Web sustav za evidenciju i ocjenjivanje studenata

Nenadić, Petar

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://urn.nsk.hr/urn:nbn:hr:200:185352

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij

Smjer Informatika

WEB APLIKACIJA ZA EVIDENCIJU I OCJENJIVANJE STUDENTA

Završni rad

Petar Nenadić

Osijek, 2017.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PREGLED UPOTRIJEBLJENIH TEHNOLOGIJA 2

2.1. HTML5 3

2.2. CSS3 5

2.3. PHP 6

2.4. SQL 10

2.5. JavaScript 14

2.6. AJAX 15

2.7. JSON 16

3. RAZVOJ WEB APLIKACIJE ZA EVIDENCIJU I OCJENJIVANJE STUDENATA 18

3.1 Korisnički zahtjevi 18

3.2 Struktura projekta 19

3.3 Sloj pristupa bazi podataka (*engl. Data Access Layer*) 21

4. ZAKLJUČAK 25

LITERATURA 26

SAŽETAK 27

ABSTRACT 28

ŽIVOTOPIS 29

1. UVOD

Razvojem Interneta i tehnologije, pojavljuju se aplikacije sa različitim namjenama. Iz tog razloga porasla je potreba za aplikacijama koje uvelike olakšavaju svakodnevni rad i štede vrijeme korisniku. Također, sve više aplikacija se uvode u sustav obrazovanja kako bi se olakšala komunikacija između studenata i profesora.

Glavni cilj ovog završnog rada je razviti samostalnu i potpuno funkcionalnu Internet aplikaciju koja će zadovoljiti današnje standarde i korisničke zahtjeve. Za izradu rada koriste se znanja stečena tijekom studiranja iz područja baza podataka i programiranja [1].

Ovaj dokument se sastoji od dva dijela: teorijski i praktični dio. Teorijski dio se sastoji od navođenja i objašnjenja izbora pojedinih tehnologija koje su korištene pri izradi ove Internet aplikacije. Također, objašnjava se zašto je i na koji način korištena implementacija pojedinih tehnologija u izradi ove aplikacije.

1.1. Zadatak završnog rada

Razviti sustav za evidenciju studenata u kojem je moguće pratiti napredak studenata kroz semestar. Sustav omogućuje prijavu studenata i nastavnika, unos bodova za svaku studentsku aktivnost i računanje ukupne ocjene. Studenti imaju mogućnost prijave na ispit te potvrdu slažu li se s ocjenom ako su kolokvirali.

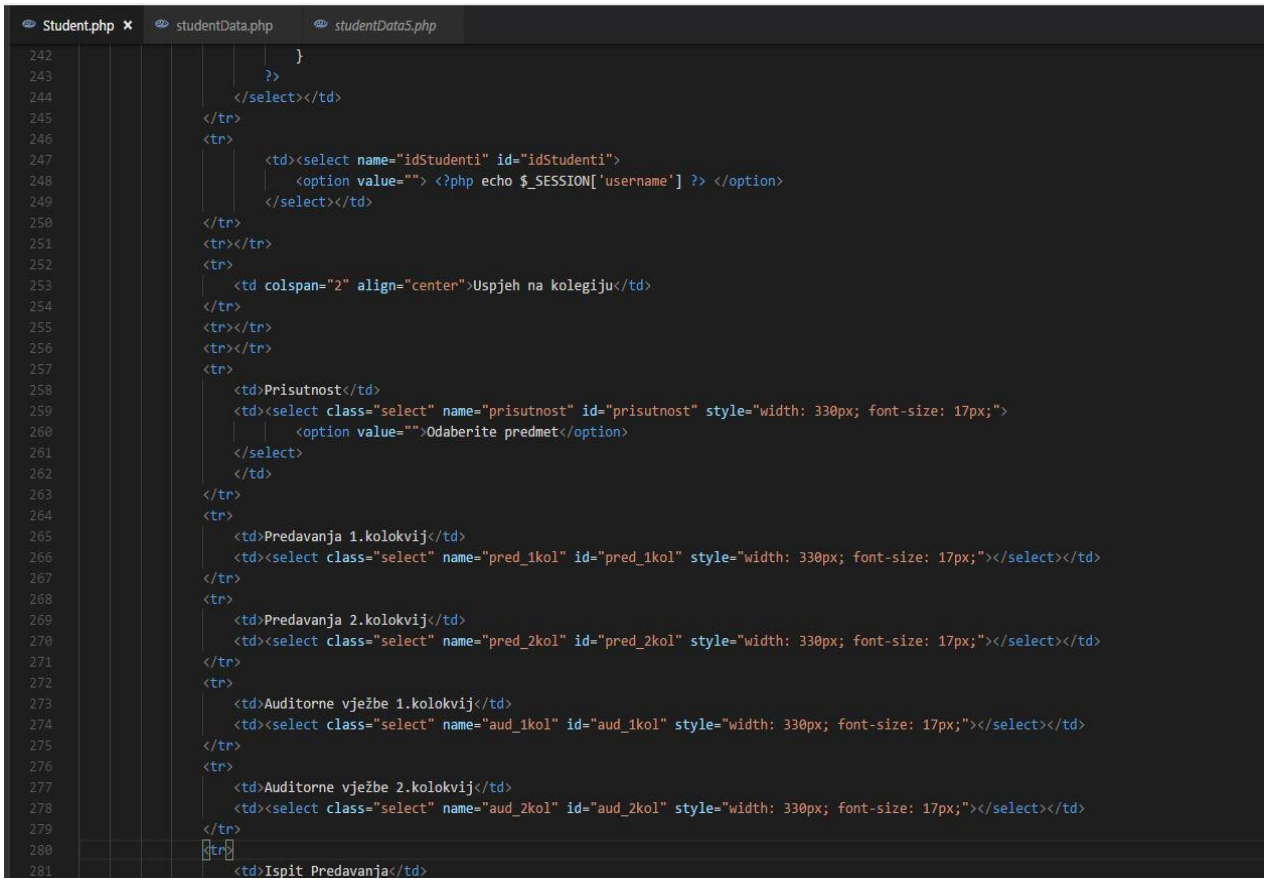
2. PREGLED UPOTRIJEBLJENIH TEHNOLOGIJA

Prema naslovu završnog rada se može zaključiti da će izrada zahtijevati specifične tehnologije. Kako bi se napravila sama jezgra aplikacije potrebno je uvesti neki poslužiteljski jezik. Najbolje rješenje za to se pokazao PHP, a samim time potreban je i MySQL za kreiranje baze podataka. Kako bi se zadovoljile backend strane aplikacije, koristi se aplikacija XAMPP iz razloga što PHP i MySQL dolaze u paketu [2].

Za pisanje HTML5, CSS3, JavaScript, AJAX i JSON koristi se uređivač teksta Visual Studio Code [3], djelomično zbog jasnoće čitanja kôda, a djelomično zbog jednostavnosti korištenja.

2.1. HTML5

Za razvoj ove aplikacije većinom je korišten jezik za označavanje hipertekstualnih dokumenata HTML5 (*engl.* ***H****yper****T****ext* ***M****arkup* ***L****anguage*) [4], [5]. HTML5 je izabran iz razloga što je osnovni jezik za pisanje Internet stranica i jedinstven je za sve platforme. HTML nije programski jezik, već jezik koji označava hipertekstualne dokumente koji se objavljuju preko World Wide Web servisa (WWW). Pomoću svojih oznaka, HTML govori web pregledniku kako treba rasporediti elemente na stranici, te kako ona treba izgledati.



Sl. 2.1. *Izgled skripte napisan u jeziku HTML5.*

Pri izradi stranice upotrijebljen je izgled tablice koji sadrži mjesta za unos (<*input class=“text“ type=“text“ name=“pr“ />*) koje pojedini profesor ispunjava za svakog studenta koji je upisan na predmet u tekućoj akademskoj godini. Profesor odabire predmet (<*td>Odaberite predmet</td>*), te ime i prezime studenta (<*td>Student</td>*). Zatim unosi rezultate za pojedini kolokvij (ako je student kolokvirao) ili za ispit (<*input type=“text“ />*) i potvrđuje unos klikom na tipku *Potvrdite* (<*input type=“button“ />*). Nakon toga, klikom na određenu tipku (npr. *Predavanja*) obavlja se funkcija izračunavanja ocjene za svaku aktivnost (<*input class=“button“ type =“button“ value=“Predavanja“*).

Ista implementacija je napravljena i za studente, ali s malim izmjenama. Student na svojoj stranici ima pravo vidjeti ocjenu za pojedinu aktivnost, potvrditi ocjenu ako se s njom slaže, klikom na tipku *Potvrdi ocjenu*. Ako se ne slaže s ocjenom ima mogućnost prijave ispita, odabirom jedne od vrijednosti iz padajućeg izbornika (*Predavanja, Auditorne vježbe* ili *Cijeli kolegij*). Nakon izbora ispita potvrđuje sa klikom na tipku *Prijavite ispit.* Zatim se u bazu unosi vrijednost koja odgovara pojedinoj vrsti ispita koju je student odabrao.

2.2. CSS3

Kako HTML5 jezik ima skromne naredbe za stilsku konfiguraciju stranice upotrijebljen je CSS3 (*engl.* ***C****ascading* ***S****tyle* ***S****heets*) [5], [6]. To su stilski predlošci koji omogućuju mijenjanje boja na stranici, oblika pojedinih elemenata, izgled i veličinu teksta, itd. Pohranjuju se u posebne datoteke koje se učitavaju kao vanjski izvori Web stranice, ili se mogu pohraniti u zaglavlje Web stranice (unutar oznake *head*). U ovom projektu CSS3 je upotrijebljen za mijenjanje pozadinske boje, izgleda teksta, izgleda tablica, postavljanje sjena na pojedine objekte te mijenjanje visine i širine elemenata.

```css
table {
    height: 880px;
    width: 700px;
    border-radius: 20px;
    margin-left: auto;
    margin-right: auto;
    margin-top: 90px;
    margin-bottom: 90px;
    background-color: lavender;
}

.shadow {
    box-shadow: 40px 40px 40px whitesmoke;
}

.button2 {
    height: 35px;
    width: auto;
    background-color: lightcyan;
    border-radius: 5px;
    box-shadow: 3px 3px 3px black;
    font-size: 15px;
}

select {
    border-radius: 7px;
    height: 33px;
    width: auto;
}

td {
    padding-left: 30px;
    padding-right: 10px;
    border-radius: 7px;
    font-size: 19px;
}
```

Sl. 2.2. *CSS skripta za oblikovanje Internet stranice.*

Klasa *.button* koja predstavlja dugme je uređena u CSS3, ima svoju širinu, visinu, pozadinsku boju, sjenu i veličinu fonta teksta koji se na njoj nalazi. Također, kod klase *body* koja se odnosi na izgled cijele stranice, primijenjen je kôd za promjenu pozadinske boje. Kod klase *table* upotrijebljen je kôd za visinu, širinu, položaj margina, te pozadinska boja.

2.3. PHP

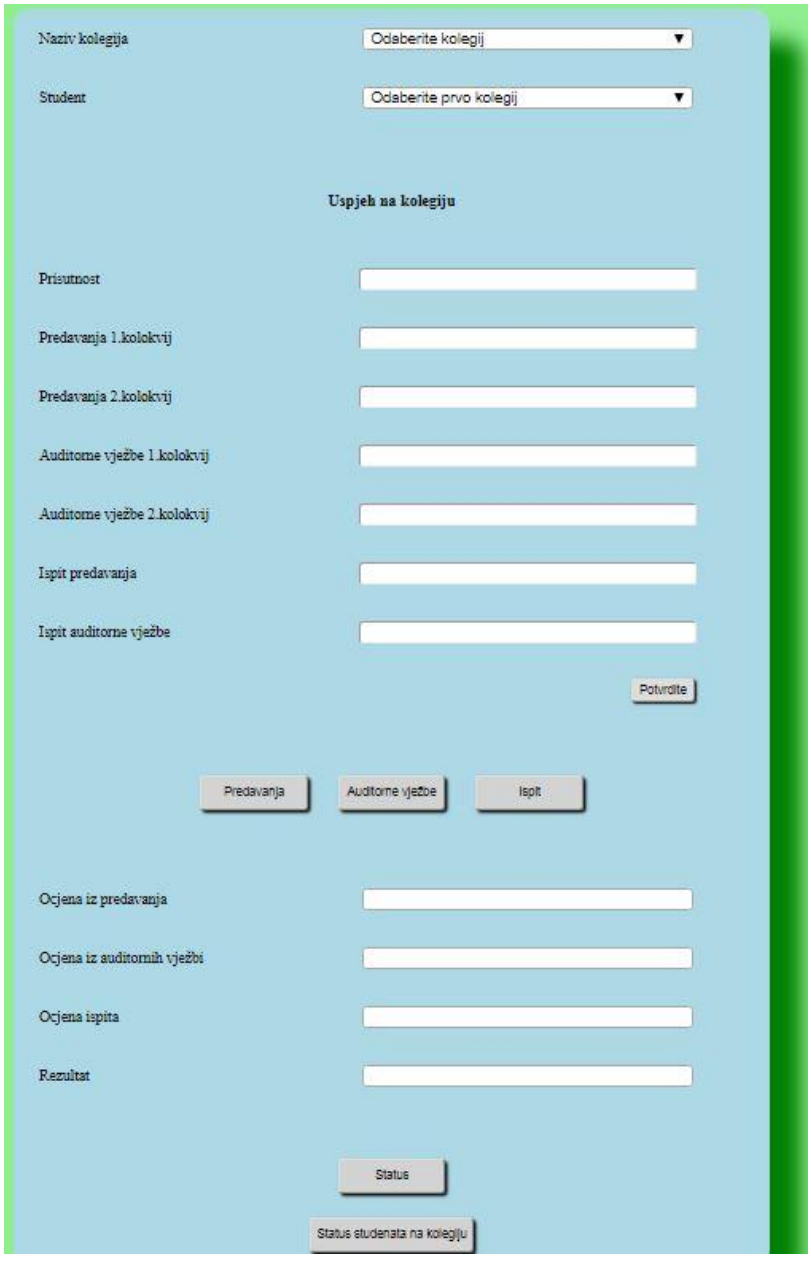
PHP (*engl. **P**hp **H**ypertext **P**reprocessor)* je, trenutno jedan od najmoćnijih i najpopularnijih skriptnih jezika dostupnih na tržištu [7], [8]. To je open-source server-side skriptni programski jezik za dinamičko generiranje HTML kôda. Pomoću PHP-a može se stvoriti web aplikacija sa velikom količinom podataka na vrlo jednostavan način. Jako je fleksibilan, brz u izvođenju i vrlo moćan jezik.

PHP je korišten u ovom projektu iz razloga što postoji potreba za prijavom na stranicu i komuniciranjem sa bazom podataka, kako ne bi došlo do greški u sustavu. Pogreška u sustavu bi se pojavila kada bi studenti i profesori imali pristup istim stranicama. Stoga se pravi stranica, pisana u PHP-u, koja omogućuje prijavu studenata i profesora, te ih usmjerava na razdvojene stranice (slika 2.3.).

```php
<?php
    session_start();
    $db = mysqli_connect("localhost", "root", "", "zavrsni");

    if(isset($_POST['login_btn'])) {
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $password = mysqli_real_escape_string($db, $_POST['password']);

        $password = md5($password); // hashirali smo lozinku prije početka
        $sql = "SELECT * FROM profesor WHERE username = '$username' AND password = '$password'";
        $result = mysqli_query($db, $sql);

        if (mysqli_num_rows($result) == 1) {
            $_SESSION['message'] = "Ulogirani ste";
            $_SESSION['username'] = $username;
            header("location: ../login/Profesor.php");
        } elseif (empty($_POST['username']) OR empty($_POST['password'])) {
            $_SESSION['message'] = "Korisničko ime i lozinka ne mogu biti prazni";
        } else {
            $_SESSION['message'] = "Korisničko ime ili lozinka su netočni!";
        }
    }
?>
```

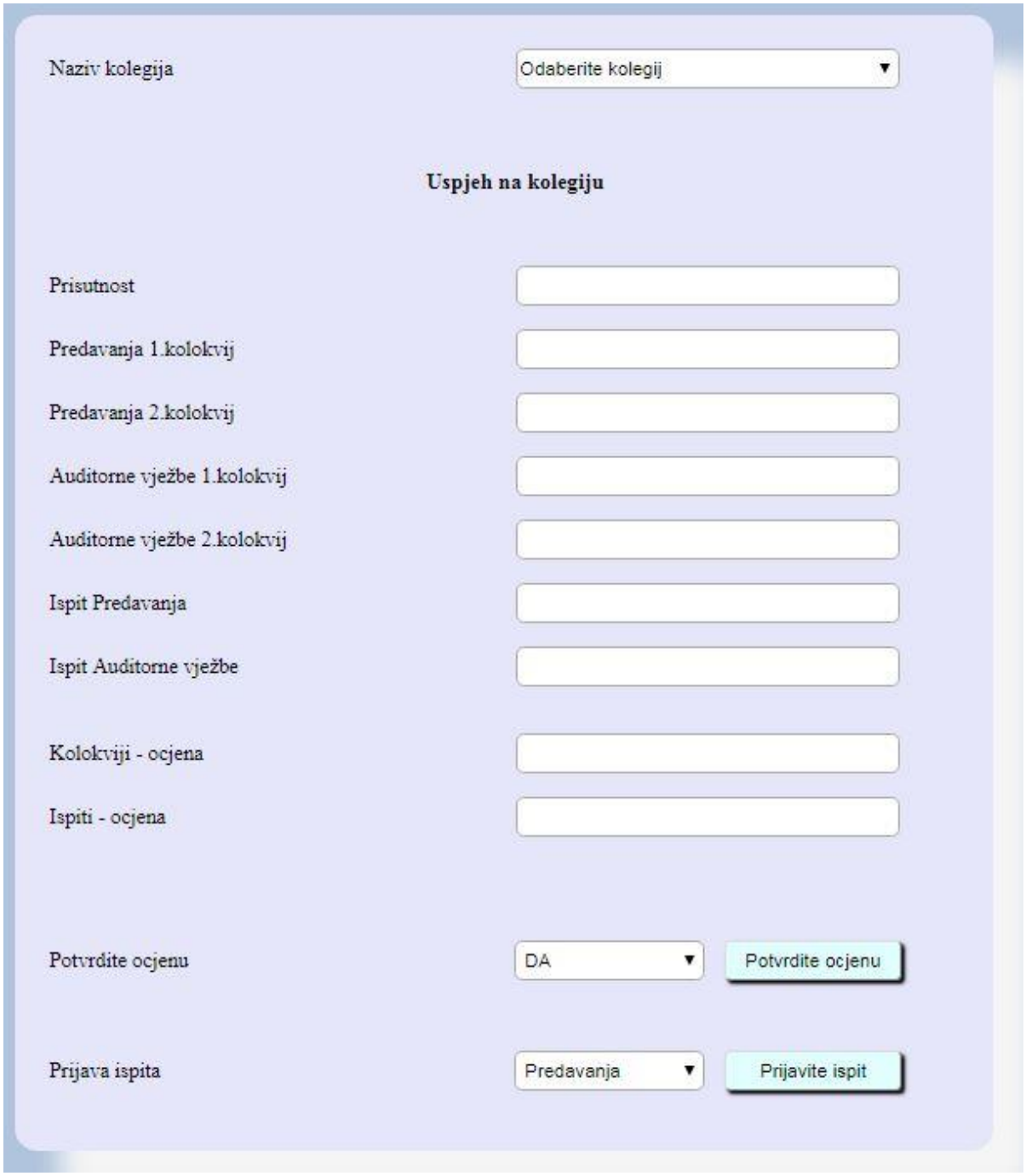
***Sl. 2.3.** PHP kôd za prijavu (login.php).*

Kod prijave profesora otvara se stranica na slici 2.4. koja omogućuje profesoru unos ocjena za pojedine aktivnosti svakog studenta. Odabirom kolegija koji odgovara prijavi profesora popunjava se padajući izbornik sa imenima studenata upisanih na taj kolegij. Nakon toga, podaci se spremaju u tablicu koja je kasnije vidljiva samo profesoru, nositelju kolegija. Tablica sadrži ime, prezime i ocjene za sve aktivnosti na kolegiju za tekući semestar, za pojedinog studenta.



Sl. 2.4. *Stranica za profesore.*

Kod prijave studenta otvara se stranica na slici 2.5. na kojoj se vide rezultati iz aktivnosti za pojedini kolegij u tekućem semestru. Nakon unosa svih aktivnosti, student može potvrditi ocjenu ili prijaviti ispit iz aktivnosti koju nije zadovoljio. Također, kada student potvrdi ocjenu ili ispit, oznaka se pojavljuje na istoj tablici koja je vidljiva samo profesoru, nositelju kolegija.



Sl. 2.5. *Stranica za studente.*

Na slici 2.6. prikazan je kôd, pisan u PHP-u, za registraciju novih korisnika. Nakon što administrator unese podatke za profesora ili studenta, omogućen je pristup stranici za evidenciju i ocjenjivanje. Također, na stranici za unos ocjena moguće je i promijeniti podatke, odnosno lozinku za pristup aplikaciji. Kod prijave administratora otvara se stranica s dvije tablice, na kojima se nalaze polja za unos podataka za profesora ili studenta.

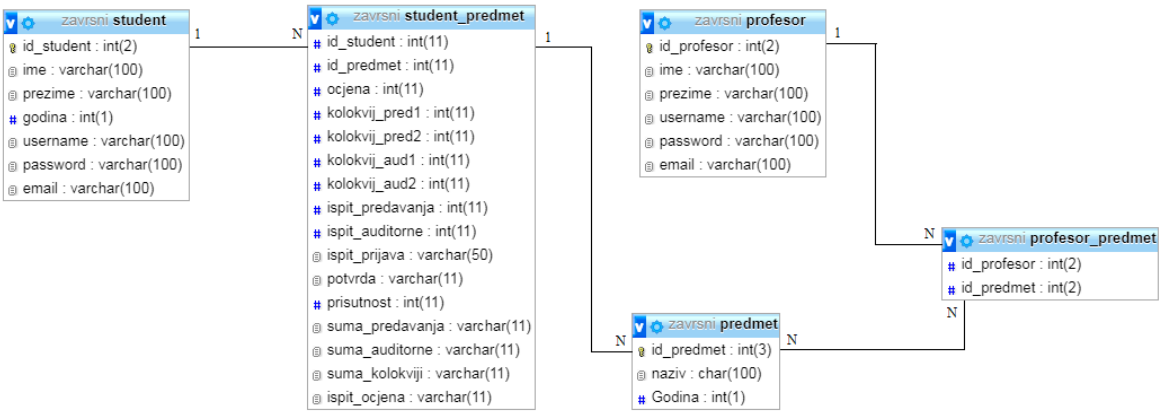
```php
<?php
    session_start();
    $db = mysqli_connect("localhost", "root", "", "zavrsni");

    if(isset($_POST['register_btn'])){
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $ime = mysqli_real_escape_string($db, $_POST['ime']);
        $prezime = mysqli_real_escape_string($db, $_POST['prezime']);
        $email = mysqli_real_escape_string($db, $_POST['email']);
        $password = mysqli_real_escape_string($db, $_POST['password']);
        $password2 = mysqli_real_escape_string($db, $_POST['password2']);
        $godina = mysqli_real_escape_string($db, $_POST['godina']);

        if($password == $password2) {
            // kreiraj korisnika
            $password = md5($password); // hashiraj lozinku prije pohrane zbog sigurnosti
            $sql = "INSERT INTO student (ime, prezime, username, password, email, godina)
                VALUES ('$ime', '$prezime', '$username', '$password', '$email', '$godina')";
            if(mysqli_query($db, $sql)){
                echo "Uspjesno!";
                $lastid = mysqli_insert_id();
                $sql = "INSERT INTO student_predmet (id_student, id_predmet) VALUES ($lastid, )";
            }
            $_SESSION['message'] = "Logirani ste.";
            $_SESSION['username'] = $username;
            header("location: ../login/Student.php"); // Usmjeri na stranicu za studente
        } else {
            // pogreška
            $_SESSION['message'] = "Lozinke se ne podudaraju.";
        }
    }
?>
```

Sl. 2.6. *PHP kôd za registraciju novih korisnika.*

2.4. SQL

Kako bi aplikacija za evidenciju i ocjenjivanje studenata bila funkcionalna potrebno je uvesti bazu podatka sa podacima o profesorima, studentima, predmetima i ispitima. Za kreiranje i upravljanje bazom korišten je standardizirani jezik za baze podataka SQL (*engl.* ***S****tructured* ***Q****uery* ***L****anguage*,) [9]. SQL pripada skupini neproceduralnih upitnih jezika koji služi za dohvat, promjenu, dodavanje i brisanje podataka, kao i za oblikovanje strukture podataka unutar relacijskih baza. Zasnovan je na relacijskom računu, ali je matematička notacija zamijenjena ključnim riječima nalik na engleski govorni jezik. Osim zadavanja upita, jezik omogućuje i definiranje i ažuriranje relacija, sortiranje i formatiranje ispisa, utjecaj na fizičku građu baze, neke aritmetičke operacije s podacima, definiranje pogleda, te kontrolu sigurnosti [10].

Baza podataka za ovu aplikaciju sastavljena je od 5 tablica:



Sl. 2.7. *Tablice za baze podataka.*

Prva tablica, imena *Student*, sadrži podatke o studentu, tj. *id_student*, *ime*, *prezime*, *godina*, *username*, *password* i *email*. *Id_student* je jedinstveni broj koji se pridružuje studentu kako bi se lakše pristupalo podacima o aktivnostima na kolegiju. *Godina* označava godinu studija na koju je student upisan. Korisničko ime (*engl. Username*) i lozinka (*engl. Password*) označavaju vrijednosti koje će se upotrebljavati pri prijavi na stranicu za studente. Za tipove podataka upotrijebljeni su INT i VARCHAR [8].

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_student**	int(2)			No	*None*		AUTO_INCREMENT
2	**ime**	varchar(100)	latin1_swedish_ci		No	*None*		
3	**prezime**	varchar(100)	latin1_swedish_ci		No	*None*		
4	**godina**	int(1)			No	*None*		
5	**username**	varchar(100)	latin1_swedish_ci		No	*None*		
6	**password**	varchar(100)	latin1_swedish_ci		No	*None*		
7	**email**	varchar(100)	latin1_swedish_ci		No	*None*		

Sl. 2.8. *Tablica Student.*

Druga tablica, imena *Profesor*, sadrži podatke o profesorima: *ime*, *prezime*, *id_profesor*, *username*, *password* i *email*. Svaki profesor ima svoj *id_profesor* kako bi mogao pristupiti svom kolegiju i unijeti podatke za aktivnost studenata. Također se koriste korisničko ime (*engl. Username*) i lozinka (*engl. Password*) kako bi profesori imali pristup stranici za profesore. Tipovi podataka korišteni u ovoj tablici su, također, INT i VARCHAR.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_profesor**	int(2)			No	*None*		AUTO_INCREMENT
2	**ime**	varchar(100)	latin1_swedish_ci		No	*None*		
3	**prezime**	varchar(100)	latin1_swedish_ci		No	*None*		
4	**username**	varchar(100)	latin1_swedish_ci		No	*None*		
5	**password**	varchar(100)	latin1_swedish_ci		No	*None*		
6	**email**	varchar(100)	latin1_swedish_ci		No	*None*		

Sl. 2.9. *Tablica Profesor.*

Treća tablica, naziva *Predmet*, sadrži podatke o *naziv* predmeta, *id_predmet, godina, pred_koef* i *aud-koef*. *Godina* u ovoj tablici odnosi se na godinu na kojoj se kolegij izvodi. Vrijednosti *pred_koef* i *aud_koef* su decimalni brojevi koji se množe sa rezultatima iz aktivnosti kako bi se dobila konačna ocjena za pojedini kolegij. Tipovi podataka korišteni u ovoj tablici su INT, CHAR i DECIMAL.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_predmet**	int(3)			No	*None*		AUTO_INCREMENT
2	**naziv**	char(100)	latin1_swedish_ci		Yes	*NULL*		
3	**godina**	int(11)			No	*None*		
4	**pred_koef**	decimal(2,1)			Yes	*NULL*		
5	**aud_koef**	decimal(2,1)			Yes	*NULL*		

Sl. 2.10. *Tablica Predmet.*

Četvrta tablica, naziva *Student_predmet*, sadrži podatke o *id_student, id_predmet, ocjena, kolokvij_pred1, kolokvij_pred2, kolokvij_aud1, kolokvij_aud2, ispit_predavanja, ispit_auditorne, ispit_prijava, potvrda, prisutnost, suma_predavanja, suma_auditorne, suma_kolokviji* i *ispit_ocjena*. Potreba za ubacivanjem *id_student* i *id_predmet* u tablicu *Student_predmet* pojavljuje se zbog toga što je potrebno povezati sve tablice u funkcionalnu cjelinu. *Id_sudent* je strani ključ iz tablice *student*, a *id_predmet* je strani ključ iz tablice *predmet. Kolokvij_pred1* i *kolokvij_pred2* označavaju ocjene koje je student ostvario na kolokvijima iz predavanja. *Kolokvij_aud1* i *kolokvij_aud2* označavaju ocjene koje je student ostvario na kolokvijima iz auditornih vježbi. *Ispit_predavanja* i *ispit_auditorne* označavaju ocjene koje je student ostvario na ispitima iz predavanja ili auditornih vježbi. *Potvrda* je vrijednost koja se mijenja u ovisnosti jeli student potvrdio konačnu ocjenu iz kolegija ili ne. *Prisutnost* je postotak koji je student ostvario dolaskom na predavanja i auditorne vježbe. *Suma_predavanja, suma_auditorne* označavaju zbroj ocjena koje je student ostvario polaganjem kolokvija iz predavanja i auditornih vježbi. *Suma_kolokviji* je zbroj ocjena koje je student postigao polaganjem kolokvija. *Ispit_ocjena* je

konačna ocjena koju je student ostvario polaganjem kolegija na kolokvijima ili ispitima. Tip podataka u ovoj tablici je INT i VARCHAR.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_student	int(11)			No	None		
2	id_predmet	int(11)			No	None		
3	ocjena	int(11)			Yes	NULL		
4	kolokvij_pred1	int(11)			Yes	NULL		
5	kolokvij_pred2	int(11)			Yes	NULL		
6	kolokvij_aud1	int(11)			Yes	NULL		
7	kolokvij_aud2	int(11)			Yes	NULL		
8	ispit_predavanja	int(11)			Yes	NULL		
9	ispit_auditorne	int(11)			Yes	NULL		
10	ispit_prijava	varchar(50)	latin1_swedish_ci		Yes	NULL		
11	potvrda	varchar(11)	latin1_swedish_ci		Yes	NULL		
12	prisutnost	int(11)			Yes	NULL		
13	suma_predavanja	varchar(11)	latin1_swedish_ci		Yes	NULL		
14	suma_auditorne	varchar(11)	latin1_swedish_ci		Yes	NULL		
15	suma_kolokviji	varchar(11)	latin1_swedish_ci		Yes	NULL		
16	ispit_ocjena	varchar(11)	latin1_swedish_ci		Yes	NULL		

Sl. 2.11. *Tablica Student_predmet.*

Peta tablica, *profesor_predmet,* koja se nalazi na slici 2.12. sadrži podatke o *id_profesor* i *id_predmet.* Ova tablica je potrebna iz razloga što je potrebno povezati profesora sa kolegijem na kojem predaje. *Id_profesor* je strani ključ iz tablice *Profesor*, a *id_predmet* je strani ključ iz tablice *Predmet.* Također, tip podataka je INT kao i u tablicama *profesor* i *predmet*.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id_profesor	int(2)			No	None	
2	id_predmet	int(2)			No	None	

Sl. 2.12. *Tablica profesor_predmet.*

2.5. JavaScript

JavaScript je jednostavan objektno orijentiran programski jezik koji ima velike mogućnosti, primjenjiv je za klijentske i serverske web aplikacije [11], [12], [13]. Web aplikaciji daje dinamičnost i interaktivnost kao što je prikazivanje korisničkih poruka, provjera ispravnosti sadržaja popunjenog obrasca, obavljanje matematičkih izračuna, animiranje slika ili kreiranje slika promjenjivih pomicanjem miša, detektiranje *plug-in* modula instaliranih u pregledniku, itd. JavaScript je danas najzastupljeniji klijentski skriptni jezik i podržavaju ga svi popularni web preglednici. Iako ima slično ime i sintaksu kao i jezik JAVA, jako se razlikuju u semantici i korištenim objektnim modelima.

JavaScript je korišten za različite animacije na stranici, kao što je mijenjanje studenata na kolegijima. Odnosno, kada se promijeni kolegij na stranici profesor, u padajućem izborniku se mijenja popis studenata upisanih na taj kolegij. Za tu akciju nije potrebno osvježavanje stranice.

```
$(document).ready(function() {
    $('#predmet').on('change',function() {
        var idpredmet = $(this).val();
        if(idpredmet) {
            $.ajax({
                type:'POST',
                url:'ajaxData.php',
                data:'id_predmet='+idpredmet,
                success:function(html){
                    $('#studenti').html(html);
                }
            });
        } else {
            $('#studenti').html('<option value="">Molimo odaberite kolegij</option>');
        }
    });
    //
});
```

Sl. 2.13. *JavaScript kôd za promjenu studenata upisanih na kolegij kod stranice za profesore.*

JavaScript (JSON) je korišten i na stranici za studente. Kod promjene predmeta, na kojeg je student prijavljen, mijenjaju se vrijednosti u tablici aktivnosti.

2.6. AJAX

AJAX (*engl. **A**synchronous **J**avaScript **A**nd **X**ML)* predstavlja grupu tehnologija namijenjenih za dinamičko kreiranje Web stranica [12], [13]. Korištenjem AJAX-a poboljšava se kvaliteta interaktivnosti s korisnikom, s tim da stranica što više sliči *desktop* aplikacijama. Ideja AJAX-a je da stranica na kojoj se odvija Web aplikacija učita samo jednom, a da se, istodobno, svaka slijedeća komunikacija sa serverom izvršava bez blokiranja i ponovnog učitavanja stranice.

AJAX naredbe upotrijebljene su kod unosa vrijednosti za aktivnosti studenata na pojedinom kolegiju (slika 2.14.). Pritiskom na tipku (<*input type="button"*>) vrijednost iz unosa (<*input type="text"*>) se unosi u tablicu *student_predmet*, nakon čega student može vidjeti svoju ocjenu, odnosno rezultat.

```
<td align="center">
    <input type="text" id="pred2" name="pred2" style="width: 270px; font-size: 19px;">
    <input class="button3" type="button" id="unesi3" value="Potvrdi" />
</td>
```

Sl. 2.14. *Kôd za unos vrijednosti u bazu podataka.*

Također, kod stranice za studente AJAX se koristi za potvrđivanje ocjena ili prijavu ispita. Student odabire vrijednost *Da* ili *Ne* za potvrđivanje ocjene ili prijavljivanje ispita (*Predavanja, Auditorne vježbe, Cijeli kolegij*) i klikom na *Potvrdite ocjenu* ili *Prijavite ispit* u bazu se unosi određeni odabir.

Sl. 2.15. *Potvrđivanje ocjene i prijava ispita.*

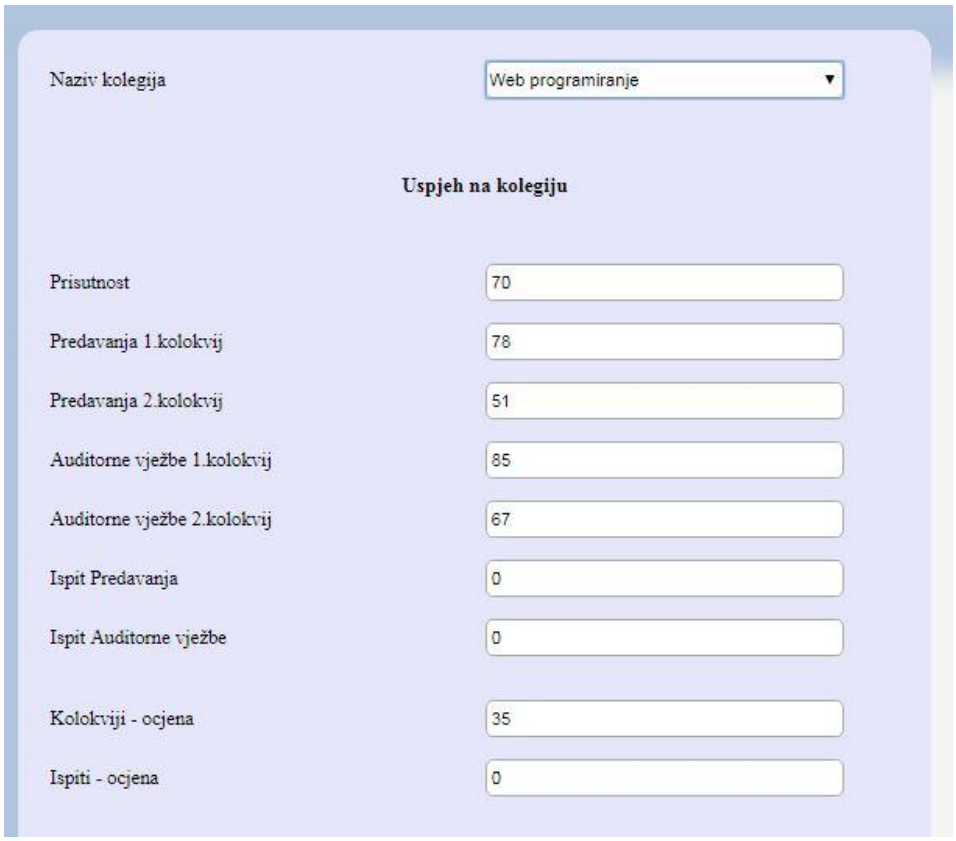
2.7. JSON

JSON (*engl. **J**ava**S**cript **O**bject **N**otation*) je tekstualno baziran standard dizajniran za razumljivu razmjenu podataka [13], [14]. Ima krati kôd, pa je lagan za pisanje i razumljiv je za čitanje. Služi za razmjenu podataka između servera i Web stranice, a podaci mogu biti samo u tekstualnom obliku. Svaki programski jezik može pročitati JSON, iako koristi JavaScript sintaksu.

JSON je upotrijebljen na stranici za studente kod ispisa bodova za aktivnosti na kolegiju. Kod promjene kolegija u padajućem izborniku pomoću JSON-a stranica komunicira sa bazom i poziva podatke za pojedine aktivnosti za odgovarajući kolegij. Kôd za JSON je sličan kao i kôd za AJAX, s razlikom da je kod JSON-a ubačen *dataType: 'JSON'*, naredba da umjesto AJAX podataka tražimo JSON podatke kao odgovor na upit.

```
$('#predmet').on('change',function() {
    var idpredmet = $(this).val();
    var idStudent = $('#idStudenti').val();
    var data = {'id_predmet': idpredmet,
                'id_student': idStudent};
    console.log(data);
    if(idpredmet) {
        $.ajax({
            data: data,
            type:'post',
            url:'studentData.php',
            dataType: 'JSON',
            success:function(data){ // !!!!!!!
                console.log(data);
                var student_object = data[0];
                $("#prisutnost").find(":selected").html(student_object.prisutnost);
            }
        });
    } else {
        $('#prisutnost').html('<option value="">Molimo odaberite kolegij</option>');
    }
```

Sl. 2.16. *JSON kôd za pozivanje podataka iz baze.*



Sl. 2.17. *Tablica student nakon primjene JSON-a.*

3. RAZVOJ WEB APLIKACIJE ZA EVIDENCIJU I OCJENJIVANJE STUDENATA

3.1 Korisnički zahtjevi

Na početku svakog rada potrebno je jasno definirati korisničke zahtjeve kako bi korisnici dobili proizvod kakav žele, odnosno kako bi programeri dobili jasniju predodžbu kako bi proizvod trebao izgledati. Korisnički zahtjevi ujedno služe kao upute za rad, ali i zaštita od zloupotrebe ove aplikacije.

S obzirom da se radi o web aplikaciji za evidenciju i ocjenjivanje studenata, od velike je važnosti da stranica posjeduje i sustav za prijavu, odnosno registraciju. Nadalje, treba korisniku omogućiti da povrati svoju lozinku ako je zaboravi ili izgubi. Korisnički računi su privatna stvar, stoga je jako važno da aplikacija bude robusna i kvalitetna kako bi svi njeni korisnici bili zaštićeni u potpunosti.

Potpuni podaci o studentima i profesorima su dostupni samo administratoru, što je i logično jer samo administrator ima mogućnost unosa podataka o profesorima i studentima. Profesor ima mogućnost pregledavanja statusa svih studenata na svom kolegiju, što uključuje i njihove ocjene, prisutnost na kolegiju, te potvrdu da li se slažu s ocjenom ili žele ići na ispit.

S druge strane, studenti imaju uvid u vlastiti status na svim kolegijima koje su upisali u tekućoj akademskoj godini. Također, imaju mogućnost potvrde ocjene te prijave ispita, ako nisu položili pojedini dio iz kolegija (predavanja ili auditorne vježbe) ili gradivo iz cijelog kolegija. Ocjene ostalih studenata nisu dostupne na pregled.

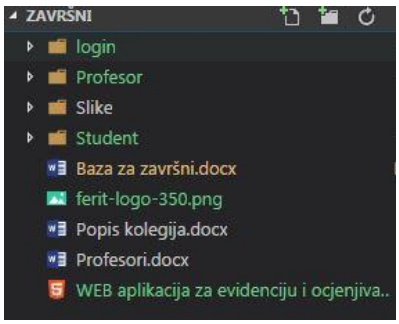
Nakon prijave na stranicu u lijevom kutu se pojavljuje korisničko ime (*engl. username*) profesora ili studenta. Početna lozinka koju dobivaju profesori i studenti je u obliku *ime.prezime,* koja nakon promjene nije dostupna čak ni administratoru. U slučaju gubitka lozinke, potrebno je kontaktirati administratora koji resetira podatke o lozinki i vraća je u oblik *ime.prezime*. Ispod korisničkog imena nalazi se link za odjavu sa stranice.

3.2 Struktura projekta

Zbog jednostavnosti prikaza strukture aplikacije, ista je rastavljena na nekoliko manjih cjelina. Struktura aplikacije je prikazana na slikama 3.1. i 3.2. Na slici 3.1 je prikazana struktura za web aplikaciju, a na slici 3.2. je prikazana struktura PHP kôda za prijavu, registraciju, itd.

Prikazana struktura na slici 3.1. podijeljena je, kao što se vidi, na 5 dijelova. Ti dijelovi su:

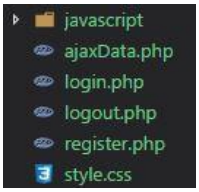
- / -root –datoteka koja sadrži sve ostale datoteke i mape. Tu se vide stranice za administratora, profesora, studenta i stranice za unos novih studenata i profesora.
- /login/ - mapa koja sadrži datoteke potrebne za prijavu i registriranje novih korisnika.
- /Profesor/ - u ovoj mapi se nalaze stranica za profesore sa svim pripadajućim funkcijama pisanim u PHP kôdu
- /slike/ - mapa u kojoj se nalaze slike koje su korištene za izradu aplikacije, stoga će sve poveznice na slike počinjati sa „*./slike/<ime slike>.<ekstenzija>* “.
- /Student/ - mapa koja sadrži stranicu za studente i pripadajuće funkcije pisane u PHP kôdu.



Sl. 3.1. *Struktura projekta 1.*

Slika 3.2. prikazuje strukturu stranice za registriranje i prijavu. Podijeljena je na 6 dijelova, a ti dijelovi su:

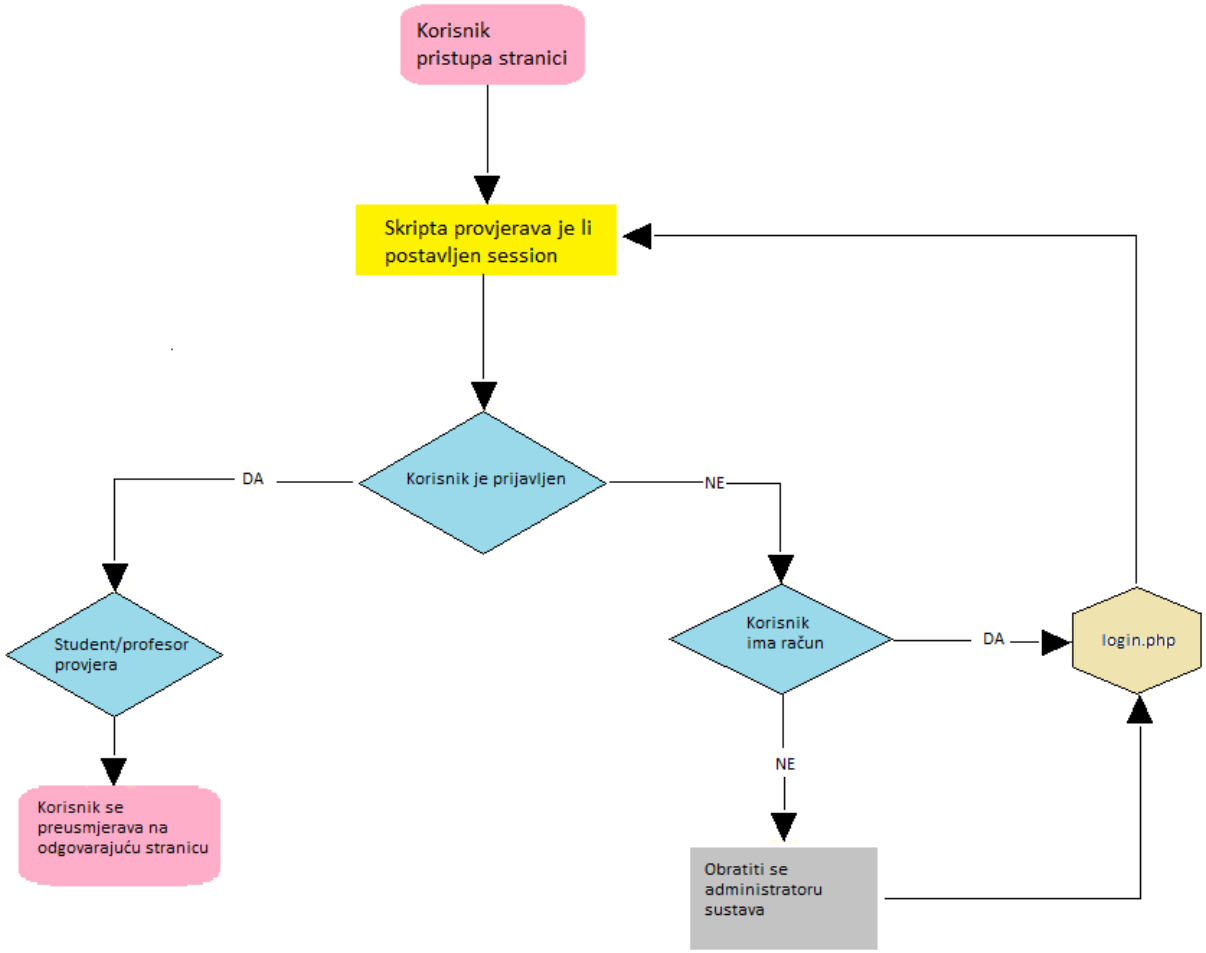
- /root/ - mapa koja sadrži stranice potrebne za prijavu i registriranje studenata i profesora
- /javascript/ - mapa u kojoj se nalaze skripte za upravljanje funkcijama
- /ajaxData.php – PHP kôd za upravljanje stranicom za prijavu
- /login.php – stranica pisana u PHP kôdu za prijavu na stranice *Profesor* ili *Student*
- /logout – PHP skripta za uklanjanje kolačića na korisničkim imenima
- /style.css – datoteka koja sadrži kôd pisan u jeziku CSS3 za oblikovanje stranice



Sl. 3.2. *Struktura projekta 2.*

3.3 Sloj pristupa bazi podataka (*engl. Data Access Layer*)

Ovaj dio aplikacije je odgovoran za pristup bazi podataka. Dijagram ispod teksta najbolje dočarava strukturu prijave na aplikaciju.



Sl. 3.3. *Dijagram toka za sustav prijave na aplikaciju.*

Na slici ispod je prikazan kôd za registriranje studenta na aplikaciju. Prilikom registracije studenta na stranicu dodjeljuje mu se jedinstveni ključ (*id_student*). Ako postoje dva studenta s istim imenom i prezimenom, dodjeljuje im se ovaj ključ kako bi se razlikovali u bazi podataka.

```php
if($password == $password2) {
    // kreiraj korisnika
    $password = md5($password); // hashiraj lozinku prije pohrane zbog sigurnosti
    $sql = "INSERT INTO student (ime, prezime, username, password, email, godina)
    VALUES ('$ime', '$prezime', '$username', '$password', '$email', '$godina')";
    if(mysqli_query($db, $sql)){
        echo "Uspjesno!";
        $lastid = mysqli_insert_id();
        $sql = "INSERT INTO student_predmet (id_student, id_predmet) VALUES ($lastid, )";
    }
    $_SESSION['message'] = "Logirani ste.";
    $_SESSION['username'] = $username;
    header("location: ../login/Student.php"); // Usmjeri na stranicu za studente
} else {
    // pogreška
    $_SESSION['message'] = "Lozinke se ne podudaraju.";
}
```

Sl. 3.4. *PHP skripta za unos studenta.*

Kada student unese svoje podatke u bazu podataka, na ekran se ispisuje poruka za uspješan unos, ili, ako je došlo do pogreške, poruka za grešku. Isti uvjeti vrijede i za unos profesora.

```php
if($password == $password2) {
    // kreiraj korisnika
    $password = md5($password); // hashiraj lozinku prije pohrane zbog sigurnosti
    $sql = "INSERT INTO profesor (ime, prezime, username, password, email)
    VALUES ('$ime', '$prezime', '$username', '$password', '$email')";
    if(mysqli_query($db, $sql)) {
        echo "uspjesno";
        $lastid = mysqli_insert_id();
        $sql = "INSERT INTO profesor_predmet (id_profesor, id_predmet) VALUES ($lastid, $kolegij)";
        mysqli_query($db, $sql);
    } else {
        echo 'Ne mogu dodati';
    }

    // mysqli_query...
    $_SESSION['message'] = "Logirani ste. " .mysqli_insert_id($db);
    $_SESSION['username'] = $username;
    header("location: ../login/Profesor.php"); // Usmjeri na stranicu za profesore
} else {
    // pogreška
    $_SESSION['message'] = "Lozinke se ne podudaraju.";
}
```

Sl. 3.5. *PHP skripta za unos profesora.*

Nakon uspješnog unosa u bazu podataka profesor, odnosno student, se može prijaviti na stranicu i pristupiti podacima koji odgovaraju njegovom statusu na kolegiju. Na slijedećoj stranici je slika na kojoj je prikazan kôd za prijavu na stranicu.

```php
<?php
    session_start();
    $db = mysqli_connect("localhost", "root", "", "zavrsni");

    if(isset($_POST['login_btn'])) {
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $password = mysqli_real_escape_string($db, $_POST['password']);

        $password = md5($password); // hashirali smo lozinku prije početka
        $sql = "SELECT * FROM profesor WHERE username = '$username' AND password = '$password'";
        $result = mysqli_query($db, $sql);

        if (mysqli_num_rows($result) == 1) {
            $_SESSION['message'] = "Ulogirani ste";
            $_SESSION['username'] = $username;
            header("location: ../login/Profesor.php");
        } elseif (empty($_POST['username']) OR empty($_POST['password'])) {
            $_SESSION['message'] = "Korisničko ime i lozinka ne mogu biti prazni";
        } else {
            $_SESSION['message'] = "Korisničko ime ili lozinka su netočni!";
        }
    }
?>
```

Sl. 3.6. *PHP skripta za prijavu na aplikaciju.*

Ako korisnik nije prijavljen u sustav skripta registrira grešku i na ekran se ispisuje poruka „*Ispričavamo se, niste registrirani. Molimo vas da kontaktirate administratora*“. Također, mjesta za unos lozinke i korisničkog imena ne smiju biti prazna, te se kod takve prijave pojavljuje poruka: *„Korisničko ime i lozinka ne mogu biti prazni“*. Kod netočne kombinacije korisničkog imena i lozinke na ekran se ispisuje poruka: *„Korisničko ime ili lozinka su netočni“*.

4. ZAKLJUČAK

Testiranjem ove aplikacije došlo se do kraja završnog rada te je obrađen nastanak aplikacije od početka do kraja. Od baze podataka, njezine implementacije na stranicu, do same strukture stranice. Tijekom razvoja aplikacije, došlo se do zaključka da je korišteno jako puno tehnologija, bez kojih sama aplikacija ne bi mogla funkcionirati.

Kreiranje baze podataka se čini kao jednostavan zadatak. Zbog lakoće pisanja tablica, programer može dobiti osjećaj sigurnosti, ali nakon par sati pisanja aplikacije može se uvidjeti da neki dijelovi nedostaju. Zbog toga je potrebno vratiti se nekoliko koraka unatrag i prepraviti tablice kojima nedostaju neki elementi.

Tijekom pisanja aplikacije došlo se do još jednog zaključka, a to je da je mogućnost pisanja projekta od nule, jako važna programerima jer im daje uvid u sve elemente projekta. Pišu kôdove koje, možda, i ne rade svakodnevno, a sada su prisiljeni.

LITERATURA

[1] Lukić, Ivica; Köhler, Mirko – Osnove Internet programiranja, ETFOS,2011., http://bib.irb.hr/prikazi-rad?&rad=510808, 28.lipnja 2017

[2] XAMPP Apache + MariaDB + PHP + Perl, https://www.apachefriends.org/index.html, 16. lipnja 2017

[3] Visual Studio Code, Source Code Editor, Microsoft, https://code.visualstudio.com/, 01.svibnja 2017.

[4] W3 Schools, HTML5 Tutorial, https://www.w3schools.com/html/default.asp, 16. svibnja 2017.

[5] A. Harris HTML5 and CSS3 All-in-one For Dummies, 3rd Edition, 2014, John Wiley & Sons

[6] W3 Schools, CSS Tutorial https://www.w3schools.com/css/default.asp, 27. svibnja 2017.

[7] W3 Schools, PHP Tutorial, https://www.w3schools.com/php/default.asp, 06.lipnja 2017.

[8] A. Forbes, The Joy of PHP: Deep Dive into Session Variables, Plum Island Publishing, 2013.

[9] M. Vedriš, Uvod u SQL, priručnik za polaznike, Sveučilište u Zagrebu, 2005.

[10] W3 Schools, SQL General Data Types https://www.w3schools.com/sql/sql_datatypes_general.asp, 01. svibnja 2017.

[11] J.N. Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript and Web Graphics, 4th Edition; O'Reilly Media, 2012.

[12] M. Moncur, P. Ballard; AJAX, JavaScript and PHP; Sams Publishing, 2009;

[13] A. Harris, JavaScript and AJAX for dummies, Wiley Publishing, 2010;

[14] E. Elliot, Programming JavaScript Applications: Robust Web architecture with Node, Html5 and Modern JavaScript libraries; O'Reilly Media; 2014;

SAŽETAK

Zadatak ovog završnog rada je izraditi web aplikaciju za evidenciju i ocjenjivanje studenata. Potrebno je bilo napraviti plan izrade, izraditi bazu podataka i arhitekturu datoteka. Također, trebalo je unaprijed zadati vrste i svrhe objekta, razine prava korisnika te detaljno vođenje ocjena za svaki kolegij kojeg student pohađa.

Nakon logičkog djela, posvećena je pažnja korisničkom iskustvu te dizajnirano web sučelje koje ima mogućnost manipuliranja bazom podataka i pristup svim svojstvima aplikacije.

Ključne riječi: web, aplikacija, evidencija, ocjenjivanje, studenti, profesori.

ABSTRACT

Web application for registering and evaluating students

The premise of this paper was to create Web application for registering and evaluating students. It was required to make plan for whole process, design database and folder architecture. Also, the types and purposes of the object, the level of user rights and the detailed assessment of each course attended by the student had to be specified in advance.

After the logical work, attention is paid to user experience and web interface was designed that has the ability to manipulate the database and access all the features of the application.

Keywords: web, application, records, appraisal, students, professors.

ŽIVOTOPIS

Petar Nenadić rođen je 12. veljače 1992. godine u Splitu, RH. Završio je osnovnu školu Stjepan Radić u Imotskom 2006. godine nakon čega je iste godine upisao Opću Gimnaziju dr. Mate Ujevića u Imotskom. Nakon završetka srednje škole 2010. godine upisao je Fakultet elektrotehnike, strojarstva i brodogradnje u Splitu. U akademskoj godini 2014./2015. prebacuje se na Elektrotehnički fakultet u Osijeku.

Petar Nenadić

_____