

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jaka Šporin

**Spletna aplikacija za prijavo  
študentov na mednarodno izmenjavo**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Alenka Kavčič

Ljubljana, 2014



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 00398 / 2013  
Datum: 3.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JAKA ŠPORIN**


Naslov: **SPLETNA APLIKACIJA ZA PRIJAVO ŠTUDENTOV NA MEDNARODNO  
IZMENJAVO  
WEB APPLICATION FOR STUDENTS TO APPLY FOR AN  
INTERNATIONAL EXCHANGE**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:


V okviru diplomske naloge razvijte spletno aplikacijo, ki omogoča prijavo študentov na razpis za mednarodno izmenjavo v okviru programa Erasmus+. Aplikacija naj nudi intuitiven uporabniški vmesnik za enostaven vpis vseh potrebnih osebnih podatkov študenta ter podatkov o izbranih tujih institucijah. Pragtako je pomemben tudi administracijski del, ki naj omogoča enostaven izvoz podatkov o prijavljenih študentih v različnih formatih, spreminjanje navodil za oddajo prijave, brisanje podatkov o prijavah, spreminjanje seznama institucij in podobno. Posebno pozornost pri razvoju aplikacije posvetite tudi varnosti, predvsem avtentikaciji uporabnikov in preprečevanju morebitnega nepooblaščenega dostopa do podatkov prijavljenih študentov. Predvidite tudi možnost povezave vaše aplikacije z obstoječimi informacijskimi rešitvami fakultete z uporabo spletnih servisov.

Mentor:

  
viš. pred. dr. Alenka Kavčič



Dekan:

  
prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Jaka Šporin, z vpisno številko **63090417**, sem avtor diplomskega dela z naslovom:

*Spletna aplikacija za prijavo študentov na mednarodno izmenjavo*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Alenke Kavčič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. januarja 2014

Podpis avtorja:



*Zahvaljujem se mentorici viš. pred. dr. Alenki Kavčič za prijaznost, popravke in koristne nasvete pri izdelavi diplomske naloge. Posebna zahvala gre tudi moji družini za vso podporo v času študija.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>O aplikaciji</b>	<b>3</b>
2.1	Namen aplikacije . . . . .	3
2.2	Uporabniški del aplikacije . . . . .	5
2.3	Administracija . . . . .	5
<b>3</b>	<b>Razvojne tehnologije in orodja</b>	<b>7</b>
3.1	Potrebna programska orodja . . . . .	7
3.1.1	Spletni strežnik Apache . . . . .	7
3.1.2	Prevajalnik PHP . . . . .	8
3.1.3	MySQL . . . . .	8
3.2	Uporabljene tehnologije . . . . .	9
3.2.1	HTML . . . . .	9
3.2.2	CSS/CSS3 . . . . .	9
3.2.3	PHP . . . . .	10
3.2.4	JavaScript . . . . .	11
3.2.5	SQL . . . . .	11
3.2.6	XML . . . . .	12
3.2.7	AJAX . . . . .	12

## KAZALO

<b>4</b>	<b>Razvoj aplikacije</b>	<b>13</b>
4.1	Analiza zahtev in arhitektura sistema . . . . .	14
4.2	Izdelava uporabniškega vmesnika . . . . .	15
4.3	Opis funkcionalnosti aplikacije . . . . .	22
4.3.1	Prijava v aplikacijo . . . . .	22
4.3.2	Implementacija prijavnih obrazcev . . . . .	25
4.3.3	Shranjevanje podatkov v podatkovno bazo . . . . .	31
4.3.4	Kreiranje dokumentov PDF . . . . .	33
4.3.5	Administracija . . . . .	35
4.4	Testiranje aplikacije . . . . .	39
<b>5</b>	<b>Sklepne ugotovitve</b>	<b>43</b>
	<b>Literatura</b>	<b>45</b>
	<b>Seznam slik</b>	<b>47</b>

# Seznam uporabljenih kratic in simbolov

- AJAX (angl. *Asynchronous JavaScript and XML*) - skupina tehnologij za ustvarjanje dinamičnih spletnih strani,
- CSS (angl. *Cascading Style Sheets*) - stilne predloge, v katerih je zapisana oblika spletne strani,
- CSV (angl. *Comma Separated Values*) - format tekstovne datoteke, v kateri so vsi podatki ločeni z enim ločilom,
- HTML (angl. *Hyper Text Markup Language*) - označevalni jezik za izdelavo spletnih strani,
- JavaScript - objektni skriptni programski jezik za razvoj interaktivnih spletnih strani,
- jQuery - knjižnica za skriptni jezik JavaScript, ki prinaša zbirko funkcij za hitrejši razvoj spletnih aplikacij,
- LDAP (angl. *Lightweight Directory Access Protocol*) - programski protokol za poizvedovanje in spreminjanje imeniških storitev, ki teče preko TCP/IP,
- MySQL (angl. *My Structured Query Language*) - odprtokodni sistem za upravljanje zbirk podatkov,

- PDF (angl. *Portable Document Format*) - odprt standard za izmenjavo elektronskih dokumentov,
- PHP (angl. *PHP Hypertext Preprocessor*) - skriptni programski jezik, namenjen izdelavi dinamičnih spletnih strani,
- SQL (angl. *Structured Query Language*) - strukturirani povpraševalni jezik za delo s podatkovnimi bazami,
- SSL (angl. *Secure Sockets Layer*) - kriptografski protokol, ki omogoča varno komunikacijo na medmrežju,
- XAMPP (angl. *X(cross-platform) Apache MySQL PHP Perl*) - odprtokodna programska oprema za izdelavo dinamičnih spletnih strani,
- XML (angl. *Extensible Markup Language*) - razširljiv označevalni jezik.

# Povzetek

V diplomski nalogi je predstavljen razvoj spletne aplikacije, ki je namenjena prijavi študentov Fakultete za računalništvo in informatiko na Erasmus študijsko izmenjavo ali praktično usposabljanje.

V prvem delu naloge se osredotočimo predvsem na namene in cilje izdelave aplikacije. Na kratko je predstavljen tudi dosedanji postopek prijave študentov na študijsko izmenjavo in prakso, ki je potekal brez spletne aplikacije.

V glavnem delu naloge so podrobneje opisani glavni koraki izdelave spletne aplikacije, tako z uporabniškega kot tudi s tehničnega vidika. Uporabniški vidik zajema predstavitev glavnih funkcionalnosti aplikacije in prikazuje, kaj aplikacija omogoča in kako poteka delo z njo. S tehničnega vidika so predstavljene vse uporabljene tehnologije in programska orodja. Podrobneje smo predstavili tudi razvoj nekaterih ključnih komponent aplikacije, ki so bistvene za njeno delovanje, in pa večje težave, s katerimi smo se soočali tekom izdelave aplikacije.

V zadnjem delu naloge so predstavljena še glavna spoznanja, do katerih smo prišli med izdelavo aplikacije. Presodili smo, ali so bili realizirani vsi začetni cilji in ima aplikacija vse funkcionalnosti, ki smo si jih zadali pred izdelavo. V tem delu naloge so na kratko predstavljene tudi nekatere možnosti nadaljnjega razvoja in nadgradnje aplikacije.

**Ključne besede:** spletna aplikacija, Erasmus, študijska izmenjava, praksa, PHP, MySQL



# Abstract

The thesis presents the development of a web application, intended for the application of students of the Faculty of Computer and Information Science for an Erasmus study exchange or practical training.

In the first part of the thesis we mainly focus on the purposes and aims of the application development. We also briefly present the so far application procedure of students for the study exchange and training, which took place without the web application.

In the main part of the thesis are thoroughly described the substantial steps of the web application development from the user's and technical point of view. The user's point of view includes the presentation of the application's main features, namely its functionality and workflow. From the technical point of view are presented all used technologies and software. We also thoroughly presented the development of some key components of the application that are crucial for its functioning, and larger issues, that we were facing during its development.

In the final part of the thesis we present the main findings, to which we came during the development of the application. We assessed, if all initial aims were realized and if the application has all the functionalities that we have set before the development. In this part of the thesis are also briefly presented some possibilities for further development of the application.

**Keywords:** web application, Erasmus, study exchange, practical training, PHP, MySQL



# Poglavje 1

## Uvod

Uporaba računalnikov in spleta se je v zadnjih letih močno povečala. Danes velja že skoraj pravilo, da se večino obrazcev in ostalih dokumentov prenaša iz papirnate v elektronsko obliko. Tako vse več prijav poteka preko različnih spletnih aplikacij, ki omogočajo lažje in hitrejše izpolnjevanje obrazcev, zmanjšujejo možnost za napačne vnose, po drugi strani pa omogočajo tudi veliko lažje upravljanje z obsežnimi množicami pridobljenih podatkov. Podatke, ki jih dobimo preko spletnih obrazcev, je veliko lažje shranjevati, urejati in arhivirati, poleg tega pa jih je mogoče tudi enostavno prenašati med različnimi aplikacijami. Vse to daje podatkom veliko večjo vrednost.

Po drugi strani pa pri elektronskem komuniciranju obstaja veliko večja možnost zlorabe podatkov, saj lahko le-ti hitro pridejo v roke tretje osebe, ki sploh ne bi smela imeti dostopa do njih. Ta oseba lahko podatke uporabi v svojo korist ali pa jih samo spremeni, s pa tem naredi določeno škodo uporabniku aplikacije. Da je možnosti za takšno zlorabo podatkov čim manj, je treba poskrbeti za varno delovanje spletnih aplikacij, predvsem pa za varen prenos in shranjevanje podatkov. S tem poskrbimo, da lahko do podatkov dostopajo in jih po potrebi spreminjajo samo osebe, ki imajo za to dovoljenje.

Z namenom lažjega upravljanja s podatki o prijavah je bila narejena tudi spletna aplikacija za prijavo študentov na Erasmus študijsko izmenjavo in prakso. Ker so do sedaj prijave potekale preko papirnatih obrazcev, je bilo

težko voditi natančno evidenco o prijavljenih študentih, poleg tega je bilo treba vse podatke ročno vnašati v računalnik. Seveda pa bo aplikacija tudi študentom precej olajšala prijavo in izpolnjevanje prijavnih obrazcev.

Pred izdelavo spletne aplikacije smo si zadali več ciljev, ki bi jih lahko razdelili na tri dele:

1. Izdelava spletnih obrazcev, ki bodo nadomestili dosedanje papirnate prijavnice. Preko obrazcev bo aplikacija pridobila vse podatke o posameznem študentu, ki so potrebni za prijavo na Erasmus študijsko izmenjavo ali prakso.
2. Izdelava administracije, preko katere lahko administrator pregleduje vse podatke o prijavljenih študentih. Administracija mora omogočati tudi izvoz podatkov o prijavah in upravljanje z bistvenimi deli baze podatkov preko uporabniškega vmesnika.
3. Končni izdelek mora biti povsem delujoča aplikacija, pripravljena za uporabo. Poskrbljeno mora biti tudi za varno delovanje in predvsem za ustrezno varnost podatkov, ki jih pridobimo s pomočjo spletne aplikacije.

# Poglavje 2

## O aplikaciji

Glavni namen aplikacije je spletna prijava študentov Fakultete za računalništvo in informatiko na Erasmus študijsko izmenjavo ali prakso. Delo z aplikacijo bo zasnovano za dva tipa uporabnikov: študente (uporabniški del) in administratorje (administracija).

### 2.1 Namen aplikacije

Z razvojem spletne aplikacije smo poskušali nadomestiti in predvsem čim bolj poenostaviti dosedanji postopek prijave študentov na praktično usposabljanje in študijsko izmenjavo.

Do sedaj so se študenti na izmenjave in prakse prijavljali s pomočjo papirnatih prijavnic, ki jih je bilo treba do določenega datuma izpolniti in nato oddati na fakulteti. Prijavnici sta bili za obe vrsti prijave povsem ločeni. Študenti, ki so se želeli prijaviti tako na prakso kot tudi na izmenjavo, so morali izpolniti obe prijavnici, kljub temu, da sta si obe prijavi vsebinsko zelo podobni. Prijavnici se namreč razlikujeta le v manjšem delu, ki se izrecno nanaša na podatke o praksi ali izmenjavi.

Pri prijavi študentov na študijsko izmenjavo igra pomembno vlogo tudi seznam institucij, na katere se je možno prijaviti. S tega seznama lahko posamezen študent izbere le tiste institucije, ki izvajajo izmenjave za sto-

pnjo študija, v kateri bo študent v času izmenjave. Tako so morali študenti s pomočjo seznama najprej poiskati ustrezne institucije, na katere se lahko prijavijo. Seveda pa seznam institucij ni stalen, saj se z leti nekoliko spreminja.

Ko se je prijavni rok iztekel, so morali na fakulteti vse podatke o prijavljenih študentih ročno vnašati v računalnik. To je bilo precej zamudno opravilo, poleg tega pa je lahko prišlo tudi do napak pri vnašanju podatkov. V primeru velikega števila papirnatih prijavnic je imela fakulteta razmeroma slab pregled nad vsemi podatki, hkrati pa je bilo oteženo tudi razvrščanje študentov glede na vrsto prijave.

Z vidika študentov bo aplikacija precej poenostavila postopek prijave, saj bodo morali določene podatke vnesti le ob prvi prijavi. Ob morebitni naslednji prijavi bodo ti podatki že shranjeni v podatkovni bazi, zato bo študent moral vnesti le specifične podatke, ki se nanašajo na samo izmenjavo ali prakso. Močno bo olajšan tudi sam izbor institucije za izmenjavo. Aplikacija bo v podatkovni bazi hranila seznam vseh institucij in bo že sama izbrala tiste, na katere se lahko posamezen študent prijavi glede na njegovo stopnjo študija.

Največjo vlogo pa bo spletna aplikacija imela pri obdelavi prijav. Za razliko od papirnatih prijavnic bo omogočala zelo dober pregled nad vsemi pridobljenimi podatki. Vsi podatki o prijavah bodo namreč od začetka do konca prijav zbrani na enem mestu, in sicer v podatkovni bazi. Ročno vnašanje v računalnik tako ne bo več potrebno, saj bo omogočen tudi izvoz podatkov v takšnem formatu, ki bo omogočal njihovo nadaljnjo obdelavo.

## 2.2 Uporabniški del aplikacije

Jedro aplikacije predstavljajo trije ločeni prijavnimi obrazci, preko katerih se študenti prijavijo na študijsko prakso ali izmenjavo.

S pomočjo prvega obrazca, ki je skupen za obe prijavi, bo aplikacija pridobila vse potrebne osnovne podatke o študentu (osebni podatki in podatki o študiju).

Dva manjša obrazca sta namenjena pridobivanju bolj specifičnih podatkov, ki se bodo nanašali na samo vrsto prijave (praksa ali izmenjava). V primeru prijave na Erasmus izmenjavo bo moral študent v obrazec vnesti podatke o treh izbranih ustanovah, ki bodo prikazane glede na stopnjo študija. Če se bo študent prijavljal na študijsko prakso, bo v obrazec vnesel vse potrebne podatke o izvajanju prakse (trajanje, institucija ipd.).

Ko bo študent uspešno izpolnil in oddal prijavo, se bodo vsi podatki shranili v podatkovno bazo. Izpolnjeno prijavnico bo treba tudi natisniti in podpisano oddati. V ta namen se bodo vsi potrebni podatki o prijavi izpisali tudi v dokument PDF, ki ga bo študent ob koncu prijave prejel po elektronski pošti. Aplikacija omogoča tudi pregled in tiskanje prijave kar preko uporabniškega vmesnika.

## 2.3 Administracija

Drugi del aplikacije je administracija, ki je namenjena le določenim osebam (administratorjem), medtem ko študenti do nje nimajo dostopa. Administracija omogoča pregled in urejanje podatkov o prijavah preko enostavnega uporabniškega vmesnika.

Glavne funkcionalnosti administracije so:

- Tabelarni prikaz podatkov o vseh prijavljenih študentih skupaj, o študentih, ki so prijavljeni na študijsko prakso in o študentih, ki so prijavljeni na študijsko izmenjavo.
- Izvoz vseh zgoraj naštetih podatkov v Excelovo datoteko.

- Brisanje vseh podatkov o prijavljenih študentih iz podatkovne baze.
- Posodobitev in pregled seznama institucij, na katere se lahko študenti prijavijo v okviru študijske izmenjave.
- Posodobitev oz. vnos navodil za končno oddajo natisnjene prijave, ki se študentu izpišejo po uspešno izpolnjeni spletni prijavi.
- Sprememba gesla za prijavo v administracijo.

## Poglavje 3

# Razvojne tehnologije in orodja

Za razvoj spletnih strani in aplikacij se uporabljajo različna orodja. V nadaljevanju so na kratko predstavljena vsa programska orodja in tehnologije, ki smo jih uporabili pri razvoju naše spletne aplikacije.

### 3.1 Potrebna programska orodja

#### 3.1.1 Spletni strežnik Apache

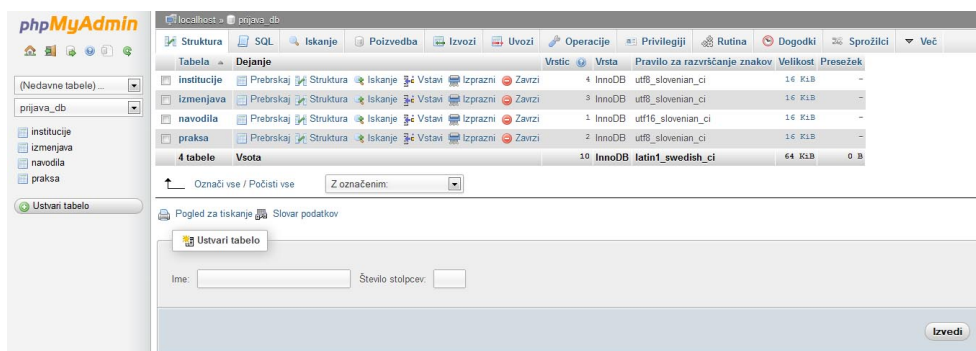
Ker smo razvijali spletno aplikacijo v programskem jeziku PHP, ki primarno teče na spletnem strežniku, smo morali najprej namestiti strežnik Apache. Strežnik skrbi za prejem zahtev, obiskovalcu spletišča pa odgovarja z ustreznimi datotekami HTML [3]. Za potrebe razvoja naše aplikacije v lokalnem okolju smo uporabili kar paket XAMPP [16]. XAMPP je odprtokoden sistem, s katerim lahko postavimo lasten spletni strežnik. Programski paket vsebuje vse, kar potrebujemo za delo z dinamičnimi spletnimi stranmi - od podpore programskemu jeziku PHP in Apache strežnika do podatkovne baze MySQL.

### 3.1.2 Prevajalnik PHP

Jezik PHP za prevajanje in interpretiranje kode potrebuje podporo PHP, ki je tudi že vključena v paket XAMPP [16]. Z namestitvijo PHP dodamo spletnemu strežniku sposobnost, da se poveže s programsko opremo PHP. Bistveno pri tem je, da strežnik obiskovalcu strani ne bo odgovoril neposredno z datoteko PHP, ampak jo bo najprej poslal prevajalniku PHP. Prevajalnik bo ustrezno prevedeno datoteko (v obliki HTML) poslal nazaj strežniku in ta datoteka bo prikazana tudi obiskovalcu spletne strani.

### 3.1.3 MySQL

Za shranjevanje podatkov o prijavljenih študentih se uporablja podatkovna baza MySQL [10], zato aplikacija potrebuje tudi strežnik MySQL. Za dostop do podatkovne baze poskrbijo funkcije, ki so že vgrajene v PHP, medtem ko smo za uvoz in izvoz podatkovne baze namestili programsko orodje PhpMyAdmin [12]. Velika prednost omenjenega orodja je enostaven uporabniški vmesnik (slika 3.1), ki omogoča upravljanje s podatkovnimi bazami in enostaven pregled nad vsemi podatki v bazi.



Slika 3.1: Pregleden uporabniški vmesnik orodja PhpMyAdmin.

## 3.2 Uporabljene tehnologije

### 3.2.1 HTML

HTML (angl. *HyperText Markup Language*) [2] je označevalni jezik, ki je namenjen izdelavi spletnih strani. Struktura in vsebina spletne strani sta predstavljeni z dokumentom, v katerem morajo biti pravilno uporabljene (ugnezdene) značke HTML. S pomočjo tako strukturiranega dokumenta spletni brskalnik nedvoumno sestavi izgled spletne strani. Kot prikazuje tudi izsek kode 3.1, vsak dokument HTML praviloma sestoji iz dveh glavnih delov. Prvi del je glava (angl. *head*), ki vsebuje meta oznake, povezave do zunanjih dokumentov in razne skripte. Drugi del strani je telo (angl. *body*), ki vsebuje celoten vidni del spletne strani in je ponavadi razdeljen z večimi elementi tipa `div`.

```
1 <html>
2 <head>
3 <title>Prijava na Erasmus</title>
4 <link rel="stylesheet" type="text/css" href="css/style.css" />
5 </head>
6 <body>
7 <div id="prijava">Obrazec za prijavo</div>
8 </body>
9 </html>
```

Izsek kode 3.1: Primer HTML strukture enostavne spletne strani.

### 3.2.2 CSS/CSS3

CSS (angl. *Cascade Style Sheet*) [1] so kaskadne stilske podloge, s pomočjo katerih točno deklariramo oblikovna pravila za posamezne skupine elementov HTML. Določamo lahko barve, velikosti, odmike, poravnave, obrobe in vrsto drugih atributov. Prav tako lahko nadziramo aktivnosti, ki jih uporabnik

nad elementi strani izvaja (npr. obarvanje povezave ob prehodu z miško). Še nekoliko več atributov lahko določamo z novejšo različico CSS3. Pomembna vloga uporabe CSS je predvsem ločitev strukture in vsebine spletne strani od njene oblikovne predstavitve. To nam močno olajša urejanje stilov, hkrati pa poskrbimo za večjo preglednost datoteke HTML. S pomočjo CSS pravil precej zmanjšamo ponavljanje kode, saj lahko na množici strani uporabimo iste podloge. CSS pravila določamo na podlagi dveh skupin elementov (class in id), ki ju določimo posameznemu HTML elementu, in številnih selektorjev (p, a, ul, li . . .) (izsek kode 3.2).

```
1 p
2 {
3   color:red;
4   text-align:center;
5 }
```

Izsek kode 3.2: Primer uporabe CSS pravil za selektor p.

### 3.2.3 PHP

PHP (angl. *PHP HyperText Preprocessor*)[8] je odprtokodni programski jezik, ki se uporablja za strežniške programe, predvsem za izdelavo dinamičnih spletnih strani in aplikacij. Za prevajanje programske kode potrebujemo ustrezno različico PHP in strežnik. Datoteke PHP se prevajajo na strežniku, odjemalec pa dobi rezultat v obliki HTML. Koda programskega jezika PHP se v datoteki nahaja med oznakama `<?php` in `?>`, kar prikazuje tudi izsek kode 3.3.

```
1 <?php
2 function writeMsg(){
3     echo "Hello world!";
4 }
```

```
5 writeMsg(); // call the function
6 ?>
```

Izsek kode 3.3: Primer enostavne funkcije v programskem jeziku PHP, ki izpiše besedilo 'Hello World'.

### 3.2.4 JavaScript

JavaScript je skriptni programski jezik [4], ki je namenjen izdelavi dinamičnih spletnih strani in aplikacij. JavaScript koda se vstavi med elemente HTML, kar je podobno kot pri jeziku PHP. Pomembna razlika pa je, da se JavaScript izvaja v brskalniku, ki teče na strani odjemalca. Veliko se uporablja predvsem v povezavi s knjižnico jQuery [6], ki programerju zelo olajša programiranje spletnih aplikacij (izsek kode 3.4).

```
1 $(document).ready(function(){
2     $('input[type="text"]').addClass("idleField");
3 });
```

Izsek kode 3.4: Enostavna jQuery funkcija, ki HTML polju tipa input doda razred 'idleField'.

### 3.2.5 SQL

SQL (angl. *Structured Query Language*) je najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnimi bazami [3].

```
1 INSERT INTO prijave (ime_studenta, priimek_studenta)
2 VALUES (ime, priimek);
```

Izsek kode 3.5: Enostavna poizvedba SQL za vnašanje podatkov v podatkovno bazo.

SQL omogoča raznoliko delo s podatki. Osnovni koraki so vnos (izsek kode 3.5), urejanje in brisanje podatkov. Z nekoliko zahtevnejšimi poizvedbami je možno izvajati tudi iskanja, grupiranja, sortiranja in številne druge operacije.

### 3.2.6 XML

XML (angl. *Extensible Markup Language*) [17] je razširljiv označevalni jezik, ki predstavlja format za opisovanje strukturiranih podatkov, arhitekturo za prenos podatkov in izmenjavo podatkov med omrežji. Zaradi enostavne in pregledne zgradbe, ki je prikazana tudi v izseku kode 3.6, je zelo primeren za komuniciranje med različnimi spletnimi aplikacijami.

```
1 <?xml version="1.0"?>
2 <student>
3     <ime>Janez</ime>
4     <priimek>Novak</priimek>
5 </student>
```

Izsek kode 3.6: Struktura enostavnega dokumenta XML.

### 3.2.7 AJAX

AJAX (asinhroni Javascript in XML) [4] je ime za skupino medsebojno povezanih razvojnih tehnik, ki se uporabljajo za razvoj interaktivnih spletnih aplikacij. Bistvena vloga tehnologije AJAX je izmenjevanje podatkov med spletnim strežnikom in aplikacijami, ki poteka asinhrono v ozadju, brez potrebe po ponovnem nalaganju strani. Podatki se prenašajo s pomočjo objektov `XMLHttpRequest`. To uporabniku omogoča hitrejšo in bolj tekočo spremljanje ter spreminjanje vsebine na spletnih straneh.

## Poglavje 4

# Razvoj aplikacije

Ker so bile zahteve za razvoj naše aplikacije znane vnaprej in dobro definirane, smo uporabili metodologijo razvoja, ki temelji na klasičnem zaporednem modelu. Ker delo ni potekalo v skupini, smo metodologijo prilagodili tako, da je bil razvoj bolj fleksibilen. S tem namenom smo združili načrtovanje modulov z implementacijo. Razvoj aplikacije je tako potekal v treh korakih:

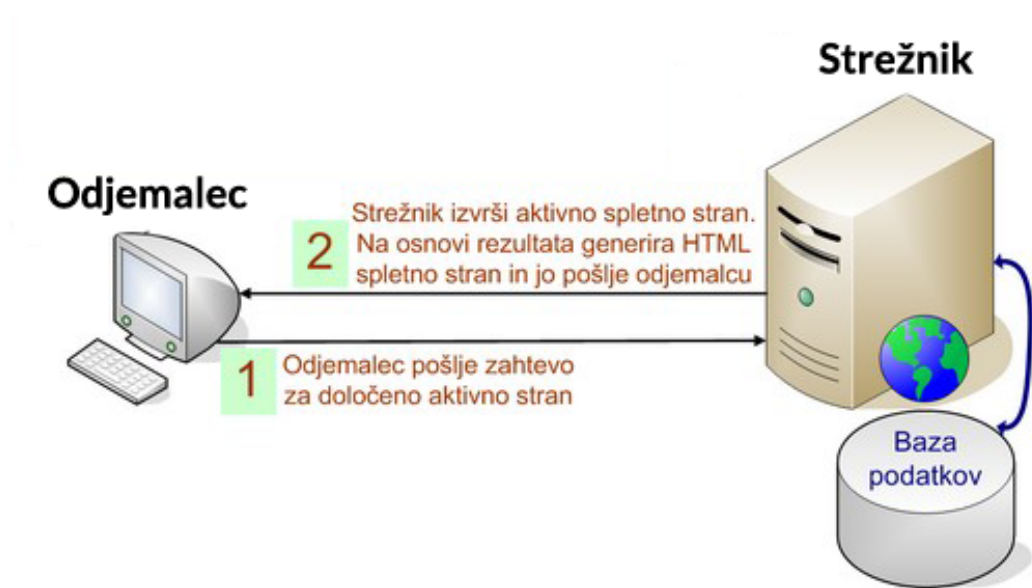
- analiza zahtev in načrt arhitekture sistema,
- inkrementalni razvoj aplikacije, ki zajema izdelavo oz. oblikovanje uporabniškega vmesnika, programiranje funkcionalnosti in varno shranjevanje podatkov v podatkovno bazo,
- testiranje in namestitvev aplikacije v delovno okolje.

V nadaljevanju so opisani glavni koraki izdelave naše spletne aplikacije, predstavljene pa so tudi nekatere težave, s katerimi smo se srečevali tekom razvoja.

## 4.1 Analiza zahtev in arhitektura sistema

Uporabniške zahteve so bile jasno določene in v celoti znane že pred začetkom razvoja. Na podlagi dobro definiranih zahtev smo si zastavili tudi glavne cilje pri razvoju aplikacije in naredili okvirni načrt izdelave, ki vključuje vse zgoraj naštetih korake.

Pred začetkom razvoja aplikacije smo morali načrtovati tudi arhitekturo sistema. To je bila razmeroma lahka naloga, saj je za tovrstne spletne aplikacije najprimernejša arhitektura odjemalec-strežnik, ki je zelo poenostavljeno prikazana na sliki 4.1. Omenjeno arhitekturo [5] bi lahko definirali kot porazdeljeno procesiranje (delitev procesiranja in podatkov) med enim ali več odjemalčevimi računalniki, na katerih se izvaja aplikacija, in enim strežniškim računalnikom, ki nudi storitve vsakemu izmed odjemalcev. Vsi računalniki so ponavadi med seboj povezani v omrežje.



Slika 4.1: Zelo poenostavljen prikaz delovanja arhitekture odjemalec-strežnik [19].

Odjemalce bi lahko definirali kot enouporabniške delovne postaje, na katerih se izvaja ustrezen uporabniški vmesnik, ki zagotavlja predstavitev in

obdelavo podatkov, povezovanje ter storitve podatkovne baze [5].

Strežnik lahko opredelimo kot en ali več večuporabniških računalnikov z deljenim pomnilnikom, ki omogoča obdelavo, povezovanje, storitve baze in ustrezen vmesnik [5].

## 4.2 Izdelava uporabniškega vmesnika

Pri načrtovanju uporabniškega vmesnika smo se trudili, da bi bil vmesnik čimbolj preprost, intuitiven in enostaven za uporabo. Zato smo morali pri načrtovanju in oblikovanju vmesnika paziti na nekatere pomembne dejavnike, ki vplivajo na preglednost, izgled in uporabnost spletne aplikacije. Glede na to, da osnovo aplikacije predstavljajo obrazci za prijavo, je bilo treba poskrbeti predvsem za ustrezno oblikovanje in postavitev vnosnih polj znotraj obrazcev.

Za izdelavo in oblikovanje ogrodja uporabniškega vmesnika smo uporabili HMTL in CSS/CSS3. Ker gre za spletno aplikacijo, do katere uporabniki dostopajo preko spletnega brskalnika, je izgled prilagojen formatu spletne strani.

Pri načrtovanju in oblikovanju uporabniškega vmesnika smo skušali zagotoviti predvsem naslednjim pravilom [18]:

- **Prilagodi se realnemu svetu**

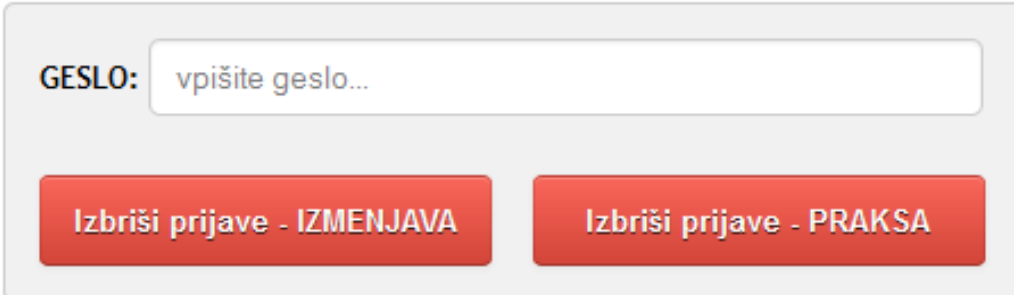
Aplikacija temelji na papirnatih obrazcih, ki so se uporabljali do sedaj, zato smo pri opisu vnosnih polj za podatke uporabili standardno terminologijo (npr. ime, priimek...). Za pomembne povezave smo uporabili gumbe, ponekod tudi ikone (puščice za naprej), kar poveča jasnost in nedvoumnost vmesnika.

- **Konsistentnost in standardi**

Konsistentnost v aplikaciji zagotavljamo predvsem tako, da imajo namensko sorodni elementi tudi podoben izgled, ne glede na to, kje se

nahajajo (enak izgled vnosnih polj, gumbov ipd.) Za gumbе so uporabljene standardne oblike (izbočenost, gradienti), kar zagotavlja podobnost z ostalimi aplikacijami. Barve gumbov so prilagojene njihovim funkcijam; rdeča barva za brisanje pomembnih podatkov (slika 4.2) in modra barva za ostale funkcionalnosti.

**S klikom na spodnje gumbе boste izbrisali vse prijave iz baze podatkov!**



The image shows a user interface for a login form. At the top, there is a label "GESLO:" followed by a text input field containing the placeholder text "vpišite geslo...". Below the input field, there are two prominent red buttons with white text. The left button is labeled "Izbrši prijave - IZMENJAVA" and the right button is labeled "Izbrši prijave - PRAKSA". The entire form is enclosed in a light gray rounded rectangle.

Slika 4.2: Gumba za brisanje podatkov o prijavah sta drugačne barve od ostalih gumbov, kar uporabnika opozori na večjo previdnost pri brisanju podatkov.

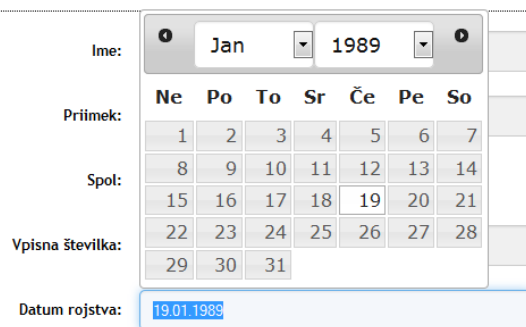
- **Izogibanje napakam pri vnosu podatkov, pomoč in dokumentacija**

Pri načrtovanju spletnih obrazcev je zelo pomembno, da že vnaprej v čim večji meri preprečimo napačne vnose. Kot napačne vnose lahko smatramo podatke v napačnem formatu, vsebinsko neustrezne podatke in prazna polja. Da bi v naši spletni aplikaciji uporabnik takoj vedel, v kakšnem formatu mora vnašati podatke, smo v nekaterih vnosnih poljih uporabili namige. Kjer so možni odgovori že vnaprej znani in je treba le izbrati eno izmed možnosti (v našem primeru letnik študija, naziv institucije ipd.), smo namesto vnosnih polj uporabili spustne sezname. Pri vnosu datumov se uporabniku prikaže koledar (slika 4.3), iz katerega je treba le izbrati željen datum. Uporabniku s tem zelo olajšamo izbiro,

poleg tega pa tudi zagotovimo, da so pridobljeni podatki v pravilnem formatu in tudi vsebinsko ustrezajo zahtevam.

### Osebni podatki

Ime:	<input type="text" value="Jan"/> <input type="text" value="1989"/>
Priimek:	<input type="text"/>
Spol:	<input type="text"/>
Vpisna številka:	<input type="text"/>
Datum rojstva:	<input type="text" value="19.01.1989"/>



Slika 4.3: Ob kliku na vnosno polje za izbiro datuma se uporabniku prikaže koledar.

S pomočjo natančnih navodil smo uporabniku tudi jasno nakazali, kateri podatki so obvezni in katere mora vnesti le pod določenim pogojem. Posamezna vnosna polja so namreč vsebinsko odvisna od predhodnih polj. Takšna polja v določenih primerih sploh niso potrebna, zato jih aplikacija dinamično prikazuje in skriva glede na izbrano možnost v predhodnem polju. V nekaterih primerih lahko aplikacija takšna polja glede na uporabnikove vnose v predhodnih poljih tudi sama predizpolni. V naši aplikaciji je takšen primer izračun števila mesecev med dvema datumoma (slika 4.4), ki ju izbere uporabnik. S tem aplikacija v veliki meri prepreči napačne vnose, saj že sama poskrbi, da je vnesena vrednost v pravilnem formatu in še vsebinsko ustrezna.

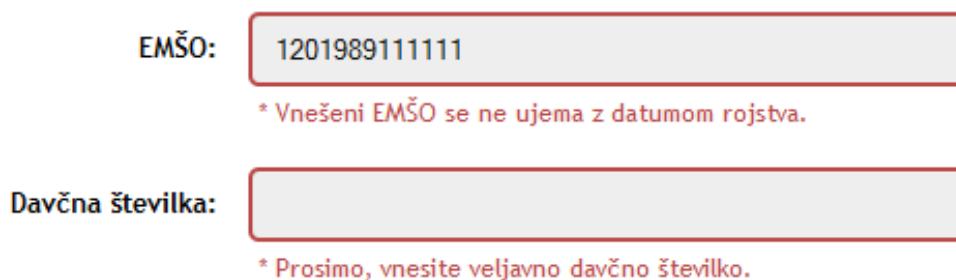
Predvideno trajanje študijske prakse: od  do  / (  mesecev )

Slika 4.4: Ko uporabnik vnese pravilne datume trajanja prakse ali izmenjave, aplikacija sama izračuna število mesecev med obema datumoma.

Uporabniški vmesnik je načrtovan tako, da je možno posamezne konstante (vrednosti spustnih seznamov, imena vnosnih polj ipd.) enostavno popraviti v posebni konfiguracijski datoteki. Za urejanje konfiguracijske datoteke, namestitvev aplikacije in uporabo administracije smo pripravili tudi kratko dokumentacijo, v kateri so jasno podana vsa potrebna navodila.

- **Javljanje napak, diagnoza in njihovo reševanje**

Kljub temu, da že vnaprej skušamo preprečiti napačne vnose, so le-ti še vedno možni, zato jih je treba odpraviti še pred oddajo obrazca. Pri spletnih prijavah za to poskrbi validacija, s pomočjo katere pravočasno preverimo pravilnost vnesenih podatkov. V našem primeru aplikacija uporabnika opozori na vsak napačen vnos in mu prepreči zaključek prijave, dokler niso vsa polja pravilno izpolnjena. Kot prikazuje slika 4.5, se napačno izpolnjena vnosna polja ob poskusu oddaje prijave obarvajo rdeče, pod vnosnim poljem pa se izpiše tudi kratko besedilo z obvestilom o napaki.



EMŠO:

\* Vnešeni EMŠO se ne ujema z datumom rojstva.

Davčna številka:

\* Prosimo, vnesite veljavno davčno številko.

Slika 4.5: Če uporabnik pozabi izpolniti obvezno vnosno polje ali je vnos v napačnem formatu, potem aplikacija vnosno polje obarva rdeče in izpiše obvestilo o napaki.

- **Grupiranje**

Pri obsežnejših obrazcih je vnosna polja smiselno razdeliti v več skupin oz. podobrazcev, ki vsebujejo vsebinsko sorodna polja, medsebojno pa se čim bolj izključujejo. S tem namreč precej izboljšamo preglednost vmesnika. V našem primeru je prijavní obrazec razdeljen na tri vsebinsko sorodne sklope: osebni podatki, podatki o študiju (slika 4.6) in podatki, ki se navezujejo na izmenjavo ali prakso.

**Podatki o študiju**

Naziv članice: UL Fakulteta za računalništvo in informatiko

Zadnji zaključen letnik: izberite letnik študija

Stopnja študija: izberite stopnjo študija

Stopnja študija v času izmenjave: izberite stopnjo študija \* samo ob prijavi na študijsko izmenjavo

Naziv študijskega programa: izberite študijski program

Področna koda: 48 Computing

Znanje tujih jezikov:

Mentor na fakulteti:

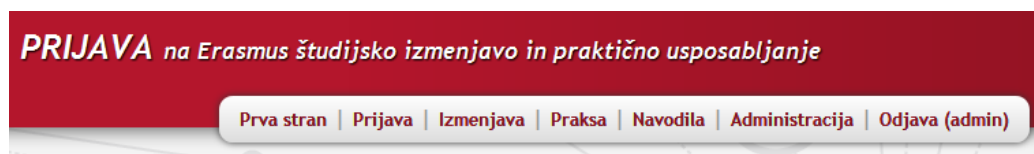
Prijava na Erasmus študijsko izmenjavo (SMS) → Prijava na Erasmus praktično usposabljanje (SMP) →

Slika 4.6: Primer grupiranja vsebinsko sorodnih vnosnih polj, ki so namenjena pridobivanju podatkov o študiju.

- **Raje prepoznaj, kot si zapomni, uporabnikov nadzor**

Z uporabo spustnih seznamov namesto tekstovnih polj, kjer je to mogoče, precej razbremenimo uporabnikov spomin. Osnovne in najbolj pomembne funkcionalnosti aplikacije morajo biti uporabniku vedno na voljo in čim bolj jasno prikazane. Zato je najbolje, če so prikazane na vsaki podstrani ter na čim bolj vidnem mestu. Med te funkcionalnosti v našem primeru spadata npr. odjava iz aplikacije in povezava

na začetno stran. V ta namen smo v glavo aplikacije, ki je prikazana na vsaki podstrani, dodali manjšo navigacijo s pomembnejšimi povezavami. Vsebina navigacije se prilagaja vrsti uporabnika aplikacije, saj administratorji potrebujejo več hitrih povezav kot študenti (slika 4.7).



Slika 4.7: Administrator lahko do vseh pomembnih strani hitro dostopa preko enostavne navigacije v glavi aplikacije.

Premiki med posameznimi vnosnimi polji so možni tudi z uporabo tipke ENTER, kar uporabniku omogoča hitro prehajanje skozi celotno prijavo.

Ko se uporabnik že nahaja na drugem delu obrazca, se lahko s pomočjo gumba vedno vrne nazaj na prvi del in popravi že izpolnjena vnosna polja. Preden se prijava dokončno zaključi in se podatki shranijo v podatkovno bazo, aplikacija na zaslon ponovno izpiše vse vnesene podatke. S tem uporabniku omogočimo, da še enkrat preveri pravilnost vseh podatkov o prijavi, po potrebi pa se lahko vrne tudi nazaj na obrazce in popravi morebitne napake.

- **Estetika in minimalistično načrtovanje**

Spletna aplikacija bo namenjena le študentom Fakultete za računalništvo in informatiko Univerze v Ljubljani. S tega vidika je bilo najbolj smiselno, da se za osnovno grafično podobo uporabijo podobni grafični elementi, kot so na ostalih fakultetnih straneh ali aplikacijah. Tako smo pri izdelavi naše aplikacije uporabili podobne barve, teksture in logotipe, kot so v novem študijskem informacijskem sistemu FRI. Barve ozadij vsebinskega dela aplikacije so nevpadljive (sivi in beli odtenki),

medtem ko so barve nekaterih pomembnih elementov (gumbov) poudarjene z bolj živimi barvami. Večjo preglednost vmesnika smo dosegli tudi z ustrezno poravnano vseh gradnikov aplikacije, ki je prikazana na sliki 4.8. Vsa vnosna polja v prijavnih obrazcih so vertikalno poravnana, poleg tega pa je vsako polje v svoji vrstici. S takšno postavitevijo smo dosegli večjo preglednost prijavnega obrazca. Hkrati smo zagotovili tudi dovolj "belega prostora", zaradi katerega je uporabnikov pogled vedno usmerjen na bistvene dele obrazca (vnosna polja).

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko

**PRIJAVA** na Erasmus študijsko izmenjavo in praktično usposabljanje

Prva stran | **Prijava** | Izmenjava | Praksa | Navodila | Administracija | Odjava (admin)

Ali ste že opravili Erasmus izmenjavo:

Če da, kje in kdaj:

Naziv izvajalca študijske prakse:  #

Naslov izvajalca študijske prakse:  #

Država izvajalca študijske prakse:  #

Ime odgovorne (kontaktne) osebe izvajalca študijske prakse:  #

Predvideno trajanje študijske prakse: od  do  / (  mesecev)

# Če delodajalca oz. prakse še nimate izbrane, pustite prazno oz. napišite le okvirne informacije. Podatke boste kasneje lahko tudi spremenili.

FRI | Prijava na Erasmus študijsko izmenjavo in praktično usposabljanje

Slika 4.8: Del prijavnega obrazca, ki je namenjen vnosu podatkov o praktičnem usposabljanju; vnosna polja so vertikalno poravnana, barve vmesnika so nevpadljive, medtem ko so gumbi poudarjeni z bolj živimi barvami.

## 4.3 Opis funkcionalnosti aplikacije

Uporabniški vmesnik omogoča interakcijo med uporabnikom in aplikacijo. Za pravilno delovanje aplikacije pa je treba sprogramirati še vse ostale funkcionalnosti, ki tečejo v ozadju in uporabniku neposredno niso vidne. Seveda pa načrtovanja uporabniškega vmesnika in razvoja funkcionalnosti ne moremo obravnavati kot dve povsem ločeni enoti. Za učinkovito delovanje aplikacije mora razvoj obeh delov potekati vzporedno.

Razvoj najpomembnejših funkcionalnosti aplikacije je potekal inkrementalno. Glavni inkrementi so bili:

- implementacija prijave v aplikacijo,
- implementacija delovanja obrazcev za prijavo na prakso in izmenjavo,
- varno shranjevanje podatkov o prijavah v podatkovno bazo,
- implementacija funkcionalnosti za kreiranje dokumentov PDF, ki so primerni za tisk,
- implementacija administracije.

Za razvoj glavnih funkcionalnosti spletne aplikacije smo uporabili tehnologije PHP, JavaScript, AJAX in podatkovno bazo MySQL.

### 4.3.1 Prijava v aplikacijo

Prijavna stran se mora uporabniku prikazati pred začetkom uporabe aplikacije, zato smo kreirali datoteko `index.html`, ki uporabnika najprej preusmeri na to stran. Avtentifikacija in avtorizacija uporabnikov potekata preko storitve Active Directory, ki že predstavlja nekakšen imenik z vsemi uporabniškimi imeni in gesli. Za povezavo smo uporabili protokol LDAP (angl. *Lightweight Directory Access Protocol*), ki omogoča enostavno iskanje podatkov po drevesni strukturi imenika (izsek kode 4.1). Ob prijavi aplikacija tako najprej vzpostavi povezavo s strežnikom LDAP, nato pa s prijavo v Active Directory

preveri pravilnost vnesenega uporabniškega imena in gesla. Če prijava uspe, potem aplikacija s pomočjo ustrezno grajenega iskalnega drevesa preveri še, če se prijavni podatki nahajajo tudi v ustreznem poddirektoriju imenika (v našem primeru je to direktorij z vsemi študenti Fakultete za računalništvo in informatiko). Če se uporabniško ime in geslo nahajata v ustreznem poddirektoriju, potem je študent uspešno prijavljen v aplikacijo. Implementacija prijave v aplikacijo je bila tako relativno enostavna, saj se študenti prijavijo kar z obstoječimi Active Directory uporabniškimi imeni in gesli. Registracija v aplikacijo zato ni potrebna.

```
1 //povezava s streznikom LDAP
2 $ldapconn = ldap_connect($adServer, $ldapport)
3 or die("Could not connect to LDAP server.");
4
5 //povezava s sistemom Active Directory
6 if($username!="" && $password!=""){
7     $ldapbind = ldap_bind($ldapconn, $username, $password);
8 }
9 //v primeru uspesne povezave poiscemo uporabnika
10 if ($ldapbind) {
11     $result = ldap_search($ldapconn,$ldaptree,$filter,$attr)
12     or die ("Error in search query: ".ldap_error($ldapconn));
13
14     $data = ldap_get_entries($ldapconn, $result);
15     ...//tukaj se izvede ostala koda
16 }
17 //na koncu zapremo povezavo
18 ldap_close($ldapconn);
```

Izsek kode 4.1: Del kode (ogrodje) za povezavo aplikacije s storitvijo Active Directory preko protokola LDAP.

Prijavni obrazec za prijavo v aplikacijo, ki je prikazan na sliki 4.9, je sicer skupen za študente in administratorje, vendar avtentifikacija administratorjev ne poteka preko storitve Active Directory. Za prijavo v administracijo smo zato kreirali posebno uporabniško ime in geslo, ki ga je po potrebi možno tudi enostavno spremeniti.

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko

**PRIJAVA** na Erasmus študijsko izmenjavo in praktično usposabljanje

### Prijava v aplikacijo

Prijavite se s svojim Active Directory uporabniškim imenom in geslom.

Uporabniško ime:

Geslo:

FRI | Prijava na Erasmus študijsko izmenjavo in praktično usposabljanje

Slika 4.9: Obrazec za prijavo v spletno aplikacijo.

Na podlagi vnesenega uporabniškega imena in gesla aplikacija preveri, če gre za navadnega uporabnika, v našem primeru študenta, ali administratorja. Študent je po prijavi preusmerjen na začetno stran aplikacije (slika 4.10), kjer lahko vidi morebitne že obstoječe prijave na študijsko prakso oz. izmenjavo, ali pa ob kliku na gumb začne z novo prijavo. Administratorja aplikacija takoj po prijavi preusmeri v administracijo. Zaradi varnosti mora aplikacija ves čas preverjati vrsto uporabnika, saj imajo administratorji dostop do vseh strani, medtem ko študentom dostop do administracije ni dovoljen.



Slika 4.10: Takoj po prijavi v aplikacijo lahko študent vidi svoje dosedanje prijave (tudi dokument PDF), če seveda le-te obstajajo.

### 4.3.2 Implementacija prijavnih obrazcev

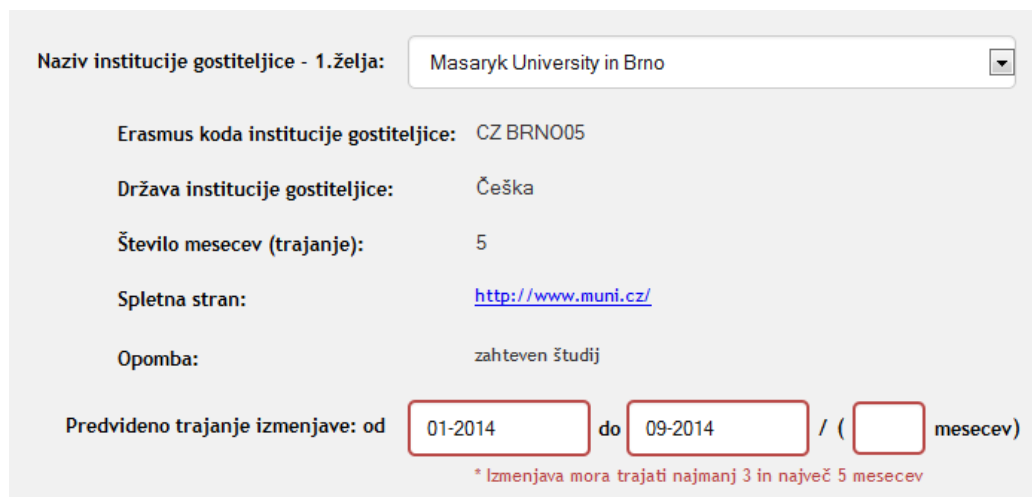
Prijavne obrazce smo najprej izdelali in oblikovali s pomočjo HTML in CSS. Po izdelavi obrazcev smo morali implementirati še ustrezno validacijo vnosov in vse ostale pomembne funkcionalnosti, kot sta dinamičen prikaz podatkov o institucijah za izmenjavo in samodejna predizpolnitev obrazcev.

#### Validacija obrazcev

Pri spletnih obrazcih je ključnega pomena dobra validacija, s katero preverimo pravilnost in ustreznost vnesenih podatkov. Pravilnost podatkov lahko preverimo na strani odjemalca (npr. s pomočjo jezika JavaScript) ali pa ob pošiljanju podatkov na strežnik (npr. z uporabo PHP). Validacija ob pošiljanju podatkov na strežnik je sicer nekoliko počasnejša, saj se mora spletna stran pred preverjanjem napak najprej ponovno naložiti. Prednost takšne validacije pa je veliko večja zanesljivost, kar je pri naši aplikaciji, ktere jedro predstavljajo prijavni obrazci, zelo pomembno. JavaScript lahko

uporabnik v brskalniku onemogoči, v tem primeru pa se tudi sama validacija na strani odjemalca ne izvede. Pri validaciji, ki se izvaja na strani strežnika, uporabniške nastavitve v brskalniku ne igrajo nobene vloge.

V sklopu naše aplikacije smo uporabili kombiniran način validacije. Vsa vnosna polja so namreč validirana s pomočjo jezika PHP na strežniški strani, pri nekaterih vnosnih poljih pa se izvede še validacija na strani odjemalca s pomočjo jezika JavaScript (slika 4.11). Validacija je tako zelo zanesljiva ne glede na uporabniške nastavitve v brskalniku. Poleg tega je uporabnik pri ključnih vnosnih poljih že med izpolnjevanjem obrazca obveščen o napakah, kar precej izboljša uporabniško izkušnjo.



Naziv institucije gostiteljice - 1. želja: Masaryk University in Brno

Erasmus koda institucije gostiteljice: CZ BRN005

Država institucije gostiteljice: Češka

Število mesecev (trajanje): 5

Spletna stran: <http://www.muni.cz/>

Opomba: zahteven študij

Predvideno trajanje izmenjave: od 01-2014 do 09-2014 / ( ) mesecev

\* Izmenjava mora trajati najmanj 3 in največ 5 mesecev

Slika 4.11: S pomočjo sprotne validacije uporabnika že med izpolnjevanjem obrazca opozorimo na napačen vnos časa trajanja študijske izmenjave. Dolžina trajanja izmenjave je odvisna od izbrane institucije.

Naša aplikacija z validacijo preverja, če so izpolnjena vsa obvezna vnosna polja in če so vsi vneseni podatki v pravilnem formatu. Pri podatkih, ki imajo že vnaprej točno določen format (davčna številka, vpisna številka, EMŠO), lahko z validacijo preverimo tako sestavo (številke, pravilnost prvih sedmih številke EMŠO) kot tudi dolžino vnosa (izsek kode 4.2).

```
1 //preverjanje pravilnosti emsa
2 $('input[name="emso"]').change(function(){
3     var rojstvo = $('input[name="datum_rojstva"]').val();
4     var emso = $('input[name="emso"]').val();
5     var arr = rojstvo.split('.');
6     var dan=arr[0]; var mesec=arr[1]; var leto=arr[2];
7
8     var datum=dan + mesec +leto.substr(1,3);
9     var emsodat=emso.substr(0, 7);
10
11     if(emsodat!= datum){
12         $("#emso-note").text("* Vneseni EMSO se neujema
13             z vnesenim datumom rojstva.");
14         $('input[name="emso"]').addClass("error");
15     }else if(emso.length!=13 || !$.isNumeric(emso)){
16         $("#emso-note").text("* Vneseni EMSO ni v pravilnem
17             formatu.");
18         $('input[name="emso"]').addClass("error");
19     }else{
20         $("#emso-note").text("");
21         $('input[name="emso"]').removeClass("error");
22     }
23 });
```

Izsek kode 4.2: Preverjanje pravilnosti EMŠO s pomočjo jezika JavaScript.

Pri ostalih vnosih (npr. imena) aplikacija preveri samo, če vsebujejo ustrezne znake in če polje ni prazno.

Z dobro validacijo tako poskrbimo, da uporabnik ne more zaključiti oz. oddati obrazca, dokler niso pravilno vneseni vsi zahtevani podatki.

## Dinamičen prikaz podatkov o institucijah

Seznam institucij, ki jih lahko študent izbere za izmenjavo, je shranjen v podatkovni bazi. V prijavnem obrazcu jih spletna aplikacija prebere iz baze in izpiše v obliki spustnega seznama. Ker je izbor institucije odvisen tudi od stopnje študija študenta v času izmenjave, je bilo treba poskrbeti za izpis le ustreznih institucij. S tem namenom se izbrana stopnja študija (v času izmenjave) shrani v sejno spremenljivko. Pri obrazcu za izbor institucij aplikacija prebere vrednost te spremenljivke in iz podatkovne baze izbere le tiste institucije, ki ustrezajo izbrani stopnji.

Ko študent izbere posamezno institucijo s spustnega seznama, mu aplikacija izpiše tudi nekatere podrobnosti o izboru, ki so prikazane na sliki 4.12.



The screenshot shows a web form with the following details:

Naziv institucije gostiteljice - 1.želja:	Masaryk University in Brno
Erasmus koda institucije gostiteljice:	CZ BRN005
Država institucije gostiteljice:	Češka
Število mesecev (trajanje):	5
Spletna stran:	<a href="http://www.muni.cz/">http://www.muni.cz/</a>
Opomba:	zahteven študij

Slika 4.12: Ob izbiri institucije za izmenjavo se pod spustnim seznamom izpiše nekaj uporabnih podatkov o izbrani instituciji.

Za implementacijo prikaza podrobnosti smo uporabili AJAX, zato se podatki o institucijah izpišejo dinamično (brez ponovnega nalaganja strani). To je omogočeno s pomočjo klica funkcije (izsek kode 4.3), ki ob vsaki spremembi izbora v spustnem seznamu preveri izbrano institucijo. Funkcija nato kreira ustrezen objekt `XMLHttpRequest`, pošlje poizvedbo na strežnik in na koncu doda parameter z ID-številko izbrane institucije v URL-naslov. V posebni datoteki PHP, ki jo pokliče JavaScript, se zgodi poizvedba v podatkovni bazi. Kot rezultat funkcija vrne tabelo v obliki HTML, v kateri so podatki

o izbrani instituciji.

```
1 function showData(str){
2   if (str==""){
3     document.getElementById("podrobno").innerHTML="";
4     return;
5   }
6   if (window.XMLHttpRequest)
7     xmlhttp=new XMLHttpRequest();
8   else
9     xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
10  xmlhttp.onreadystatechange=function(){
11    if (xmlhttp.readyState==4 && xmlhttp.status==200)
12      {
13        document.getElementById("podrobno").innerHTML=
14        xmlhttp.responseText;
15      }
16  }
17  xmlhttp.open("GET","getinfo.php?q="+str,true);
18  xmlhttp.send();
19  }
20 }
```

Izsek kode 4.3: AJAX funkcija, ki omogoča dinamičen izpis podrobnosti o izbrani instituciji.

### Samodejna predizpolnitev obrazcev

Delovanje spletne aplikacije je zasnovano tako, da se v določenih primerih del obrazcev samodejno predizpolni.

Ko se študent prvič prijavi v aplikacijo in še nima nobene odprte prijave na izmenjavo ali prakso, aplikacija najprej preveri, če obstaja ustrezen dokument XML z nekaterimi osebnimi podatki o prijavljenem študentu. Ta

dokument mora biti ustrezno strukturiran in poimenovan. Struktura dokumenta je povsem enostavna, pomembno je le, da so posamezni elementi (angl. *Tag*) ustrezno poimenovani, kot prikazuje tudi izsek kode 4.4. Dokument bo priskrbela spletna storitev, ki bo vse potrebne podatke pridobila iz študijskega informacijskega sistema FRI. Če ustrezen XML obstaja, potem ga aplikacija razčleni in s pridobljenimi podatki predizpolni ustrezna vnosna polja. Omenjena spletna storitev sicer še ni implementirana, ker je študijski informacijski sistem še v razvoju, a se implementacija pričakuje v bližnji prihodnosti.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <student>
3     <ime>Janez</ime>
4     <priimek>Novak</priimek>
5     <spol>M</spol>
6     <vpisna_stevilka>12345678</vpisna_stevilka>
7     <datum_rojstva>16/01/1989</datum_rojstva>
8     <kraj_rojstva>Celje</kraj_rojstva>
9     <drzavljanstvo>Slovensko</drzavljanstvo>
10    <naslov_sp>Slovenska cesta 123</naslov_sp>
11    <posta_sp>1000 Ljubljana</posta_sp>
12    <naslov_ob>Dunajska cesta 123</naslov_ob>
13    <posta_ob>1000 Ljubljana</posta_ob>
14    <telefon>031 123 123</telefon>
15    <naziv_stud_prog>BVS RI</naziv_stud_prog>
16 </student>
```

Izsek kode 4.4: Struktura dokumenta XML, ki ga lahko aplikacija razčleni in s pridobljenimi podatki predizpolni vnosna polja.

Če je študent že izpolnil prijavo na izmenjavo ali prakso, potem aplikacija namesto iz dokumenta XML podatke prebere iz podatkovne baze. V tem primeru se skupna vnosna polja (za prijavo na izmenjavo in prakso) samo-

dejno predizpolnijo s podatki, ki so shranjeni v podatkovni bazi od prejšnje prijave.

Vsa predizpolnjena vnosna polja lahko študent po potrebi tudi popravi. Če aplikacija ne dobi ustreznega dokumenta XML ter podatkov o študentu še ni v podatkovni bazi, so ob prvi prijavi seveda vsa vnosna polja prazna in jih mora študent samostojno izpolniti.

### 4.3.3 Shranjevanje podatkov v podatkovno bazo

Ko študent v prijavnici obrazec uspešno vnese vse potrebne podatke in zaključi prijavo, mora aplikacija vse podatke varno shraniti v podatkovno bazo. Le tako lahko te podatke kasneje ponovno uporabimo in jih urejamo.

Naša aplikacija vse podatke o prijavah shranjuje v podatkovno bazo MySQL, saj ima programski jezik PHP že vgrajene vse osnovne funkcije za delo s tovrstnimi bazami. Za shranjevanje podatkov o prijavah sta v bazi namenjeni dve tabeli, in sicer ena za podatke o študentih prijavljenih na izmenjavo in druga za študente, ki so prijavljeni na študijsko prakso. Za dve povsem ločeni tabeli smo se odločili, ker je bil ta način najprimernejši za izvajanje poizvedb v kodi, hkrati pa podatkovna baza nikoli ne bo vsebovala toliko podatkov, da bi to kakorkoli vplivalo na hitrost izvajanja poizvedb in delovanja aplikacije. Ker v tabelah ni veliko polj, ki bi bila pogosto prazna, jih tudi ni smiselno deliti na manjše enote.

### Varnost podatkov v podatkovni bazi

Kot smo že omenili, je velika prednost programskega jezika PHP enostavnost, s katero lahko komunicira s podatkovnimi bazami, predvsem z MySQL. Seveda pa pri tem obstaja nevarnost za potencialno velike varnostne težave, na katere lahko naletimo. Najpogostejša nevarnost, s katero se soočamo pri interakciji s podatkovno bazo, je t.i. vrivanje stavkov SQL (angl. *SQL injection*) [14]. Do takšnega napada lahko pride, če uporabnik izkoristi varnostno napako za izvajanje svojih poizvedb SQL v naši podatkovni bazi. S tem lahko uporabnik pridobi vse podatke iz podatkovne baze in jih (zlonamerno)

izkoristi v svojo korist, nenazadnje pa lahko tudi izbriše celotno podatkovno bazo. Na srečo je ta varnostna ranljivost razmeroma enostavna in jo lahko s pravilnim programiranjem dokaj hitro odpravimo. Zelo pomembno je, da iz podatkov, ki jih vnašamo v podatkovno bazo, odstranimo vse nedovoljene znake (predvsem navednice). Že s tem lahko v veliki meri preprečimo izvajanje neželenih poizvedb SQL v naši podatkovni bazi.

Enostavnejši, a ne vedno učinkovit način za preprečevanje tovrstnih napadov je uporaba funkcije `mysqli_real_escape_string()` [7]. Funkcija iz uporabniških vnosov odstrani posebne znake, ki bi lahko vplivali na izvajanje poizvedb SQL.

Najbolj zanesljivo pa se vrivanju SQL izognemo tako, da namesto običajnih stavkov oz. poizvedb SQL, v katerih so uporabniški vnosi vstavljeni neposredno, uporabimo t.i. pripravljene stavke (angl. *prepared statements*) [13]. Ta način je sicer nekoliko bolj zapleten za izvedbo, a precej bolj zanesljiv.

Pri uporabi pripravljenih stavkov [13] spletna aplikacija pripravi predlogo stavka SQL in ga pošlje sistemu za upravljanje s podatkovnimi bazami. Nekatere vrednosti, ki jih imenujemo parametri, ostanejo nedefinirane in jih nadomestimo z znakom `"?"`. Sistem stavek razčleni in ga prevede ter shrani, ne da bi ga izvedel. Ker je stavek shranjen, mu lahko aplikacija kasneje poda manjkajoče parametre, sistem pa ga izvrši in vrne rezultate. Ker morajo biti iz parametrov, ki se prenesejo preko drugega protokola, pravilno odstranjeni vsi posebni znaki (angl. *Escape Characters*), so ti stavki odporni na vrivanje SQL.

Pri razvoju naše aplikacije smo za delo s podatkovno bazo uporabili modul `mysqli`, ki je izboljšana verzija modula `mysql` in za razliko od starejše verzije omogoča oba načina preprečevanja vrivanja stavkov SQL. Tehniko pripravljenih stavkov smo uporabili tam, kjer je ranljivost za tovrstne napade največja. V naši aplikaciji so to primeri, kjer pridejo uporabniki v interakcijo s podatkovno bazo; npr. pri shranjevanju podatkov iz prijavnega obrazca v podatkovno bazo (izsek kode 4.5). V ostalih primerih, kjer uporabnik nima neposrednega vpliva na izvrševanje poizvedb, smo uporabili

funkcijo `mysqli_real_escape_string()`.

```
1  if ($stmt = $connection->prepare("INSERT INTO
2  izmenjava (ime, priimek) VALUES (?,?)")) {
3
4      // Dodamo parametre tipa String v SQL stavek.
5      $stmt->bind_param("ss", $izmenjava->ime,
6      $izmenjava->priimek);
7
8      // Izvedemo SQL stavek.
9      $stmt->execute();
10
11     // Zapremo pripravljene stavek.
12     $stmt->close();
13 }
```

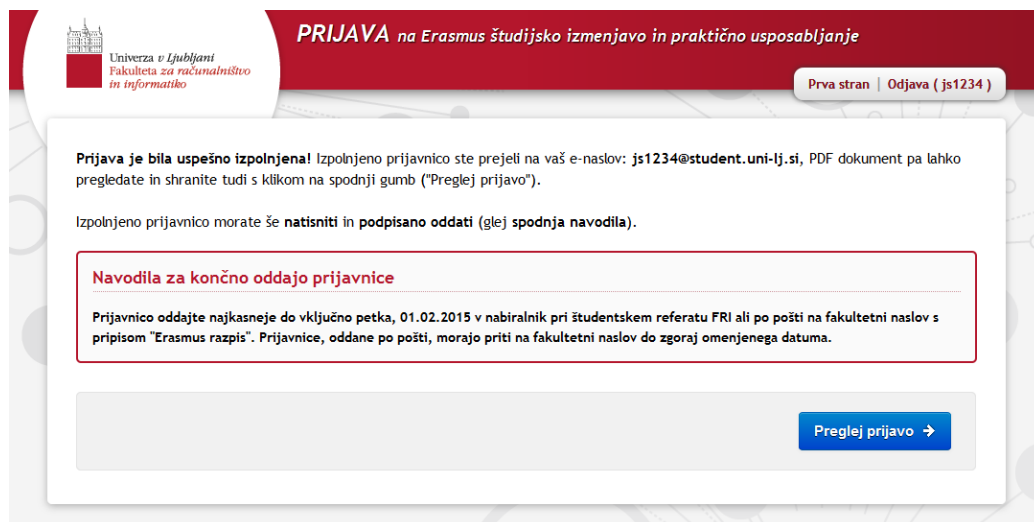
Izsek kode 4.5: Primer uporabe metode pripravljenih stavkov pri shranjevanju podatkov v podatkovno bazo.

#### 4.3.4 Kreiranje dokumentov PDF

Ker morajo študenti izpolnjene prijavnne obrazce na koncu še natisniti in podpisane oddati, aplikacija omogoča tudi generiranje primerne dokumenta za tisk. V ta namen smo uporabili odprtokodni razred TCPDF [15], ki omogoča kreiranje dokumentov PDF v programskem jeziku PHP. Sicer obstaja več podobnih razredov za kreiranje tovrstnih dokumentov, vendar so se pri ostalih pojavljale težave pri podpori kodiranju UTF-8. To postane problem pri šumnikih, ki so bili v kreiranih dokumentih PDF napačno prikazani oz. jih sploh ni bilo. Pri uporabi razreda TCPDF teh problemov ni, saj že v osnovi podpira kodiranje UTF-8. Poleg tega razred vsebuje tudi nekaj lastnih pisav, ki omogočajo prikazovanje šumnikov.

Kot predlogi za kreiranje dokumentov PDF smo pripravili dve ustrezno

oblikovani datoteki HTML; eno za prijavo na izmenjavo in eno za študijsko prakso. V datotekah so prikazani vsi zahtevani podatki o prijavi posameznega študenta, ki jih aplikacija prebere iz podatkovne baze. TCPDF na podlagi predlog kreira dokument PDF, ki ima enako strukturo in izgled kot vnaprej pripravljene datoteki HTML. Ker mora aplikacija kreirati dokumente PDF na večih mestih, smo za to naredili kar posebno funkcijo. Študent lahko dokument PDF shrani oz. natisne takoj po zaključeni prijavi (slika 4.13) in ob prijavi v aplikacijo (če je že prijavljen na kakšno izmenjavo ali prakso). Poleg tega aplikacija po vsaki zaključeni prijavi pošlje študentu tudi elektronsko pošto z ustreznim dokumentom PDF v priložnosti.



Slika 4.13: Po uspešnem zaključku prijave na prakso ali izmenjavo aplikacija uporabnika preusmeri na stran z nadaljnjimi navodili, kjer je možno tudi pregledati in shraniti dokument PDF s podatki o prijavi.

Za pošiljanje elektronske pošte smo uporabili knjižnico PHPMailer [11], ki je prosto dostopna na spletu. Sicer ima programski jezik PHP že vgrajeno tudi lastno funkcijo za pošiljanje e-pošte, ki pa ni najbolj zanesljiva. Tudi pošiljanje priložnosti je z omenjeno funkcijo težje realizirati, kot pa z uporabo knjižnice PHPMailer.

### 4.3.5 Administracija

Administracija določenim uporabnikom (administratorjem) omogoča podroben pregled vseh podatkov o prijavljenih študentih, izvoz teh podatkov v Excelovo datoteko ter urejanje baze podatkov. Vse to poteka preko enostavnega uporabniškega vmesnika.

#### Pregled in izvoz podatkov o prijavah

Aplikacija omogoča prikaz in izvoz podatkov o prijavljenih študentih v treh različnih oblikah:

- podatki o prijavah na študijsko izmenjavo,
- podatki o prijavah na študijsko prakso,
- podatki o prijavah na študijsko izmenjavo in prakso skupaj.

Izpis podatkov o prijavah

Izmenjava (SMS)
Praksa (SMP)
Vse prijave
Institucije

➔ **Vse prijave**

#	Študent	Vpisna št.	Študij	Tren. status	Stopnja	1. želja	2. želja	3. želja	Odhod
1	Janez Novak	12345678	BUN-RI	2.	prva	FH Joanneum Gesellschaft mbH	-	-	12-2013
2	Anja Novak	12345678	BVS-RI	1.	prva	FH Joanneum Gesellschaft mbH	Management Center Innsbruck	-	12-2013
3	Simon Hočevar	11111111	BUN-RI	2.	druga				12-2013
4	Peter Novak	12345678	BVS-RI	2.	prva				12-2013

Izvozi v Excel (.xlsx) ➔

Slika 4.14: Administracija omogoča pregled in izvoz podatkov v treh različnih formatih; vse prijave skupaj, prijave na izmenjavo in prijave na prakso.

Sam pregled podatkov je implementiran s pomočjo treh gumbov. Aplikacija ob kliku na posamezen gumb prebere ustrezne podatke iz podatkovne baze in jih v obliki pregledne tabele izpiše na zaslon (slika 4.14).

Izvoz podatkov v Excel smo implementirali s pomočjo dodatne knjižnice PHPExcel [9], ki omogoča branje in pisanje datotek Excel preko programskega jezika PHP (izsek kode 4.6). Ker se strukture posameznih datotek Excel nekoliko razlikujejo, smo pripravili tri predloge (datoteke PHP), na podlagi katerih razred PHPExcel generira ustrezno strukturirane in oblikovane tabele Excel s podatki.

```
1 $objPHPExcel = new PHPExcel();
2 $result = $connection->query($izmenjava);
3
4 $rowCount=2;
5 while ($row = mysqli_fetch_array($result)) {
6     $column = 'A';
7     for($j=0; $j<count($izmenjava_titles);$j++){
8         if(isset($row[$j])){
9             $objPHPExcel->getActiveSheet()->setCellValue(
10                 $column.$rowCount,strip_tags($row[$j]));
11             $column++;
12         }
13     }
14     $rowCount++;
15 }
```

Izsek kode 4.6: Del kode za zapisovanje podatkov iz podatkovne baze v datoteko Excel s pomočjo jezika PHP.

Administracija omogoča tudi tabelarni prikaz podatkov o vseh institucijah, ki so v podatkovni bazi in na katere se lahko študenti prijavijo v okviru študijske izmenjave.

## Upravljanje z bazo podatkov

Pomemben del administracije, ki je prikazan na sliki 4.15, je namenjen upravljanju z bistvenimi deli podatkovne baze.

Brisanju podatkov o prijavljenih študentih sta namenjena dva gumba, eden za izmenjave in eden za prakse. Ob gumbih je še polje za vnos varnostnega gesla. Uporabnik mora namreč pred brisanjem podatkov iz podatkovne baze najprej vpisati pravilno varnostno geslo, sicer se brisanje ne izvede. S tem smo zagotovili, da nihče ne more po pomoti izbrisati vseh podatkov o prijavih.

Upravljanje z bazo podatkov

**→ Posodobitev seznama institucij**

izberite ustrezno .csv datoteko s seznamom institucij in posodobite bazo.

Datoteka ni izbrana.

**→ Brisanje prijav**

S klikom na spodnje gumba boste izbrisali vse prijave iz baze podatkov!

**GESLO:**

**→ Posodobitev navodil**

Vpišite navodila za oddajo prijavnice.

Prijavnico oddajte najkasneje do vključno petka, 01.02.2015 v nabiralnik pri študentskem referatu FRI ali po pošti na fakultetni naslov s pripisom "Erasmus razpis". Prijavnice, oddane po pošti, morajo priti na fakultetni naslov do zgoraj omenjenega datuma.

---

Vnesite staro geslo:

Vnesite novo geslo:

Slika 4.15: Del administracije, ki je namenjen upravljanju z bazo podatkov.

Seznam institucij, na katere se lahko študenti prijavijo v okviru študijske izmenjave, se pogosto spreminja. Zato smo implementirali tudi možnost enostavnega posodabljanja seznama v podatkovni bazi. Za posodobitev institucij je treba pripraviti le ustrezno formatirano datoteko CSV, ki vsebuje vse po-

trebne podatke o institucijah (slika 4.16). Aplikacija izbrano datoteko odpre in razčleni s pomočjo PHP funkcije `fgetcsv()`. Funkciji je treba kot argument podati tudi ločilo, s katerim so ločeni podatki v datoteki CSV. Pri tem smo imeli kar nekaj težav, saj je vrsta ločila v tovrstnih datotekah odvisna od lokalnih nastavitvev računalnika oz. Excela, ki se najpogosteje uporablja za kreiranje datotek CSV. Excel s slovenskimi nastavitvami datoteko CSV naredi tako, da vse podatke loči s podpičjem, medtem ko Excel z angleškimi nastavitvami podatke loči z vejico. Da smo zagotovili pravilno delovanje v različnih okoljih, smo ločilo podali v obliki konstante (privzeto nastavljeno ločilo je `”;`), ki jo je možno enostavno popraviti v konfiguracijski datoteki. Aplikacija glede na podano ločilo razčleni datoteko CSV, podatki pa se nato shranijo v podatkovno bazo, kjer je posebna tabela z seznamom institucij.

```
INSTITUCIJA;ERASMUS KODA;DRŽAVA;ŠTEVILO ŠTUDENTOV;ŠTEVILO MESECEV;STOPNJA ŠTUDIJA;OPOMBA;SPLETNA STRAN
Management Center Innsbruck;A INNSBRU08;Avstrija;2;6;prva;;http://www.mci.edu
FH Joanneum Gesellschaft mbH;A GRAZ09;Avstrija;2;6;prva, druga;;http://www.fh-joanneum.at
Graz University ;A GRAZ02;Avstrija;2;10;druga;obvezno znanje nemškega jezika;http://portal.tugraz.at
University - Varna;BG VARNA02;Bolgarija;3;5;prva, druga;;http://www.tu-varna.acad.bg
```

Slika 4.16: Primer pravilno strukturirane datoteke CSV za posodobitev podatkov o institucijah v podatkovni bazi.

Poleg seznama institucij se občasno spreminjajo tudi navodila za končno oddajo prijavnice. V administraciji je vnos in posodabljanje teh navodil namenjeno dodatno vnosno polje.

Administrator lahko po potrebi spremeni tudi svoje geslo za prijavo v aplikacijo. To lahko stori v spodnjem delu administracije, in sicer z vpisom starega in novega gesla ter potrditvijo. Geslo, ki je shranjeno v podatkovni bazi, je zaradi varnosti kriptirano. Za kriptiranje smo uporabili funkcijo `crypt()`, ki pa ob običajni uporabi ne omogoča dovolj dobre zaščite gesla. Funkcijo smo zato dodatno izboljšali z algoritmom `Blowfish`, ki za kodiranje uporabi dodaten kodirni ključ s spremenljivo vsebino in dolžino. S tem smo odpravili nevarnost razkritja gesla naključnemu uporabniku ob dostopu oz. morebitnem napadu na podatkovno bazo.

## 4.4 Testiranje aplikacije

Pred začetkom resne uporabe aplikacije smo morali še čim bolj natančno testirati njeno delovanje. Testiranje glavnih funkcionalnosti je sicer potekalo že med samim razvojem aplikacije, zato večjih težav s tem na koncu ni bilo. Pred končnim testiranjem smo morali aplikacijo namestiti še v delovno okolje, v našem primeru je bil to strežnik Linux. Sprotno in končno testiranje je potekalo z več vidikov:

- **Delovanje aplikacije v različnih brskalnikih**

Ker gre za spletno aplikacijo, ki se uporabniku prikazuje v spletnih brskalnikih, je bilo treba testirati tudi delovanje in prikaz aplikacije v različnih brskalnikih.

Pri testiranju spletnih strani ali aplikacij v brskalnikih je v ospredju predvsem izgled uporabniškega vmesnika, ki je ponavadi v vsakem brskalniku prikazan nekoliko drugače. Do takšnih razlik prihaja, ker nekateri brskalniki, predvsem pa starejše različice, ne podpirajo nekaterih novejših pravil CSS oz. jih različno prikazujejo. Pri oblikovanju naše aplikacije smo uporabljali tudi novejši CSS3, zato je bilo nekaj več težav s prikazom posameznih oblikovnih elementov v nekaterih brskalnikih. Večje napake smo odpravili, v določenih malenkostih (npr. prikaz barvnih gradientov) pa se prikaz v različnih brskalnikih še vedno nekoliko razlikuje, vendar to nikakor ne vpliva na samo kakovost delovanja aplikacije.

Poleg podpore pravil CSS se brskalniki nekoliko razlikujejo tudi v izvajanju JavaScript kode. Pri nekaterih JavaScript funkcijah smo zato morali dodati še nekaj dodatne kode, ki omogoča pravilno delovanje aplikacije v različnih brskalnikih.

- **Delovanje aplikacije na strežniku**

Razvoj aplikacije je potekal na lokalnem strežniku v operacijskem sistemu Windows. Z namestitvijo aplikacije na Linux strežnik smo pre-

verili, če aplikacija brezhibno deluje tudi na ciljnim strežniku. Poleg tega je nekatere funkcionalnosti, kot sta pošiljanje elektronske pošte in prijava v aplikacijo preko storitve Active Directory, veliko lažje testirati na ciljnim strežniku. Ker smo za razvoj aplikacije v lokalnem okolju uporabljali strežnik Apache, ki deluje tako v Windows kot Unix okolju, aplikacija seveda brezhibno deluje tudi na Linux strežniku.

Nekaj težav smo na začetku imeli le s pošiljanjem elektronske pošte, ki je na strežniku Linux potekalo zelo počasi. Za pravilno delovanje smo morali ustrezno konfigurirati poštni strežnik `Sendmail`, ki na strežnikih Linux skrbi za pošiljanje e-pošte. Konfiguracija je obsegala predvsem nastavitve ustreznih naslovov IP in strežniških imen v nekaterih datotekah na strežniku.

Dodatno smo morali na strežnik namestiti tudi podporo LDAP, ki omogoča povezavo s storitvijo Active Directory in enostavno iskanje podatkov po drevesni strukturi imenika.

- **Uporabnost in varnost aplikacije**

Naša spletna aplikacija je namenjena predvsem pridobivanju in shranjevanju nekaterih podatkov o študentih. Ob testiranju smo zato še enkrat preverili, če aplikacija od uporabnika pridobi vse podatke, ki so potrebni za uspešno prijavo na izmenjavo ali prakso. Prav tako smo preverili, če so vsi podatki varno shranjeni v podatkovni bazi in naključen uporabnik ne more dostopati do njih.

Z vidika varnosti smo morali zagotoviti tudi, da imajo dostop do aplikacije samo prijavljeni uporabniki in da lahko do administracije res dostopajo le osebe, ki imajo za to dovoljenje (administratorji). V ta namen smo prijavo v aplikacijo že v testni fazi povezali s storitvijo Active Directory, kjer se nahajajo tudi vsa uporabniška imena in gesla.

Za varen prenos podatkov med odjemalcem in strežnikom smo na strežniku omogočili tudi varno povezavo preko protokola SSL (angl. *Secure Sockets Layer*), ki zakodira in s tem zaščiti podatke med samim procesom

---

pošiljanja. Na strežnik smo dodali tudi ustrezno datoteko `.htaccess`, ki se uporablja za enostavno upravljanje različnih strežniških nastavitv. S pomočjo omenjene datoteke oz. njene vsebine smo zagotovili, da uporabnik aplikacijo vedno uporablja preko varne povezave.



## Poglavje 5

### Sklepne ugotovitve

Spletna aplikacija za prijavo na Erasmus študijsko izmenjavo in prakso bo študentom vsekakor zelo poenostavila sam postopek prijave. Še nekoliko večjo vlogo bo aplikacija imela pri samem pregledu in urejanju podatkov o prijavah. Ker se podatki po prijavi posameznega študenta shranijo v podatkovno bazo, jih ni treba ročno vnašati v računalnik, kot je bilo to potrebno pri prijavah na papirju. Poleg tega aplikacija omogoča tudi enostaven izvoz podatkov v Excel, kar je pomembno predvsem za nadaljnjo obdelavo prijav.

Z izdelavo spletne aplikacije nam je uspelo zadostiti vsem ciljem, ki smo si jih zadali na začetku. Uspešno smo implementirali prijavne obrazce, s pomočjo katerih aplikacija pridobi vse potrebne podatke o prijavljenih študentih. Administracija vključuje vse potrebne funkcionalnosti, ki omogočajo enostaven pregled in izvoz podatkov ter upravljanje s podatkovno bazo. Končni izdelek diplomskega dela je popolnoma delujoča spletna aplikacija, ki je že nameščena v delovno okolje in pripravljena na takojšnjo uporabo.

Kljub temu, da je aplikacija že primerna za uporabo, pa so v prihodnosti še možne nekatere izboljšave in nadgradnje. Le-te so možne predvsem na področju povezave aplikacije z drugimi sistemi preko spletnih storitev. Številne podatke o študentih bo že pred izpolnjevanjem obrazcev mogoče pridobiti kar iz študijskega informacijskega sistema FRI. Aplikacija namreč že ima implementirano funkcionalnost za branje datotek XML s podatki o študentih,

ki jih bo pridobila preko spletne storitve. Ko bo spletna storitev na voljo, študentom ob prijavi ne bo treba vnašati vseh podatkov, ampak le tiste, ki niso na voljo v študijskem informacijskem sistemu. Poleg osnovnih podatkov bi aplikacija (preko spletnih storitev) lahko pridobila tudi nekatere podatke, ki jih sicer ne pridobivamo s prijavnimi obrazci, ampak so vseeno potrebni za nadaljnjo obdelavo prijav. To so predvsem ocene študentov po posameznih letnikih in stopnjah študija, na podlagi katerih bo možno izračunati tudi povprečja ocen, ki so glavni kriterij fakultete pri izboru študentov za izmenjavo. Dokler pa omenjene spletne storitve ne bodo na voljo, bo treba te podatke ročno vnašati v računalnik.

# Literatura

- [1] D. Cederholm, CSS3 For Web Designers, New York: A Book Apart, 2010
- [2] R. Larsen, Beginning HTML and CSS, Indianapolis, IN : J. Wiley & Sons, 2013.
- [3] M. Štrancar, S. Klemen, PHP in MySQL na spletnem strežniku Apache, Ljubljana : Pasadena, 2002.
- [4] T. Negrino, S. Dori, JavaScript and Ajax : for the web, Berkeley : Peachpit Press, 2007.
- [5] (2013) Aplikacije 'odjemalec - strežnik'. Dostopno na:  
[http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/PODATKOVNE\\_BAZE/aplikacije\\_odjemalec\\_strenik.html](http://colos1.fri.uni-lj.si/ERI/RACUNALNISTVO/PODATKOVNE_BAZE/aplikacije_odjemalec_strenik.html)
- [6] (2013) jQuery - write less, do more. Dostopno na:  
<http://jquery.com/>
- [7] (2013) mysqli\_real\_escape\_string. Dostopno na:  
<http://php.net/manual/en/mysqli.real-escape-string.php>
- [8] (2013) PHP: Hypertext Preprocessor. Dostopno na:  
<http://php.net/>
- [9] (2013) PHPExcel knjižnica. Dostopno na:  
<http://phpexcel.codeplex.com/>

- 
- [10] (2013) PHP MySQL Introduction. Dostopno na:  
[http://www.w3schools.com/php/php\\_mysql\\_intro.asp](http://www.w3schools.com/php/php_mysql_intro.asp)
- [11] (2013) PHPMailer knjižnica. Dostopno na:  
<https://github.com/Synchro/PHPMailer>
- [12] (2013) phpMyAdmin. Dostopno na:  
[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)
- [13] (2013) Prepared statements and stored procedures. Dostopno na:  
<http://php.net/manual/en/pdo.prepared-statements.php>
- [14] (2013) SQL Injection. Dostopno na:  
<http://php.net/manual/en/security.database.sql-injection.php>
- [15] (2013) TCPDF knjižnica. Dostopno na:  
<http://www.tcpdf.org/>
- [16] (2013) XAMPP. Dostopno na:  
<http://www.apachefriends.org/en/xampp.html>
- [17] (2013) XML Tutorial. Dostopno na:  
<http://www.w3schools.com/xml/>
- [18] (1995) 10 Usability Heuristics for User Interface Design (Jakob Nielsen). Dostopno na: <http://www.nngroup.com/articles/ten-usability-heuristics/>
- [19] (2013) Aktivne spletne strani. Dostopno na:  
<http://drenovec.tsckr.si/splet/aktivne.htm>

# Slike

3.1	Pregleden uporabniški vmesnik orodja PhpMyAdmin. . . . .	8
4.1	Zelo poenostavljen prikaz delovanja arhitekture odjemalec-strežnik [19]. . . . .	14
4.2	Gumba za brisanje podatkov o prijavah sta drugačne barve od ostalih gumbov, kar uporabnika opozori na večjo previdnost pri brisanju podatkov. . . . .	16
4.3	Ob kliku na vnosno polje za izbiro datuma se uporabniku prikaže koledar. . . . .	17
4.4	Ko uporabnik vnese pravilne datume trajanja prakse ali izmenjave, aplikacija sama izračuna število mesecev med obema datumoma. . . . .	17
4.5	Če uporabnik pozabi izpolniti obvezno vnosno polje ali je vnos v napačnem formatu, potem aplikacija vnosno polje obarva rdeče in izpiše obvestilo o napaki. . . . .	18
4.6	Primer grupiranja vsebinsko sorodnih vnosnih polj, ki so namenjena pridobivanju podatkov o študiju. . . . .	19
4.7	Administrator lahko do vseh pomembnih strani hitro dostopa preko enostavne navigacije v glavi aplikacije. . . . .	20
4.8	Del prijavnega obrazca, ki je namenjen vnosu podatkov o praktičnem usposabljanju; vnosna polja so vertikalno poravnana, barve vmesnika so nevpadljive, medtem ko so gumbi poudarjeni z bolj živimi barvami. . . . .	21

---

4.9	Obrazec za prijavo v spletno aplikacijo. . . . .	24
4.10	Takoj po prijavi v aplikacijo lahko študent vidi svoje dosedanje prijave (tudi dokument PDF), če seveda le-te obstajajo. . . . .	25
4.11	S pomočjo sprotne validacije uporabnika že med izpolnjevanjem obrazca opozorimo na napačen vnos časa trajanja študijske izmenjave. Dolžina trajanja izmenjave je odvisna od izbrane institucije. . . . .	26
4.12	Ob izbiri institucije za izmenjavo se pod spustnim seznamom izpiše nekaj uporabnih podatkov o izbrani instituciji. . . . .	28
4.13	Po uspešnem zaključku prijave na prakso ali izmenjavo aplikacija uporabnika preusmeri na stran z nadaljnjimi navodili, kjer je možno tudi pregledati in shraniti dokument PDF s podatki o prijavi. . . . .	34
4.14	Administracija omogoča pregled in izvoz podatkov v treh različnih formatih; vse prijave skupaj, prijave na izmenjavo in prijave na prakso. . . . .	35
4.15	Del administracije, ki je namenjen upravljanju z bazo podatkov. . . . .	37
4.16	Primer pravilno strukturirane datoteke CSV za posodobitev podatkov o institucijah v podatkovni bazi. . . . .	38