

Smer: JAVA Development

Tema: Web aplikacija – Naučna konferencija “Etran”

Mentor : Polaznik:

Darko Živanović Dejana Rapaić

Beograd, 2016.

Sadržaj

[1. Opis aplikacije 3](#_Toc462052638)

[2. Korišćeni alati 3](#_Toc462052639)

[2.1 NetBeans 3](#_Toc462052640)

[2.2 Java 4](#_Toc462052641)

[2.3 HTML I CSS 4](#_Toc462052642)

[2.4 MySQL 4](#_Toc462052643)

[3. Upustvo za korisnike 5](#_Toc462052644)

[4. Upustvo za administratora 8](#_Toc462052645)

[5. Baza podataka 9](#_Toc462052646)

[6. Upustvo za programere 10](#_Toc462052647)

[6.1 Uvod u izradu aplikacije 10](#_Toc462052648)

[6.2 Izrada aplikacije 11](#_Toc462052649)

[6.2.1 Izrada korisničkog dela aplikacije 13](#_Toc462052650)

[6.2.2 Izrada administratorskog dela aplikacije 18](#_Toc462052651)

[7. Literatura 19](#_Toc462052654)

# Opis aplikacije

Tema ove aplikacije je naučna konferencija “Etran”. Osnovne informacije o održavanju konferencije su date na statičkim stranicama ovog web sajta. Ideja je da se, pored pružanja informacija, korisnicima omogući da se ovim putem prijave za učešće na konferenciji tako što se prvobitno registruju na sajt, nakon što popune određeni formular sa ličnim podacima i podacima o zaposlenju. Nakon registracije, korisnik može da se prijavi na svoj nalog, a dodatna mogućnost koju stiče je mogućnost prijave rada za učešće na konferenciji, što podrazumeva upload fajla i popunjavanje formulara vezanog za rad. Kada prođe rok za prijavu rada, administratori dostavljaju te prijave recenzentima, a nakon izvršene recenzije administratori svakom radu dodeljuju komentar i odluku o učešću na konferenciji. Interakcija između administrator sajta i recenzenta nije prikazana u projektu. Administrator je taj koji na stranici sa radovima unosi odluku i komentar recenzenta. On takođe može da briše prijave radova i korisničke naloge. Podatke ne može da menja, ali može da ih menja sam korisnik.

# Korišćeni alati

Za izradu ovog projektnog rada korišćeni su sledeći alati: NetBeans, Java programski jezik, HTML, CSS, MySql baza podataka.

### 2.1 NetBeans

Softver koji sam koristila kao razvojno okruženje za web aplikaciju je Net Beans IDE (integrated development environment - IDE), jedno od najpopularnijih pored Eclipse i IntelliJIDEA. Net Beans je besplatan, radi na raznim operativnim sistemima (cross-platform) - Microsoft Windows, Mac OS X, Linux, Solaris i drugim platformama koje podržavaju Java Virtualnu Mašinu (JVM). Vlasništvo je Oracle korporacije.

NetBeans je razvojno okruženje prvenstveno namenjeno razvoju Java tehnologija, ali isto tako pruža dosta dodatnih mogućnosti koje mu omogućavaju da se jednako efikasno može koristiti za razvoj računarskih programa I u ostalim programskim jeyicima kao što su C, C++, PHP.. Podržava različite tehnologije i alate koji poboljšavaju razvojni process aplikacije.

### 2.2 Java

Programski jezik Java je jedan od najmoćnijih programskih jezika današnjice, sveobuhvatan, podržan od strane velike zajednice. I serverski i klijentski deo su pisani u Javi. JSP tehnologija je Javina web tehnologija kojom pravimo dinamičke web strane integracijom HTML koda sa Java kodom.

Java je objektni orjentisani programski jezik, koji je razvila kompanija  [Sun Microsystems](https://sr.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) početkom devedesetih godina. Java je višenamenski programski jezik. Osnovne karakteritike su mu portabilnost, jednostavnost, pouzdanost, objektna orjentacija. Ovaj programski jezik nije vezan za operativni system.

### 2.3 HTML I CSS

HTML (HyperText Markup Language) je platformski neutralan jezik koji omogudava izvršavanje programa na daljinu. Danas ovaj jezik predstavlja standard za izradu Internet dokumenata.

CSS (akronim od Cascading Style Sheets) je style sheet jezik koji se koristi za opis prezentacione semantike dokumenta pisanog u opisnom jeziku (markup language). Jednostavnije rečeno, CSS opisuje, tj. uređuje izgled i formatiranja bilo kog elementa na stranici. HTML i CSS imaju potpuno drugačiju namenu, ali se idealno dopunjuju. HTML nosi sadržaj, dok CSS taj isti sadržaj uređuje.

### 2.4 MySQL

Baza podataka MySQL je open sors višenitni, višekorisnički SQL sistem za upravljanje bazama podataka. Sistem radi kao server, obezbjeđujući višekorisnički interfejs za pristup bazi podataka. MySQL je besplatna baza, ima veliku zajednicu korisnika. Sada je vlasništvo Oracle korporacije.

Softver koji sam koristila kao razvojno okruženje baze podataka je *MySQL Workbench*. Softver ima tri namene: SQL razvoj (pisanje upita, skriptova, izrada baze i objekata baze i upravljanje njima), modelovanje baze (pravljenje ER modela, forward i reverse engineering, ...) i administracija SQL servera (podešavanje server baze, podešavanje korisničkih naloga, praćenje logova...).

# 3. Upustvo za korisnike

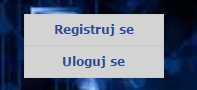
Prilikom prve posete sajta, otvara se naslovna strana koja sadrži osnovne informacije o konferenciji. Navigacioni meni za neregistrovanog korisnika je prikazan na slici 1.



*Slika 1.*

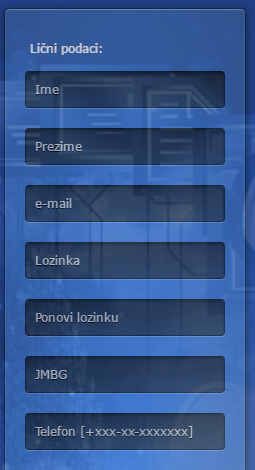
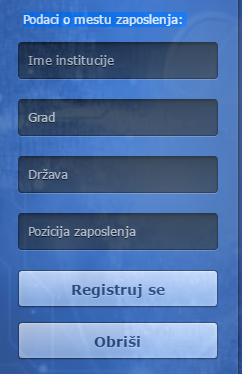
Detaljnije informacije u vezi konferencije, kao što su predsedništvo, tehnički komitet i savet se nalaze na stranicama koje su smeštene u delu navigacije pod imenom “komiteti”. Stranice “Poziv za radove” i “Uputstvo za autore” pružaju detaljne informacije neophodne autorima koji žele da se prijave za učešće na konferenciji. Na stranici “odobreni radovi” nalazi se spisak odobrenih radova za učešće na konferenciji sa imenima autora.

U desnom delu prozora nalaze se linkovi za registraciju i logovanje (slika 2).



*Slika 2.*

Da bi korisnik mogao da se uloguje, potrebno je prvo da se registruje, a formular za registraciju prikazan je na slici 3.

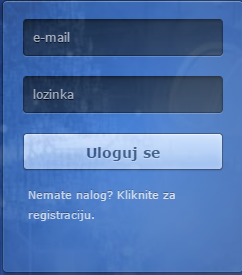
 

*Slika 3.*

Korisnik može samo jednom da se registruje sa određenom e-mail adresom i zbog toga se prilikom registracije proverava da li već ostoji nalog sa navedenom e-mail adresom. Ukoliko postoji, korisnik se obaveštava da već postoji registrovan nalog sa navedenom e-mail adresom.

Prijava je uspešna tek kada se podaci u formularu popune ispravno. Potrebno je popuniti sva polja; e-mail adresa mora biti uneta u formatu aaa@bbb.ccc; polja “Lozinka” i “Ponovi lozinku” moraju da se poklapaju; JMBG mora da sadrži samo numeričke vrednosti i tačno 13 cifara; broj telefona mora biti ukucan u prikazanom formatu (+xxx-xx-xxxxxxx) i može da sadrži 11 ili 12 cifara.

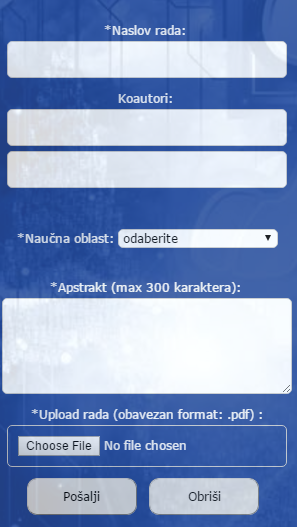
Na slici 4 prikazan je formular za prijavu na korisnički nalog (logovanje).

**

*Slika 4.*

Nakon unosa e-maila i lozinke korisnik je uspešno prijaljen na svoj nalog. Detalje svog profila može videti klikom na “Moj profil”. Na toj stranici može se izvršiti izmena ličnih podataka (klikom na “Izmeni”). Jedini podatak koji se ne može promeniti je e-mail adresa. Klikom na “Obriši” trajno se poništava korisnički nalog, tj. iz baze podataka se trajno brišu .

Dodatna mogućnost koja se dobija je prijavljivanje radova za učešće na konferenciji, klikom na “Prijava rada”. Prikaz formulara dat je na slici 5.



*Slika 5.*

Ovavezna polja su označena simbolom “\*”.

Autor može prijaviti maksimalno 2 rada, kao što je i navedeno na stranici Uputstvo za autore. U slučaju da se na korisničkom nalogu nalazi više od 2 rada, biće uzeta u obzir poslednja 2 prijavljena. Takođe, na konferenciju se mogu prijaviti samo radovi koji, pored glavnog autora, mogu imati još maksimalno 2 koautora. Polja za unos imena koautora su jedina koja nisu obavezna.

Moguće je izvršiti samo upload fajlova sa ekstenzijom .pdf. Fajlovi se smeštaju u folder RadoviUpload koji se nalazi u okviru projekta u folderu Web Pages.

Prilikom prijavljivanja rada, u bazu podataka se upisuje datum prijave rada. Zato je potrebno da autori poštuju rok za prijavu koji je naveden na stranici “Poziv za radove”, jer radovi koji su prijavljeni nakon navedenog roka se neće uzeti u razmatranje i biće obrisani od strane administratora.

Na stranici “Moji radovi” tabelarno je prikazan spisak radova gde se dalje mogu videti detalji u vezi svakog rada ponaosob (uz dodatni komentar o radu i odluku u vezi učešća rada na konferenciji), može se pregledati upload-ovani fajl,a mogu se i izmeniti podaci o radu (klikom na “izmeni”). Takođe se može u potpunosti poništiti prijava određenog rada klikom na polje “Obriši”. U tabeli je data informacija o statusu rada koji dodeljuje administrator sajta – “odobren” ili “neodobren” za učešće na konferenciji. Sve dok administrator ne unese odluku o statusu rada, ovo polje će imati vrednost “Nije još dodeljen status”. Odluke se donose nakon što prođe rok za prijavljivanje radova.

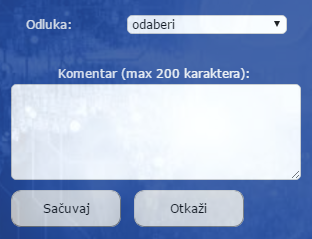
# 4. Upustvo za administratora

Administrator, putem url-a <http://localhost:8080/KONFERENCIJA/admin.jsp>, dospeva na stranicu na kojoj može da se uloguje kao administrator, koristeći korisničko ime i lozinku koji su dati u tabeli (u bazi) pod nazivom “admini”. Administrator nema mogućnost izmene podataka, već samo brisanja naloga, brisanja prijave rada i postavljanjem komentara i odluke o statusu rada.

Klikom na “Spisak autora” dobija tabelarni prikaz svih registrovanih korisnika (ID, ime, prezime i e-mail). Klikom na:

* “Detalji” - otvara se stranica sa svim informacijama o korisniku;
* “Obriši” – briše se korisnički nalog;
* “Spisak radova” – dobija se tabelarni prikaz prijavljenih radova (ID rada, naslov rada, opcija “Detalji o radu”, status rada (odobren/neodobren/nije još dodeljen status) i opcija “Obriši rad”.

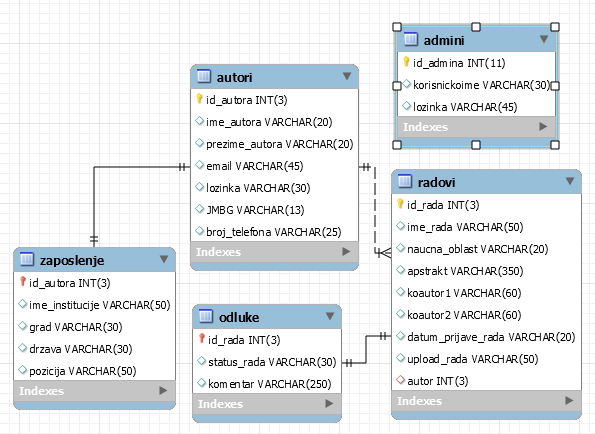
Na stranici sa detaljima o radu stoje sve informacije u vezi prijave rada i sam rad koji se može pogledati klikom na link koji stoji posle naziva “RAD”. Ovo je ujedno i stranica na kojoj admin dodeljuje status radu uz prateći komentar - klikom na “Dodeli status” otvara se stranica sa formularom (slika 6).



*Slika 6.*

Kako se koji rad proglasi odobrenim, on se pojavljuje na stranici „Odobreni radovi“ kojoj može prisupiti i registrovani i neregistrovani korisnik. Administrator može da menja odluku i prateći komentar uz rad svaki put kada odabere opciju „Dodeli status“.

# 5. Baza podataka



*Slika 7.*

Prema zadatim parametrima, odnosno informacijama od značaja, pravi se baza podataka pod nazivom “konferencija\_baza”. Ona se sastoji od 5 tabela, kao što se vidi na slici 7.

Glavna tabela se zove “autori” u kojoj se nalaze osnovne informacije o registrovanom korisniku. U tabeli “zaposlenje” se nalaze dodatne informacije o korisniku i ona sadrži strani ključ kojim je povezana sa tabelom “autori” (veza tipa “one-to-one” (1:1)). Tačnije, primarni ključ u tabeli “zaposlenje” određen je primarnim ključem tabele “autori”. U pitanju su kolone sa istim nazivom u obe tabele: id\_autora. Obe tabele se popunjavaju prilikom registracije.

Prilikom registracije, korisnik unosi u odgovarajuća polja za unos ime, prezime, e-mail, lozinku, JMBG, broj telefona (navedeni podaci se smeštaju u tabelu “autori”), zatim unosi naziv institucije u kojoj je trenutno zaposlen, ime grada I države u kojoj se ta institucija nalazi kao I poziciju na kojoj se nalazi u okviru institucije (navedeni podaci se smeštaju u tabelu “zaposlenje”, a primarni ključ ove tabele se nasleđuje automatski od tabele “autori”) .

Nakon registracije korisnik može da se uloguje na osnovu e-maila I lozinke, a tada mu se nudi mogućnost da popuni formular za prijavu rada, čiji se podaci smeštaju u tabelu “radovi”. Svaki registrovan korisnik, tj autor. može da prijavi više radova, tako da veza između tabela “autori” i “radovi” je tipa “one-to-many” (1:n). To znači da tabela “radovi” sadrži strani ključ prema tabeli autori koji povezuje kolone autori (tabela “radovi”) i id\_autora(tabela autori). Tabela radovi takođe ima svoj primarni ključ id\_rada koji je auto-inkrementalan.

Tabela “odluke” se automatski formira prilikom prijave rada zato što je ona vezana za tabelu “radovi” vezom “one-to-one”, njen primarni ključ je određen primarnim ključem tabele “radovi – id\_rada. Prilikom kreiranja tabele “odluke”, kolone status\_rada i komentar tabel se popunjavaju podrazumevanim stringovima "Nije još dodeljen status!" i "Nije dodeljen komentar!".

Tabela “admini” je zasebna i sadrži samo korisničko ime i lozinku administratora.

# 6. Upustvo za programere

### 6.1 Uvod u izradu aplikacije

S obzirom da je reč o web aplikaciji, korišćena verzija jave je Java EE - nasuprot Javi SE koja se koristi za razvoj desktop aplikacija. Razlika u verzijama Jave je u bibliotekama, pošto Java EE ima biblioteke koje proširuju funkcionalnost standardne verzije zbog razvoja aplikacija za web, poput JPA (Java Persistance API - specifikacija programerskog interfejsa za rad sa bazama podataka).

Takođe, za razliku od programiranja u Javi SE, web programiranje u Javi EE razlikuje 2 celine aplikacije: serverska i klijentska logika. Java klase koje razvijamo kao serversku logiku se zovu servleti. To je serverski kod, naš "motor" web aplikacije. Web server koji obrađuje serversku logiku je (u slučaju Net Beans-a) GlassFish.

Kada korisnik web sajta, odnosno veb aplikacije, napravi zahtev, kao što će to u našem slučaju biti unos podataka u web formu i slanje podataka u bazu, web server će pokrenuti serverski kod i opslužiti zahtev klijenta. Izborom metode obraćanja serveru mi određujemo kako će server reagovati na zahtev klijenta. Postoje 2 najčešće korištene metode obraćanja serveru preko internet HTTP protokola: to su GET i POST. GET metoda zahteva podatke sa web resursa, dok POST metoda šalje podatke serveru na obradu. Naša POST metoda preuzima podatke iz kontrola za unos podataka na našoj web stranici i šalje ih serveru parametrizacijom tj. transferom parametara koji se nalaze u telu zahteva. POST metoda nema ograničenje na količinu podataka koju šaljemo server, tako da se, za razliku od GET medote, koristi za prenos fajlova.

Klijentsku logiku pokreće naš pregledač veb-sadržaja poput Mozilla firefox, Google chrome, IE... Međutim, ono što razlikuje statički sadržaj kakav ima HTML od dinamičkog je to što je u dinamičkom sadržaju kakav je na JSP veb stranicama (Java Server Pages) je u sam kod "utkana" i logika koja izvršava određene funkcionalnosti, stvarajući tu "dinamiku".

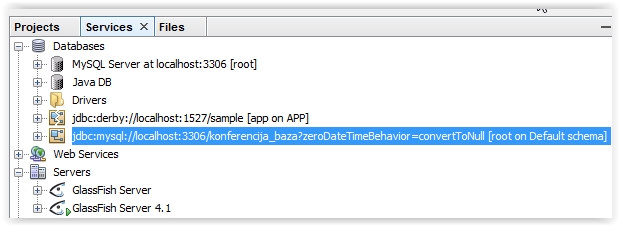
JSP je tehnologija koja u samu stranicu implementira tagove i skriplete kako bi se kod ubacio u stranicu, a da pritom klijentu nije vidljiv. JSP je upravo ono što sam koristila pri izradi aplikacije jer zadatak i funkcionalnost veb aplikacije nameće dinamičke stranice sa rad sa podacima preko web brauzera, a to JSP omogućava. Zajednički naziv za Java servlete i JSP je Java Web Components.

Karakteristika Jave je postojanje klasa, dok je karakteristika baze podataka postojanje tabela. Proces koji premošćava jaz između relacioniog i objektno orijentisanog modela zove se ORM (objektno-relaciono mapiranje). Alati koji automatizuju ovaj process nazivaju se ORM alati. U ovom projektu je korišćen *Hibernate*. To je *open source* alat za objektno-relaciono mapiranje u Javi. On omogućava razvoje perzistentnih klasa, koristeći opšte kocepte Jave. *Hibernate* vodi računa o mapiranju Java klasa i tabela u bazi podataka i omogućava kreiranje upita. Ovi upiti mogu biti napisani SQL upitnim jezikom ili *Hibernate* upitnim jezikom HQL – *Hibernate Query Language*.

### 6.2 Izrada aplikacije

Prvi korak u izradi projekta bio je kreiranje baze podataka koristeći *MySQL Workbench*. Za rad sa bazom podataka neophodno je učitavanje biblioteke MySQL JDBC Driver. Kreira se konekcija na bazu kako bi mogli da se u sistem povlače podaci iz baze i pohranjuju u bazu. Program treba da vrši spoj podataka iz forme za prikupljanje podataka kao i manipulaciju i izmenu istih.

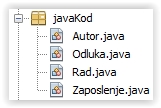
Konekcija na bazu se iz Net beans-a vrši preko JDBC MySQL konektora preko porta 3306 (slika 8).



*Slika 8.*

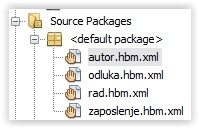
Drugi korak u izradi projekta je bio dizajn JSP stranica korišćenjem HTML-a i CSS-a (folder Web Pages).

Nakon toga se prave klase objekata u Java programu koji odgovaraju bazi. One se nalaze u paketu javaKod, kao što je prikazano na slici 9.



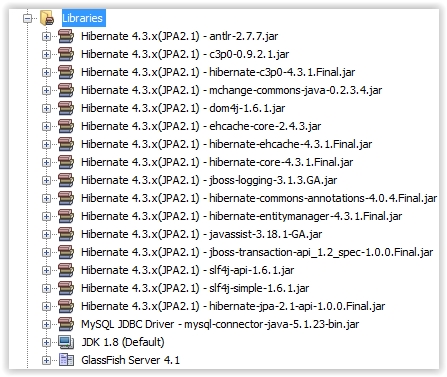
*Slika 9*

Da bi se uspostavila veza između ovih objekata u javi i odgovarajućih tabela, izvršila sam mapiranje klasa pomoću XML-a. Mapirajuće datoteke se nalaze u paketu <default package> i imaju ekstenziju hbm.xml (slika 10).



*Slika 10*

Da bi se omogućilo korišćenje *Hibernate*-a, izvršeno je učitavanje biblioteke Hibernate 4.3.x(JPA2.1). Učitane biblioteke prikazane su na slici 11.



*Slika 11*

Nakon ovih koraka Hibernate konfiguracioni fajl izgleda kao na slici 12.



*Slika 12*

### 6.2.1 Izrada korisničkog dela aplikacije

HTML kod za navigacioni meni se nalazi na stranici „header.jsp“, a u ostale jsp stranice je ugrađen kroz jsp *include* direktivu: <jsp:include page="header.jsp" />.

Korisnički deo aplikacije prikazuje se u 2 varijante – za registrovanog (ulogovanog) i neregistrovanog korisnika. Registrovani korisnik može da pristupi dvema dodatnim stranicama u odnosu na neregistrovanog korisnika, a to su „prijava rada.jsp“ i „mojiRadovi.jsp“.

Nakon što korisnik ispuni formular za registraciju koji se nalazi na stranici „registracija.jsp“, JavaScript funkcija validateForm() proverava ispravnost unosa, i kada vrati vrednost „true“ podaci iz forme šalju se serveru POST metodom. Na stranici „registracijaIspis.jsp“ u okviru java skripleta otvara se sesija i započinje Hibernate transakcija (slika 13).



*Slika 13*

Kao što se vidi iz koda sa slike, parametri iz formulara se čuvaju u objektima Autor i Zaposlenje. Pre cuvanja podataka u bazi, vrši se provera da li u bazi već postoji navedena e-mail adresa (pošto ona mora biti jedinstvena). To se ostvaruje pozivanjem metode checkEmail() koja se nalazi u klasi proveraEmaila.java i vraća vrednost „true“ ukoliko taj e-mail već postoji u bazi. Ukoliko ova metoda vrati vrednost „false“, *Hibernate* će se pobrinuti da svi podaci iz formulara budu smešteni u odgovarajuće tabele u bazi.

Nakon uspešne registracije, korisnik može da se uloguje tako što u na stranici „login strana.jsp“ unese svoj email i lozinku. I ovom prilikom JavaScript funkcija proverava da li su oba polja popunjena i da li je ispravan format email adrese. Uneti podaci se ovaj put obrađuju putem Java servleta LoginServlet.java (slika 14).

**

*Slika 14*

U okviru klase LoginDao.java poziva se metoda validate() u okviru koje se uspostavlja konekcija sa bazom putem JDBC-a i proverava se da li uneti podaci odgovaraju podacima iz baze (u tom slučaju funkcija vraća vrednost „true“). Ukoliko odgovaraju, atribut „logged“ dobije vrednost „true“, a korisniku se dodeljuje sesija koja nosi podatke o email-u i ID-ju korisnika. Ti podaci se čuvaju u atributima „ID“ i „email“ i njihove vrednosti se mogu dohvatiti po potrebi sve dok je korisnik ulogovan, naredbom: request.getSession(true).getAttribute("email"); analogno je za ID.

Atribut „logged“ se koristi da se određene stavke prikažu zavisno od toga da li je korisnik ulogovan ili nije, što se vidi na slici 15, u delu koda sa stranice „header.jsp“. To se realizuje korišćenjem JSTL (Java Standard Tag Library) tagova.



*Slika 15*

Biblioteke tagova se dele po kategorijama i potrebno je u okviru stranice na kojoj se koriste tagovi učitati deskriptor za svaku biblioteku. U ovom slučaju se radi o bibliteci „Core“: <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>.

Klikom na polje „Izloguj se“ korisnik može u svakom trenutku da prekine svoju sesiju, a to se ostvaruje naredbom: request.getSession(true).invalidate(); vrednost atributa „logged“ dobija vrednost null.

Dok je ulogovan, korisnik može da popuni formular za prijavu rada, na stranici „prijava rada.jsp“ i upload-uje svoj fajl, isključivo u pdf formatu: accept="application/pdf,application/vnd.ms-excel. JavaScript funkcija obezbeđuje da sva polja budu popunjena i podaci se šalju na server isključivo POST metodom, pošto GET metoda ne podržava slanje binarnih podataka. Podatke obrađuje Java servlet Upload\_rada.java i čuva ih u bazi pozivajući metodu save() koja se nalazi u klasi RadDao.java. Takođe se u bazu upisuje podatak o trenutnom vremenu i datumu, dobijen nizom sledećih naredbi:

DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

Date date = new Date();

String datumPR = dateFormat.format(date);

Što se tiče upload-ovanog fajla, on se smešta u folder RadoviUpload (u okviru projekta), a putanja je definisana u klasi Config.java i glasi:

C:\\Users\\dejana.rapaic\\Desktop\\Projekti\\KONFERENCIJA\_VERZIJA\_103\_1\\web\\RadoviUpload

U tabeli u bazi se čuva samo naziv tog fajla. Deo koda koji je zalužan za obradu i čuvanje upload-ovanog fajla se takođe nalazi u okviru servleta Upload\_rada.java i prikazan je na slici 16:



*Slika 16*

RadDao.java je možda i najvažnija java klasa u ovom projektu jer se u njoj nalaze najvažnije metode za rad sa bazom a to su:

* getConnection() – uspostavlja se veza sa bazom putem JDBC-a;
* save(Rad r) – u tabelu „radovi“ se smeštaju podaci naredbom INSERT;
* update(Rad r) – u tabeli „radovi“ se menjaju podaci naredbom UPDATE u koloni sa tačno određenim ID-jem. Poziva se u okviru servleta Upload\_izmenaRada.java - to je servlet koji obrađuje podatke iz formulara na stranici mojRad\_update.jsp, gde se po potrebi menjaju podaci uneti uz upload-ovan fajl ;
* delete(int id) – brišu se svi podaci o radu sa određenim ID-jem, kao i kolona u tabeli „odluke“ koja mu odgovara. Poziva se u okviru servleta RadDeleteServlet.java – to je servlet koji se „odaziva“ kada korisnik želi da obriše svoj rad ;
* getRadById(int id) – ova metoda vraća podatke o radu na osnovu ID-ja. Poziva se gde god je potrebno dobiti ID rada nad kojim se vrši neka akcija (npr. na stranici mojRad\_detalji.jsp) ;

Spisak odobrenih radova sa imenima autora (na stranici ListaOdobrenihRadova.jsp) se dobija pozivom metode getOdobreniRadovi() u klasi odobreniRadovi, što je vidljivo i registrovanom i neregistrovanom korisniku.

Kada korisnik želi da promeni lične podatke klikom na opciju „izmeni“ (stranica „mojProfil.jsp“), otvara mu se stranica mojProfil\_update.jsp gde može da u okviru formulara izmeni sve podatke osim email-a. Obradu podataka iz ovog formulara vrši servlet autorUpdateServlet.java i vrši izmene u bazi pozivajući metodu update() koja se nalazi u klasi autorUpdate.java.

Kada korisnik želi da obriše svoj nalog klikom na polje „Obriši nalog“, akciju izvršava autorDeleteServlet.java koji, pozivom metode delete() u klasi autorDelete, briše sve podatke u svim tabelama u bazi koji su vezani za ovog korisnika.

### 6.2.2 Izrada administratorskog dela aplikacije

# Kao što je već rečeno, administratorskom delu sajta se pristupa uz pomoć url-a <http://localhost:8080/KONFERENCIJA_VERZIJA_104/admin.jsp>. adminLoginServlet.java vrši obradu zahteva za prijavljivanje, tako što poziva metodu validate() u klasi admin\_login\_provera.java koja proverava da li uneti podaci odgovaraju podacima iz tabele admini. Ukoliko je tako, atribut „alogged“ ima vrednost „true“ sve dok se administrator ne izloguje.

# Na stranici admin\_spisakAutora.jsp prikazan je tabelarno spisak svih registrovanih korisnika sajta. To je realizovano na sledeći način: formira se lista objekata tipa Autor pozivanjem metode getSviAutori() koja se nalazi u klasi SviAutori.java (paket domain) i korišćenjem for petlje ispisuju se u svakom redu pojedinačno podaci o korisniku. Klikom na opciju „obriši“ iz baze se brišu svi podaci iz tabela koji su vezani za tog korisnika, a taj zahtev obrađuje java servlet admin\_autorDeleteServlet.java. Ovaj servlet briše podatke tako što poziva metodu delete() klase autorDelete.java, koja briše podatke o korisniku na osnovu ID-ja autora.

Lični podaci o autoru su prikazani na stranici admin\_detaljiAutora.jsp. Poziva se metoda getAutorByID() u klasi getAutorByIDautora.java koja vraća objekat tipa Autor i ti podaci se ispisuju na stranici naredbom out.println.

Na stranici admin\_spisakRadovaPoAutoru.jsp tabelarno je prikazan spisak radova za određenog korisnika, a podaci su dobijeni pozivom metode getRadoviByAutor() koja vraća listu objekata tipa Rad, u klasi MojiRadovi.java. Svaki rad se može obrisati klikom na opciju „Obriši“. Taj zahtev obrađuje servlet admin\_RadDeleteServlet.java pozivom metode delete() u klasi RadDao.java. Detalji rada su prikazani na stranici admin\_detaljiRada.jsp. Klikom na opciju „Dodeli status“ otvara se stranica admin\_odlukaIkomentar.jsp gde administrator u padajućem meniju bira jednu od opcija (odobren/neodobren), popunjava polje za komentar i klikom na „Sačuvaj“ servlet admin\_donesiOdluku.java obrađuje zahtev, pozivom metode save\_o() u klasi admin\_sacuvajOdluku.java. Odluku i komentar može da menja svaki put kada odabere opciju „Dodeli status“. Čim bude odobren, rad se automatski pojavljuje na stranici ListaOdobrenihRadova.jsp.

# 

# 7. Literatura

[1] Darko Živanović – Slajdovi sa predavanja ITAcademy 2015/16.

[2] Vuk Ninić – Slajdovi sa predavanja ITAcademy 2014/15.

[3] http://www.w3schools.com

[4] http://www.javatpoint.com

[5] http://www.stackoverflow.com/