

Web aplikacija za praćenje uspjeha studenata na određenom predmetu

Glavaš, Lucija

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:481398>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Stručni studij

**WEB APLIKACIJA ZA PRAĆENJE USPJEHA
STUDENATA NA ODREĐENOM PREDMETU**

Završni rad

Lucija Glavaš

Osijek, 2018.

Sadržaj

1. UVOD	3
1.1. Zadatak i struktura rada	3
2. KORIŠTENE TEHNOLOGIJE	4
2.1. HTML	4
2.2. CSS	4
2.3. Bootstrap	5
2.4. PHP	5
2.5. MySQL	6
2.6. JavaScript	6
3. BAZA PODATAKA	6
3.1. Utvrđivanje i analiza zahtjeva	7
3.2. Konceptualno oblikovanje	9
3.3. ER dijagram	10
3.4. Fizičko oblikovanje	12
3.5. Unos i brisanje podataka	13
4. IZRAČUN ZAVRŠNE OCJENE	15
4.1. Odabir podataka iz baze	15
4.2. Bodovi ostvareni na kontrolnim zadaćama	17
4.3. Bodovi ostvareni na laboratorijskim vježbama	18
4.4. Bodovi ostvareni u domaćim zadaćama	19
4.5. Dolasci	19
4.6. UTS bodovi i završna ocjena	20
5. KORISNIČKO SUČELJE	22
6. ZAKLJUČAK	26
LITERATURA	27
SAŽETAK	28
WEB APPLICATION FOR STUDENT COURSE RECORDS	29
Abstract	29
ŽIVOTOPIS	30
PRILOZI	31

1. UVOD

Cilj ovog završnog rada dizajnirati je i razviti web aplikaciju koja će omogućiti praćenje uspjeha studenata na predmetima (kolegijima). Aplikacija mora omogućiti unos podataka o studentima, kolegijima, kriterijima ocjenjivanja i uspjesima svih studenata te prikaz tih podataka na korisničkom sučelju. Sučelje mora biti pregledno i pružati informacije relevantne za korisnika. Za izradu ove web aplikacije korištene su tehnologije: HTML5, CSS3, Bootstrap, PHP, MySQL i JavaScript. U radu su detaljno opisani pojedini dijelovi aplikacije, njihova funkcionalnost, specifične funkcije korištene za njihovu izradu i načini na koje funkcije manipuliraju podacima. U prvom će dijelu rada biti opisana svaka tehnologija i način na koji je primijenjena u aplikaciji. U drugom dijelu rada biti će prikazana konkretna izvedba aplikacije, tj. dijelovi koda koji čine njenu funkcionalnost.

1.1. Zadatak i struktura rada

Važan dio ovog rada podešavanje je baze podataka i njena optimizacija na konkretnom web serveru. Za rad je korišten Linux server. Baza podataka kreirana je na MySQL-u. Pristup bazi podataka i njenoj manipulaciji ima samo administrator sustava.

2. KORIŠTENE TEHNOLOGIJE

Za izradu ove web aplikacije korištene su sljedeće tehnologije: HTML5, CSS3, Bootstrap, PHP, MySQL i JavaScript. U sljedećih ću nekoliko poglavlja opisati navedene tehnologije i njihovu primjenu u radu.

2.1. HTML

HTML (engl. *Hyper Text Markup Language*) prezentacijski je jezik koji se koristi za izradu web stranica. HTML-om oblikuje se struktura stranice, njen sadržaj i konstrukcija te se stvaraju hiperveze. Hiperveze, tj. poveznice unutar HTML dokumenta, čine hijerarhijsku strukturu web stranice i na taj način određuju koji korisnik doživljava njen sadržaj i na koji način. HTML vrlo je jednostavan za korištenje i lako se uči pa je njegova prihvaćenost iznimno velika. HTML je danas najrašireniji jezik za izradu web stranica i web aplikacija. Temeljna je zadaća HTML-a dati pregledniku uputu o prikazu sadržaja stranice, a da pri tome stranica izgleda jednako na svim računalima, web preglednicima i operacijskim sustavima.

Osnovni su element svake web stranice napisane na HTML-u oznake (engl. *tags*). Oznake opisuju kako će se nešto prikazivati na stranici. Prilikom izrade ove aplikacije bilo je važno koristiti odgovarajuće oznake kako bi konstrukcija stranice bila smisljena i kako bi se svi elementi pravilno prikazivali. Za konstrukciju većih dijelova stranice korištena je oznaka *div* (engl. *division*). Pomoću te oznake pojedini dijelovi koda lako se grupiraju u skupine i na taj način tvore izgled stranice. Još jedna bitna oznaka korištena u ovome radu je *form*. Ta se oznaka koristi za stvaranje obrazaca koji korisniku omogućuju unos podataka. S obzirom na to da je jedna od glavnih funkcionalnosti ove aplikacije unos podataka u bazu, bilo je vrlo važno napraviti smislene i praktične forme koje će olakšati rad s aplikacijom.

2.2. CSS

CSS (engl. *Cascading Style Sheets*) jezik je kojim se vrši stiliziranje dokumenta napisanog u HTML-u. U početku se prilikom izrade web stranica koristio samo HTML, ali razvojem

tehnologije i Interneta brzo se stvorila potreba za jezikom koji će riješiti problem stilske ograničenosti i omogućiti manipulaciju prikaza elemenata stranice. CSS omogućio je prikaz web stranica na uređajima različitih zaslona, što je jako važno u današnje vrijeme kada su web stranice dostupne na računalima, prijenosnim računalima, mobilnim uređajima i pločastim uređajima. Naredbama CSS-a elementi web stranice prilagođavaju se zaslonima i prikazuju na način što ugodniji ljudskom oku.

Prilikom izrade ove aplikacije bilo je važno omogućiti njen prikaz na svim uređajima. Naime, svrha ove aplikacije pružanje je detaljnog uvida u uspjeh studenta na određenim kolegijima. To znači da student u bilo kojem trenutku i s bilo kojeg uređaja mora imati uvid u svoje bodove, izostanke i ostale kriterije koji utječu na njegovu ocjenu. Tu je također bilo važno voditi računa o preglednosti. Pomoću CSS-a stvorena je stilski jednostavna, ali vrlo pregledna web aplikacija modernog dizajna, jednostavna za korištenje.

2.3. Bootstrap

Bootstrap je okvir za dizajniranje web stranica i web aplikacija. Sadrži predefinirane predloške napisane u HTML-u i CSS-u. Predlošci sadrže komponente korisničkog sučelja kao što su tipografski elementi, gumbi, navigacijske trake itd. Predlošci se vrlo jednostavno implementiraju na stranicu i tako olakšavaju i ubrzavaju njen razvoj. Jedna od glavnih funkcionalnosti Bootstrap-a, i razlog zbog kojega je korišten u izradi ove aplikacije, pružanje je sustava rešetki.

Sustav rešetki omogućuje skaliranje elemenata stranice na uređajima različitih dimenzija. Svojim je svojstvima ubrzao izradu aplikacije jer nije bilo potrebno samostalno kreirati mrežu; korištenjem klasa CSS-a kreirana je mreža prilagodljiva različitim zaslonima.

2.4. PHP

PHP (engl. *PHP: Hypertext Preprocessor*) programski je jezik namijenjen programiranju dinamičkih web stranica. Koristi se za izvršavanje skripti na serverskoj strani. To znači da izvrši neku radnju na serveru, a korisnik ima dojam da se radnja izvršila u njegovom

pregledniku. Web stranica koju korisnik vidi u pregledniku rezultat je rada PHP-a na serveru.

Za izradu ove web aplikacije korišten je PHP kako bi se omogućilo dodavanje, mijenjanje i prikazivanje podataka upisanih u bazu. Podatke u bazu upisuju administratori, a korisnici ih mogu samo pregledati. Podaci se odnose na osnovne informacije o studentima (ime, prezime, JMBAG), osnovne informacije o kolegijima (naziv kolegija, broj predavanja itd.) i kriterije ocjenjivanja pomoću kojih sustav računa ocjenu, tj. uspjeh studenta na kolegiju.

2.5. MySQL

MySQL (engl. *My Structured Query Language*) sustav je za upravljanje bazom podataka. MySQL baza optimizirana je kako bi bila brza i funkcionalna. Vrlo je stabilna i ima dobro definirane module te podržava PHP. MySQL baze idealne su za pohranu većeg broja podataka. Omogućavaju jednostavno dodjeljivanje prava pristupa i zbog toga je MySQL korišten pri izradi ove aplikacije.

2.6. JavaScript

JavaScript programski je jezik koji se izvršava u pregledniku na strani korisnika. Zajedno s HTML-om i CSS-om čini skup osnovnih tehnologija za razvoj web stranica. Omogućuje kreiranje interaktivnih web stranica. Ugrađen je u sve web preglednike tako da se može koristiti bez instalacije dodatka.

U ovoj je web aplikaciji JavaScript korišten za potrebe kreiranja navigacijske trake i skočnih prozora.

3. BAZA PODATAKA

Baza podataka (engl. *database*) organizirana je zbirka podataka pohranjena na sustavan način. Sadrži sheme, tablice, upite, poglede i druge objekte koji omogućuju upravljanje podacima. Baza omogućuje definiciju, kreiranje, unos, dohvaćanje i ažuriranje podataka koji su u nju upisani.

Svaka je baza podataka izgrađena pomoću tablica. Tablice se sastoje od atributa i entiteta. Entitet tablice jest pojam o kojem se skupljaju podaci. Entitet sadrži određene attribute koji ga opisuju. Baze podataka uglavnom se sastoje od više tablica koje su međusobno povezane. Veza se uspostavlja između dva ili više entiteta. Takve baze podataka koje se sastoje od skupa povezanih tablica (relacija) nazivaju se relacijske baze podataka.

Za potrebe ove web aplikacije kreirana je jedna takva baza. Svaka cjelina aplikacije ima svoju funkciju i za rad su joj potrebni različiti podaci pa je sukladno tome napravljeno nekoliko tablica koje su međusobno smisleno povezane. Ovakav relacijski model baze vrlo je popularan jer olakšava unos, pretragu i održavanje sustava. U sljedećih ću nekoliko poglavlja dati uvid u bazu podataka kreiranu za potrebe ovoga rada.

3.1. Utvrđivanje i analiza zahtjeva

Podaci koji se skupljaju u bazu odnose se na osnovne informacije o studentima, profesorima i kolegijima.

Za svakog studenta bitno je definirati njegov JMBAG, ime, prezime i status (redovni ili izvanredni). Studenti se upisuju na kolegije i istovremeno mogu biti upisani na više kolegija.

Za svakog je profesora bitno navesti njegovo ime i prezime (s titulom), matični broj znanstvenika, njegovu funkciju (docent, predavač, asistent itd.), katedru na kojoj se nalazi, kontakt (adresu e-pošte) i lozinku koju koristi za prijavu u sustav te ga je potrebno dodati na kolegije. Profesor može biti na više kolegija istovremeno.

Za kolegije je bitno upisati naziv, oznaku, studij na kojem se nalaze, semestar u kojem se polažu, broj ECTS bodova i kriterije ocjenjivanja. Kriteriji se odnose na ukupan broj predavanja, laboratorijskih vježbi, broj ostvarivih bodova itd.

Podatke o studentima, profesorima i kolegijima ispunjava super-administrator sustava.

Super-administrator osoba je (npr. zaposlenik referade) koja u sustav unosi studente, profesore, definira kolegije, dodjeljuje profesore na kolegije i na njih upisuje studente. Super-administrator ima najveća prava pristupa podacima. Ono što super-administrator unese u sustav niti jedan drugi korisnik ne može mijenjati ili brisati.

Administratora sustava može biti više. To su profesori koje je super-administrator dodao u sustav i dodijelio im neki kolegij. Administratori mogu voditi evidenciju o studentima za one kolegije na koje su dodijeljeni i za one studente koji su upisani na te kolegije, ali ne mogu mijenjati ostale podatke o kolegijima ili studentima.

Analizom specifikacija mogu se uočiti dolje navedeni entiteti, atributi i veze.

Entiteti su:

- *studenti*, *studenti_status*, *studenti_upisani_kol*,
- *profesori*, *profesori_titule*, *profesori_funkcije*, *profesori_dod_kol*,
- *kolegiji*, *studiji*, *kolegij_kriterij_pr*, *kolegij_kriterij_av*, *kolegij_kriterij_lv*, *kolegij_kriterij_dz*, *kolegij_kriterij_kz*, *kolegij_kriterij_ispit*, *evidencija_pr*, *evidencija_av*, *evidencija_lv*, *evidencija_dz*, *evidencija_kz* i *evidencija_ispit*.

Atributi za entitet:

- *studenti* su: id, jmbag, ime, prezime, status Stud_id;
- *studenti_status* su: id, naziv;
- *studenti_upisani_kol* su: id, student_id, kolegij_id;
- *profesori* su: id, maticni_br, titula_id, ime, prezime, funkcija_id, kontakt, lozinka;
- *profesori_titule* su: id, naziv;
- *profesori_funkcije* su: id, naziv;
- *profesori_dod_kol* su: id, profesor_id, kolegij_id;
- *kolegiji* su: id, oznaka, naziv, studij_id, semestar, ects;
- *studiji* su: id, naziv;
- *kolegij_kriterij_pr* su: id, kolegij_id, br_pr;
- *kolegij_kriterij_av* su: id, kolegij_id, br_av;
- *kolegij_kriterij_lv* su: id, kolegij_id, br_lv, max_bod_lv;
- *kolegij_kriterij_dz* su: id, kolegij_id, br_dz, max_bod_dz, min_bod_dz;
- *kolegij_kriterij_kz* su: id, kolegij_id, max_bod_kz1, max_bod_kz2;
- *kolegij_kriterij_ispit* su: id, kolegij_id, max_bod;
- *evidencija_pr* su: id, student_id, kolegij_id, pr, dolazak;
- *evidencija_av* su: id, student_id, kolegij_id, av, dolazak;
- *evidencija_lv* su: id, student_id, kolegij_id, bodovi, lv, dolazak;
- *evidencija_dz* su: id, student_id, kolegij_id, dz_bodovi;

- *evidencija_kz* su: id, student_id, kolegij_id, bodovi_kz1, bodovi_kz2;
- *evidencija_ispit* su: id, student_id, kolegij_id, bodovi.

Veze se nalaze između:

- *studenti* i *studenti_status*,
- *studenti_upisani_kol*, *studenti* i *kolegiji*,
- *profesori* i *profesori_titule*,
- *profesori* i *profesori_funkcije*,
- *profesori_dod_kol*, *profesori* i *kolegiji*,
- *kolegiji* i *studiji*,
- *kolegiji* i *profesori*,
- *kolegij_kriterij_pr* i *kolegiji*,
- *kolegij_kriterij_av* i *kolegiji*,
- *kolegij_kriterij_lv* i *kolegiji*,
- *kolegij_kriterij_dz* i *kolegiji*,
- *kolegij_kriterij_kz* i *kolegiji*,
- *kolegij_kriterij_ispit* i *kolegiji*,
- *evidencija_pr*, *studenti* i *kolegiji*,
- *evidencija_av*, *studenti* i *kolegiji*,
- *evidencija_lv*, *studenti* i *kolegiji*,
- *evidencija_dz*, *studenti* i *kolegiji*,
- *evidencija_kz*, *studenti* i *kolegiji*,
- *evidencija_ispit*, *studenti* i *kolegiji*.

3.2. Konceptualno oblikovanje

Na osnovu analize specifikacije dobiveni su podaci prikazani u *Tablici 3.2.1*. Tablica sadrži popis entiteta i veza između njih, odgovarajuće funkcionalnosti veza i pripadne atribute. Podvučeni atributi u *Tablici 3.2.1*. predstavljaju kandidate za ključ.

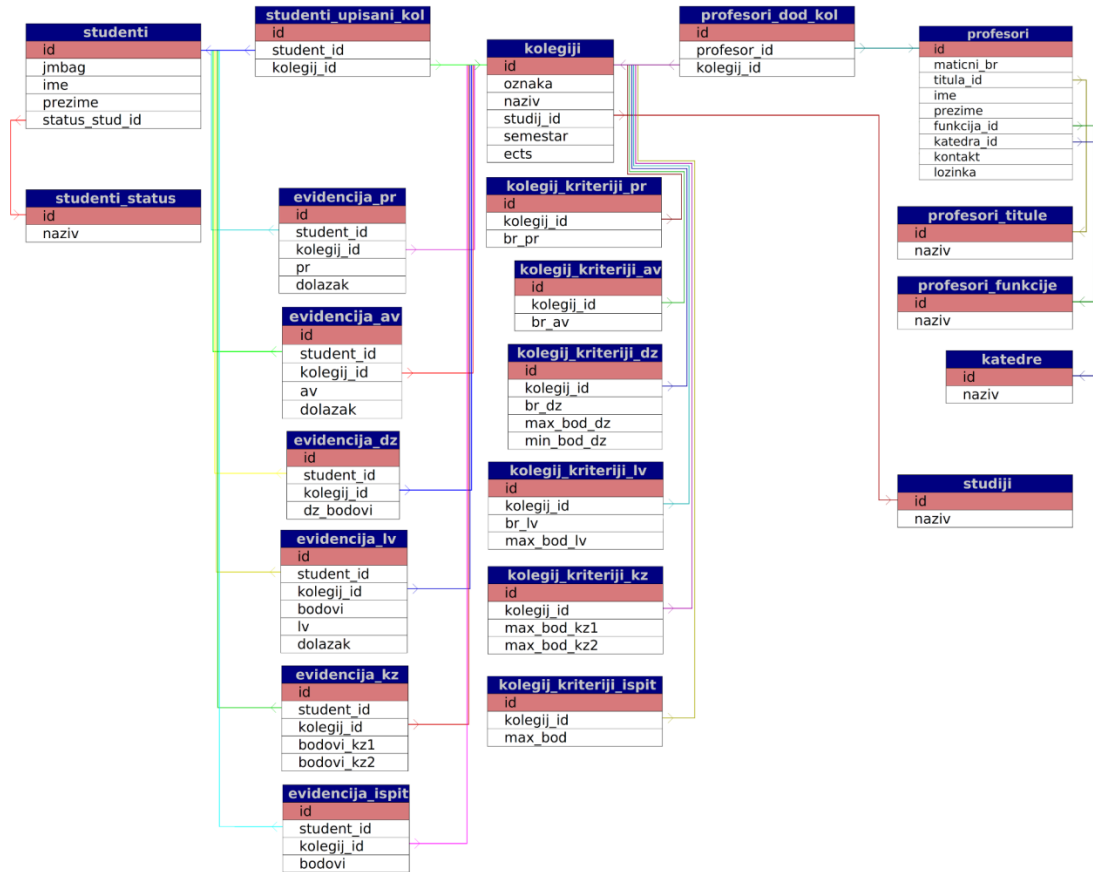
Tab. 3.2.1. Pripadni atributi entiteta/veza baze podataka

ENTITET/VEZA	ATRIBUT
studenti	<u>id</u> , jmbag, ime, prezime, status_stud_id

studenti_status	<u>id</u> , naziv
studenti_upisani_kol	<u>id</u> , student_id, kolegij_id
profesori	<u>id</u> , maticni_br, titula_id, ime, prezime, funkcija_id, kontakt, lozinka
profesori_titule	<u>id</u> , naziv
profesori_funkcije	<u>id</u> , naziv
profesori_dod_kol	<u>id</u> , profesor_id, kolegij_id
kolegiji	<u>id</u> , oznaka, naziv, studij_id, semestar, ects
studiji	<u>id</u> , naziv
kolegij_kriterij_pr	<u>id</u> , kolegij_id, br_pr
kolegij_kriterij_av	<u>id</u> , kolegij_id, br_av
kolegij_kriterij_lv	<u>id</u> , kolegij_id, br_lv, max_bod_lv
kolegij_kriterij_dz	<u>id</u> , kolegij_id, br_dz, max_bod_dz, min_bod_dz
kolegij_kriterij_kz	<u>id</u> , kolegij_id, max_bod_kz1, max_bod_kz2
kolegij_kriterij_ispit	<u>id</u> , kolegij_id, max_bod
evidencija_pr	<u>id</u> , student_id, kolegij_id, pr, dolazak
evidencija_av	<u>id</u> , student_id, kolegij_id, av, dolazak
evidencija_lv	<u>id</u> , student_id, kolegij_id, bodovi, lv, dolazak
evidencija_dz	<u>id</u> , student_id, kolegij_id, dz_bodovi
evidencija_kz	<u>id</u> , student_id, kolegij_id, bodovi_kz1, bodovi_kz2
evidencija_ispit	<u>id</u> , student_id, kolegij_id, bodovi

3.3. ER dijagram

Prevođenjem konceptualnog modela (*Tablica 3.2.1.*) u relacijski model dobivena je relacijska shema baze podataka. Baza je prikazana ER dijagramom na *Slici 3.3.1.*



Sl. 3.3.1. ER dijagram

3.4. Fizičko oblikovanje

Fizička shema baze podataka ostvarena je SQL naredbama za implementaciju. Stvaranje veza između tablica prikazano je u prikazu *Programskog koda 3.4.1.* i *3.4.2.* Zbog složenosti baze podataka i preglednosti ovog rada, fizičko oblikovanje baze podataka demonstrirano je samo na tablicama *kolegiji* i *kolegij_kriterij_pr*.

Programski kod 3.4.1. Fizičko oblikovanje baze podataka – tablice *kolegiji* i *kolegij_kriterij_pr*

```
CREATE TABLE IF NOT EXISTS `kolegiji` (  
  `id` int(11) NOT NULL,  
  `oznaka` varchar(10) COLLATE utf8_unicode_ci NOT NULL,  
  `naziv` varchar(50) COLLATE utf8_unicode_ci NOT NULL,  
  `studij_id` int(11) NOT NULL,  
  `semestar` int(11) NOT NULL,  
  `ects` decimal(3,1) NOT NULL,  
  `nositelj_id` int(11) NOT NULL,  
  `br_upisanih_stud` int(11) NOT NULL  
);  
  
ALTER TABLE `kolegiji`  
  ADD PRIMARY KEY (`id`), ADD UNIQUE KEY `oznaka` (`oznaka`), ADD KEY  
`kolegiji_studij_id_foreign` (`studij_id`), ADD KEY `kolegiji_nositelj_id_foreign`  
(`nositelj_id`);  
  
CREATE TABLE IF NOT EXISTS `kolegij_kriteriji_pr` (  
  `id` int(11) NOT NULL,  
  `kolegij_id` int(11) NOT NULL,  
  `br_pr` int(11) NOT NULL  
);  
  
ALTER TABLE `kolegij_kriteriji_pr`  
  ADD PRIMARY KEY (`id`), ADD KEY `kolegij_kriteriji_pr_kolegij_id_foreign`  
(`kolegij_id`);
```

Programski kod 3.4.2. Stvaranje veze između tablica *kolegiji* i *kolegij_kriterij_pr*

```
ALTER TABLE `kolegij_kriteriji_pr`  
ADD CONSTRAINT `kolegij_kriteriji_pr_kolegij_id_foreign` FOREIGN KEY  
(`kolegij_id`) REFERENCES `kolegiji` (`id`) ON DELETE CASCADE;
```

3.5. Unos i brisanje podataka

Unos i brisanje podataka u bazu može izvršavati samo super-administrator sustava preko korisničkog sučelja aplikacije. Unos i brisanje podataka prikazani su u prikazu *Programskog koda 3.5.1.* i *3.5.2.* Zbog složenosti baze podataka i preglednosti ovog rada fizičko oblikovanje baze podataka demonstrirano je samo na tablici *kolegiji*.

Programski kod 3.5.1. Unos podataka u tablicu *kolegiji*

```
<?php  
include('../././database/db-connect.php');  
  
$oznaka = $_POST['oznaka'];  
$naziv = $_POST['naziv'];  
$studij = $_POST['studij_id'];  
$semestar = $_POST['semestar'];  
$sects = $_POST['ects'];  
$nositelj = $_POST['nositelj_id'];  
  
$query = $conn->prepare("INSERT INTO kolegiji (oznaka, naziv, studij_id, semestar,  
ects, nositelj_id) VALUES (?, ?, ?, ?, ?, ?)");  
$query->bind_param('ssiidi', $oznaka, $naziv, $studij, $semestar, $sects, $nositelj);  
$query->execute();  
  
mysqli_close($conn);  
  
?>
```

Programski kod 3.5.2. Brisanje podataka iz tablice *kolegiji*

```
<?php
include('../database/db-connect.php');

$id = $_GET['id'];

$query = $conn->prepare("DELETE FROM kolegiji WHERE id = ?");
$query->bind_param('i', $id);
$query->execute();

mysqli_close($conn);

$host=$_SERVER["HTTP_HOST"];
$path=rtrim(dirname($_SERVER["PHP_SELF"]), "\\\");
header("Location: http://$host$path/");
exit;
?>
```

4. IZRAČUN ZAVRŠNE OCJENE

Kada super-administrator u sustav unese sve potrebne profesore, studente i kolegije, administratori mogu unositi evidenciju. Podatke unose za svakog studenta posebno, ovisno o tome na koji su kolegij dodijeljeni. Student svoj uspjeh na kolegiju može pregledati unosom svog JMBAG-a na početnoj stranici aplikacije.

Unosi se broj predavanja i auditornih vježbi na kojima je student bio prisutan, koliko je bodova ostvario na pojedinoj laboratorijskoj vježbi i je li bio na vježbi, koliko je bodova ostvario na domaćim zadaćama, koliko je bodova ostvario na kontrolnim zadaćama (kolokvijima) i koliko je bodova ostvario na ispitu. Bodovi s ispita unose se samo ako student nije ostvario dovoljan broj bodova na kolokvijima.

Kada se unesu svi potrebni podaci, administrator na svom sučelju ima tablicu s pregledom evidencije za svaki kolegij na koji je dodijeljen. U tablici se računa konačni prijedlog ocjene kolegija za svakoga studenta. Ocjena se računa prema *Okviru kriterija ocjenjivanja studenata FERIT-a za 2016. i 2017. godinu*. Kao pomoć pri sastavljanju funkcija i uvjeta za pojedinu ocjenu korištene su i tablice s evidencijom ocjena kolegija: Baze podataka (stručni studij) za 2015. godinu i Objektivno orijentirano programiranje (stručni studij) za 2016. godinu. Okvir za ocjenjivanje i tablice s evidencijom dodane su kao reference u poglavlje *Prilozi* na kraju ovog rada.

4.1. Odabir podataka iz baze

Za izračun ocjena svakoga studenta aplikacija mora pročitati podatke upisane u bazu, izračunati ocjenu i ispisati ju na zaslonu. Tablice koje se pri tome koriste su: *studenti* i *studenti_upisani_kol* kako bismo mogli provjeriti jesu li studenti upisani na kolegij za koji se računa uspjeh; *kolegij_kriteriji_pr*, *kolegij_kriteriji_av*, *kolegij_kriteriji_kz*, *kolegij_kriteriji_lv*, *kolegij_kriteriji_dz* i *kolegij_kriteriji_ispit* kako bismo dobili podatke o kriterijima bodovanju na kolegiju; *evidencija_pr*, *evidencija_av*, *evidencija_kz*, *evidencija_lv*, *evidencija_dz* i *evidencija_ispit* kako bismo dobili informaciju o ostvarenim bodovima. Za odabir tih podataka iz tablice korištena je *select* naredba prikazana u *Programskom kodu 4.1.1*.

Programski kod 4.1.1. SELECT naredba za dohvaćanje kriterija i evidencije bodova kolegija

```
"SELECT s.*, suk.student_id AS suk_stud,
k_kz.max_bod_kz1 AS kz1_max, k_kz.max_bod_kz2 AS kz2_max,
e_kz.bodovi_kz1 AS e_kz_bodovi_kz_1, e_kz.bodovi_kz2 AS e_kz_bodovi_kz_2,
k_lv.br_lv AS br_lv, k_lv.max_bod_lv AS max_bod_lv,
k_dz.br_dz AS br_dz, k_dz.max_bod_dz AS max_bod_dz,
k_pr.br_pr AS br_pr,
k_av.br_av AS br_av,
k_is.max_bod AS ispit_max_bod

FROM studenti s, studenti_upisani_kol suk, kolegij_kriteriji_kz k_kz, evidencija_kz
e_kz, kolegij_kriteriji_lv k_lv, kolegij_kriteriji_dz k_dz, kolegij_kriteriji_pr k_pr,
kolegij_kriteriji_av k_av, kolegij_kriteriji_ispit k_is

WHERE s.id = suk.student_id AND suk.kolegij_id = $kolegijID AND
k_kz.kolegij_id = $kolegijID AND e_kz.kolegij_id = $kolegijID AND
e_kz.student_id = suk.student_id AND k_lv.kolegij_id = $kolegijID AND
k_dz.kolegij_id = $kolegijID AND k_pr.kolegij_id = $kolegijID AND
k_av.kolegij_id = $kolegijID AND k_is.kolegij_id = $kolegijID

ORDER BY prezime ASC
"
```

4.2. Bodovi ostvareni na kontrolnim zadaćama

Kada su iz baze dohvaćeni svi potrebni podaci, mogu se provjeravati kriteriji. Prvi izračunat kriterij odnosi se na bodove ostvarene u kontrolnim zadaćama. Kako bi uspješno položio kolegij, student mora ostvariti između 20% i 50% bodova iz svake kontrolne zadaće pojedinačno i mora imati najmanje 50% bodova komulativno. Ako student nije zadovoljio uvjet na kontrolnim zadaćama, mora položiti pismeni ispit s najmanje 50% ostvarenih bodova.

Za izračun ukupnog broja bodova ostvarenih na kontrolnim zadaćama koristi se funkcija prikazana u *Programskom kodu 4.2.1.*

Programski kod 4.2.1. Funkcija koja računa ukupan broj bodova ostvaren na kontrolnim zadaćama i ispitu

```
$kz_ukupno = round(((( $row['e_kz_bodovi_kz_1'] / $row['kz1_max']) +  
($row['e_kz_bodovi_kz_2'] / $row['kz2_max'])) * 50, 2);  
  
$ispit_ukupno = round(($row_is['bodovi'] / $row['ispit_max_bod']) * 100, 2);
```

Dakle, ako je student iz prve kontrolne zadaće ostvario 39 od 40 bodova, a iz druge 26 od 30, iz toga je vidljivo da ima ukupno 92.08% bodova i da je zadovoljio uvjet.

Ako je, naime, student u prvoj kontrolnoj zadaći ostvario 8 od 40 bodova, a u drugoj 0 od 30, to znači da je student zadovoljio uvjet na prvoj kontrolnoj zadaći, ali nije na drugoj te da ima ukupno 10% bodova pa mora izaći na pismeni ispit.

4.3. Bodovi ostvareni na laboratorijskim vježbama

Kod laboratorijskih je vježbi potrebno izračunati broj bodova koje je student ostvario. Funkcija koja računa broj bodova ostvaren na laboratorijskim vježbama prikazana je u *Programskom kodu 4.3.1.*

Programski kod 4.3.1. Funkcija koja računa ukupan broj bodova ostvaren na laboratorijskim vježbama

```
$result_lv = mysqli_query($conn, "SELECT SUM(bodovi) AS lv_bodovi FROM
evidencija_lv e_lv WHERE e_lv.kolegij_id = $kolegijID AND e_lv.student_id = " .
$row['suk_stud']);

$row_lv = mysqli_fetch_array($result_lv);

$lv_ukupno = round(($row_lv['lv_bodovi'] / $row['max_bod_lv']) * 100, 2);
```

Iz tablice vidimo da aplikacija prvo u varijablu *result_lv* naredbom *SUM* sprema zbroj bodova ostvarenih na svakoj laboratorijskoj vježbi, a zatim u varijablu *lv_ukupno* sprema ukupan broj bodova ostvaren na vježbama. Varijabla *lv_bodovi* predstavlja zbroj bodova iz vježbi, a varijabla *max_bod_lv* predstavlja maksimalan ostvariv broj bodova. Rezultat se zatim množi sa sto kako bi se dobio postotak, a funkcija *round* zaokružuje rezultat na dvije decimale.

4.4. Bodovi ostvareni u domaćim zadaćama

Tijekom semestra studenti su u mogućnosti ostvariti dodatne bodove u vidu domaćih zadaća. Student nije obavezan predati niti jednu zadaću, ali one mu mogu popraviti konačnu ocjenu. Funkcija koja računa ukupan broj bodova ostvaren na domaćim zadaćama prikazana je u *Programskom kodu 4.4.1.*

Programski kod 4.4.1. Funkcija koja računa ukupan broj bodova ostvaren u domaćim zadaćama

```
$dz_ukupno = round(($row_dz['dz_bodovi'] / $row['max_bod_dz']) * 100, 2);
```

4.5. Dolasci

Pri računanju konačne ocjene važno je provjeriti je li student zadovoljio uvjete pohađanja nastave. Student na nastavi mora ostvariti najmanje 70% nazočnosti, a na laboratorijskim vježbama 100%, tj. mora biti prisutan na svim vježbama). Postotak pohađanja nastave računa se prema funkciji prikazanoj u *Programskom kodu 4.5.1.*

Programski kod 4.5.1. Funkcija koja računa postotak pohađanja nastave

```
$dolasci_pr = round(($row_pr['pr_dolasci'] / $row['br_pr']) * 100, 2);  
  
$dolasci_av = round(($row_av['av_dolasci'] / $row['br_av']) * 100, 2);  
  
$dolasci_lv = round(($row_lv_d['lv_dolasci'] / $row['br_lv']) * 100, 2);  
  
$dolasci_uk = round((0.4 * $dolasci_pr) + (0.2 * $dolasci_av) + (0.4 * $dolasci_lv), 2);
```

U tablici vidljivo je da aplikacija prvo računa postotak pohađanja predavanja, auditornih vježbi i laboratorijskih vježbi posebno, a zatim računa ukupan postotak pohađanja nastave.

4.6. UTS bodovi i završna ocjena

Kako bi se utvrdila završna ocjena, prvo se mora izračunati ukupan broj bodova, tj. UTS. U izračun UTS bodova ubraja se: broj bodova ostvaren na kontrolnim zadaćama (ili ispitu), broj bodova ostvaren na laboratorijskim vježbama, broj bodova ostvaren u domaćim zadaćama i broj bodova ostvaren pohađanjem nastave:

$$S_{uts} = \text{round}((0.4 * \$kz_ukupno) + (0.25 * \$lv_ukupno) + (0.25 * \$dz_ukupno) + (0.1 * \$dolasci_ukupno), 2);$$

Ako student nije položio ni kontrolne zadaće ni ispit, nije ostvario dovoljno bodova za prolazak kolegija.

Pri računanju završne ocjene važno je voditi računa o nekoliko kriterija:

- kako bi položio kolegij, student mora imati 50% ili više bodova iz kontrolnih zadaća (ili pismenog ispita) i mora imati 100% dolaznosti na laboratorijskim vježbama;
- ako student ima manje od 100% dolaznosti na laboratorijskim vježbama, mora ponavljati kolegij u sljedećoj akademskoj godini;
- ako je student prisustvovao svim vježbama, ali nije zadovoljio uvjet od 50% bodova na kolokvijima, treba izaći na pismeni ispit.

Ukupna se ocjena određuje prema bodovnom sustavu napisanom u *Tablici 4.6.1.*

Tab. 4.6.1. Utvrđivanje ukupnog broja bodova i konačne ocjene

UTS	Konačna ocjena
$90 \leq \text{UTS} \leq 100$	izvrstan (5)
$75 \leq \text{UTS} < 90$	vrlo dobar (4)
$60 \leq \text{UTS} < 75$	dobar (3)
$\text{UTS} < 60$	dovoljan (2)

Aplikacija koristi sve prethodno navedene parametre i pomoću *IF/ELSE* petlje računa konačnu ocjenu. Izračun je prikazan u *Programskom kodu 4.6.1.*

Programski kod 4.6.1. IF/ELSE petlja za izračun konačne ocjene

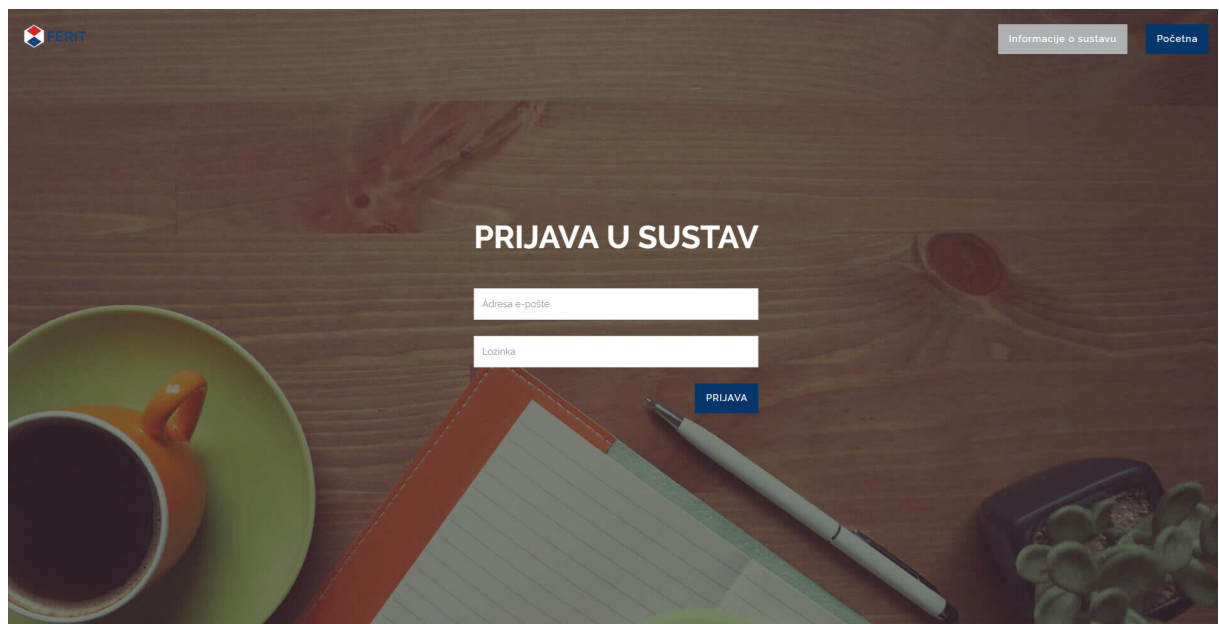
```
if (($kz_ukupno > 49 || $ispit_ukupno > 49) && $dolasci_lv == 100) {
    $komentar = "Položeno";
    if ($kz_ukupno > 49 && $ispit_ukupno < 50) {
        $ocjena_ukupno = (0.6 * $uts) + (0.4 * $kz_ukupno);
        if ($ocjena_ukupno >= 90 && $ocjena_ukupno <= 100) {
            $ocjena = "izvrstan (5)";
        }
        if ($ocjena_ukupno >= 75 && $ocjena_ukupno < 90) {
            $ocjena = "vrlo dobar (4)";
        }
        if ($ocjena_ukupno >= 60 && $ocjena_ukupno < 75) {
            $ocjena = "dobar (3)";
        }
        if ($ocjena_ukupno < 60) {
            $ocjena = "dovoljan (2)";
        }
    }
    if ($kz_ukupno < 50 && $ispit_ukupno > 49) {
        $ocjena_ukupno = (0.6 * $uts) + (0.4 * $ispit_ukupno);
        if ($ocjena_ukupno >= 90 && $ocjena_ukupno <= 100) {
            $ocjena = "izvrstan (5)";
        }
        if ($ocjena_ukupno >= 75 && $ocjena_ukupno < 90) {
            $ocjena = "vrlo dobar (4)";
        }
        if ($ocjena_ukupno >= 60 && $ocjena_ukupno < 75) {
            $ocjena = "dobar (3)";
        }
        if ($ocjena_ukupno < 60) {
            $ocjena = "dovoljan (2)";
        }
    }
}

if ($dolasci_lv < 100 && ($kz_ukupno < 50 && $ispit_ukupno < 50)) {
    $komentar = "LV neodrađene";
    $ocjena_ukupno = "-";
    $ocjena = "-";
}

if ($dolasci_lv == 100 && ($kz_ukupno < 50 && $ispit_ukupno < 50)) {
    $komentar = "Ponovno ispit";
    $ocjena_ukupno = "-";
    $ocjena = "-";
}
```

5. KORISNIČKO SUČELJE

Korisničko sučelje aplikacije izrađeno je na HTML-u i Bootstrap-u. Sastoji se od tri glavna dijela: super-administracija, administracija i dio za studente. Kako bi super-administrator i administratori pristupili sučelju, moraju izvršiti prijavu u sustav. Stranica za prijavu izgleda kako je prikazano na *Slici 5.1*.



Sl. 5.1. Prijava korisnika u sustav

Kako bi se prijavili u sustav, administratori u obrazac za prijavu unose svoju adresu e-pošte i lozinku. Prilikom dodavanja profesora u sustav, super-administrator dodaje im korisničku oznaku (adresu e-pošte) i lozinku. S tim se podacima profesori prijavljuju u sustav, a nakon prve prijave mogu promijeniti zadanu lozinku.

Super-administracijom sučelju pristup ima odgovorna osoba (npr. zaposlenik referade) koji u sustav, nakon izvršene prijave, unosi sve kolegije, profesore i studente. Super-administrator postavlja kriterije kolegija, dodaje profesore na kolegije i upisuje studente. Prikaz super-administracijskog sučelja prikazano je na *Slikama 5.2. i 5.3.*

FERIT NADZORNA PLOČA

Popis studenata

JMBAG	Ime i prezime	Status studenta
165060018	Maja Đurković	Redovni student
165060247	Lucija Glavaš	Redovni student
149210607	Tomislav Grgurić	Redovni student
165060044	Bruno Matković	Redovni student

Popis profesora

Matični broj znanstvenika	Ime i prezime	Funkcija	Katedra	Kontakt
266160	doc.dr.sc. Damir Blažević	docent	Katedra za računalno inženjerstvo	damir.blazevic@etfos.hr

Popis kolegija

Naziv	Studij	Semestar	ECTS
Objektno orijentirano programiranje (SR303-15)	Stručni studij informatike	4.	6.5

Sl. 5.2. Super-administracija - Nadzorna ploča

FERIT ADMINISTRACIJA STUDENATA

Dodaj studenta

JMBAG	Ime i prezime	Status studenta	Upiši na kolegij	Ispisi s kolegija	Uredi	Obriši
165060018	Maja Đurković	Redovni student	+	x		
165060247	Lucija Glavaš	Redovni student	+	x		
149210607	Tomislav Grgurić	Redovni student	+	x		
165060044	Bruno Matković	Redovni student	+	x		

Sl. 5.3. Super-administracija - Popis studenata

Administracijskom sučelju imaju pristup svi profesori dodani u sustav. Kada se prijave u sustav, profesori imaju pregled svih kolegija na koje su dodani i mogu vršiti evidenciju studenata koji su upisani na njihove kolegije. Evidencija se svodi na praćenje uspjeha studenata na predavanjima, domaćim zadaćama, provjerama znanja i laboratorijskim vježbama. Administracijsko je sučelje prikazano na *Slikama 5.4. i 5.5.*

The screenshot shows the 'NADZORNA PLOČA' (Dashboard) interface. On the left is a navigation menu with options: 'Vodenje evidencije', 'Pregled evidencije', and 'Odjava'. The main content area is divided into two sections: 'Moji kolegiji' and 'Moji studenti'.

Moji kolegiji

Oznaka	Naziv	Studij	Semestar	ECTS
SR303-15	Objektno orijentirano programiranje	Stručni studij informatike	4.	6,5

Moji studenti

JMBAG	Ime i prezime	Status studenta
165060247	Lucija Glavaš	Redovni student
165060018	Maja Đurković	Redovni student
165060044	Bruno Matković	Redovni student
149210607	Tomislav Grgurić	Redovni student

Sl. 1.4. Administracija - Nadzorna ploča

The screenshot shows the 'PREGLED EVIDENCIJE' (Evidence Review) interface. The left navigation menu is the same as in the previous screenshot. The main content area displays a detailed report for the course 'Objektno orijentirano programiranje (SR303-15)'.

Objektno orijentirano programiranje (SR303-15)

JMBAG	Ime	Prezime	KZ1	KZ2	KZ ukupno [%]	LV [%]	DZ [%]	Dolasci PR • AV [%]	Dolaci LV [%]	PI [%]	Komentar	Ocjena
165060018	Maja	Đurković	8/40	0/30	10	62,86	67,5	51,11	100	0	Ponovno ispit	-
165060247	Lucija	Glavaš	39/40	26/30	92,08	100	100	77,78	100	0	Položeno	izvrstan (5)
149210607	Tomislav	Grgurić	0/40	0/30	0	54,29	32,5	51,11	85,71	0	LV neodrađene	-
165060044	Bruno	Matković	0/40	0/30	0	85,71	0	48,89	100	80	Položeno	dobar (3)

Sl. 5.5. Administracija - Pregled evidencije

Sučelje za studente odnosi se na početnu stranicu aplikacije. Na početnoj stranici unosom JMBAG-a studenti mogu vidjeti svoje uspjehe na upisanim kolegijima. Uspjeh se odnosi na konačan prijedlog ocjene, ali i na sve parametre koji utječu na ocjenu, što studentima omogućava detaljan uvid u ostvarene bodove. Sučelje za studente prikazano je na *Slikama 5.6. i 5.7.*

Za provjeru uspjeha ovdje upišite svoj JMBAG

Pronadi moj JMBAG

JMBAG: 165060247 | Ime i prezime: Lucija Glavaš | Status: Redovni student

Naziv kolegija	KZ1	KZ2	KZ ukupno [%]	LV [%]	DZ [%]	Dolasci PR i AV [%]	Dolasci LV [%]	Komentar	PI [%]	Ocjena
Objektno orijentirano programiranje (SR303-1g)	39/40	26/30	92.08	100	100	77.78	100	Položeno	0	Izvrstan (g)

Sl. 5.6. Početna stranica - Prikaz uspjeha

Natrag na početnu

KOLEGIJ: OBJEKTNO ORIJENTIRANO PROGRAMIRANJE

JMBAG: 165060247 | Ime i prezime: Lucija Glavaš | Status: Redovni student

Bodovi ostvareni tijekom semestra

PREDAVANJA		DOMAĆE ZADAĆE		AUDITORNE VJEŽBE		LABORATORIJSKE VJEŽBE				DOLASCI UKUPNO
Dolasci	%	Bodovi	%	Dolasci	%	Dolasci	%	Bodovi	%	
20 / 30	66.67	40 / 40	100	15 / 15	100	7 / 7	100	35 / 35	100	86.67 %

Kontrolne zadaće i pismeni ispit

KZ1		KZ2		KZ UKUPNO	PISMENI ISPIT	
Bodovi	%	Bodovi	%		Bodovi	%
39 / 40	97.5	26 / 30	86.67	92.08 %	/ 100	0

Ukupan broj bodova i ocjena

UTS	KOMENTAR	OCJENA	
		%	Broj
95.5	Položeno	94.132	Izvrstan (g)

Sl. 5.7. Početna stranica - Detaljan uvid u uspjeh na kolegiju

6. ZAKLJUČAK

Web aplikacija za praćenje uspjeha studenata na određenom predmetu napravljena je u sklopu završnog rada na stručnom studiju Informatike. Svrha aplikacije računanje je završne ocjene na pojedinom kolegiju. Aplikacija se sastoji od super-administratorskog sučelja na kojemu se unose sve bitne informacije o studentima, profesorima i kolegijima, od administratorskog sučelja na kojemu profesori vrše evidenciju tokom semestra te od javnog sučelja na kojemu studenti pomoću svog JMBAG-a mogu vidjeti svoj uspjeh na pojedinim kolegijima.

Aplikacija je napisana u HTML-u i CSS, a baza je kreirana na Linux serveru pomoću MySQL-a. Aplikacija je optimizirana i može joj se pristupiti sa svih uređaja i operacijskih sustava.

LITERATURA

[1] Doc.dr.sc. Ivica Lukić, „Baze podataka – Uvodno predavanje, modeli baza podataka i osnovni pojmovi“, PowerPoint prezentacija, 2016. (Zadnji pristup: 16. srpnja 2017.)

- https://loomen.carnet.hr/pluginfile.php/506150/mod_resource/content/2/BP%2001%20Uvodno%20predavanje.pdf

[2] Doc.dr.sc. Ivica Lukić, „Baze podataka – Relacijski model podataka“, PowerPoint prezentacija, 2016. (Zadnji pristup: 2. rujna 2017.)

- https://loomen.carnet.hr/pluginfile.php/506151/mod_resource/content/1/BP%2002%20Relacijski%20model.pdf

[3] Wikipedia, „HTML“, članak s web stranice (Zadnji pristup: 15. lipnja 2017.)

- <https://en.wikipedia.org/wiki/HTML>

[4] Wikipedia, „MySQL“, članak s web stranice (Zadnji pristup: 15. lipnja 2017.)

- <https://en.wikipedia.org/wiki/MySQL>

[5] SmartDraw, „Entity Relationship Diagram“, web stranica, 2017. (zadnji pristup: 2. rujna 2017.)

- <https://www.smartdraw.com/entity-relationship-diagram/>

SAŽETAK

Web aplikacija za praćenje uspjeha studenata na određenom predmetu napravljena je prema pravilniku ocjenjivanja Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek iz 2017. Aplikacija točno računa uspjeh studenata pomoću kriterija ocjenjivanja i evidencije unesene u bazu podataka.

Za kreiranje klijentskog dijela aplikacije korišteni su HTML, CSS, Bootstrap i JavaScript, a na poslužiteljskoj je strani korišten PHP i MySQL na stvarnom Linux serveru. Za upravljanje bazom podataka može se koristiti cPanel ili super-administratorski dio unutar same aplikacije.

Ključne riječi: HTML, CSS, Bootstrap, PHP, MySQL, server, baza podataka, obrazac

WEB APPLICATION FOR STUDENT COURSE RECORDS

Abstract

A Web Application for Student Course Records was made according to the Rules of Assessment of the Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, from 2017. The application accurately calculates student success using the course rating criteria and student records entered in a database.

Client side of the application was created with HTML, CSS, Bootstrap and JavaScript. Server side of the application was created with PHP and MySQL on a real Linux server. Database can be manipulated through cPanel or through super-admin application interface.

Keywords: HTML, CSS, Bootstrap, PHP, MySQL, server, database, form

ŽIVOTOPIS

Lucija Glavaš rođena je 24. prosinca 1994. i Osijeku. Osnovnu je školu i prirodoslovno-matematičku gimnaziju završila je u Osijeku. Nakon završene srednje škole, u 2013. godini, upisala je stručni studij Informatike na Elektrotehničkom fakultetu Osijek (danas: Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek).

Lucija Glavaš

PRILOZI

Svi prilozi korišteni za izradu ovoga rada nalaze se na CD-u priloženom uz rad.