Izrada web aplikacije za pomoć u odabiru šminke

Čačija, Ana

Undergraduate thesis / Završni rad

2019

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:200:507351

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-08**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij računarstva

IZRADA WEB APLIKACIJE ZA POMOĆ U ODABIRU ŠMINKE

Završni rad

Ana Čačija

Osijek, 2019.

SADRŽAJ

1. UVOD.....1

1.1. Zadatak završnog rada.....1

2. PRIMIJENJENE TEHNOLOGIJE.....2

2.1. HTML.....2

2.2. CSS.....3

2.3. PHP.....3

2.4. Bootstrap.....4

2.5. jQuery.....4

2.6. MySQL.....5

3. REALIZACIJA WEB APLIKACIJE.....7

3.1. Struktura projekta.....7

3.1. Baza podataka.....8

3.2. Registracija i prijava korisnika.....8

3.3. Generiranje podataka za trenutnog korisnika.....12

3.4. Izgled sučelja.....15

4. ZAKLJUČAK.....21

LITERATURA.....22

SAŽETAK.....23

ABSTRACT.....24

ŽIVOTOPIS.....25

1. UVOD

Cilj završnog rada ogleda se u izradi web aplikacije za pomoć pri odabiru šminke. Temelji se na ideji da zainteresiranim korisnicima omogući lakše snalaženje prilikom izbora određene vrste kozmetike na osnovi njihove strukture lica. Aplikacija sadrži registraciju i prijavu korisnika, ostvarenih programskim jezikom PHP, dok su za izgled korišteni Bootstrap, HTML i CSS. Također, sadrži pregled fotografija i tehnika potrebnih za ostvarenje najboljeg odabira šminke za pojedino svojstvo lica korisnika. Zamišljeni pregled prethodno spomenutih dijelova ostvaruje se pomoću jQuery-a. U drugom poglavlju, detaljnije su opisane spomenute tehnologije, kao i njihova važnost u izvršenju početne ideje. Treće poglavlje, realizacija web aplikacije, sadržava rad s bazom podataka, registraciju i prijavu u sustav, te funkcije koje osiguravaju pravilan prikaz podataka trenutnom korisniku. U ovom poglavlju je, također, predočen krajnji izgled sučelja web aplikacije.

1.1. Zadatak završnog rada

Zadatak završnog rada uključuje izradu web aplikacije koja treba omogućiti lakše snalaženje prilikom odabira kozmetike. Navedeno je potrebno ostvariti koristeći PHP, HTML, CSS, MySQL, i slične web-tehnologije. Web aplikacija se sastoji od više zasebnih stranica kako bi se omogućila preglednost, a ujedno i ostvarile zadane funkcionalnosti.

2. PRIMIJENJENE TEHNOLOGIJE

Kako bi se izradila što preglednija i funkcionalnija web aplikacija, potrebno je koristiti se tehnologijama za razvoj web aplikacija koje će to osigurati. Spomenute tehnologije predstavljaju opisne jezike i pakete koje računala koriste za komunikaciju.

2.1. HTML

HTML (engl. *HyperText Markup Language*) opisni je jezik korišten za kreiranje sadržaja web stranice, kao i njegovo vizualno predstavljanje. Uzimajući u obzir jednostavnost i dostupnost ovog opisnog jezika, vidljivo je zbog čega je jedan od najraširenijih i najkorištenijih. Prema [1], hipertekst označuje tekstualnu strukturu sastavljenu od međusobno povezanih jedinica informacije, a prikazanu na elektroničkom uređaju. Temeljna zadaća HTML jezika je uputiti web preglednik u način prikazivanja hipertekst dokumenta, pri čemu je vrlo važno povesti računa o tome da izgleda jednako na svim web preglednicima, računalima i operacijskim sustavima.[2] Važno je spomenuti da HTML nije programski jezik te se, koristeći ga, ne može izvršiti nikakva zadaća. Služi samo za opis hipertekstualnih dokumenata, koristeći oznake (engl. *tags*) za opis sadržaja.[3] Spomenute oznake se, u većini slučajeva, pojavljuju u paru, gdje prva oznaka predstavlja početak, a druga završetak. Prilikom kreiranja HTML dokumenta, na početku se treba postaviti <!DOCTYPE> element, čime se definira inačica standarda korištenog pri izradi dokumenta. Nadalje, <html> element predstavlja početak dokumenta. Sastoji se od <head> i <body> elemenata, od kojih <head> služi za zaglavlje u kojem se definiraju naslov, stilska obilježja stranice, skripte kreirane u JavaScript jeziku, i sl. Element <body> sadrži glavni sadržaj HTML dokumenta. Također, postoji mogućnost pisanja komentara koja se može ostvariti na bilo kojem mjestu unutar dokumenta, a sam tekst komentara neće biti prikazan na stranici. Još jedna mogućnost HTML dokumenata je korištenje meta oznaka. One sadrže ključne riječi, opis, informacije o vlasništvu, i sl. Točnije, sadrže podatke koje tražilice upotrebljavaju kako bi zapisale informacije o stranici.

Godine 1993. objavljena je prva verzija HTML jezika, a razvoj se nastavlja objavljivanjem verzija 2.0 te 3.0. HTML4 predstavljen je 1997. godine, a HTML5 donosi brojne nove mogućnosti koje prijašnje verzije nisu imale, kao što su mogućnost upravljanja pomoću tipkovnice, brojne nove elemente, i dr.

2.2. CSS

CSS (engl. *Cascading Style Sheets*) je stilski jezik, korišten za oblikovanje web stranica. Predstavlja osnovnu tehnologiju na kojoj se temelje današnje web aplikacije. Glavna ideja CSS-a je odvajanje prezentacijskog koda u zasebne datoteke, te njegovo definiranje pomoću jednostavnih pravila koja mogu biti primijenjena na sve elemente, ili samo na određene.[4] Prije pojave CSS-a, definiranje se odvijalo unutar HTML-a, čime je dolazilo do problema miješanja sadržaja i strukture s kodom čija je jedina svrha bila prezentacija. Drugi problem bio je taj što se HTML kod za definiranje izgleda trebao iznova postavljati na svakom elementu, i svakoj stranici.[5] Danas, CSS se obično piše u zasebnoj datoteci, a onda povezuje s HTML dokumentom, što se čini koristeći HTML element *link*. Svako CSS pravilo započinje selektorom, koji određuje za koji HTML element to pravilo vrijedi.

```
html{
  background-color: white;
}
```

Sl.2.1. *Primjer korištenja CSS selektora*

Prva verzija CSS-a definirana je krajem 1996. godine. Trenutačna verzija je CSS3, koja donosi brojne novitete i poboljšanja, kao što je npr. mogućnost prilagodbe stranice (ili elemenata) prema razlučivosti ekrana. Uzimajući u obzir korištenje mobilnih uređaja kao primarni način pristupa Internetu u današnje vrijeme, ova mogućnost olakšava prilagodbu dokumenta mobilnim, ali i ostalim uređajima. [6], [7]

2.3. PHP

Skriptni jezik PHP (engl. *PHP: Hypertext Preprocessor*) namijenjen je programiranju dinamičkih web stranica. Sadrži podršku raznih baza podataka, internet protokola te raspolaže brojnim programskim knjižnicama. Zbog toga je jedna od najraširenijih poslužiteljskih skriptnih tehnologija, dok broj web stranica na kojima se koristi svakodnevno raste. Izvršava se na poslužitelju, čime se razlikuje od klijentskih skriptnih jezika. To znači da se može upotrebljavati za dinamičko generiranje sadržaja stranice, slanje i primanje kolačića (engl. *cookie*), prikupljanje podataka iz obrazaca, i sl. PHP može biti korišten na svim glavnim operacijskim sustavima, uključujući Microsoft Windows, Linux, mnoge inačice Unix-a, macOS, i slično. Najznačajnija mogućnost PHP-a je podrška velike količine baza podataka, a uz to podržava komunikaciju s ostalim servisima koristeći protokole kao što su HTTP, IMAP, COM, i dr.

PHP datoteke sadrže nastavak „.php“, a svaka skripta počinje korištenjem oznake <?php i završava s oznakom ?>. Naredbe završavaju znakom „ ; “, a sve korištene varijable ispred sebe moraju sadržavati znak „$“ kao što je prikazano na slici 2.1.

```php
<?php
$example = "Hello world!";
echo $example;
?>
```

Sl. 2.1. *Jednostavan primjer korištenja PHP varijable*

Prilikom izrade ove web aplikacije, PHP se koristio za postavljanje podataka u MySQL bazu podataka i dohvaćanje istih iz baze.

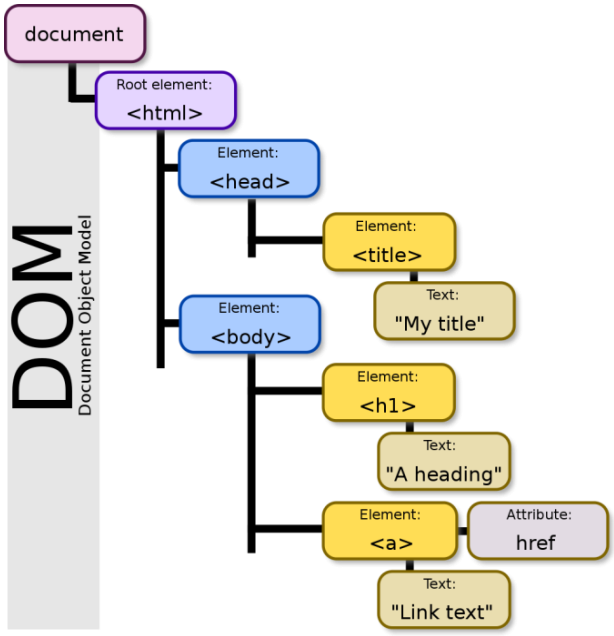
2.4. Bootstrap

Bootstrap je besplatno CSS okruženje (engl. *framework*) namijenjeno razvoju Internet stranica. Sadrži predloške za obrasce, dugmad (engl. *button*), navigaciju, itd. temeljene na CSS-u i JavaScript-u. Prvenstvena svrha dodavanja Bootstrapa u Internet projekt je primjena njegovog izbora stilova za taj projekt. Nakon što je dodan u projekt, pruža osnovne stilske definicije svim HTML elementima. CSS klase definirane Bootstrapom mogu se dalje mijenjati i prilagođavati sadržaju. Svaki njegov sastavni dio sadržava HTML strukturu, CSS deklaracije, a u nekim slučajevima i JavaScript kod. Nadalje, podržava i potiče izradu prilagodljivih (engl. *responsive*) web stranica korištenjem strukture rešetke (engl. *grid*), iz čega proizlazi da se vrlo lako može prilagođavati ekranima raznih dimenzija, što je vrlo bitna karakteristika prilikom izrade web aplikacije.

Nakon određenih problema prilikom dosljednosti u izgledu sučelja, zaposlenici Twitter-a razvili su Bootstrap kako bi se osigurala ujednačenost i bolja preglednost. Bootstrap 2 objavljen je 2012. godine, sadržavajući nekoliko novih dijelova, kao i promjene brojnih postojećih dijelova. Godinu dana nakon toga, Bootstrap 3 upotrijebio je pristup „primarno mobitelima“ (engl. *mobile-first approach*), a Bootstrap 4 donio je značajne promjene kao što su dodavanje prilagodljivih razmaka, povećanje veličine slova, unaprijeđeni stil izbornika, slikovnih klasa, obrazaca, itd. [8]

2.5. jQuery

jQuery predstavlja JavaScript knjižnicu napravljenu kako bi pojednostavila manipulaciju DOM (engl. *Document Object Model*) elementima, kao i rukovanje događajima i CSS animacijama te razvoj AJAX aplikacija. DOM označuje sučelje koje se odnosi prema HTML ili XML dokumentu kao prema strukturi stabla, gdje je svaki čvor objekt koji predstavlja dio dokumenta.



Sl. 2.2. Prikaz DOM strukture

Prema [9], najrasprostranjenija je JavaScript knjižnica, a koristi ga 73% od 10 milijuna najpopularnijih web stranica. jQuery može biti korišten u potrazi za specifičnim elementom u dokumentu, za mijenjanje njegovih atributa ili rukovanje događajima. Ova knjižnica najčešće se koristi kao JavaScript datoteka koja definira sva sučelja, uključujući DOM, događaje (engl. *Events*) i AJAX funkcije. Unutar web stranice uključuje se povezivanjem s mjesnom (engl. *local*) kopijom ili jednom od trenutno dostupnih kopija preko javnih poslužitelja. jQuery objektne metode (engl. *object methods*) uobičajeno vraćaju jQuery objekt, koji omogućuje upotrebu lančanih metoda (engl. *method chains*), tj. izvršavanje više metoda na jednom objektu.

2.6. MySQL

Kada je riječ o bazama podataka, MySQL baze relacijskog tipa jedan su od najčešćih izbora baze za projekte. To je besplatni sustav za upravljanje bazom podataka otvorenog koda (engl. *open source*). Prije samog početka rada s bazom podataka, treba izraditi odgovarajući

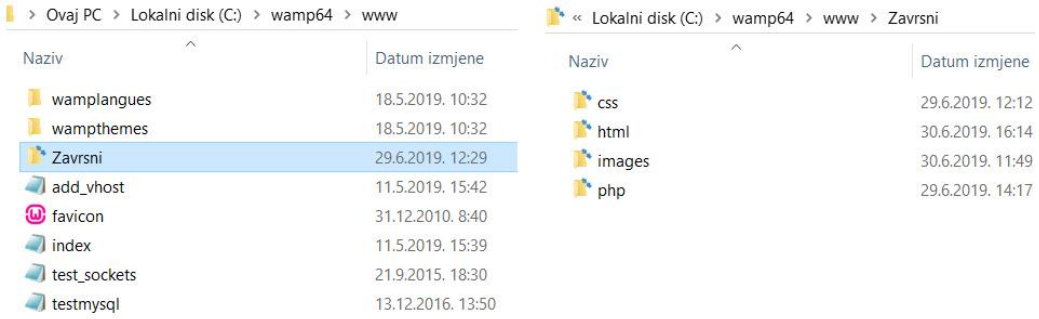
izgled same baze, koji će se kasnije upotrebljavati kao određen broj tablica korištenih za pohranu podataka. Osnovni pojmovi prilikom korištenja baza podataka su entiteti i relacije. Entiteti su stvarni objekti, osobe, i sl., o kojima pohranjujemo informacije, dok relacije predstavljaju poveznice među entitetima. MySQL sustav pohranjuje podatke unutar tablica sastavljenih od redova i kolona, pri čemu se redovi još nazivaju i slogovima.

3. REALIZACIJA WEB APLIKACIJE

Kreirana aplikacija sastoji se od naslovne stranice na kojoj su pojašnjeni svrha aplikacije i način njenog funkcioniranja, odvojenih stranica namijenjenih za registraciju ili prijavu korisnika, odvojenih stranica koje sadrže fotografije i objašnjenja pravilnog odabira šminke uzimajući u obzir podatke korisnika upisane u bazi podataka, te zasebne stranice na kojoj se nalaze izvori preuzetih fotografija. Početna stranica sadrži i odjeljak za istraživanje, namijenjen korisnicima koji nisu registrirani, a prije same registracije žele pogledati o čemu se zapravo radi. U nastavku je detaljnije objašnjen postupak izrade web aplikacije, počevši s bazom podataka.

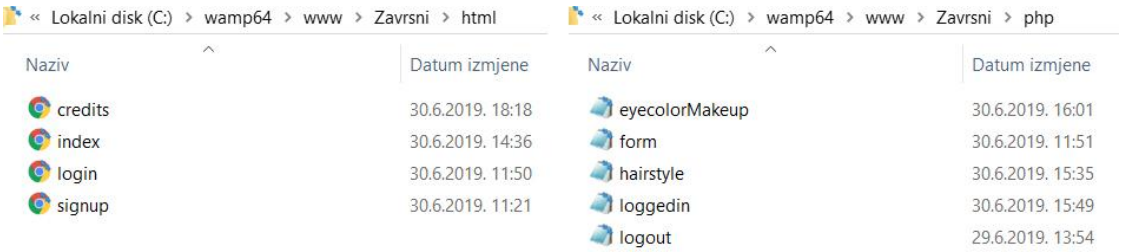
3.1. Struktura projekta

Projekt se sastoji od dokumenata sortiranih u datoteke prema tipu dokumenta. Tako postoje „html“, „css“ i „php“ datoteke, kao i „images“, namijenjena spremanju slika. Sve se one nalaze u datoteci završnog rada prikazanoj slikom 3.1.



Sl.3.1. *Osnovna struktura projekta*

Unutar „html“ datoteke, nalaze se svi dokumenti s nastavkom „.html“. Unutar „php“ datoteke, svi dokumenti nastavka „.php“, i tako dalje. Primjer „html“ i „php“ datoteka prikazan je slikom 3.2.



Sl.3.2. *Prikaz datoteka*

3.1. Baza podataka

S obzirom na to da je ova web aplikacija namijenjena radu s različitim korisnicima, potrebno je njihove podatke spremati u bazu podataka. Prilikom izrade, korišten je alat PHPMyAdmin te MySQL baza podataka, kao i Wamp serverski programski paket verzije 3.1.9. Nakon pokretanja MySQL-a i Apache servera iz Wamp-a, potrebno je pokrenuti phpMyAdmin iz kontrolne ploče Wamp-a ili upisivanjem *localhost/phpmyadmin* u Internet preglednik. Za potrebe aplikacije, izrađena je tablica „users“ koja za primarni ključ koristi korisničko ime (engl. *username*) korisnika. Sadrži i podatke poput e-pošte, zaštićene zaporke, boje i oblika očiju te oblika lica korisnika.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	username	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	2	email	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	password	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	faceshape	varchar(10)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
☐	5	eyeshape	varchar(15)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
☐	6	eyecolor	varchar(6)	latin1_swedish_ci		Yes	*NULL*			Change Drop More

Sl. 3.3. *Prikaz tablice „users“*

Pomoću ove tablice, ostvarit će se registracija i prijava koje su prikazane u nastavku.

3.2. Registracija i prijava korisnika

PHP skriptni jezik, uz ostalo, ima mogućnost spajanja na bazu podataka te tako omogućuje unos, izmjenu i dohvaćanje podataka iz baze. Na samom početku pisanja PHP skripte, potrebno je pozvati funkciju *session_start()*, koja kreira novu ili nastavlja trenutnu sesiju (engl. *session*). Nakon toga, ostvaruje se povezivanje s bazom podataka funkcijom *mysqli_connect(host, username, password, dbname)*, gdje *host* predstavlja adresu servera, *username* korisničko ime MySQL baze, a *password* sukladno tome zaporku korištenu za MySQL bazu. Nadalje, *dbname* predstavlja *database name*, tj. ime baze podataka na koju se spaja.

```
$db = mysqli_connect('localhost', 'root', '', 'users');
```

Sl. 3.4. *Povezivanje s bazom podataka*

Kao što je prethodno spomenuto, host vrijednost u ovom slučaju je „localhost“, jer se koristi mjesni (engl. *local*) pristup poslužitelju. Korisničko ime za pristup prilikom postavljanja zadano

je kao „root“, dok zaporka nije zadana, zbog čega se u dijelu za zaporku ostavlja prazan prostor. Ime baze podataka korištene u ovoj web aplikaciji je „users“.

Prije same registracije korisnika, vrlo je važno provjeriti ispunjenost glavnih uvjeta poput popunjenih polja u formi, podudarnosti s drugim korisnikom, jednakosti zaporke i potvrdne zaporke, i tako dalje.

```
if ($password != $confirmpassword) {
  $message = "The two passwords must be identical.\\n";
   echo ("<script type='text/javascript'>alert('$message');
   location.href='/Zavrsni/html/signup.html'; </script>");
}
```

Sl. 3.5. *Provjera jednog od potrebnih uvjeta*

Kako je vidljivo sa slike 3.5., provjera ispravnosti unesenih podataka odvija se pomoću *if...else* iskaza (engl. *statement*). U konkretnom slučaju, provjerava se je li potvrdna zaporka jednaka prvotno unesenoj nakon ispunjenja obrasca, te u slučaju da nije, korisnik se preusmjerava na odvojenu stranicu gdje treba ispuniti zadani uvjet kako bi se registrirao. Sukladno navedenom, ostatak provjera obavlja se na sličan način. Nakon ispunjenja ovih uvjeta, potrebno je u bazi podataka provjeriti postoji li već korisnik unesenog korisničkog imena, koristeći *SELECT* iskaz kojim se odabire redak u kojem je *username* jednak unesenom, te funkcije *mysqli_query()* i *mysqli_fetch_assoc()*. Ukoliko je ispunjen i ovaj uvjet, zaporka korisnika se zaštićuje korištenjem *md5()* funkcije te se podaci popunjeni u obrascu postavljaju u bazu podataka na način prikazan slikom 3.6.

```
$password1 = md5($password);
$query1 = "INSERT INTO users (username, email, password, faceshape, eyeShape, eyecolor)
          VALUES('$username', '$email', '$password1', '$faceShape', '$eyeShape', '$eyeColor')";
mysqli_query($db, $query1);
$_SESSION['username'] = $username;
$_SESSION['faceShape'] = $faceShape;
$_SESSION['eyeShape'] = $eyeShape;
$_SESSION['eyeColor'] = $eyeColor;
$_SESSION['success']="You are now registered.";
$message = "Thank you for your registration.\\n";
echo ("<script type='text/javascript'>alert('$message');
location.href='/Zavrsni/php/loggedin.php'; </script>");
```

Sl. 3.6. *Registracija korisnika*

Ova registracija odvija se popunjavanjem obrasca dokumenta „signup.html“, a poziva se putem „form.php“ dokumenta. Dio programskog koda obrasca prikazan je u nastavku.

```
<div class="signup-form">
    <link rel="stylesheet" href="/Zavrsni/css/signup.css">
    <form action="/Zavrsni/php/form.php" method="post">
        <h2>Sign Up</h2>
        <p>Fill this form to create an account!</p>
        <hr>
        <div class="form-group">
            <div class="input-group">
                <span class="input-group-addon">
                    <i class="fa fa-user"></i>
                </span>
                <input type="text" class="form-control" name="username"
                        placeholder="Username" required="required">
            </div>
        </div>
```

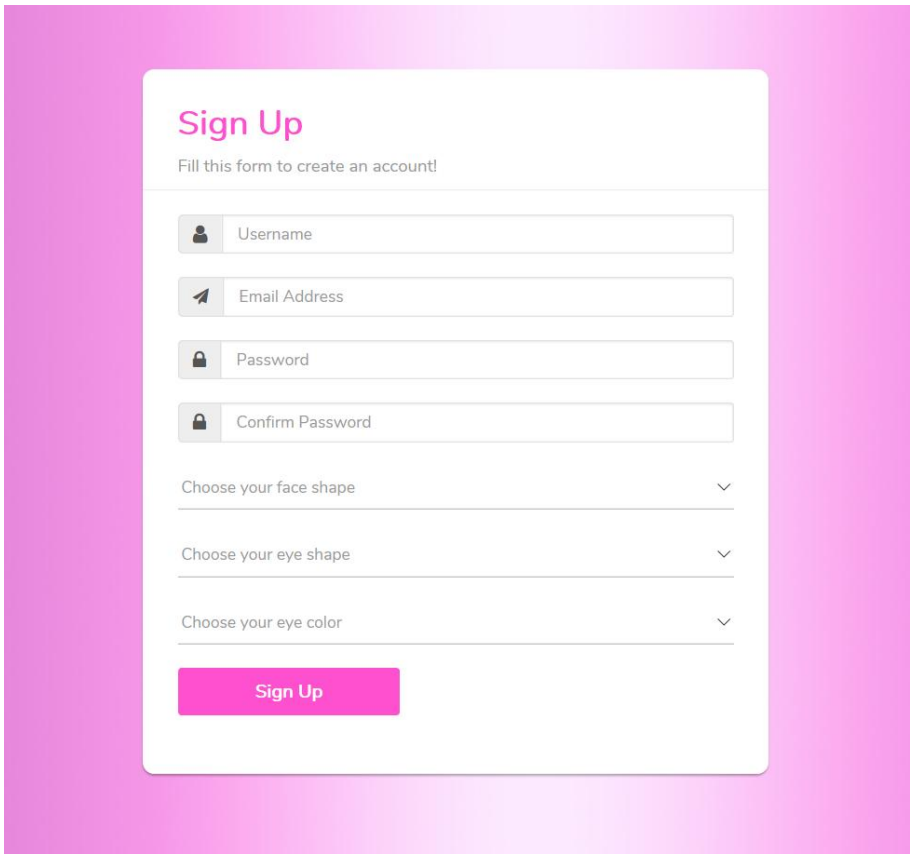
Sl. 3.7. *Programski kod obrasca za registraciju korisnika*

Na sličan način, napravljena su i ostala polja obrasca, koja sadrže drugačije nazive i tipove, npr. ukoliko je riječ o polju za upis zaporke, koristit će se *„type="password""*. Kao što je vidljivo iz slike 3.7., obrazac je povezan s odvojenim CSS dijelom korištenim za stiliziranje izgleda polja i ostalih njegovih elemenata.

```
.signup-form {
    width: 35%;
    margin: 0 auto;
    padding: 30px 0;
}
.signup-form form {
    margin-top: 5%;
    color: #999;
    border-radius: 10px;
    margin-bottom: 15px;
    background: #fff;
    box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
    padding: 30px;
}
.signup-form .input-group-addon {
    max-width: 42px;
    text-align: center;
}
```

Sl. 3.8. *Dio programskog koda dokumenta „signup.css"*

CSS se temelji na ciljanju pojedinih elemenata obrasca, kako bi se osigurao različit izgled njegovih dijelova. U predočenom programskom kodu, stilizira se izgled obrasca općenito, a ostatak CSS koda napisan je na takav način uključujući različita svojstva za pojedine elemente. Prikaz obrasca predočen je slikom 3.9. u nastavku.



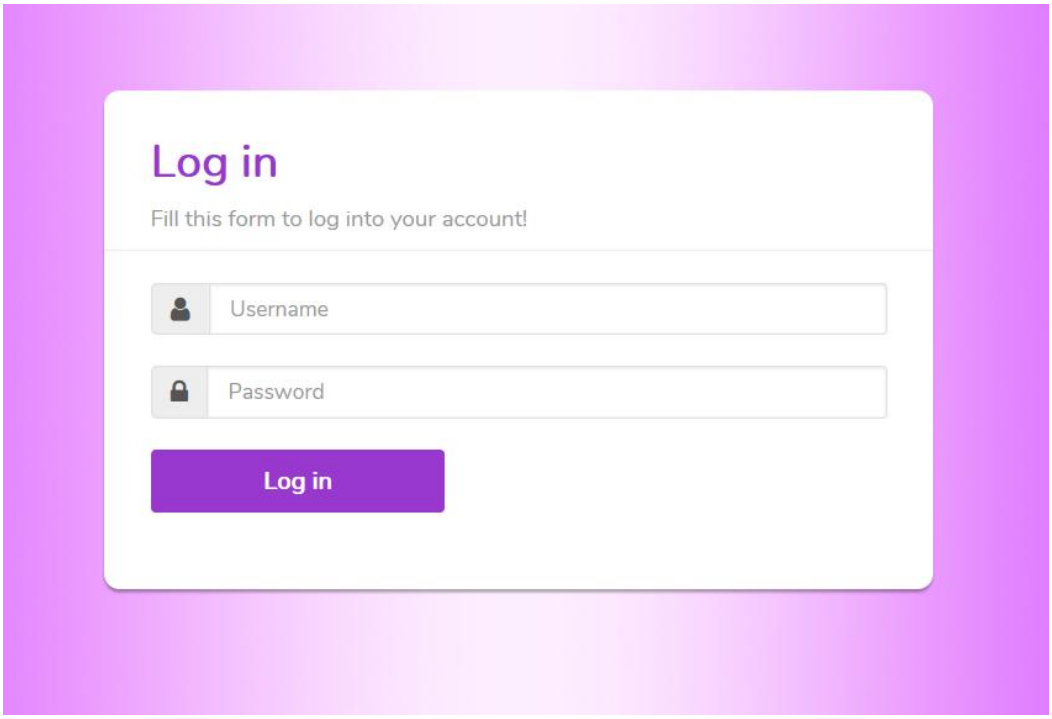
Sl. 3.9. *Izgled obrasca za registraciju korisnika*

Kao i prilikom registracije korisnika u sustav, prije same prijave potrebno je PHP-om provjeriti jesu li ispunjena polja koja su potrebna za obavljanje prijave. Ta polja uključuju korisničko ime i zaporku, koji su nužni za raspoznavanje i razlikovanje korisnika. Nadalje, potrebno je provjeriti za kojeg korisnika se korisničko ime i zaporka podudaraju, te onda tog korisnika prijaviti i preusmjeriti na drugu stranicu (u ovom slučaju, to je „loggedin.php“). Slikom 3.10. prikazan je dio PHP koda koji je korišten za ostvarivanje prijave korisnika u sustav.

```php
$password1 = md5($password);
$query = "SELECT * FROM users WHERE username='$username'";
$result = mysqli_query($db, $query);
$row=mysqli_fetch_array($result);
if ($row['password']==$password1) {
  $_SESSION['username'] = $row['username'];
  $_SESSION['faceShape'] = $row['faceshape'];
  $_SESSION['eyeShape'] = $row['eyeshape'];
  $_SESSION['eyeColor'] = $row['eyecolor'];
  $message = "You are now logged in.\\n";
  echo ("<script type='text/javascript'>alert('$message');
  location.href='/Zavrsni/php/loggedin.php'; </script>");
```

Sl. 3.10. *PHP kod za prijavu korisnika*

Ukoliko se jave poteškoće prilikom pronalaska korisnika podudarajućih korisničkog imena i zaporke, korisnika se, tada, o tome obavještava slično kao sa slike 3.10., a zatim je potrebno preusmjeriti ga na početnu stranicu „index.html“. Slikom 3.11. prikazan je izgled obrasca prijave korisnika.



Sl. 3.11. *Izgled obrasca za prijavu korisnika*

3.3. Generiranje podataka za trenutnog korisnika

Ovaj dio web aplikacije zamišljen je na način da se pomoću jQuery-a te PHP-a, prilikom učitavanja stranice, određeni dijelovi programskog koda prikazuju korisniku, dok ostali dijelovi ostaju skriveni. U tu svrhu, koristan je bio *switch...case* blok korištena u PHP-u kako bi se provjerili podaci korisnika u bazi podataka. Unutar PHP-a, prisutan je jQuery programski kod. Prije početka uporabe jQuery-a, ova knjižnica prvo se mora uključiti u dokument unutar <head> dijela dokumenta. Ukoliko se ovaj korak preskoči, željeni rezultati neće se moći ostvariti. Uključivanje se čini na način prikazan slikom 3.12. u nastavku.

```
<script src="https://code.jquery.com/jquery-1.10.2.js"></script>
```

Sl. 3.12. *Uključena jQuery knjižnica*

Unutar jQuery-a, za potrebe ove web aplikacije korištene su opcije za prikazivanje i skrivanje određenih elemenata dokumenta koristeći funkcije *show()* i *hide()*. Uobičajeni poziv navedenih funkcija izvodi se kao sa slike 3.13.

```
$(selector).hide(speed,callback);
$(selector).show(speed,callback);
```

Sl. 3.13. *Sintaksa jQuery „show()“ i „hide()“ funkcija*

Pri tome, *speed* označava brzinu kojom se određena funkcija odvija, a može sadržavati vrijednosti „slow“, „fast“ ili broj u milisekundama. *Callback* predstavlja funkciju koja treba biti izvršena nakon izvršenja *show()* ili *hide()* funkcija. U izradi web aplikacije, korištena je i funkcija *toggle()*, vrlo slična *hide()* i *show()* funkcijama. Ona omogućuje promjenu stanja elementa na temelju njegove trenutne vidljivosti (engl. *visibility*). Ako je element skriven (engl. *hidden*), primjenjuje show() funkciju, i obratno.

Uzimajući u obzir to da se funkcijama *hide()* i *show()* prikaz elementa (engl. *display*) automatski postavi na blok-prikaz (engl. *block*), bilo je potrebno vratiti ga na željenu vrijednost, koja je u slučaju ove web aplikacije bila *flex*. Ukoliko se to ne učini, velika je vjerojatnost da element na kojem se rade promjene neće biti prikazan pravilno u Internet pregledniku. Taj se problem može riješiti dodavanjem ključne riječi *!important* unutar definicije prikaza elementa u CSS-u, jer spomenute funkcije tada neće prepisati (engl. *override*) određenu vrijednost. No, u ovoj web aplikaciji koristila se druga metoda, koja zahtijeva korištenje metode *css()* nakon pozivanja *hide()* ili *show()* funkcija.

```
css("propertyname","value");
```

Sl. 3.14. *Sintaksa jQuery „css()“ metode*

Osnovna sintaksa te metode prikazana je slikom 3.14., a ona služi za postavljanje ili vraćanje jedne ili više stilskih postavki (engl. *style-property*) određenog elementa. Slikom 3.15. detaljnije je prikazano korištenje spomenutih funkcija i metoda zajedno, na *div* elementu klasnog naziva „downturned“.

```
<script type='text/javascript'>
$('.downturned').show();
$('.downturned').css('display', 'flex');
</script>
```

Sl. 3.15. *Korištene jQuery metode*

Prikazani dio programskog koda odnosi se na oblik očiju korisnika, a nalazi se unutar prethodno spomenutog *switch...case* bloka PHP skripte. Ova web aplikacija provjerava strukturu lica unesenu prilikom registracije korisnika, a kako se sastoji od više povezanih stranica, svaka od

njih zadužena je za provjeru određenog dijela. Tako postoji odvojena stranica koja, između ostalog, sadrži provjeru boje očiju, te stranica koja sadrži provjeru oblika lica korisnika. Sve provjere odvijaju se na sličan način, a u nastavku, slikama 3.16. i 3.17. prikazana je dio provjere oblika lica korisnika koristeći PHP i jQuery.

```php
<?php
    switch($_SESSION['faceShape'])
    {
        case 'Square':
            echo ("<script type='text/javascript'>
            $('.square').show();
            $('.square').css('display', 'flex');
            </script>");
        break;
        case 'Diamond':
            echo ("<script type='text/javascript'>
            $('.diamond').show();
            $('.diamond').css('display', 'flex');
            </script>");
            break;
```

Sl. 3.16. *Dio programskog koda provjere oblika lica*

Ukoliko korisnik prilikom registracije odabere opciju „Other“, koja je ponuđena za slučaj da ne zna kakvu strukturu posjeduje, prikazat će mu se svi oblici, a ne samo određeni. Ovaj pristup vrijedi i za ostale provjere unutar web-aplikacije, a programski kod je sličan za sve provjere. Slikom 3.17. predočen je prikaz svih postojećih oblika lica.

```php
case 'Other':
    echo ("<script type='text/javascript'>
    $('.square').show();
    $('.diamond').show();
    $('.round').show();
    $('.oval').show();
    $('.heart').show();
    $('.square').css('display', 'flex');
    $('.diamond').css('display', 'flex');
    $('.round').css('display', 'flex');
    $('.oval').css('display', 'flex');
    $('.heart').css('display', 'flex');
    </script>");
    break;
```

Sl. 3.17. *Programski kod u slučaju odabira opcije „Other“*

Sličan programski kod nalazi se u početnoj stranici „index.html“, koja nudi mogućnost istraživanja (engl. *explore*). Ova opcija ponuđena je korisnicima koji nisu sigurni žele li se

registrirati, a zanima ih što bi im registracijom bilo ponuđeno. Klikom na gumb, prikazuje se većina fotografija korištenih na svakoj od odvojenih stranica, ali bez detaljnijih objašnjenja i usmjerenja. Na taj način osigurano je da ne-registrirani korisnici nemaju mogućnosti registriranih, ali ipak mogu vidjeti o čemu se radi. Slikom 3.18. prikazan je kod primijenjen u svrhu izvršenja navedenog ponašanja.

```
$(document).ready(function() {
$("#explore").click( function() {
$('.textExplore') .fadeToggle();
});
});
```

Sl. 3.18. *Prikaz ponašanja „Explore“ dugmeta*

3.4. Izgled sučelja

Prilikom izrade web aplikacije, korišten je CSS da bi ona bila što privlačnija i jednostavnija za uporabu. Kako bi bila prilagodljiva mobilnim uređajima i tabletima, unutar CSS-a korištena su svojstva *@media* (engl. *Media query*). Ona se najčešće koriste za provjeru veličine ekrana korištenog uređaja ili za provjeru orijentacije uređaja, te za određivanje primijenjene razlučivosti (engl. *resolution*). U ovoj web aplikaciji, korišteno je svojstvo (engl. *value*) „max-width“, koje označava najveću moguću širinu prikaza, kao što je prozor preglednika (engl. *browser window*). Također je upotrebljeno svojstvo „screen“, koje se koristi za ekrane računala, tableta, mobilnih telefona, i dr. Korištenje *@media* svojstva opisano je slikom 3.19. u nastavku.

```
@media (max-width: 600px){
    header{
        font-family: 'Nunito', sans-serif;
        padding:1px;
        width:100%;
        height:70%;
        background-image: url("../images/rot5.jpg");
        background-size: cover;
    }

    .container{
        margin: auto;
        max-width: 60%;
        padding: 20px;
    }
    a{
      background-size: 10px 4px;
    }
```

Sl. 3.19. *Korištenje @media svojstva*

Ovim dijelom programskog koda opisan je izgled pojedinih dijelova stranice pri čemu je širina veličine 600 piksela. Pri tome se mijenja raspored stranice, neki elementi počinju zauzimati više prostora, dok ostali mogu ostati isti. Bitno je napomenuti da prilikom korištenja ovog svojstva, ne moraju se ispočetka pisati CSS svojstva pojedinih elemenata, nego je potrebno samo prepisati ona svojstva koja želimo promijeniti. Za prilagodbu sučelja mobilnim telefonima, korišteno je *@media* svojstvo s postavljenim minimalnim i maksimalnim veličinama, prikazano slikom 3.20.

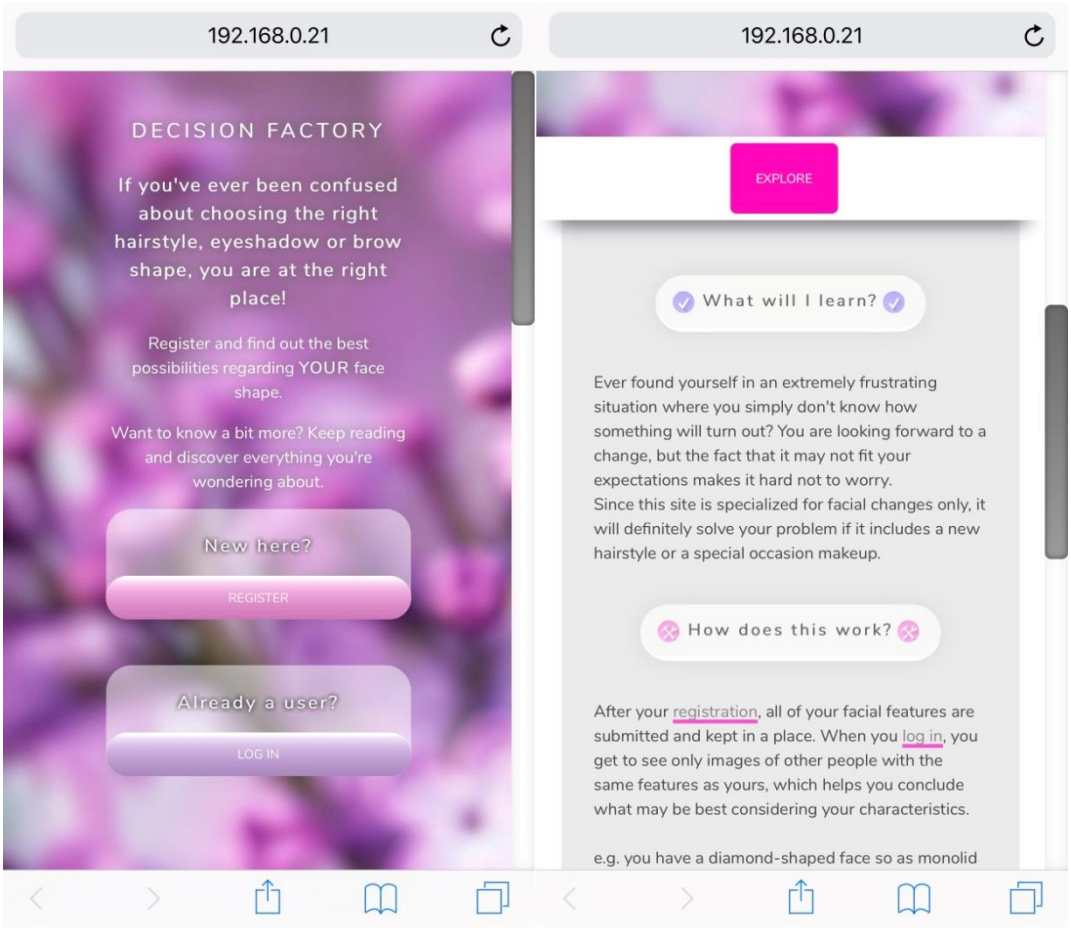
```
@media (min-width: 320px) and (max-width: 480px) {

  header{
    font-family: 'Nunito', sans-serif;
    padding:1px;
    width:100%;
    height:100%;
    background-image: url("../images/rot5.jpg");
    background-size: cover;
}
.container{
    margin: auto;
    max-width: 60%;
    padding: 20px;

}
.col{
  padding: 1px;
}
.myImg{
  border-radius: 8px;
}
```

Sl. 3.20. *Prilagodba mobilnim uređajima*

Slikama 3.19. i 3.20., prikazani su samo dijelovi programskog koda, uzimajući u obzir to da je kompletan kod vrlo dugačak, zbog čega nije prikazan cijeli.

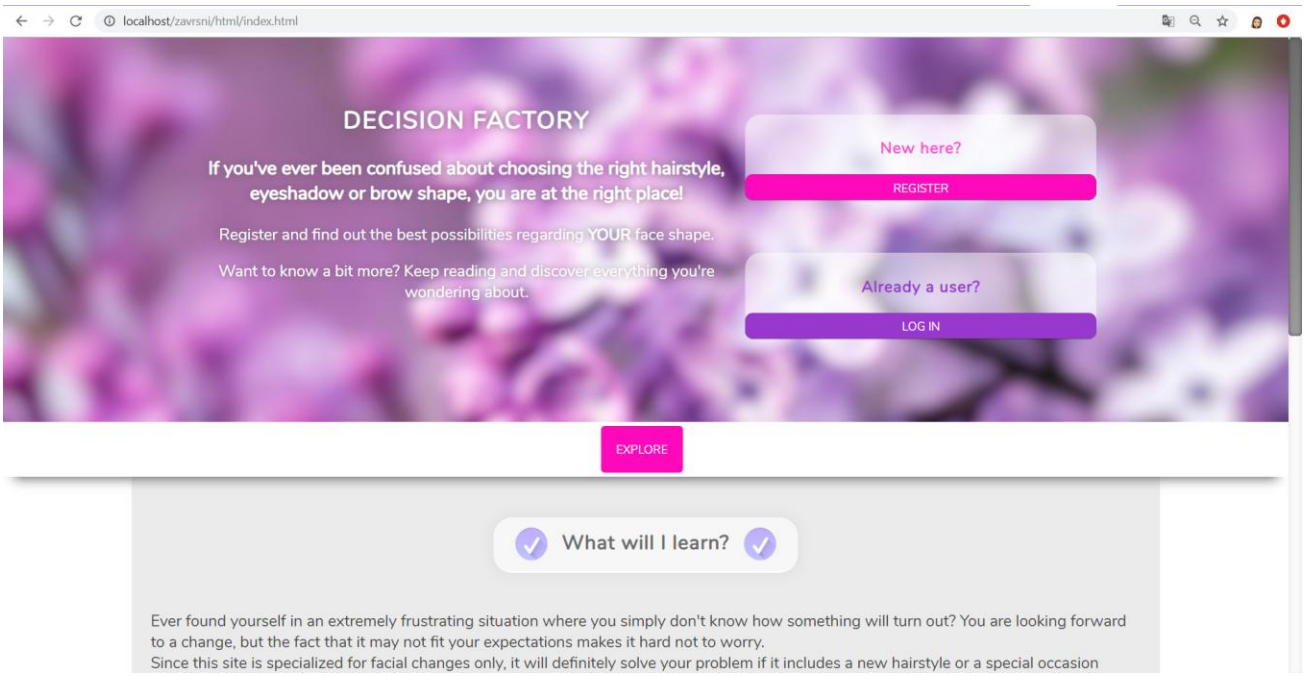
Kako bi se omogućilo pregledavanje ove web aplikacije putem mobilnog uređaja, bilo je potrebno osigurati mogućnost spajanja vanjskih uređaja na *localhost.*Nakon toga, u Internet preglednik (u ovom slučaju, to je Safari) se trebala upisati IP adresa (engl. *IP address*) te pronaći put do željenog dokumenta, koji je isti onom putu upisanom u Internet pregledniku na računalu.

Primjeri prikaza početne stranice na mobilnom uređaju dani su slikom 3.21.



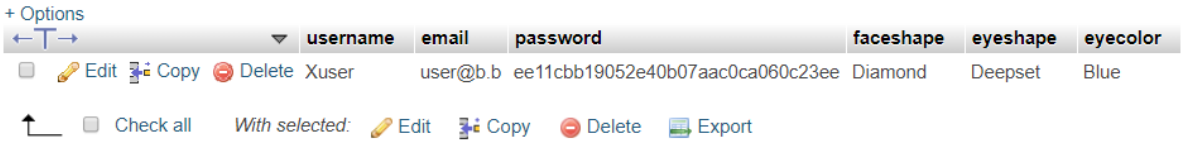
Sl. 3.21. *Prikaz web aplikacije na mobilnom uređaju*

Kako bi aplikacija bila preglednija, prikaz na mobilnom uređaju drugačiji je od prikaza na računalu. Uvažavajući veličinu ekrana mobilnog uređaja, ideja je bila napraviti sučelje lako za snalaženje i korištenje. U nastavku, slikom 3.22., prikazana je razlika između izgleda aplikacije na računalu i manjim ekranima.



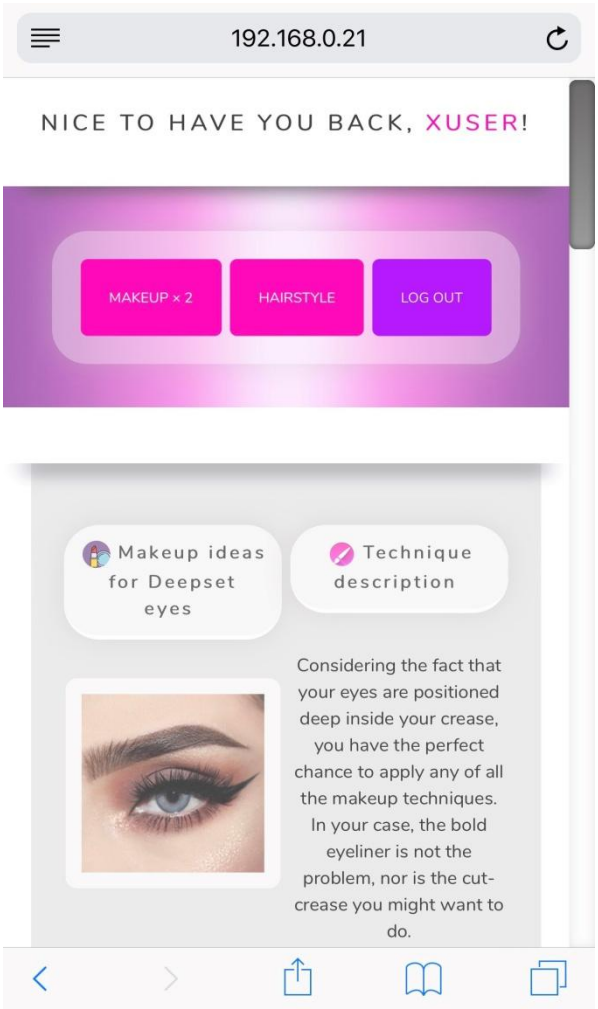
Sl. 3.22. *Prikaz web aplikacije na računalu*

Nakon prijavljivanja ili registracije u sustav, korisnik je preusmjeren na stranicu „loggedin.php“, preko koje može pristupiti ostalim željenim stranicama navigacijom. Također, postoji mogućnost odjave iz sustava. Slike 3.24. i 3.25. pokazat će izgled odvojenih navigacijskih stranica na mobilnom uređaju i računalu, a za ovaj prikaz koristio se testni korisnik Xuser (njegovi podaci iz tablice podataka prikazani su slikom 3.23.)

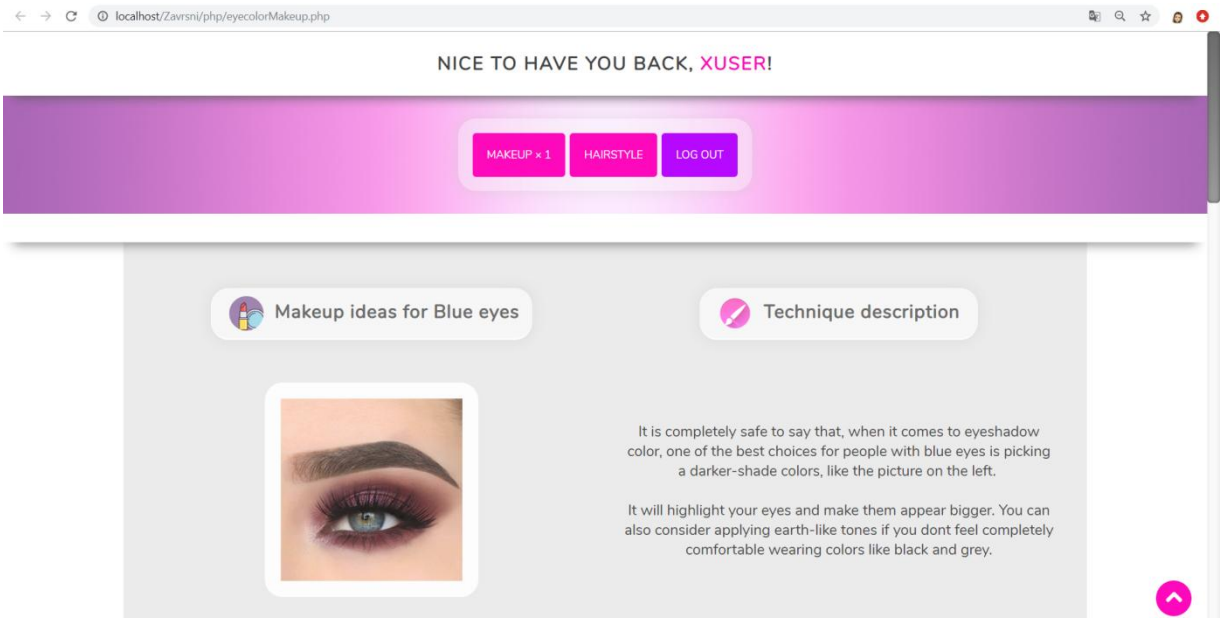


Sl. 3.23. *Podaci korisnika Xuser*

Kao što je vidljivo iz slike 3.23., tj. iz baze podataka „users“, korisnik Xuser posjeduje specifične karakteristike oblika lica, očiju te boje očiju. Na temelju tih parametara će se u nastavku, kao što je prikazano prethodnim potpoglavljem, određivati elementi koji će se korisniku prikazivati i kojima će imati pristup.



Sl. 3.24. *Prikaz stranice prijavljenog korisnika na mobilnom uređaju*



Sl. 3.25. *Prikaz stranice prijavljenog korisnika na računalu*

Važno je napomenuti da su fotografije prikupljene prilikom izrade web aplikacije zaštićene autorskim pravima, a svi izvori s kojih su prikupljene, navedeni su u zasebnom dokumentu naziva „credits.html“.

```
<section class="loginNav">
    <div class="l">
        <div class="navv" id="in">
            <div class="loginNavPart">
            <button id="eyeshape"
            onClick="document.getElementById('creditPhotos').scrollIntoView();">Makeup CR1</button>
            <button id="eyeshape"
            onClick="document.getElementById('creditPhotos2').scrollIntoView();">Makeup CR2</button>
            <button id="eyeshape"
            onClick="document.getElementById('creditPhotos3').scrollIntoView();">Hairstyle CR</butto
            </div></div></div></section>
<section class="text">
    <div class="textPart creditPhotos" id="creditPhotos">
        <div class="l">
            <h2 class="h2Title">Credits x Section 1</h2>
        </div>
        <p><a href="https://www.pinterest.com/pin/739575570038896777/">Green eyes 3</a></p>
        <p><a href="https://www.pinterest.com/pin/625648573212059574/">Green eyes 4</a></p>
```

Sl. 3.26. *Prikaz dijela dokumenta „credits.html“*

U tom dokumentu se kroz jednostavnu navigaciju može pristupiti svakoj izvornoj fotografiji koja je preuzeta radi funkcionalnosti web aplikacije.

4. ZAKLJUČAK

Realizacijom ovog završnog rada, prikazan je i obrazložen postupak izrade web aplikacije za pomoć pri odabiru šminke. To je učinjeno koristeći određene web tehnologije poput PHP-a, jQuery-a, dok su HTML, CSS i Bootstrap korišteni za izgled stranice. Opisani su razlozi i načini korištenja navedenih tehnologija, a tako i proces kreiranja i upravljanja bazom podataka u svrhu spremanja i dohvaćanja podataka. Svakom korisniku omogućena je registracija, a onda i pregled stranice na temelju kojega može donijeti željenu odluku. Aplikacija je u konačnici testirana u preglednicima Google Chrome i Safari.

LITERATURA

[1], [2], [3] HTML - https://hr.wikipedia.org/wiki/HTML [6.9.2019.]

[4], [5] CSS - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c220_polaznik.pdf [6.9.2019.]

[6] CSS - https://hr.wikipedia.org/wiki/CSS [6.9.2019.]

[7] CSS - https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c220_polaznik.pdf [6.9.2019.]

[8] Bootstrap - https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework) [6.9.2019.]

[9] jQuery – https://en.wikipedia.org/wiki/JQuery [6.9.2019.]

SAŽETAK

Ovim završnim radom kreirana je web aplikacija namijenjena za pomoć u odabiru šminke pomoću HTML, CSS i PHP jezika. Aplikacija korisnicima pruža mogućnosti registracije i prijave. Može se koristiti u svrhu edukacije u području koje obuhvaća, te u svrhu zabave. Spomenute su i objašnjene korištene tehnologije poput HTML-a, MySQL-a, Bootstrap-a, itd. Također, detaljno je pojašnjen postupak realiziranja web aplikacije.

Ključne riječi: PHP, HTML, CSS, Bootstrap, MySQL, jQuery, web aplikacija, baza podataka

ABSTRACT

Created web application, with a purpose of helping users decide on desired makeup choices, was made using HTML, CSS and PHP languages. Application provides registration and login possibilities. It can be used in educational purposes within a specific domain, so as in the fun purposes. Technologies used for creating this application like HTML, MySQL, Bootstrap, etc. were mentioned and described. The procedure of creating the application was also explained in detail.

Key words: PHP, HTML, CSS, Bootstrap, MySQL, jQuery, web application, database

ŽIVOTOPIS

Ana Čačija rođena je 7.4.1998. godine u Osijeku. Nakon završetka osnovne škole „Tin Ujević“, 2012. godine upisuje III. gimnaziju u Osijeku. Maturira 2016. godine, te iste godine upisuje preddiplomski sveučilišni studij Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija.