Web aplikacija za studentske radove

Vlajčić, Armando

Undergraduate thesis / Završni rad

2016

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:200:049340

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Preddiplomski studiji računarstva

Web aplikacija za studentske radove

Završni rad

Armando Vlajčić

Osijek, 2016.

ETFOS
ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 27.06.2016.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Armando Vlajčić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3596, 02.08.2013.
Mentor:	Doc.dr.sc. Ivan Aleksi
Sumentor:	
Naslov završnog rada:	Web aplikacija za studentske radove
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 Postignuti rezultati u odnosu na složenost zadatka: 3 Jasnoća pismenog izražavanja: 3 Razina samostalnosti: 3
Datum prijedloga ocjene mentora:	27.06.2016.
Datum potvrde ocjene Odbora:	11.07.2016.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

ETFOS
ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

IZJAVA O ORIGINALNOSTI RADA

Osijek, 19.07.2016.

Ime i prezime studenta:	Armando Vlajčić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3596, 02.08.2013.
Ephorus podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za studentske radove**

izrađen pod vodstvom mentora Doc.dr.sc. Ivan Aleksi

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

1.2. Poveznica na internet stranicu 1

2. KORIŠTENE TEHNOLIGIJE I ALATI 2

2.1. HTML 2

2.2. CSS 2

2.3. JavaScript 2

2.4. PHP 2

2.5. MySQL 3

2.6. jQuery 3

2.7. Bootstrap 3

2.8. XAMPP 3

3. IZRADA APLIKACIJE 4

3.1. Baza podataka 4

3.2. Izrada web aplikacije i PHP skripte 5

3.2.1. Registracija i prijava korisnika 5

3.2.2. Pretraživanje radova 7

3.2.3. Dodavanje novih radova 9

4. ZAKLJUČAK 12

1. UVOD

Cilj ovog završnog rada je izrada web aplikacije koja će studentima omogućiti objavljivanje svojih radova. Osim objavljivanja svojih radova, studenti će moći pregledati radove drugih studenata iz različitih područja znanosti. U konačnoj verziji web aplikacije, korisnicima će biti omogućena registracija, prijava, dodavanje radova te njihovo pretraživanje po različitim parametrima. U dodavanju svojih radova, korisnici će se moći poslužiti različitim multimedijalnim sadržajima kao što su tekstovi, formule, tablice, slike, video zapisi te 3D modeli.

1.1. Zadatak završnog rada

U ovom završnom radu potrebno je napraviti web aplikaciju za studentske radove. Radovi bi se pisali u HTML5 formatu s unaprijed definiranim stilovima oblikovanjima i prikazima multimedijalnih sadržaja, koji uključuju: tekst, formule, tablice, slike, zvučne i video zapise te 3D modele.

1.2. Poveznica na internet stranicu

http://zavrsni.net23.net/index.php

2. KORIŠTENE TEHNOLIGIJE I ALATI

2.1. HTML

HTML (Hyper Text Markup Language) je osnovni jezik weba i služi za izradu web stranica. On nije programski, već opisni jezik i za njegovu upotrebu se koriste oznake (eng. *tags*) koje najčešće dolaze u paru tako da prva oznaka označava početak, a druga kraj (npr.). Trenutna verzija HTML-a je HTML5 te se s tom verzijom HTML-a ovaj rad izrađuje. Razlog za to je što se u HTML5 dodaju nove oznake koje služe za prikaz novih multimedijalnih formata.

2.2. CSS

CSS (Cascading Style Sheets) je stilski jezik koji služi za promjenu izgleda HTML dokumenta. Pomoću CSS-a možemo urediti prikaz HTML oznaka onako kako nama odgovara. CSS se koristi kako bi se odvojio sadržaj od vizualnog dijela.

2.3. JavaScript

JavaScript je skriptni programski jezik te je trenutno najpopularniji programski jezik za programiranje na webu te je uz HTML i CSS najvažnija tehnologija za izradu web stranica. Pomoću JavaScripta, web stranici omogućujemo interakciju s korisnikom. JavaScript se izvodi na računalu korisnika te ga nije potrebno prevoditi. Danas, svi preglednici podržavaju JavaScript što pridonosi njegovoj popularnosti.

2.4. PHP

PHP je skriptni programski jezik koji se koristi za kreiranje dinamičkih web stranica te povezivanje i manipuliranje s bazom podataka. Većina sintakse i mogućnosti je preuzeta iz programskih jezika C, Java i Perl [1]. PHP se izvodi na poslužitelju te krajnji korisnik ne može vidjeti njegov kod. Trenutno, PHP se kao programski jezik na strani poslužitelja koristi na više od 80% web stranica za koje znamo koji jezik koriste [2]. Razlog tomu je velika podrška za njega, kao i potpuna i jasno razumljiva dokumentacija.

2.5. MySQL

MySQL je jedan od najkorištenijih sustava za upravljanje relacijskim bazama podataka. Također, to je sustav otvorenog koda što znači da je u potpunosti besplatan za korištenje i izmjenjivanje njegova koda. Pomoću MySQL-a možemo kreirati bazu podataka koja će nam služiti za spremanje podataka o radovima i korisnicima.

2.6. jQuery

jQuery je biblioteka napisana u JavaScriptu te sadrži mnoštvo gotovih funkcija pomoću kojih možemo brže i jednostavnije izraditi različite mogućnosti naše web stranice. Kao i MySQL, jQuery je softver otvorenog koda. Danas, jQuery je najdominantnija biblioteka za JavaScript te se koristi na otprilike 70% web stranica gdje se koristi neka JavaScript biblioteka [3].

2.7. Bootstrap

Bootsrap je skup alata (eng. *framework*) koji se koristi za izradu web stranica. Za razliku od nekih drugih alata, Bootstrap se bazira isključivo na olakšavanju izrade izgleda web stranice. On sadrži predloške za gumbe, forme, navigaciju i druge stvari napisane u HTML-u i CSS-u. Iako se u ovom radu više baziramo na funkcionalnost, a ne na izgled, Bootstrap nam omogućava da uz malo napora dobijemo osnovni izgled web aplikacije kako bi bila što više prolagođena korisniku za korištenje.

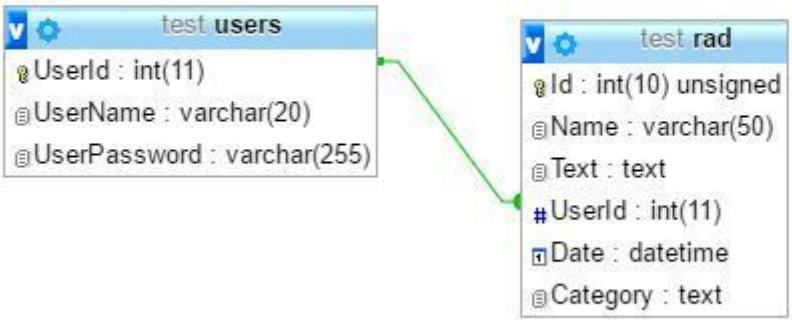
2.8. XAMPP

XAMPP je skup alata koji se koriste za izradu i korištenje besplatnog web poslužitelja. Dolazi za Apache web poslužiteljem, MariaDB baze podataka te prevoditeljima za PHP i Perl skripte. Pomoću njega možemo s lakoćom podesiti lokalni poslužitelj na kojem možemo testirati web aplikaciju i njene mogućnosti.

3. IZRADA APLIKACIJE

3.1. Baza podataka

Za potrebe ovog rada, u bazi podataka su nam potrebne samo dvije tablice, jedna za korisnike i jedna za radove. U njih ćemo spremati sve potrebne podatke o njima.

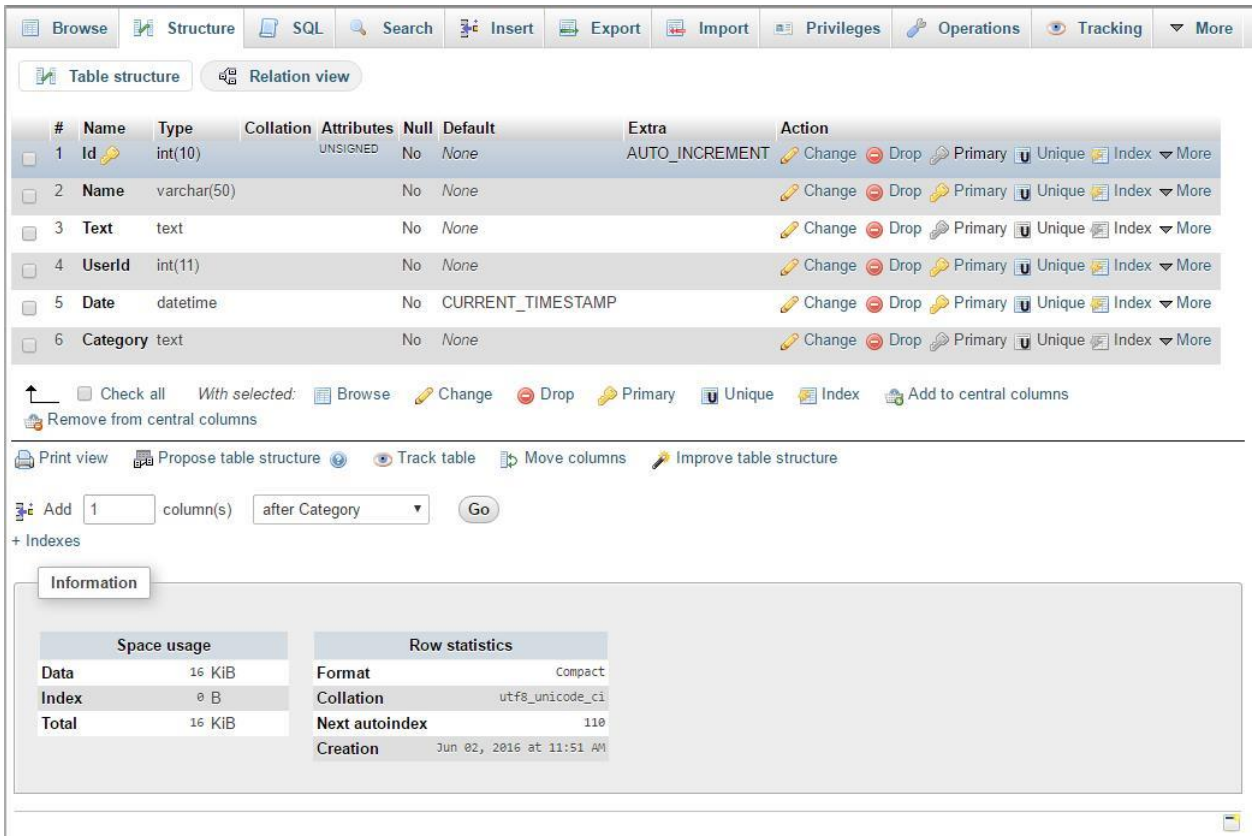


Slika 3.1 Prikaz modela baze podataka

Na slici 3.1 vidimo model baze podataka koji se koristi za izradu ovog rada. U tablici za korisnike se nalaze 3 stupca, UserId - služi za razlikovanje korisnika te je ujedno i primarni ključ, UserName – stupac u koji se sprema korisničko ime te UserPassword – stupac u koji se sprema lozinka korisnika. Razlog za korištenje duljine teksta od 255 kod lozinke korisnika je taj što će se koristiti funkcija koja kriptira tekst koji je korisnik unio za svoju lozinku kako bi se povećala sigurnost.

Druga tablica sadrži 6 stupaca, a to su: Id - služi za razlikovanje radova, Name - sadrži ime rada, Text – sadrži tekst rada tj. njegov cjelokupni sadržaj zapisan u HTML formatu, UserId – stupac u koji se sprema oznaka korisnika koji je objavio rad, također, ovaj stupac je strani ključ koji se veže sa stupcem UserId u tablici users, Date – stupac koji sadrži datum i vrijeme kada je rad objavljen te zadnji stupac Category u koji se sprema kategorija rada.

Izrada ove baze podataka se obavlja u alatu phpMyAdmin koji dolazi uz XAMPP. Pomoću njega ne moramo pisati duge MySQL naredbe, već imamo grafičko sučelje koje nam olakšava kreiranje i korištenje baze podataka.



Slika 3.2 Izgled sučelja phpMyAdmin alata

3.2. Izrada web aplikacije i PHP skripte

3.2.1. Registracija i prijava korisnika

Registracija korisnika se obavlja na jednostavan način. Nakon što se unesu korisnički podaci, potrebno je samo te podatke ubaciti u bazu podataka.

```
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);
$sql = "INSERT INTO `users` (`UserId`, `UserName`, `UserPassword`) VALUES (NULL,
'$username', '$password')";
    if(mysqli_query($con, $sql)){
            $_SESSION['login'] = false;
            $_SESSION['register'] = true;
            $_SESSION['userName'] = $username;
            header("Location:login.php");
    } else{
     echo "ERROR: Could not able to execute $sql. " . mysqli_error($con);
    }
```

Slika 3.3 Kod za registraciju korisnika

Prva stvar koja je potrebna je spremanje vrijednosti koje je korisnik unio u varijable. Korisničke vrijednosti se nalaze u $_POST varijabli te ih pomoću te varijable možemo spremiti u neke druge.

Bitno je primijetiti da se za lozinku koristi PHP funkcija password_hash() koja kriptira korisničku lozinku u tekst dužine 255 znakova.

Nakon spremanja unesenih vrijednosti, te vrijednosti spremamo u bazu podataka pomoću SQL naredbe INSERT. Ukoliko je upit na bazu uspješan, podešavaju se varijable sesije koje nam omogućavaju da u svakom trenutku znamo da li je neki korisnik prijavljen te koje je njegovo korisničko ime.

```
if(isset($_SESSION['login']) && $_SESSION['login']==true){
              $bLoggedIn = true;
       }
elseif(isset($_POST['login'])){
        $username = $_POST['username'];
        $password = $_POST['password'];
        $q="SELECT userId, userName, userPassword FROM users
WHERE(userName='$username')";
        $res = mysqli_query($con, $q);
        while($row=mysqli_fetch_array($res)) {
              if (password_verify($password, $row['userPassword'])) {
              $_SESSION['login']=true;
              $_SESSION['userId'] = $row['userId'];
              $bLoggedIn = true;
              header("Location:index.php");
              }
              else
               echo "<label>Unešeni su krivi podaci</label>";
              }
       }
elseif(isset($_SESSION['register']) && $_SESSION['register'] == true){
            $_SESSION['login']=true;
            $q="SELECT userId FROM users WHERE(userName='{$_SESSION['userName']}')";
            $res = mysqli_query($con, $q);
            while($row=mysqli_fetch_array($res)){
            $_SESSION['userId'] = $row['userId'];
            }
            $bLoggedIn = true;
            header("Location:index.php");
      }
```

Slika 3.4 Kod za prijavu korisnika

Prijava korisnika se odvija na sličan način, samo što ovaj put umjesto umetanja podataka u bazu pomoću naredbe INSERT, koristimo naredbu SELECT koja nam omogućuje uzimanje podataka iz baze. Tako pomoću te naredbe provjeravamo da li postoji korisnik s takvim korisničkim podacima te ukoliko postoji, postavimo varijable sesije na odgovarajuće vrijednosti. Nakon toga, korisnika vraćamo na početnu stranicu.

3.2.2. Pretraživanje radova

Pretraživati radove možemo pomoću njegova imena i/ili njegove kategorije. Oba podatka su spremljena u bazu podataka pa to možemo izvesti pomoću SQL naredbe SELECT. U ovom radu, odabrali smo da je pretraživanje po imenu opcionalno, dok je pretraživanje po kategoriji obavezno. Razlog tome je što je u nekim slučajevima teško znati ime rada koje nas zanima, ali smo sigurni za kategoriju rada. Također, ukoliko su oba polja opcionalna te se u pretraživanju ostave prazna, rezultat tog pretraživanja bi bili svi radovi. Trenutno, svi radovi se mogu naći na naslovnoj stranici tako da je tu pokrivena i takva vrsta pretraživanja.

```
<form id="search_form" action="" method="post">
       <input type="search" class="form-control" name="project_name" placeholder="Ime
       projekta" /> <br>
       <label>Teme</label> <br>
       Robotika <input type="radio" value="Robotika" name="project_category" />
       Elektronika <input type="radio" value="Elektronika" name="project_category"/>
       Elektrotehnika <input type="radio" value="Elektrotehnika"
       name="project_category"/>
       Programiranje <input type="radio" value="Programiranje"
       name="project_category"/>
       Ostalo <input type="radio" value="Ostalo" name="project_category" checked/>
            <br/> <br />
       <input type="submit" value="Pretrazi" name="submit" class="btn btn-primary" />
</form>
```

Slika 3.5 Forma za pretraživanje radova

Kako smo odredili da je pretraživanje po kategoriji obavezno, za tip unosa kategorije odabran je izborni gumb (eng. *radio button*) te smo jedan od njih („Ostalo“) postavili kao označen.

```
if(isset($_POST["submit"])){
       $project_name = $_POST['project_name'];
       $project_category = ucfirst($_POST['project_category']);
       if($_POST["project_name"] !=""){
              $res=mysqli_query($con,"SELECT rad.Id, rad.Name, rad.Text,
              rad.Date,rad.Category, users.UserId, users.UserName FROM rad,users WHERE
              rad.UserId = users.UserId AND rad.Name LIKE '$project_name' AND
              rad.Category = '$project_category' ORDER BY Id DESC");

              while($row=mysqli_fetch_array($res)){
                     echo "<h1><a
                     href=\"viewpost.php?id={$row['Id']}\">{$row['Name']}</a></h1>";
                      echo "<a
                     href=\"user.php?id={$row['UserId']}\">{$row['UserName']}</a>";
                     echo "<p style=\"font-size:80%\">{$row['Category']}
                     {$row['Date']}</p>";
                     echo "<p>{$row['Text']}</p>";
              }
       }
```

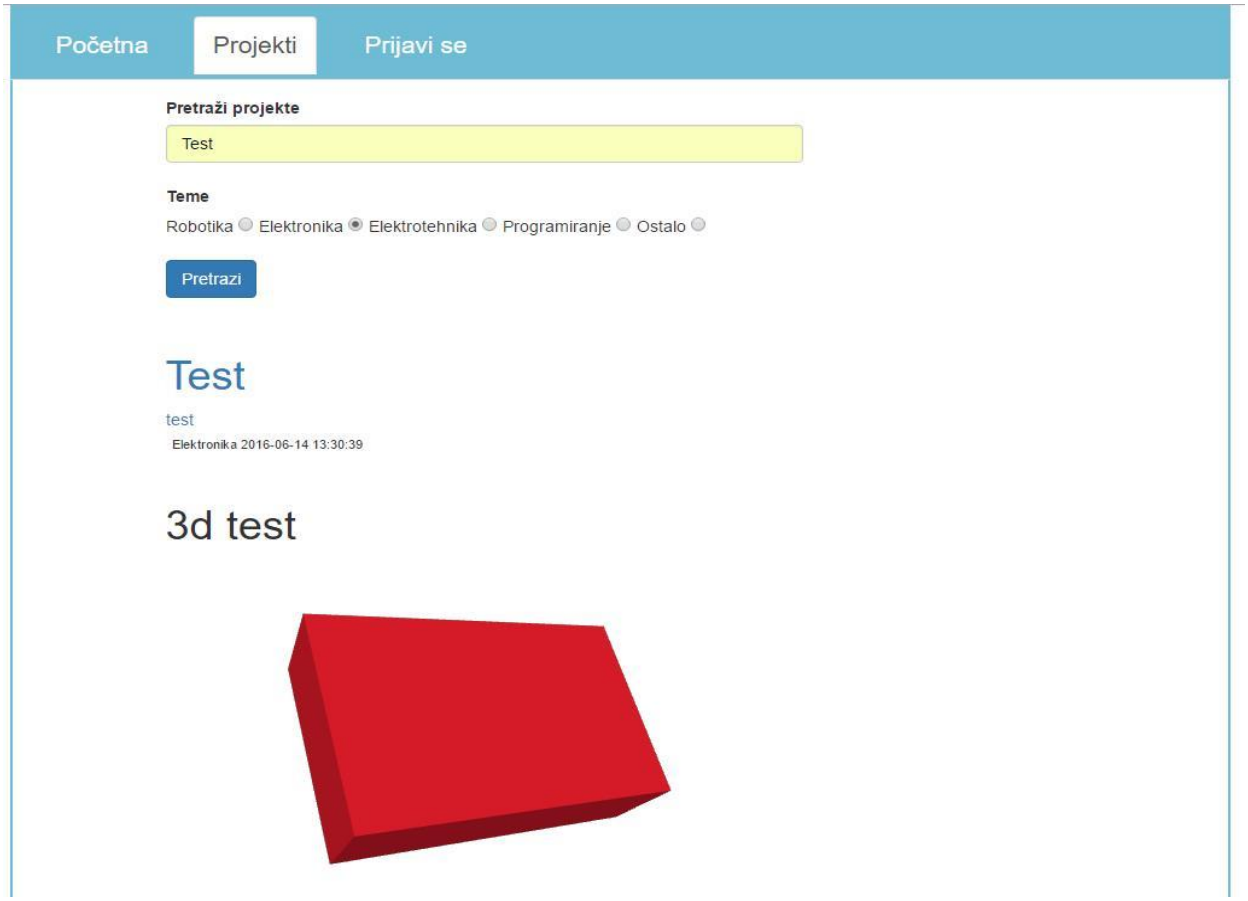
```
        else{
              $res=mysqli_query($con,"SELECT rad.Id, rad.Name, rad.Text,
              rad.Date,rad.Category, users.UserId, users.UserName FROM rad,users WHERE
              rad.UserId = users.UserId AND rad.Category = '$project_category' ORDER BY
              Id DESC");

              while($row=mysqli_fetch_array($res)){
                     echo "<h1><a
                     href=\"viewpost.php?id={$row['Id']}\">{$row['Name']}</a></h1>";
                     echo "<a
                     href=\"user.php?id={$row['UserId']}\">{$row['UserName']}</a>";
                     echo "<p style=\"font-size:80%\">{$row['Category']}
                    {$row['Date']}</p>";
                    echo "<p>{$row['Text']}</p>";
               }
        }

}
```

Slika 3.6 Kod za pretraživanje projekta

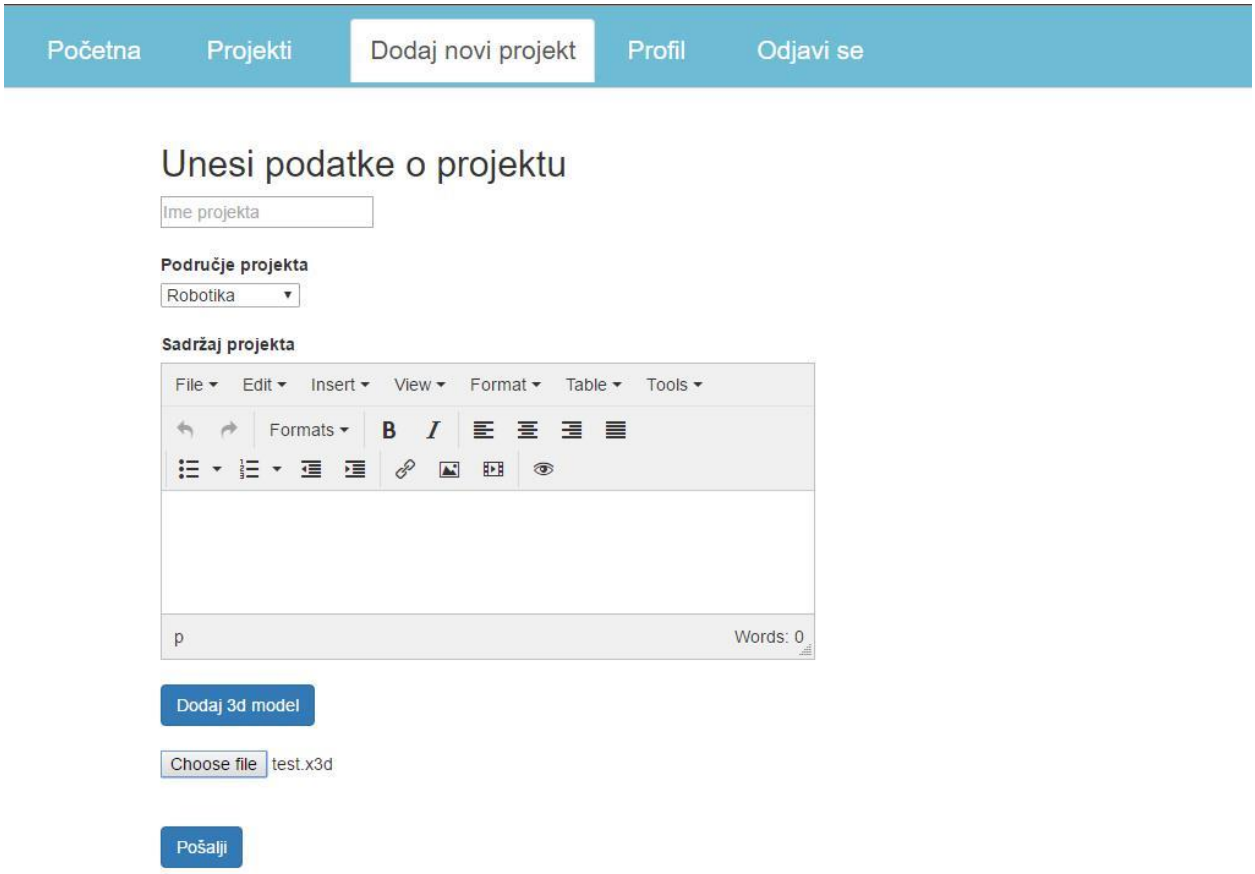
U pretraživanju prvo provjeravamo da li je unesena vrijednost za ime rada jer će nam se SQL upit razlikovati u tom slučaju. Nakon toga, pišemo naš SQL upit u ovisnosti o unesenim podacima. Jedina razlika između ta dva upita je što kod slučaja gdje korisnik pretražuje i po imenu imamo dodatni uvjet (rad.Name LIKE $project_name) u SQL upitu.



Slika 3.7 Primjer rezultata pretraživanja projekta

Nakon odrađenog upita, ispisujemo sve podatke svih projekata koji odgovaraju zadanim kriterijima. Ime projekta i ime korisnika koji je objavio rad su poveznice koje vode na stranicu o tom projektu ili o korisniku gdje se mogu vidjeti svi njegovi radovi. Na slici 3.7. možemo vidjeti primjer rezultata jednog takvog pretraživanja.

3.2.3. Dodavanje novih radova



Slika 3.8 Izgled stranice za dodavanje novog projekta

Dodavanje novih radova je najvažniji dio ovoga rada. S obzirom da korisnik u svom radu ima mogućnost korištenja više različitih multimedijalnih sadržaja, za unos sadržaja projekta koristimo TinyMCE [4] uređivač teksta. Pomoću tog uređivača korisnik ima mogućnost korištenja različitih veličina teksta, naslova, dodavanje slika, videa i tablica te još mnoštvo alata za formatiranje teksta. TinyMCE radi tako da cjelokupni sadržaj koji je napisan u njemu pretvori u HTML format što nama uvelike olakšava prikaz sadržaja rada u aplikaciji. Kako bi bili u mogućnosti koristi TinyMCE uređivač teksta, u našu datoteku moramo uključiti JavaScript skriptu koja se nalazi na njihovoj stranici [5]. Osim uključivanje skripte, možemo postaviti dodatne mogućnosti za izgled i funkcionalnost uređivača. Kompletan dio koda vezan za TinyMCE uređivač može se vidjeti na

slici 3.9. No osim osnovnih multimedijalnih sadržaja, korisnik je u mogućnosti i podijeliti svoje 3D modele. 3D modeli mogu biti jako korisni u radovima gdje se nešto izrađuje, poput robotike.

```
<script src="//cdn.tinymce.com/4/tinymce.min.js"></script>
<script>
        tinymce.init({ selector:'textarea',
        plugins: [
              'advlist autolink link image lists charmap print preview hr anchor
              pagebreak spellchecker',
              'searchreplace wordcount visualblocks visualchars code fullscreen
              insertdatetime media nonbreaking',
              'save table contextmenu directionality emoticons template paste
              textcolor'
        ],
       toolbar: 'insertfile undo redo | styleselect | bold italic | alignleft
       aligncenter alignright alignjustify | bullist numlist outdent indent | link
       image media fullpage | preview ',
        resize: 'both'
    });
</script>
```

Slika 3.9 Uključivanje TinyMCE uređivača u datoteku

Nakon klika na gumb „Dodaj 3D model“, na stranici se prikazuje polje za unos datoteke. To se izvršava pomoću jQuery skripte. U toj skripti je potrebno definirati tip podatka koji se unosi, kao i koji su formati prihvatljivi. U našem slučaju, prihvaća se samo .x3d format koji je najrašireniji format za 3D modele. Korisnik, ukoliko poželi, može dodati više 3D modela u svoj rad te nije ograničen njihovim brojem.

```
<script>

        $(document).on("click", "button", function(){
              var $modelText = "<input type=\"file\" class=\"model\" name=\"files[]\"
              accept=\".x3d\"><br><br>";

              $(".addNewInputButton").off('click').on('click', function() {
                    $($modelText).insertBefore( ".submitBtn" );
              });
        });
</script>
```

Slika 3.10 Dodavanje unosa za 3D model

Nakon što je korisnik završio sa pisanjem svojeg rada i pritisnuo na gumb „Pošalji“, sadržaj njegovog unosa se prenosi u varijable $_POST (naslov, kategorija i sadržaj) i $_FILES (3D modeli). Prije unosa u bazu podatka, datoteke se moraju prenijeti te im se treba dodijeliti odgovarajuće ime. Također, u sadržaj rada se mora dodati kod koji služi za prikazivanje 3D modela u HTML formatu. Za prikazivanje 3D modela, osim spomenutog HTML koda, potrebna nam je i JavaScript skripta koja se može pronaći na [6]. Nju moramo uključiti u svaku stranicu koja će prikazivati 3D modele. Kod za dodavanje radova u bazu podataka može se vidjeti na slici 3.11.

```
  //Postavljanje direktorija, dodijeljivanje varijabli
    define ("SITE_ROOT", realpath(dirname(__FILE__)));

    function recursive_escape(&$value) {
            if (is_array($value))
                array_map('recursive_escape', $value);
            else
                $value = mysql_real_escape_string($value);
        }
    array_map('recursive_escape', $_POST);

    $project_name = $_POST['project_name'];
    $project_user = $_SESSION['userId'];
    $project_category = ucfirst($_POST['project_category']);
    $project_text = $_POST['project_text'];

    //Upload datoteka i dodijeljivanje imena
    $uploads_dir = "/uploads/";
    foreach ($_FILES["files"]["error"] as $key => $error) {
        if ($error == UPLOAD_ERR_OK) {
              $tmp_name = $_FILES["files"]["tmp_name"][$key];
              $name = $_FILES["files"]["name"][$key];
              move_uploaded_file($tmp_name, SITE_ROOT . $uploads_dir .
              preg_replace('/\s+/', '', $project_name) . "_" . $name);

              $item = '<div class="x3d">
                        <x3d width="600px" height="400px">
                            <scene>
                                  <Viewpoint position="0.07254 0.07220 0.02757"
                                  orientation="-0.86638 0.42982
                                  -0.25425 1.39025" zNear="0.04351" zFar="0.20633"
                                  description=""></Viewpoint>
                               <Inline nameSpaceName="Deer" mapDEFToID="true" url="'.
                               $uploads_dir . preg_replace('/\s+/', '',
                               $project_name) . "_" . $name .'" />
                            </scene>
                        </x3d>
                    </div>';
            $project_text = $project_text . $item;
        }
        else
            echo $error;
    }
    // Dodavanje rada u bazu podataka
    $sql = "INSERT INTO `rad` (`Id`, `Name`, `Text`, `UserId`, `Date`, `Category`)
VALUES (NULL, '$project_name', '$project_text', '$project_user',
CURRENT_TIMESTAMP,'$project_category')";
    if(mysqli_query($con, $sql)){
        echo "Records added successfully.";
    } else{
     echo "ERROR: Could not able to execute $sql. " . mysqli_error($con);
    }
```

Slika 3.11 Kod za dodavanje radova u bazu podataka

4. ZAKLJUČAK

U ovom radu je izrađena web aplikacija za objavljivanje studentskih radova. Ovdje su prezentirane najvažnije mogućnosti aplikacije: registracija i prijava, pretraživanje te dodavanje radova. Prednost ovoga rada je olakšavanje studentima objavljivanje njihovih radova i priručnika. Također, drugim studentima pruža centraliziranu aplikaciju gdje će moći pronaći mnoštvo studentskih radova iz područja kojih ih zanimaju. Naravno, aplikacija se može dodatno urediti i poboljšati. Moguće je pružiti dodatne mogućnosti kod pretraživanja kao i poboljšati cjelokupan izgled aplikacije kako bi bila što intuitivnija za korištenje.

LITERATURA

[1] PHP Manual, General Information, http://php.net/manual/en/faq.general.php, lipanj 2016.

[2] Usage statistics and market share of PHP for websites, https://w3techs.com/technologies/details/pl-php/all/all, lipanj 2016

[3] Usage of JavaScript libraries for websites, https://w3techs.com/technologies/overview/javascript_library/all, lipanj 2016

[4] TinyMCE, https://www.tinymce.com/, lipanj 2016

[5] TinyMCE JavaScript skripta, https://cdn.tinymce.com/4/tinymce.min.js, lipanj 2016

[6] JavaScript skripta za prikazivanje 3D modela, http://www.x3dom.org/download/x3dom.js, lipanj 2016

SAŽETAK

Naslov: Web aplikacija za studentske radove

U ovom radu izrađena je web aplikacija za objavljivanje studentskih radova. Pomoću nje, napravljen je sustav koji omogućuje studentima pregled radova iz različitih kategorija poput robotike i elektronike. Sama aplikacija je napisana u programskom jeziku PHP, a za prikaz se koriste HTML i CSS kao i JavaScript. U web aplikaciji, korisniku je omogućena registracija i prijava te pretraživanje i dodavanje novih radova. Kod dodavanja radova, korisnik se može poslužiti različitim formatima poput teksta, tablica, slika, videa i 3D modela. Nakon opisanih tehnologija, u radu je dan prikaz rješavanja najvažnijih mogućnosti web aplikacije.

Ključne riječi: web aplikacija, studentski radovi, PHP, HTML5

ABSTRACT

Title: Web application for student projects

Web application was developed in this final paper. With it, students are able to view projects from different categories like robotics and electronics. Application itself was written in PHP programming language and HTML, CSS and JavaScript are used for displaying web page. Users are able to register, login, search projects and add new projects. When adding new projects, users may use different formats like text, tables, pictures, videos and 3D models. After short review about used technologies, most important features of application are explained.

Keywords: web application, student projects, PHP, HTML5

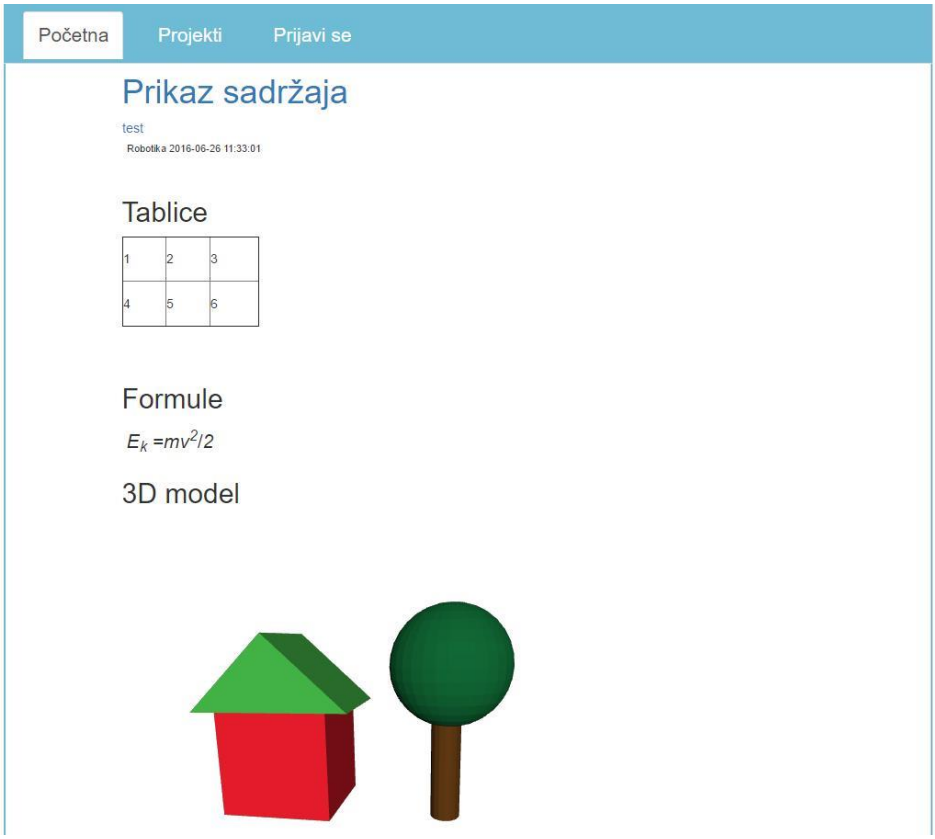
ŽIVOTOPIS

Armando Vlajčić rođen je 8. veljače 1995. godine u Osijeku. 2009. godine upisuje III. Gimnaziju u Osijeku. Tamo sudjeluje na dva županijska natjecanja iz programiranje u kategoriji Algoritmi. Osim toga, bio je član tima koji je dva puta izborio pravo nastupa na međunarodnom natjecanju ACSL. 2013. godine završava srednju školu te upisuje preddiplomski studiji računarstva na Elektrotehničkom fakultetu u Osijeku.

PRILOZI

Cjelokupan kod se može pronaći na DVD-u priloženom uz ovaj rad.

Prilog A. Izgled završene aplikacije

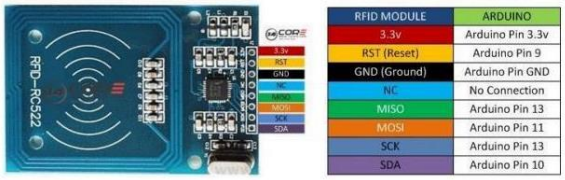




1.1. Elementi kostrukcijskih vježbi



SI.1. Mini USB Nano V3.0 CH340G Mikrokontroler za Arduino



RFID MODULE	ARDUINO
3.3v	Arduino Pin 3.3v
RST (Reset)	Arduino Pin 9
GND (Ground)	Arduino Pin GND
NC	No Connection
MISO	Arduino Pin 13
MOSI	Arduino Pin 11
SCK	Arduino Pin 13
SDA	Arduino Pin 10

SI.2. MF522-AN modul RC522 RFID senzor i potrebni pinovi.

