem value="ICEpush"
url="http://www.icepush.org" target="_blank" icon="ui-icon ui-icon-home"/>
                <ace:menuItem value="ICEpdf"
url="http://www.icepdf.org" target="_blank" icon="ui-icon ui-icon-home"/>
                <ace:menuItem value="ICESoft"
url="http://www.icesoft.com" target="_blank" icon="ui-icon ui-icon-home"/>
            </ace:submenu>

            <ace:submenu label="Metropolitan">
                <ace:menuItem value="Metropolitan"
url="http://www.metropolitan.ac.rs/" target="_blank" icon="ui-icon ui-icon-home"/>
                <ace:menuItem value="FIT"
url="http://www.metropolitan.ac.rs/osnovne-studije/

```

```
fakultet-informacionih-tehnologija/" target="_blank" icon="ui-icon ui-icon-home"/>
        <ace:menuItem value="FDU"
url="http://www.metropolitan.ac.rs/fakultet-digitalnih-umetnosti-2/"
target="_blank" icon="ui-icon ui-icon-home"/>
        <ace:menuItem value="FAM"
url="http://www.metropolitan.ac.rs/osnovne-studije/fakultet-za-menadzment/"
target="_blank" icon="ui-icon ui-icon-home"/>
        </ace:submenu>
    </ace:menu>

    </ace:panel>
</f:facet>

</ace:splitPane>

</ace:panel>

</h:form>

</h:body>

</html>
```

Sledi kod JSF datoteke potvrda.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:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ace="http://www.icefaces.org/icefaces/components">
    <h:head>
        <title>Stranica sa korisnickim podacima</title>

    </h:head>
    <h:body>
        <ace:panel header="Podaci o korisniku">
            <h:panelGrid columns="2"
                columnClasses="rightalign-bold,leftalign">
                <h:outputText value="Titula:"/>
                ${registrationBean.titula}
                <h:outputText value="Ime:"/>
                ${registrationBean.ime}
                <h:outputText value="Prezime:"/>
                ${registrationBean.prezime}
                <h:outputText value="Starost:"/>
                ${registrationBean.starost}
                <h:outputText value="Email adresa:"/>
                ${registrationBean.email}
            </h:panelGrid>
        </ace:panel>
    </h:body>
</html>
```

## PRIMER 11 - JAVA KLASA - CDI I VALIDATOR

*Slede rešenja za klasu CDI zrna i klasu za validaciju unete email adrese u formi.*

Klasa RegistrationBean.java je priložena sledećim listingom:

```
package com.metropolitan.IceFaces.vezba6;

/**
 *
 * @author Vladimir Milicevic
 */
import javax.enterprise.context.RequestScoped;
import javax.inject.Named;

@Named
@RequestScoped
public class RegistrationBean {

    private String titula;
    private String ime;
    private String prezime;
    private Integer starost;
    private String email;

    public String getTitula() {
        return titula;
    }

    public void setTitula(String titula) {
        this.titula = titula;
    }

    public String getIme() {
        return ime;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

    public String getPrezime() {
        return prezime;
    }

    public void setPrezime(String prezime) {
        this.prezime = prezime;
    }

    public Integer getStarost() {
```

```
        return starost;
    }

    public void setStarost(Integer starost) {
        this.starost = starost;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

Klasa EmailValidator.java je priložena sledećim listingom:

```
package com.metropolitan.IceFaces.vezba6;

/**
 *
 * @author Vladimir Milicevic
 */
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.component.html.HtmlInputText;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator(value = "emailValidator")
public class EmailValidator implements Validator {

    @Override
    public void validate(FacesContext context, UIComponent uiComponent, Object
value) throws ValidatorException {
        Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");
        Matcher matcher = pattern.matcher(
            (CharSequence) value);
        HtmlInputText htmlInputText
            = (HtmlInputText) uiComponent;
        String label;

        if (htmlInputText.getLabel() == null
            || htmlInputText.getLabel().trim().equals("")) {
            label = htmlInputText.getId();
        } else {
            label = htmlInputText.getLabel();
        }
    }
}
```

```
    }

    if (!matcher.matches()) {
        FacesMessage facesMessage;
        facesMessage = new FacesMessage(label
            + ": Email adresa nije korektna");

        throw new ValidatorException(facesMessage);
    }
}

}
```

## ▼ Poglavlje 12

# RichFaces biblioteka komponentata

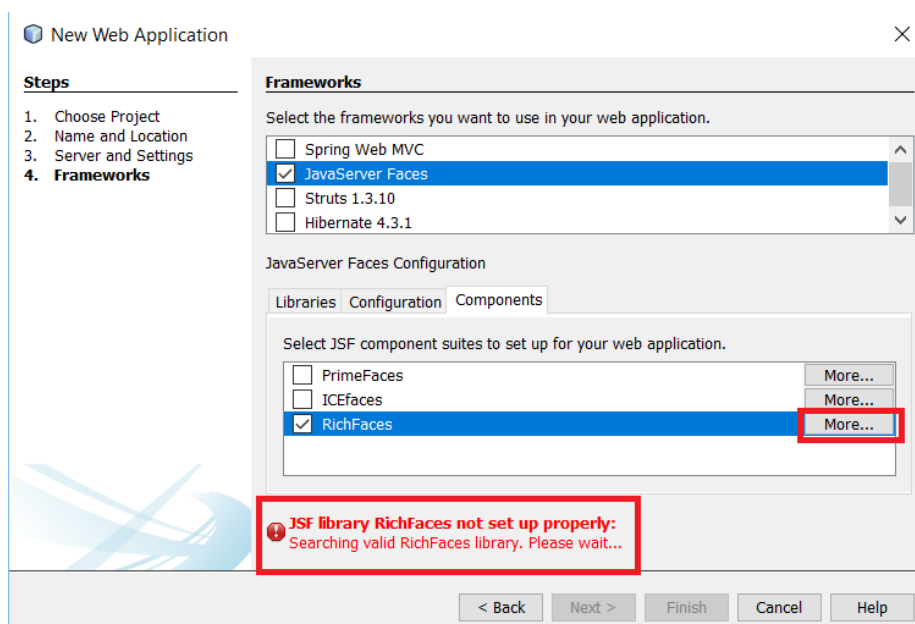
## UKLJUČIVANJE RICHFACES BIBLIOTEKE U PROJEKAT

*Neophodno je preuzeti RichFaces podršku i ugraditi je u projekat.*

Sledeća, veoma popularna, biblioteka komponentata koja će biti obrađena jeste biblioteka pod nazivom RichFaces. Kao i prethodne dve biblioteke, kada je RichFaces angažovana u razvoju veb aplikacija, koje u sebi integrišu i JSF stranice, dosta posla je automatizovano, napor koji programer ulaže u razvoj same aplikacije, i odgovarajućih JSF stranica, značajno je redukovano, a same stranice poseduju moćan i veoma atraktivan izgled.

Kada je u pitanju podrška za rad sa RichFaces komponentama, u okviru JSF stranica veb aplikacije koja se razvija, neophodno je izvršiti neke akcije koje prethode procesu razvoja aplikacije i uključivanja same RichFaces biblioteke kao podrške tom procesu. Navedeno podrazumeva da **inicijalno ne postoje JAR fajlovi dostupni razvojnom okruženju NetBeans IDEte je neophodno izvršiti njihovo preuzimanje i dodavanje u novu NetBeans biblioteku za rad sa RichFaces komponentama veb stranica.**

U narednom izlaganju će ovaj postupak biti detaljno objašnjen. Da bi RichFaces biblioteka bila korišćena neophodno je, na dosta puta obrađen i veoma poznat način, kreirati Java veb aplikaciju u okviru nekog razvojnog okruženja, ovde će to biti NetBeans IDE. **Prilikom kreiranja projekta neophodno je izabrati JavaServer Faces okvir, a u tabu Componets selektovati opciju RichFaces** (sledeća slika).



Slika 12.1 Izbor za uključivanje RichFaces biblioteke komponentata

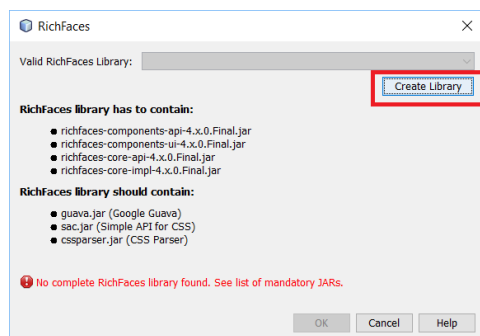
Kao što je moguće primetiti sa slike, kada se prvi put kreira veb projekat ja JSF stranicama koje zahtevaju RichFaces komponente, sama biblioteka nije podešena na adekvatan način. Iz navedenog razloga je neophodno preuzeti odgovarajuće datoteke za podršku (JAR fajlove) i dodati ih kao elemente RichFaces biblioteke u okviru razvojnog okruženja NetBeans IDE. Detaljno o ovoj temi biće govora u izlaganju koje neposredno sledi.

## PODEŠAVANJE RICHFACES BIBLIOTEKE U RAZVOJNOM OKRUŽENJU

### *Inicijalno RichFaces biblioteka nije podešena.*

Kao što je napomenuto, prilikom prvog izbora RichFaces biblioteke, za neki veb projekat čije veb stranice zahtevaju ove komponente, ona još uvek nije podešena i ne može biti korišćena. Iz navedenog razloga, sa linka <http://richfaces.jboss.org/download> ili **iz dodatnih materijala Shared Resources**, koji su sastavni deo ovog objekta učenja, moguće je preuzeti fajlove koji predstavljaju kompletnu podršku neophodnu za korišćenje RichFaces komponentata u JSF stranicama aktuelnog veb projekta. Podrška je spakovana pod ZIP datotekom pod nazivom **richfaces-distribution-4.5.17.Final.zip** i sadrži JAR fajlove koje je neophodno uključiti u novu biblioteku razvojnog okruženja za rad sa RichFaces komponentama.

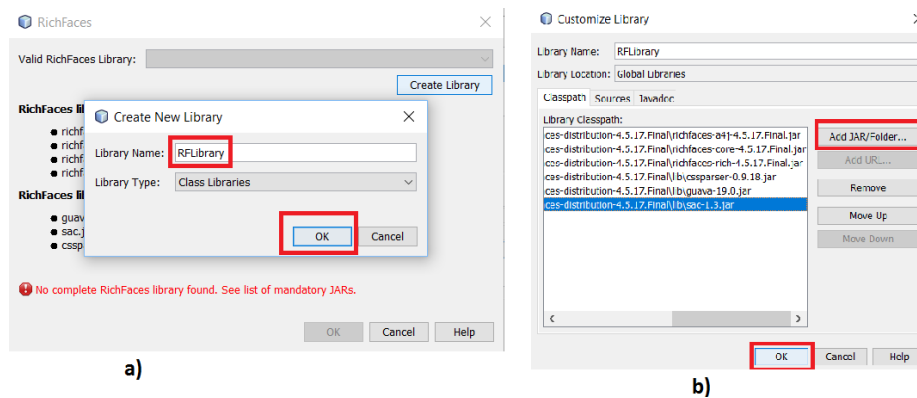
Sada je neophodno vratiti se na Sliku 1 i uočiti dugme "More" desno od **Stringa** kojim je u priloženom prozoru obeležena RichFaces biblioteka. Klikom na ovo dugme otvara se prozor priložen sledećom slikom:



Slika 12.2 RichFaces prozor za kreiranje nove biblioteke

Kao što je moguće primetiti, u ovom prozoru je omogućeno kreiranje nove NetBeans IDE RichFaces biblioteke i priložena je informacija o datotekama koje ona mora da sadrži, kao i neobaveznim datotekama koje mogu biti od koristi prilikom rada sa RichFaces komponentama.

Na jedan od dva pomenuta načina, neophodno je preuzeti **richfaces-distribution-4.5.17.Final.zip** i raspakovati je na izabranoj lokaciji na računaru. Klikom na dugme **Create Library** (Slika 2) otvara se nov prozor u kojem će biti moguće kreiranje naziva biblioteke sa RichFaces komponentama (Slika 2 - a) i dodavanje neophodnih datoteka iz preuzetog ZIP fajla u biblioteku (Slika 2 - b). Sledi navedena slika.



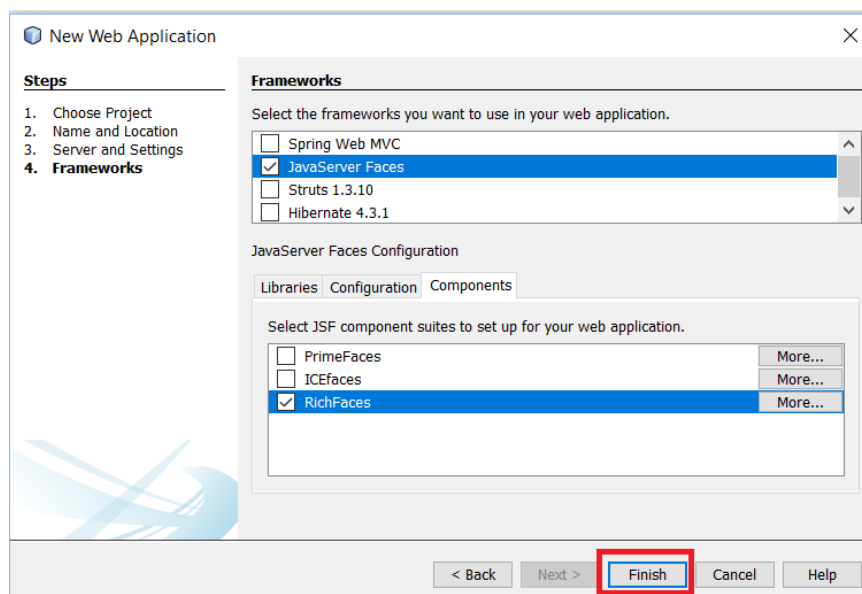
Slika 12.3 Izbor imena i dodavanje JAR fajlova za RichFaces podršku

Klikom na dugme "OK" na Slici 2 - b, kreirana je u razvojnom okruženju NetBeans IDE, biblioteka sa RichFaces komponentama, naziva RFLibrary.

## INICIJALNI RICHFACES PROJEKAT

*Sa obezbeđivanjem potpune podrške moguće je kreirati prvi RichFaces projekat.*

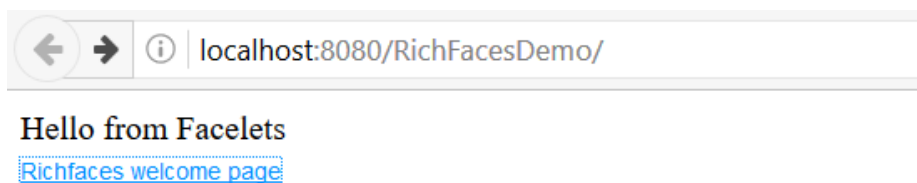
Sa obezbeđivanjem potpune podrške moguće je kreirati prvi RichFaces projekat na način što će ponovo akcenat biti na prozoru prikazanim Slikom 1. **Prozor je ponovo je aktuelan nakon što je razvojno okruženje kreiralo biblioteku RFLibrary.** Prozor sada ne poseduje informacije koje ukazuju da RichFaces biblioteka komponenta nije podešena, a to je pokazatelj da su sve prethodne akcije obavljene na adekvatan način. Klikom na dugme "Finish" (sledeća slika) biće kreiran inicijalni projekat koji je dobio naziv **RichFacesDemo**.



Slika 12.4 Kreiranje RichFaces projekta - kraj



Kada je obaljena i ova poslednja aktivnost, kreiran je veb projekat koji, slično prethodnim slučajevima, sadrži dve JSF stranice: `index.xhtml` i `welcomeRichfaces.xhtml`. Prvom stranicom je realizovana jednostavna pozdravna poruka, a to se može videti na sledećoj slici.



Slika 12.5 Početna stranica kreirane veb aplikacije

Ovoj stranici odgovara sledeći listing koji sadrži automatski generisan kod:

```
<?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:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
    <br/>
    <h:link outcome="welcomeRichfaces" value="Richfaces welcome page" />
  </h:body>
</html>
```

## PRIMER 12 - STRANICA SA RICHFACES KOMPONENTAMA

*Sledi pregled korišćenih specifičnih RichFaces komponenta.*

Na početnoj stranici postoji link pod nazivom "Richfaces welcome page" koji služi za obezbeđivanje navigacije do JSF stranice `welcomeRichfaces.xhtml` koja sadrži kod koji je sačinjen od standardnih JSF i RichFaces komponenta. Izgled navedene stranice je prikazan sledećom slikom.



Slika 12.6 JSF Stranica sa RichFace panelom

Kao što je moguće primetiti, kreirana je stranica sa predefinisanim izgledom kojeg je moguće menjati i prilagođavati u skladu sa željenim sadržajem na stranici. Da bi bilo moguće izvršiti analizu i dati objašnjenje korišćenih specifičnih RichFaces komponenta, neophodno je dati listing koda datoteke `welcomeRichfaces.xhtml` iz kojeg je i generisan izlaz koji je dostupan u veb pregledaču na način prikazan prethodnom slikom. Sledi listing JSF datoteke `welcomeRichfaces.xhtml`:

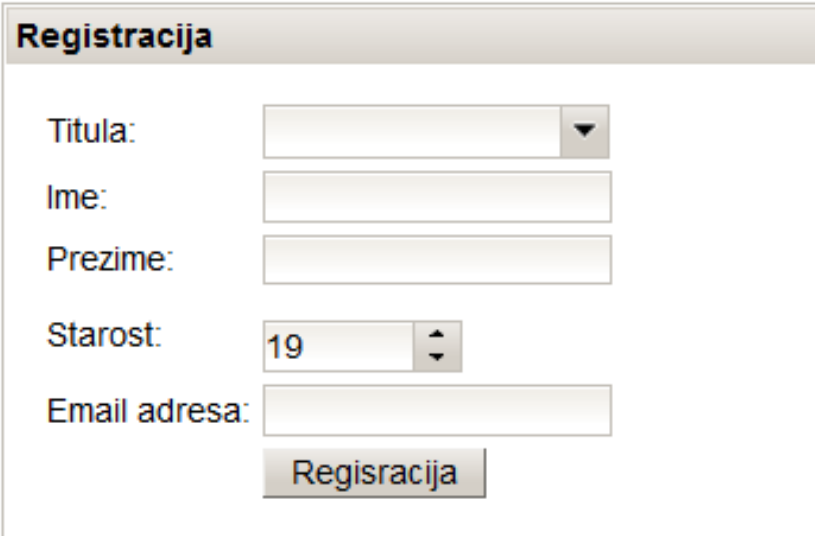
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:rich="http://richfaces.org/rich"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>Richfaces Welcome Page</title>
  </h:head>
  <h:body>
    <rich:panel header="Welcome to Richfaces">
      RichFaces is an advanced UI component framework for easily integrating
      Ajax capabilities into business applications using JSF. Check out the links below
      to learn more about using RichFaces in your application.
      <ul>
        <li><h:outputLink value="http://richfaces.org" >Richfaces Project
        Home Page</h:outputLink></li>
        <li><h:outputLink value="http://showcase.richfaces.org" >Richfaces
        Showcase</h:outputLink></li>
        <li><h:outputLink value="https://community.jboss.org/en/
        richfaces?view=discussions" >User Forum</h:outputLink></li>
        <li><h:outputLink value="http://www.jboss.org/richfaces/docs"
        >Richfaces documentation...</h:outputLink>
        <ul>
          <li><h:outputLink value="http://docs.jboss.org/richfaces/
          latest_4_X/Developer_Guide/en-US/html_single/" >Development
          Guide</h:outputLink></li>
          <li><h:outputLink value="http://docs.jboss.org/richfaces/
          latest_4_X/Component_Reference/en-US/html/" >Component Reference</h:outputLink></li>
          <li><h:outputLink value="http://docs.jboss.org/richfaces/
          latest_4_X/vdldoc/" >Tag Library Docs</h:outputLink></li>
        </ul>
      </li>
    </ul>
  </rich:panel>
</h:body>
</html>
```

Iz priloženog koda je moguće primetiti da je jedini RichFaces specifični tag, koji je korišćen, jeste tag `<rich:panel>` koji je iskorišćen za kreiranje panela koji sadrži tekst i linkove ka stranicama.

## RICHFACES KOMPONENTE - PRIMER - JSF DATOTEKA SA FORMOM

*Inicijalni projekat će biti modifikovan za demonstraciju RichFaces podrške u JSF stranicama.*

Datoteku `welcomefaces.xhtml`, koja je automatski generisana kreiranjem RichFaces projekta, moguće je iskoristiti i redefinisati tako da umesto skupa prikazanih linkova sadrži dobro poznatu formu za unos korisničkih podataka. Forma će imati izgled, definisan u okviru pomenutog `<rich:panel>` RichFaces elementa, koji je prikazan sledećom slikom.



Slika 12.7 Forma za unos podataka kreirana RichFaces komponentama

Stranica, koja prikazuje formu sa slike, realizovana je kodom kojem odgovara sledeći listing:

```
<?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:rich="http://richfaces.org/rich"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>Richfaces Welcome Page</title>
  </h:head>
  <h:body>
    <rich:panel header="Registracija">
      <h:form prependId="false">
        <h:panelGrid columns="3"
                      columnClasses="rightalign,leftalign,left
                      align">
          <h:outputLabel value="Titula: " for="titula"/>
```

```

        <rich:select id="salutation"
            value="#{registrationBean.titula}">
            <f:selectItem itemLabel="" itemValue="" />
            <f:selectItem itemLabel="St." itemValue="ST" />
            <f:selectItem itemLabel="Ma"
                itemValue="MA" />
            <f:selectItem itemLabel="MSc"
                itemValue="MMSC" />
            <f:selectItem itemLabel="Dr" itemValue="DR" />
        </rich:select>
        <rich:message for="titula" />
        <h:outputLabel value="Ime:"
            for="ime" />
        <h:inputText id="ime" label="Ime"
            required="true"
            value="#{registrationBean.ime}"
            />
        <rich:message for="ime" />
        <h:outputLabel value="Prezime:" for="prezime" />
        <h:inputText id="prezime" label="Prezime"
            required="true"
            value="#{registrationBean.prezime}"
            />
        <rich:message for="prezime" />
        <h:outputLabel for="starost" value="Starost:" />
        <rich:inputNumberSpinner id="starost" label="starost"
            value="#{registrationBean.starost}"
            minValue="19"
            maxValue="65" />
        <rich:message for="starost" />
        <h:outputLabel value="Email adresa:"
            for="email" />
        <h:inputText id="email" label="Email adresa"
            required="true"
            value="#{registrationBean.email}">
            <f:validator validatorId="emailValidator" />
        </h:inputText>
        <rich:message for="email" />
        <h:panelGroup>
            <h:commandButton id="register" value="Regisracija"
                action="confirmation" />
        </h:panelGroup>
    </h:panelGrid>
</h:form>
</rich:panel>
</h:body>
</html>

```

## JSF RICHFACES DATOTEKA SA FORMOM - DODATNA RAZMATRANJA

*Sledi analiza priloženog koda sa RichFaces komponentama.*

U narednom izlaganju sledi detaljna analiza modifikovane stranice `welcomefaces.xhtml` koja sada sadrži formu za unos korisničkih podataka. Forma odgovara po sadržaju formama prethodnih primera sa ciljem kvalitetnijeg poređenja i demonstriranja različitih biblioteka komponenta.

Ono što je na početku moguće primetiti, iz priloženog listing koda JSF stranice `welcomefaces.xhtml`, da je forma za unos korisničkih podataka enkapsulirana unutar RichFaces komponente `<rich:panel>`, a to znači da će, prilikom kreiranja sadržaja stranice, forma biti učitana u odgovarajući panel.

Dalje, u kodu je moguće primetiti sledeću RichFaces komponentu `<rich:select>`. Zadatak ove komponente jeste kreiranje **drop - down** liste za izbor titule koja se odnosi na korisnika. Ova komponenta ima određenu prednost u odnosu na standardnu JSF komponentu `<h:selectOneMenu>` budući da ona može biti podešena kao `comboBox`, a to znači da je pored izbora ponuđenih opcija, dozvoljen i unos teksta u GUI kontrolu koja je nastala primenom `<rich:select>` komponente. Da bi ova funkcionalnost bila omogućena, neophodno je podesiti vrednost atributa `enableManualInput`, komponente `<rich:select>` na `true`.

Takođe je moguće primetiti da je priloženim kodom obezbeđen mehanizam za izveštavanje u slučaju javljanja grešaka prilikom unosa. Greške mogu da nastanu na dva načina:

- nije popunjeno obavezno polje forme;
- uneti podatak nema odgovarajući tip ili format.

Upravljanje prikazivanjem grešaka povereno je skupu RichFaces komponenta koje su određene tagom `<rich:message>`.

Izbor godina korisnika je prepušten veoma atraktivnoj kontroli koja je kreirana primenom RichFaces komponente `<rich:inputNumberSpinner>`. Ovom komponentom je kreirana kontrola koja omogućava zadavanje numeričkih vrednosti, koje su u direktnoj vezi sa brojem godina korisnika, bilo direktnim unosom sa tastature ili izborom koji se vrši pomoću strelica sa desne strane kontrole.

U ovom delu lekcije su prikazane samo izabrane RichFaces komponente. Detaljno upoznavanje sa bibliotekom i odgovarajućim komponentama, moguće je angažovanjem dokumentacije koja je dostupna na linku: <http://www.jboss.org/richfaces/docs>.

U nastavku, neophodno je pokazati kako se čuvaju podaci koji se prosleđuju sa jedne stranice na drugu, kako se vrši provera korektnosti unosa RichFaces forme i izvršiti testiranje i demonstraciju veb aplikacije koja je uzeta kao primer.

## UNOS PODATAKA U FORMU I STRANICA ZA PRIKAZIVANJE REZULTATA

*Testiranje unosa u formu kreiranoj na stranici koja sadrži RichFaces komponente.*

Za početak je neophodno izabrati jednu od ponuđenih vrednosti: St. (student), Ma (master), MSc (magistar nauka) ili dr (doktor nauka) kao titulu iz kreiranog RichFaces padajućeg menija izgrađenog primenom komponente `<rich:select>`. Dalje, za potrebe demonstracije, biće popunjena sva tražena polja vodeći računa o tipovima podataka i formatu podataka koji se unose.(prikazano sledećom slikom).

Slika 12.8 Popunjena forma

Kada se popune odgovarajući podaci forme, klikom na dugme "Registracija", podaci se šalju na server, i za njihovo čuvanje je od sada zaduženo CDI zрно registrationBean koje je instanca klase RegistrationBean (više reči u narednom izlaganju). Stranica za prikazivanje podataka, pod nazivom `confirmation.xhtml`, dobija vrednosti koje su unete na formi od pomenutog CDI zrna i prikazuje ih korisniku u veb pregledaču. Sledećom slikom je prikazana stranica koja prikazuje unete korisničke podatke. Pojavljivanje podataka ukazuje da je aplikacija primera uspešno kreirana.

Slika 12.9 Podaci uneti na formi stranice sa RichFaces komponentama

Sledi kod JSF stranice `confirmation.xhtml`, koja je takođe kreirana primenom RichFaces komponenta:

```
<?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:rich="http://richfaces.org/rich"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Podaci o korisniku</title>
  </h:head>
  <h:body>
```

```
<rich:panel header="Registracija">
  <h:form prependId="false">
    <h:panelGrid columns="2"
      columnClasses="rightalign-bold,leftalign">
      <h:outputText value="Titula:"/>
      ${registrationBean.titula}
      <h:outputText value="Ime:"/>
      ${registrationBean.ime}
      <h:outputText value="Prezime:"/>
      ${registrationBean.prezime}
      <h:outputText value="Starost:"/>
      ${registrationBean.starost}
      <h:outputText value="Email adresa:"/>
      ${registrationBean.email}
    </h:panelGrid>
  </h:form>
</rich:panel>
</h:body>
</html>
```

## VALIDACIJA PODATAKA U RICHFACES STRANICI SA FORMOM

*Provera unosa forme kreirane pomoću RichFaces komponentata.*

Kao što je već dobro poznato, validacija unetih podataka u formu je veoma važan zadatak. Ozbiljna veb aplikacija ne bi trebalo da dozvoli unos nekorektnih korisničkih podataka, u smislu ne poklapanja tipova podataka, unosa email adrese koja ne odgovara standardnom email formatu adrese ili unos broja godina koji ne može biti realan.

Ukoliko je polje za unos teksta forme ima atribut **required** podešen na vrednost **true**, polje je obavezno za unos i ukoliko se ne unese odgovarajući podatak u polje u prostoru predviđenom za poruke biće prikazan odgovarajuća poruka sa greškom. Ovo je standardno za sve JSF stranice, a ovde je cilj prikazivanje poruka sa greškama u RichFaces maniru. Navedeno je prikazano sledećom slikom:

The screenshot shows a web form titled "Registracija". It has five input fields: "Titula:" with a dropdown menu showing "Dr"; "Ime:" which is empty and has a red error message "Ime: Validation Error: Value is required." next to it; "Prezime:" with the text "Milicevic"; "Starost:" with a spinner box showing "42"; and "Email adresa:" with the text "vlada@gmail.com". At the bottom of the form is a button labeled "Registracija".

Slika 12.10 0 0 Poruka sa greškom prikazana komponentom

Postoji u drugi scenario, dobro poznati scenario. U polje je unet podatak koji nema odgovarajući format, a to je opet email adresa.

Tada klasa **Validator** (priložen je listing koda klase) stupa na scenu i šalje predefinisano informaciju da je unos email adrese nekorektan. Navedeno je prikazano sledećom slikom.

Slika 12.11 Poruka o nekorektnom String formatu prikazana komponentom

```
package com.metropolitan.IceFaces;

/**
 *
 * @author Vladimir Milicevic
 */
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.component.html.HtmlInputText;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;

@FacesValidator(value = "emailValidator")
public class EmailValidator implements Validator {

    @Override
    public void validate(FacesContext context, UIComponent uiComponent, Object
value) throws ValidatorException {
        Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");
        Matcher matcher = pattern.matcher(
            (CharSequence) value);
        HtmlInputText htmlInputText
            = (HtmlInputText) uiComponent;
        String label;

        if (htmlInputText.getLabel() == null
            || htmlInputText.getLabel().trim().equals("")) {
            label = htmlInputText.getId();
        } else {
            label = htmlInputText.getLabel();
        }

        if (!matcher.matches()) {
```



```
        FacesMessage facesMessage;  
        facesMessage = new FacesMessage(label  
            + ": Email adresa nije korektna");  
  
        throw new ValidatorException(facesMessage);  
    }  
}
```

## RICHFACES - VIDEO MATERIJAL

*Ovaj deo lekcije je zaokružen odgovarajućim video materijalom.*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ZADATAK 11

*Probajte sami*

Pokušajte da realizujete JSF stranice iz primera 8 primenom RichFaces biblioteke komponenta.

## ▼ Poglavlje 13

### Domaći zadatak 3 - 2

#### ZADATAK 2

##### *Izrada domaćeg zadatka broj 2.*

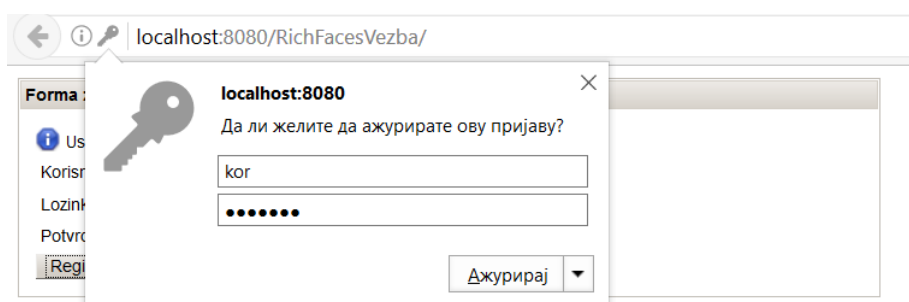
Primenom neke od obrađenih JSF biblioteka komponentata rešiti sledeći zadatak:

- Glavna stranica sadrži sledeći formu za unos i potvrdu korisničkog imena i lozinke. Primer izgleda forme dat je sledećom slikom:




Slika 13.1 Primer izgleda forme

- Unošenjem podataka, veb pregledač će vas pitati da li želite da Vam sačuva podatke (sledeća slika):



Slika 13.2 Snimanje ili ažuriranje registracije

- U slučaju da se ne podudaraju vrednosti za "Lozinka" i "Potvrdite lozinku" prikazati odgovarajuće obaveštenje, kao na primer:



Forma za registraciju

✖ Lozinka mora da odgovara!

Korisnicko ime: kor

Lozinka: •••••••

Potvrdite lozinku: •••••••

Registracija

Slika 13.3 Lozinka i potvrda se ne podudaraju

- Kreirati odgovarajuće CDI zrno za podatke forme i u njemu definisati ograničenja po pitanju minimalne i maksimalne dužine korisničkog imena i lozinke;
- Ispunjeni zadaci se boduju, u zavisnosti od stepena realizacije, do 50% od ukupnog broja predviđenih poena;
- Dodati po vlastitom izboru sadržaj u veb aplikaciju (originalnost i kompleksnost realizovanog se posebno boduje i nosi do 50% od predviđenog broja bodova)

Primer zadatka, koji može biti od koristi, priložen je u dodatnim materijalima Shared Resources uz domaće zadatke. Domaći zadatak ne sme da bude prosta kopija ovog zadatka, ali može da se oslanja na njega.

## ▼ Poglavlje 14

# Zaključak

## ZAKLJUČAK

*Lekcija se bavila kreiranjem i upotrebom JSF stranica u Java veb aplikacijama.*

Lekcija se bavila kreiranjem i upotrebom JSF stranica u Java veb aplikacijama. Lekcija je posebno istakla da JavaServer Faces (JSF) predstavlja specifičan Java okvir ( framework ) namenjen za razvoj širokog spektra Java veb aplikacija. Vodeći se navedenim, u ovoj lekciji je bio cilj pokazivanje načina na koji primena JSF pojednostavljuje razvoj veb aplikacija. U tu svrhu su bile obrađene neke značajne teme za analizu i demonstraciju ovog specifičnog Java okvira za razvoj veb aplikacija:

1. Kreiranje JSF projekta primenom izabranog razvojnog okruženja, konkretno je korišćen NetBeans IDE razvojno okruženje;
2. Postavljanje JSF tagova za korišćenje prednosti koju obezbeđuje primena taga JSF `<h:panelGrid>`;
3. Korišćenje statičke i dinamičke navigacije za definisanje navigacije između stranica;
4. Razvoj CDI (Contexts and Dependency Injection for the Java EE Platform) zrna za enkapsulaciju podataka i programske logike;
5. Implementacija JSF validatora za proveru valjanosti unosa pojedinih polja forme;
6. Kreiranje JSF šablona i isticanje prednosti njihove primene;
7. Rad sa JSF temama i isticanje prednosti njihove primene.

Posebno značajno u prethodnom izlaganju predstavlja činjenica da je lekcija insistirala na analizi i demonstraciji primenom odabranih primera koji su na najbolji način istakli i pokazali prednosti primene JSF stranica u Java veb aplikacijama. Od posebnog značaja je bila fokusiranost primera na rad sa šablonima stranica i temama za pokazivanje veoma jednostavnog načina kreiranja JSF stranica sa predefinisanim atraktivnim izgledom. Konkretni primer je, takođe, pokazao kako jedna šablon JSF datoteka na veoma jednostavan način upravlja nizom JSF datoteka klijenata šablona.

Izlaganje, započeto u teorijskom delu lekcije, posebno je zaokruženo zadacima koji su kompletno testirani i demonstrirani u materijalima za vežbe. Svaki od navedenih zadataka, odgovara kompletno urađenoj i funkcionalnoj veb aplikaciji čiji je kod priložen kao dodatni materijal u Objektu učenja "JSF - vežbe" u okviru kategorije Shared Resources.

Savladavanjem ove lekcije studenti su osposobljeni da koriste JSF tehnologiju za razvoj Java veb aplikacija pored detaljno obrađene JSP tehnologije iz prethodnih lekcija. Studenti su kroz ove dve tehnologije stekli znanja koja im omogućavaju savladavanje naprednijih koncepata

Java EE veb aplikacija čija analiza i demonstracija sledi kroz lekcije koje će biti izučavane odmah nakon ove lekcije.

## ZAKLJUČAK - NASTAVAK

*Lekcija je nastavila izlaganje o razvoju veb aplikacija primenom standardnih JSF komponenata.*

Lekcija je nastavila izlaganje o razvoju veb aplikacija primenom standardnih JSF komponenata. Posebno je u ovoj lekciji istaknuto da JSF ima jednu veoma dobru i moćnu osobinu - proširivost. JSF tehnologija dozvoljava programerima da razvijaju vlastite JSF komponente pa je u tu svrhu u lekciji bilo govora o nekoliko poznatih JSF biblioteka komponenata koje u velikoj meri olakšavaju razvoj veb aplikacija. Lekcija se, upravo, fokusirala na tri najpoznatije i najpopularnije biblioteke JSF dokumenata:

- *PrimeFaces;*
- *ICEfaces;*
- *RichFaces.*

Posebno, a u saglasnosti sa prethodno navedenim, lekcija je detaljno obradila sledeće teme, uz obavljanje pratećih zadataka:

- Primena PrimeFaceskomпоненata u JSF aplikacijama, podešavanje biblioteke razvojnog okruženja i demonstracija primerom;
- Primena IceFaces komponenata u JSF aplikacijama, podešavanje biblioteke razvojnog okruženja uz preuzimanje datoteka za podršku i demonstracija primerom;
- Primena RichFaces komponenata u JSF aplikacijama, podešavanje biblioteke razvojnog okruženja uz preuzimanje datoteka za podršku i demonstracija primerom.

Savladavanjem ove lekcije studenti su stekli, takođe, kompletniju sliku o primeni JSF stranica u Java veb aplikacijama.

## LITERATURA

*U pripremi lekcije korišćena je najnovija literatura.*

1. Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito. 2014. Java Platform, Enterprise Edition The Java EE Tutorial, Release 7, ORACLE
2. David R. Heffelfinger. 2015. Java EE7 Developomnet With NetBeans 8, PACK Publishing
3. Yakov Fain. 2015. Java 8 programiranje, Kombib (Wiley)
4. Josh JUNEau. 2015. Java EE7 Recipes, Apress
5. <https://www.tutorialspoint.com/jsf/>