Izrada web stranice za grupno financiranje ideja

Mašanović, Luka

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:200:448096

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

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

Sveučilišni studij

IZRADA WEB STRANICE ZA GRUPNO FINANCIRANJE IDEJA

Završni rad

Luka Mašanović

Osijek, 2016

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. GRUPNO FINANCIRANJE 2
3. TEHNOLOGIJE 3
3.1 PHP 3
3.1.1 PHP sintaksa 3
3.1.2 Varijable u PHP-u 4
3.1.3 Operatori 5
3.1.4 Petlje 7
3.1.5 Kontrolne strukture 8
3.2 HTML 11
3.2.1 HTML elementi 12
3.3 SASS 16
3.3.1 Sass varijable 17
3.3.2 Ugnježđivanje 17
3.3.3 Import naredba 18
3.3.4 Mixins 19
3.3.5 Nasljeđivanje 20
4. BAZA PODATAKA 22
4.1 phpMyAdmin 25
5. PAYPAL SANDBOX 27
6. IZGLED STRANICE I PRIMJENA 29
7. ZAKLJUČAK 32
LITERATURA 33
SAŽETAK 34
ABSTRACT 35
ŽIVOTOPIS 36
PRILOZI 37

1. UVOD

Cilj ovog završnog rada je izrada web stranice za grupno financiranje ideja. Potrebno je napraviti stranicu na kojoj će korisnici moći registrirati te objavljivati svoje ideje tj. projekte, za koje će moći primati donacije od zainteresiranih korisnika putem PayPal platforme. U drugom poglavlju je objašnjena sama ideja grupnog financiranja. Poglavlje tehnologije opisuje sve tehnologije koje su korištene tijekom izrade web stranice, dok poglavlje o bazi podataka objašnjava korištenje osnovnih SQL naredbi i izgled same baze. PayPal sandbox poglavlje opisuje kreiranje sandbox računa i način na koji sam sandbox funkcionira.

1.1. Zadatak završnog rada

Zadatak završnog rada je napraviti web stranicu na kojoj bi korisnici mogli objavljivati pozive za grupno financiranje (engl. *crowdfunding*) svojih projekata, a zainteresirani posjetioci će moći donirati novce putem PayPal platforme.

2. GRUPNO FINANCIRANJE

Svrha grupnog financiranja (engl. *crowdfunding*) je zapravo prikupljanje novaca od strane grupe ili pojedinih donatora u svrhu ostvarivanja određenog cilja. Općenito crowdfunding platforma predstavlja mjesto na internetu gdje se mogu naći razne zanimljive informacije, kreativne ideje i slične stvari vezane za projekte. Na taj način je lakše prikupiti sredstva za ideje koje bi inače teško imale uspjeha u tradicionalnim financijskim institucijama.

Najpoznatije crowdfunding platforme i njihov princip djelovanja je različit, ali u suštini se svodi na to da pojedinci s relativno malim iznosima financiraju projekte do kojih im je stalo te za uzvrat dobivaju određene povlastice ili nagrade.

Neki od najpopularnijih i najpoznatijih crowdfunding platforma su KickStarter, IndieGoGo te RocketHub. Kickstarter je jedan od najuspješnijih sa prikupljenih gotovo 2 milijarde dolara kroz 110000 uspješno financiranih projekata.

Postoje četiri glavne kategorije koje se temelje na načinu poticanja ulagača odnosno kako zainteresirati ulagača da donira vašem projektu:

- Donacije (GoFundMe, Crowdrise, GivenGain) – donatori očekuju da ih se spomene u promociji projekta.
- Nagrade (Kickstarter, Indiegogo, Rockethub) – nagrade se uglavnom odnose na povlašteno uživanje u rezultatima financiranog projekta.
- Posuđivanje (Kiva, Zopa) – ulagači beskamatno posuđuju sredstva, a po realizaciji i uspješnoj provedbi ulog im bude vraćen.
- Vlasnički udjeli (Crowdcube, Crowdfunder, Sellaband) - ulagači polažu pravo na dobit od projekta poput dioničara.

3. TEHNOLOGIJE

3.1 PHP

PHP (Hypertext Preprocessor) je programski jezik koji se bazira na C i Perl sintaksi, namijenjen je izradi dinamičnih web stranica. Prednosti PHP-a su široka podrška različitih baza podataka, programskih biblioteka i internet protokola. [1]

PHP je open-source server-side skriptni programski jezik za dinamičko generiranje HTML koda. PHP je skriptni programski jezik koji kreira HTML stranicu na serveru prije nego što je isporučena klijentu na njegov preglednik. Ovakav način generiranja sadržaja omogućava to da klijent ne može vidjeti skriptu nego čisti HTML kod koji je PHP generirao. [2]

3.1.1 PHP sintaksa

PHP skripta počinje sa <?php i završava sa ?>. Također može biti postavljena bilo gdje u dokumentu. PHP dokumenti inače sadrže HTML elemente te php skripte. Zadana ekstenzija php dokumenata je „ .php “

```
<!DOCTYPE html>
<html>
<body>
<h1>PHP stranica</h1>
<?php
echo “Pozdrav”;
?>
</body>
</html>
```

Sl. 3.1. *Primjer PHP dokumenta*

Na slici 3.1 prikazan je php dokument u kojem se koristi php skripta, koja ispisuje „Pozdrav“ na web stranici. Također svaka php naredba završava sa točka-zarez (;).

Bitna stvar kod pisanja koda je komentiranje, koje se ne izvršava kao dio programa nego služi kao podsjetnik. Komentiranje se koristi kada je potrebno drugoj osobi koja čita kod dati do znanja što se događa u određenom dijelu koda ili što programer misli u tom trenutku.

PHP podržava više načina komentiranja kao što je prikazano na slici 3.2

```
<?php

// Jedno linijski komentar



# Također jedno linijski komentar



/* Više linijski komentar, koji

nam omogućuju pisanje u više

redova.

*/

?>
```

Sl. 3.2. *Komentari u PHP-u*

U php-u ključne riječi nisu *case-sensitive*, odnosno nije bitno jesu li pisane velikim ili malim slovima, ali imena varijabli jesu *case-sensitive* pa na to treba obratit pažnju prilikom pisanja koda.

3.1.2 Varijable u PHP-u

Deklariranje varijabli u php-u se vrši stavljanjem znaka $ prije naziva varijable. Taj znak upućuje php prevoditelju da se radi o varijabli, a ne o običnom tekstu. Također postoje određena pravila pri pisanju PHP varijabli:

- Varijabla mora počinjati sa slovom ili znakom donja crta (_).
- Ime varijable ne smije počinjati brojem.
- Ime varijable smije sadržavati samo alfa numeričke znakove i (A-z, 0-9, i _).

Za razliku od drugih jezika kao što su C, C++ i Java, kod pisanja php varijabli nije potrebno deklarirati tip varijable prije korištenja već php to automatski radi, ovisno o vrsti varijable.

Tipovi varijabli koje PHP podržava:

Tab. 3.1. *Tipovi varijabli*

Integer	Cijeli broj
Float	Decimalni broj
String	Cijela riječ ili rečenica
Array	Niz
Boolean	Istina ili laž (0 ili 1)
NULL	Nema vrijednost

3.1.3 Operatori

Operatori se koriste kako bi obavljali određene operacija na varijablama i njihovim vrijednostima. PHP dijeli operatore na sljedeće grupe:

Tab. 3.2. *Aritmetički operatori*

Operator	Ime	Primjer	Rezultat
+	Zbrajanje	$x + $y	Zbroj od $x i $y
-	Oduzimanje	$x - $y	Razlika od $x i $y
*	Množenje	$x * $y	Produkt od $x i $y
/	Dijeljenje	$x / $y	Kvocijent od $x i $y
%	Modul	$x % $y	Ostatak dijeljenja od $x i $y

Tab. 3.3. *Operatori uspoređivanja*

Operator	Ime	Primjer	Rezultat
= =	Jednako	$x == $y	Vraća istina (true) ako je $x = $y
= = =	Identično	$x === $y	Vraća istina (true) ako je $x = $y i ako su istog tipa.
!=	Nije jednako	$x != $y	Vraća istina (true) ako $x nije jednako $y
<>	Nije jednako	$x <> $y	Vraća istina (true) ako $x nije jednako $y
!==	Nije identično	$x !== $y	Vraća istina (true) ako $x nije jednako $y i nisu isto tipa
<	Manje	$x < $y	Vraća istina (true) ako je $x manji od $y
>	Veće	$x > $y	Vraća istina (true) ako je $x veći od $y
>=	Veće jednako	$x >= $y	Vraća istina (true) ako je $x veći ili jednak $y
<=	Manje jednako	$x <= $y	Vraća istina (true) ako je $x manji ili jednak $y

Tab. 3.4. *Logički operatori*

Operator	Ime	Primjer	Rezultat
and	I	$x and $y	Istina (true) ako su $x i $y istina (true)
or	ili	$x or $y	Istina (true) ako je $x istina (true) ili ako je $y istina (true)
xor	Xor	$x xor $y	Istina (true) ako je $x istina (true) ili ako je $y istina (true), ali ne i ako su oba istina (true)
&&	I	$x && $y	Istina (true) ako su oboje $x i $y istina (true)
| |	ili	$x || $y	Istina (true)ako je $x true ili ako je $y istina (true)
!	Ne	!$x	Istina (true) ako $x nije istina (true)

Tab. 3.5. *Operatori uvećavanja i smanjivanja*

Operator	Ime	Rezultat
++$x	Preduvećavanje	Uveća $x za jedan, pa vrati $x
$x++	Naknadno uvećavanje	Vrati $x, pa uveća $x za jedan
--$x	Predsmanjivanje	Umanji $x za jedan, pa vrati $x
$x--	Naknadno smanjivanje	Vrati $x, pa umanji $x za jedan

3.1.4 Petlje

U php-u petlje izvode blok naredbi dokle god je uvjet istinit (engl. *true*). Dijele se na četiri petlje:

- While – blok naredbi se ponavlja dokle god je uvjet istinit(engl. *true*).

```
while (uvjet je istinit) {
izvršavaju se naredbe; }
```

Sl. 3.3. *Sintaksa while petlje*

- Do…While – blok naredbi se izvede jednom pa se nakon toga ispituje uvjet, što znači da bez obzira na to da li je uvjet istinit blok naredbi će se izvršiti barem jednom.

```
Do {izvršavaju se naredbe; }
while (uvjet je istinit);
```

Sl. 3.4. *Sintaksa do..while petlje*

- For – petlja for se koristi kada se unaprijed zna broj ponavljanja. Sastoji se od tri parametra. Početne vrijednosti, vrijednost sa kojom se uspoređuje te operator uvećanja ili smanjenja. Dokle god je uvjet istinit petlja će izvršavati blok naredbi, kada uvjet više ne bude istinit petlja se završava.

```
for (početna vrijednost; vrijednost sa kojom se uspoređuje; operator uvećanja ili smanjenja) {
blok naredbi; }
```

Sl. 3.5. *Sintaksa for petlje*

- Foreach – Foreach petlja se koristi za rad sa nizovima. Ona prolazi kroz svaki element niza i izvršava blok naredbi. Sprema ključ i vrijednost svakog elementa niza u posebne varijable u kojim se za svako ponavljanje petlje nalaze ključ i vrijednost elementa niza na kojem se trenutno nalazi unutarnji pokazivač. Unutarnji pokazivač se prije ulaska u petlju nalazi na 0 i svakim novim krugom u petlji povećava se za 1. Novim zvanjem foreach petlje unutarnji pokazivač se resetira. Petlja se vrti sve dok ne ostane bez elemenata niza. [5]

```
foreach ($niz as $vrijednost) {
blok naredbi; }
```

Sl. 3.6. *Sintaksa foreach petlje*

3.1.5 Kontrolne strukture

Kontrolne strukture se koriste kada je potrebno obaviti različite zadatke ovisno o različitim uvjetima. PHP podržava četiri kontrolne strukture:

- If – Izvršava određeni blok naredbi ako je uvjet istinit (engl. *true*).

```
if (uvjet) {
    blok naredbi će se izvršiti ako je uvjet istinit; }
```

Sl. 3.7. *Sintaksa if kontrolne strukture*

- If…else – Izvršava određeni blok naredbi ako je uvjet istinit, a ako nije izvršava se drugi blok naredbi.

```
if (uvjet) {
    blok naredbi će se izvršiti ako je uvjet istinit; }
else {
drugi blok naredbi koji će biti izvršen ako uvjet nije istinit; }
```

Sl. 3.8. *Sintaksa if..else kontrolne strukture*

- If...elseif...else – Izvršava različite blokove naredbi za više od dva uvjeta.

```
if (uvjet) {
    blok naredbi će se izvršiti ako je uvjet istinit; }
elseif(uvjet) {
blok naredbi će se izvršiti ako je uvjet istinit; }
else { blok naredbi će se izvršiti ako svi uvjeti nisu istiniti;  }
```

Sl. 3.9. *Sintaksa if..elseif..else kontrolne strukture*

- Switch – Koristi se kada je potrebno izvršiti različite naredbe ovisno o različitim uvjetima.

```
switch (n) {
   case label1:
      blok naredbi će se izvršiti ako je n=label1;
      break;
   case label2:
      blok naredbi će se izvršiti ako je n=label2;
      break;
   case label3:
      blok naredbi će se izvršiti ako je n=label3;
      break;
   ...
   default:
      blok naredbi će se izvršiti ako je n različit od svih labeli;
}
```

Sl. 3.10. *Sintaksa switch kontrolne strukture*

3.2 HTML

HTML (HyperText Markup Language), što znači prezentacijski jezik za izradu web stranica. Hipertekst dokument se stvara pomoću HTML-a gdje on stvara i oblikuje sadržaj. HTML je svoju popularnost stekao s tim što se lako uči, jednostavan je te je od početka zamišljen kao besplatan i dostupan svima. Prikazivanje HTML dokumenata se vrši u web pregledniku te se nastoji da bude jednako prikazan neovisno o kojem je operacijskom sustavu, web pregledniku ili računalu riječ. Također HTML nije programski jezik.

HTML datoteke imaju ekstenzije .html ili .htm, koje zapravo predstavljaju obične tekstualne datoteke. Svaka stranica se sastoji od znakova (engl. *tags*), oni opisuju kako će se nešto prikazati u web pregledniku. HTML dokumenti imaju uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranice.

HTML oznake koje se kreiraju u paru počinju sa znakom manje od (<), a završavaju znakom veće od (>). Zatvarajuća oznaka kreira se isto kao i otvarajući, ali se prije završnog znaka > dodaje i kosa crta (/) .

Svaki HTML dokument počinje sa <!DOCTYPE> elementom, kojim je označena inačica standarda koji se koristi za izradu HTML dokumenta DTD (engl. *Document Type Declaration*). Također svaki element može imati atribute kojima se definiraju svojstva tog elementa. Nakon inačice za standarda slijedi <html> element koji označava početak HTML dokumenta. Unutar <html> elementa najčešće se nalazi <head> element koji predstavlja zaglavlje HTML dokumenta u kojem se uglavnom definiraju jezične značajke dokumenta, stilska obilježja stranice, skripte kreirane u JavaScrpit jeziku kao i sam naslov stranice. U <body> elementu kreira se sadržaj dokumenta odnosno ono što se prikazuje korisniku u pregledniku. [3]

3.2.1 HTML elementi

HTML elemenata ima puno, ali četiri osnovna koja se gotovo uvijek nalaze u HTML dokumentu su prikazana u tablici 3.6.

Tab. 3.6. Osnovni *HTML elementi*

HTML element	Opis
<html> ... </html>	Početak i kraj samog HTML dokumenta
<head> ... </head>	Zaglavlje HTML dokumenta
<title> ... </title>	Naslov
<body> ... </body>	Definira tijelo dokumenta

Vrlo bitna stvar kod HTML dokumenta su naslovi. Tražilice koriste naslove kako bi indeksirali strukturu i sadržaj web stranice. Važno je koristiti naslove kako bi pokazali strukturu dokumenta.

Tab. 3.7. *HTML naslovi*

HTML element	Opis
Naslov	Naslov 1
Naslov	Naslov 2
Naslov	Naslov 3
Naslov	Naslov 4
Naslov	Naslov 5
Naslov	Naslov 6

Naslov veličine <h1> definiran je kao najznačajniji naslov i trebao bi biti korišten kao glavni naslov na stranici. Ostali naslovi bi ga trebali pratiti po značajnosti, praćen sa <h2>, pa još manje važan <h3> i tako dalje.

HTML paragrafi su definirani sa oznakom <p>. Najveći problem kod paragrafa je to što se izlazni rezultat ne može mijenjati dodavanjem dodatnih razmaka ili linija. Preglednik će automatski maknuti dodatne linije i prazan prostor. Kako bi to bilo moguće koristi se oznaka <pre> koja sačuva dodatne prazne redove i razmake.

Tab. 3.8. *HTML paragrafi*

HTML element	Opis
tekst	Definira paragraf
 	Lomi tekst u novi red
<pre> tekst </pre>	Definira preformatiran tekst

Linkovi se nalaze gotovo na svim stranicama, omogućavajući korisnicima put od stranice do stranice. HTML linkovi su hiper linkovi. Moguće je klikom na link preći na drugi dokument s time da link ne mora biti tekst, moguće je da bude slika ili neki drugi HTML element. Linkovi su definirani sa oznakom <a> unutar kojeg se nalazi *href* atribut koji određuje destinaciju linka. Također atribut *target* se koristi za određivanje gdje će link biti otvoren, on može imati vrijednosti prikazane u tablici 3.9.

Tab. 3.9. *HTML linkovi - target atributi*

Target atributi	Opis
_blank	Otvara dokument u novom prozoru
_self	Otvara dokument u istom prozoru
_parent	Otvara dokument u roditeljskom okviru
_top	Otvara dokument u punoj veličini prozora
framename	Otvara dokument u imenovanom okviru

```
<a href="http://www.google.com/" target="_blank">Google</a>
```

Sl. 3.11. *Sintaksa HTML linka*

Slike u HTML-u su definirane sa oznakom <img>. Gdje je <img> oznaka prazna, sadrži samo atribute i nema zatvarajuću oznaku. Oznaka pomoću koje se prikazuje slika je *src*, to je atribut koji definira URL slike. Jedan od obveznih atributa je i *alt* atribut koji pruža alternativni tekst u slučaju da se slika iz nekog razloga ne može prikazati. U slučaju da preglednik ne može pronaći sliku, prikazana će biti vrijednost *alt* atributa. Također je moguće koristiti i dva atributa za određivanje visine i širine slike.

```
<img src="html.gif" alt="HTML Slika" width="128" height="128">
```

Sl. 3.12. *Sintaksa HTML slike*

Definiranje tablice u HTML-u se obavlja sa oznakom <table>. Svaki novi red je definiran sa oznakom <tr>. Dok je zaglavlje definirano sa oznakom <th>, prema zadanim postavkama naslovi u tablicama su podebljani i centrirani. Podatak u tablici se definira pomoću oznake <td>.

Tab. 3.10. *Oznake za definiranje tablica u HTML-u*

Oznake	Opis
<table>	Definira tablicu
<th>	Definira zaglavlje ćelije u tablici
<tr>	Definira red u tablici
<td>	definira ćeliju u tablici
<caption>	Definira naslov tablice
<colgroup>	Specificira grupu jednog ili više stupaca radi oblikovanja
<thead>	Grupira sadržaj zaglavlja
<tbody>	Grupira sadržaj tijela
<tfoot>	Grupira sadržaj podnožja

Liste su korisne u organizaciji preglednih i strukturiranih informacija. U HTML-u postoje neoznačene, označene i opisne liste. Neoznačena lista počinje sa oznakom <ul>, a svaki novi element liste se nalazi između <li> oznaka. Elementi će biti označeni sa crnim kružićima, prema zadanim postavkama. Također je moguće birati vrstu znaka koja će označavati svaki novi element. Označene liste počinju sa <ol> oznakom te kao i kod ne označenih lista svaki novi element počinje sa oznakom <li>. Sadržaj liste će biti označen sa brojevima prema zadanim postavkama, što je također moguće promijeniti sa atributom *type* u <ol> oznaci, svi tipovi numeriranja se nalaze u tablici 3.11.. Opisna lista je lista koja za svaki element liste ima njegov opis. Oznaka <dl> definiran opisnu listu dok <dt> oznaka definira ime elementa te oznaka <dd> opis elementa.

Tab. 3.11. *Tipovi numeriranja u označenim listama*

Type atribut	Opis
type=“1“	Elementi u listi će biti numerirani brojevima
type=“A“	Elementi u listi će biti numerirani velikim slovima
type=“a“	Elementi u listi će biti numerirani malim slovima
type=“I“	Elementi u listi će biti numerirani velikim rimskim brojevima
type=“i“	Elementi u listi će biti numerirani malim rimskim brojevima

Tab. 3.12. *Oznake liste*

Oznaka	Opis
	Definira neoznačenu listu
	Definira označenu listu
	Definira element u listi
<dl>	Definira opisnu listu
<dt>	Definira stavku u listi
<dd>	Definira opis u opisnoj listi

3.3 SASS

Sass (Syntactically Awesome StyleSheets) je ekstenzija CSS-a odnosno skriptni jezik koji daje običnom CSS-u eleganciju. Također dozvoljava korištenje varijabli, izraza, funkcija, osnovnu logiku te petlje. Funkcije omogućavaju lakše manipuliranje sa bojama i ostalim vrijednostima. Kod je puno bolje organiziran i lakše ga je za čitati. Kompatibilan je sa svim verzijama CSS-a.

Sass je preprocesor koji omogućuje stvari kao što su varijable, mixine, ugnježđivanja i slično. Naravno web stranica to ne razumije, stoga Sass nakon što je instaliran na projekt vrši pretvorbu preko terminala naredbom „sass input.scss output.css“ koja pretvara Sass u standardni CSS dokument kojeg preglednik razumije.

```
$blue: #3bbfce;
$margin: 16px;

.content-navigation {
  border-color: $blue;
  color:
    darken($blue, 20%);
}
.border {
  padding: $margin / 2;
  margin: $margin / 2;
  border-color: $blue;
}
```

Sl. 3.13. *SCSS sintaksa*

```
.content-navigation {
  border-color: #3bbfce;
  color: #2b9eab;
}

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce;
}
```

Sl. 3.14. *CSS sintaksa*

Primjer sa slike 3.13 predstavlja kod koji je napisan pomoću SCSS sintakse. Na slici je prikazano korištenje varijabli koje su definirane na početku. Dalje su vidljive operacije koje se mogu

obavljati s njima kao što su dijeljenje ili čak jednostavno pozivanje varijable u kojoj je pohranjena vrijednost boje. Slika 3.14 prikazuje kako to izgleda kada je prevedeno u standardni CSS.

3.3.1 Sass varijable

Najjednostavniji način korištenja Sass-a je kroz varijable. Varijable u Sass-u počinju sa znakom $. U njih se mogu spremiti boje, fontovi ili bilo koje druge CSS vrijednost koje je kasnije moguće iskoristiti.

```
$font:    Helvetica, sans-serif;
$primarna-boja: #333;

body {
  font: 100% $font;
  color: $primarna-boja; }
```

Sl. 3.15. *SCSS sintaksa - varijable*

```
body {
  font: 100%  Helvetica, sans-serif;
  color: #333;
}
```

Sl. 3.16. *CSS sintaksa*

Kada se Sass procesira, uzima vrijednosti koje su definirane pod $font i $primarna-boja i daje normalne CSS vrijednosti kao što je prikazano na slici 3.16. Ovakvo korištenje varijabli može biti jako korisno pogotovo kada se kroz stranicu koriste iste boje, font i slično.

3.3.2 Ugnježđivanje

Kod pisanja HTML-a primjećuje se vizualna hijerarhija, što nije slučaj kod CSS-a. Sass omogućuje ugnježđivanje (engl. *nesting*) selektora, odnosno istu vizualnu hijerarhiju kao i kod HTML-a. Ovo je odličan način za organizaciju i lakšu preglednost CSS-a.

```
nav {
    ul {
        margin: 0;
        padding: 0;
        list-style: none;
        }
    li {
        display: inline-block;
        }
}
```

Sl. 3.17. *SCSS sintaksa – ugnježđivanje*

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
```

Sl. 3.18. *CSS sintaksa*

3.3.3 Import naredba

CSS omogućava korištenje uvoz (engl. *import*) naredbe koja dijeli CSS u manje, bolje održive dijelove koda. Jedina mana kod korištenja naredbe import je što CSS stvara novi HTTP zahtjev. Sass će uzeti dokument koji je potrebno uvesti i kombinirati ga sa dokumentom koji je ranije uvezen te pružiti samo jedan dokument pregledniku. Za uvezeni dokument koji ne treba biti preveden u CSS-u koristi se donja crta (_) prije naziva Sass dokumenta.

```
// _reset.scss
html,body,ul,ol {
   margin: 0;
  padding: 0; }
```

Sl. 3.19. *SCSS sintaksa*

```
// baza.scss
@import 'reset';
body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
`
```

Sl. 3.20. *SCSS sintaksa – import naredba*

```
html, body, ul, ol {
  margin: 0;
  padding: 0;
}
body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}  padding: 0; }
```

Sl. 3.21. *CSS sintaksa*

Na slici 3.19. je prikazan dio koda koji je uvezen u baza.scss dokument. Kao što je prikazano na slici nije potrebno uključiti ekstenziju .scss kod uvoza reset dokumenta, Sass je dovoljno pametan da to sam shvati. Na slici 3.21. prikazan je konačni procesuirani CSS dokument nakon korištenja naredbe import.

3.3.4 Mixins

U CSS-u postoje ključne riječi koje su teško pamtljive i često dolazi do pogreške pri pisanju tih ključnih riječi, primjer toga su vendor prefiksi. Mixini omogućuju da se naprave grupe CSS deklaracija koje se mogu koristit kroz cijelu stranicu. Također je moguće dodijeliti im vrijednost kako bi bili što fleksibilniji. Dobar primjer je korištenje mixina za vendor prefikse. Kao što je prikazano na slici 3.22.

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
    -moz-border-radius: $radius;
     -ms-border-radius: $radius;
         border-radius: $radius; }
.box { @include border-radius(10px); }
```

Sl. 3.22. S*CSS sintaksa - mixin*

Mixin se kreira sa ključnom riječi @mixin te nakon toga ime mixina. U primjeru sa slike 2.22. korišteno je ime *border-radius* te varijabla $radius unutar zagrada. Toj varijabli moguće je pridružiti bilo koju vrijednost. Nakon što je mixin kreiran, moguće ga je koristiti kao CSS deklaraciju sa naredbom @include nakon koje slijedi ime mixina.

```
.box {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  border-radius: 10px; }
```

Sl. 3.23. *CSS sintaksa – nakon korištenja mixina*

3.3.5 Nasljeđivanje

Jedna od najboljih značajki Sass-a je nasljeđivanje (engl. *Inheritance*). Naredba @extend omogućava dijeljenje CSS naredbi selektorima. Zbog tog svojstva Sass ostaje čist za čitanje odnosno pregledniji. Na slici 3.24. prikazan je primjer korištenja nasljeđivanja i naredbe @extend.

```
.poruka {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333; }
.uspjeh {
  @extend .poruka;
  border-color: green; }
.pogreška {
  @extend .poruka;
  border-color: red; }
```

Sl. 3.24. S*CSS sintaksa – naredba extend*

```
.poruka, .uspjeh, . pogreška {
  border: 1px solid #cccc;
  padding: 10px;
  color: #333; }
.uspjeh {
  border-color: green; }
.pogreška {
  border-color: red; }
```

Sl. 3.25. *CSS sintaksa*

Na slikama 3.24. i 3.25. nalaze se poruke za prikaz pogreški i uspjeha. Nasljeđivanje omogućuje da svojstva iz klase poruka budu proslijeđene klasama uspjeh i pogreška, kao što je vidljivo u krajnjem rezultatu na slici 3.25.. Što znači da nasljeđivanje omogućava izbjegavanje pisanja imena klasa i elementa više puta.

4. BAZA PODATAKA

Pri izradi stranice bilo je nužno izraditi bazu podataka u koju će biti pohranjeni imena projekata, korisnici i slično. Baze podataka se izrađuju pomoću SQL-a (Structured Query Language) koji je razvijen za rad i postavljanje upita u radu s bazom. Za upravljanje bazom korišten je MySQL.

MySQL je besplatan, open source sustav za upravljanje bazom podataka te je česti izbor baze za projekte otvorenog koda. Distribuira se kao sastavni dio serverskih Linux distribucija. MySQL baze su relacijskog tipa, koji se koristi zbog svog načina skladištenja i pretraživanja velikih količina podataka. PHP i MySQL su svoju popularnost stekli baš zbog toga što su *open source* i mogu se besplatno koristiti. [4]

Za upravljanje bazom preko Web-a korišten je PhpMyAdmin koji je također besplatan softver, napisan u PHP-u. Uz pomoć njega moguće je izvršavati mnoge MySQL operacije putem korisničkog sučelja izravno u pregledniku.

Kako bi bilo omogućeno korištenje MySQL servera, Apachea i PHP-a korišten je serverski paket za izradu dinamičkih web stranica na lokalnoj mreži pod nazivom XAMPP.

XAMPP je potpuno besplatna Apache distribucija koja sa instalacijom donosi PHP, Perl, Apache server, MySQL server i druge aplikacije korisne za razvoj i testiranje dinamičkih web stranica na osobnom računalu.

Baza podataka najčešće sadrži jednu ili više tablica. Svaka od tih tablica je definirana imenom. Tablice sadrže redove u kojima se nalaze podaci. Primjer tablice 4.1. Kupci koja se sastoji od četiri retka, svaki za jednu mušteriju te pet stupaca.

Tab. 4.1. *Tablica kupaca*

KupacID	Ime kupca	Adresa	Grad	Država
1	Thomas Hardy	120 Hanover Sq.	London	UK
2	Maria Anders	Obere Str. 57	Berlin	Njemačka
3	Marko Ivanović	Vukovarska 21	Zagreb	Hrvatska
4	Antonio Moreno	Mataderos 2312	México D.F.	Mexico

Većina naredbi za upravljanje bazom podataka obavlja se preko SQL izjava (naredbi). Naredbom na slici 4.1 odabiru se svi zapisi u tablici Kupc“. Neke od najbitnijih SQL naredbi prikaze su u tablici 4.2.

```
SELECT * FROM Kupci;
```

Sl. 4.1. *SQL naredba SELECT*

Tab. 4.2. *Tablica SQL naredbi*

SQL naredba	Opis
SELECT	Izvlači podatke iz baze
UPDATE	Ažurira podatke iz baze
DELETE	Briše podatke iz baze
INSERT INTO	Stavlja nove podatke u bazu
CREATE DATABASE	Pravi novu bazu
ALTER DATABASE	Modificira postojeću bazu
DROP TABLE	Briše postojeću bazu
CREATE INDEX	Kreira indeks (ključ za pretraživanje)
DROP INDEX	Briše indeks

MySQL također podržava različite tipove podataka. Vrsta podatka definira kakvu vrijednost stupac može sadržavati. Svaki stupac u tablici baze podataka mora imati naziv i vrstu podatka.

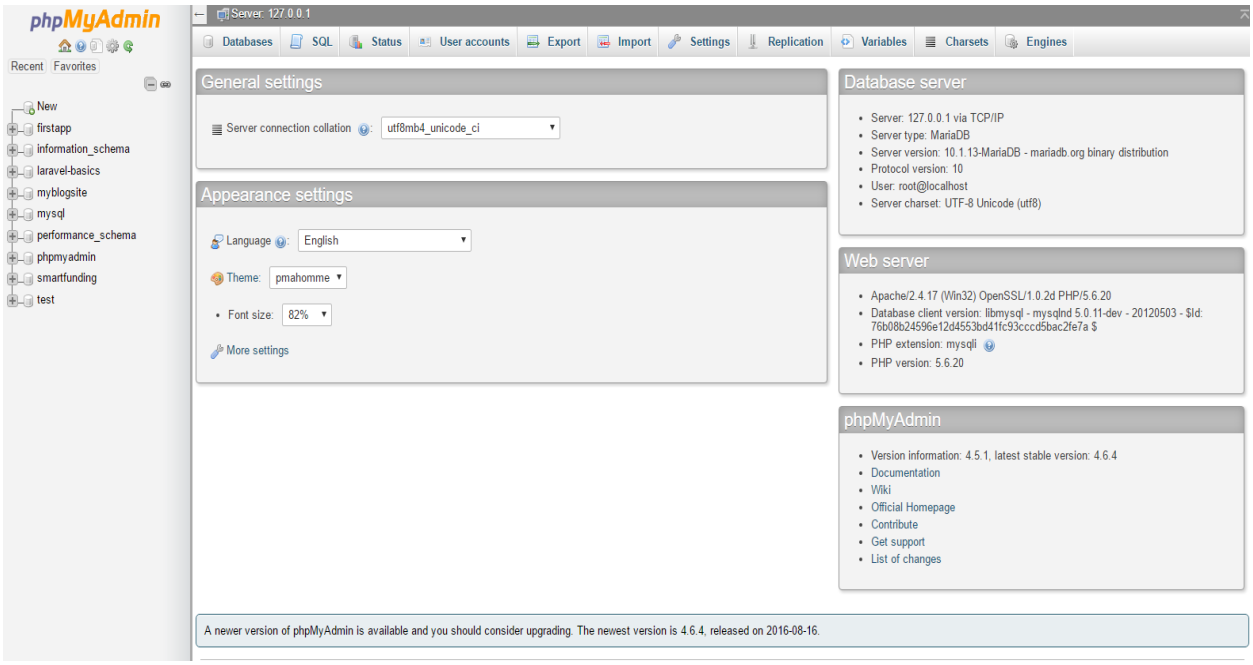
Pri kreiranju SQL tablice programer odlučuje koji tip podataka će biti spremljen u svaki stupac. Tip podatka je oznaka i vodič za SQL da razumije koji tip podatka se očekuje u svakom stupcu te također određuje kakvu će interakciju SQL imati s pohranjenim podacima. U tablici 4.3 prikazani su tipovi podataka u SQL-u.

Tab. 4.3. *Tipovi podatka u SQL*

Tip podatka	Opis
CHARACTER (n)	Niz znakova. Fiksne duljine n.
VARCHAR (n)	Niz znakova. Varijabilne duljine. Maksimalna duljina n.
BINARY (n)	Binarni niz. Fiksne duljine n.
BOOLEAN	Pohranjuje istina ili laž.
VARBINARY (n)	Binarni niz. Varijabilne duljine. Maksimalna duljina n.
INTEGER (p)	Cijeli broj. Preciznost p.
SMALLINT	Cijeli broj. Preciznost 5.
INTEGER	Cijeli broj. Preciznost 10.
BIGINT	Cijeli broj. Preciznost 19.
DECIMAL (p, s)	Decimalni broj sa p znakova i s decimala.
FLOAT (p)	Realni broj s decimalnim djelom s preciznošću p.
REAL	Realni broj. Preciznost 7.
FLOAT	Realni broj. Preciznost 16.
DATE	Sprema godinu, mjesec i dan.
TIME	Sprema sate, minute i sekunde.
TIMESTAMP	Sprema godinu, mjesec, dan, sate, minute i sekunde.
INTERVAL	Predstavlja period vremena.

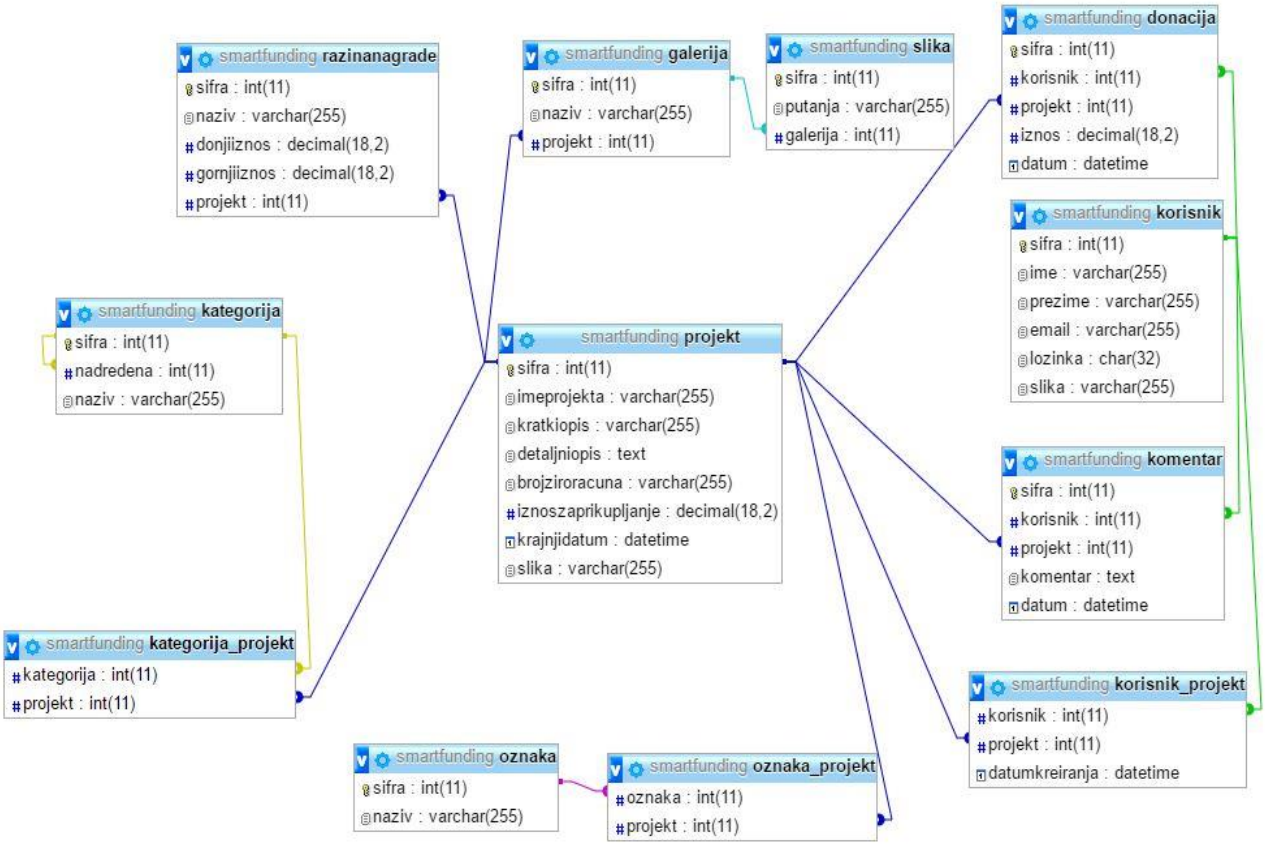
4.1 phpMyAdmin

U „XAMPP Control Panelu“ pokreću se Apache i MySQL serveri. Nakon toga u web pregledniku se upisuje „localhost/phpmyadmin“ kako bi se pristupilo grafičkom sučelju baze podataka. Početna stranica „phpMyAdmina“ je prikazana na slici 4.2.



Sl. 4.2. *phpMyAdmin početna stranica*

Baza podataka se sastoji od 12 tablica: donacija, galerija, kategorija, kategorija projekt, komentar, korisnik, korisnik projekt, oznaka, oznaka projekt, projekt, razina nagrade i slika. Na slici 4.3. prikazane su svih 12 vizualno povezanih tablica.



Sl. 4.3. *Baza podataka*

5. PAYPAL SANDBOX

PayPal Sandbox je samoodrživa testna okolina koja imitira stvarno PayPal okruženje. Pruža zaštićeni prostor u kojem je moguće pokrenuti i gledati proces prijave i zahtjeva koji je napravljen bez doticaja sa stvarnim novcem i stvarnim PayPal računima.

PayPal sandbox nudi značajke koje se nalaze na PayPal proizvodnim poslužiteljima. Dok su neke PayPal značajke ne primjenjuju na sandbox (kao što je zatvaranje računa, izdavanje mjesečnih izvješća, spremanje postavki dostave i podršku PayPal trgovine), moguće je testirati PayPal procese znajući da će se ponašati isto kao i na proizvodnim poslužiteljima, odnosno kao što to rade u sandbox okruženju.

Korištenjem izmišljenih sandbox testnih računa i njihove povezanosti sa provjerom autentičnosti vjerodajnica u PayPal API pozivima, moguće je testirati te ispravljati rad aplikacije bez spominjanja i korištenja pravih PayPal korisnika ili računa. Sandbox omogućuje rad s aplikacijom u sigurnom okruženju i pruža podešavanje PayPal rutine prije premještanja aplikacije u proizvodnju.

Kada je transakcija pokrenuta korištenjem sandbox testnih računa, PayPal stvara lažnu transakciju koja se ponaša upravo kao transakcije u pravom okruženju. Sve sandbox transakcije su praćene na testnoj stranici https://www.sandbox.paypal.com kao što se stvarne transakcije prate na https://www.paypal.com. Sandbox okruženje se sastoji od testne stranice i stranice sa testnim računima. Stranica sa testnim računima omogućava stvaranje i upravljanje skupom testnih računa, a na testna stranica se koristi za pregled transakcija.

Sandbox testni računi su virtualni te postoje samo u sandbox okruženju. Oni predstavljaju korisničke osobe koje sudjeluju u lažnim transakcijama koje stvaramo u testnoj fazi. Potrebno je napraviti barem jedan testni račun za svakog korisnika koji je zastupljen u transakciji koja je obavljena. Sandbox podržava dva tipa različitih računa:

- Osobni, predstavlja kupca ili pošiljatelja u transakciji
- Poslovni, predstavlja trgovca ili primatelja u transakciji

Također postoji opcija skupni prijenos, koji pruža način stvaranja do 40 sandbox testnih računa odjednom. Konfiguriranjem više testnih računa, moguće je simulirati sve moguće scenarije plaćanja. Konfiguracijom različitih osobnih računa moguće je simulirati lažne kupce kao potvrđene, ne potvrđene, korisnike kreditnih kartica, PayPal korisnike i slično. Konfiguriranjem

poslovnih testnih računa da predstavljaju trgovce, trgovce treće strane i API pozivatelje. Izradom poslovnih testnih računa omogućeno je pokrivanje različitih načina plaćanja koji trgovci prihvaćaju. Konfiguracijom mješavine testnih računa, moguće je pokriti cijeli niz scenarija za plaćanje u testnom okruženju.

6. IZGLED STRANICE I PRIMJENA

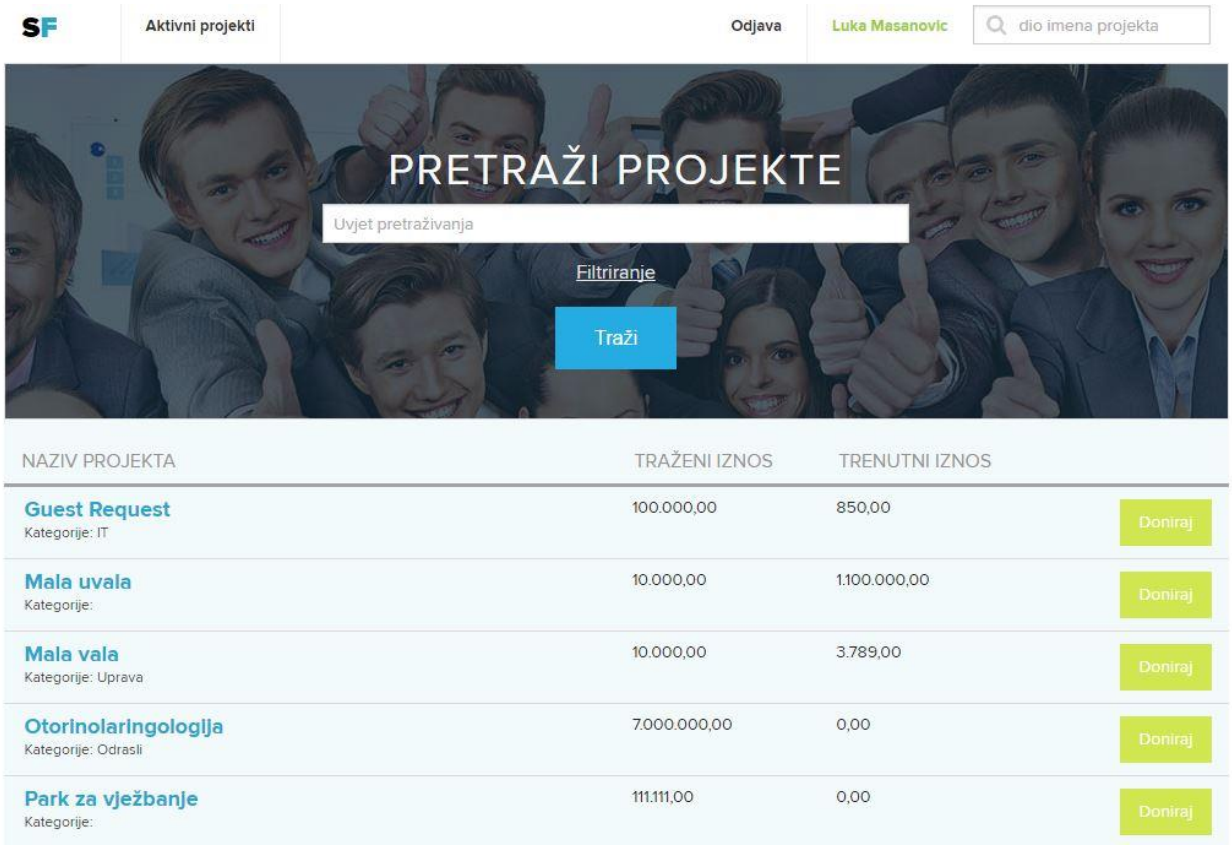
Kao rezultat ovog završnog rada je stranica za grupno financiranje ideja. Pri izradi stranice korištene su tehnologije opisane u prijašnjim poglavljima. Na početnoj stranici u navigaciji se nalazi linkovi za aktivne projekte, prijavu, registraciju te tražilica kao što je prikazano na slici 6.1.

Sl. 6.1. *Početna stranica*

Prilikom dolaska na početnu stranicu korisnik se treba registrirati ukoliko želi kreirati vlastiti projekt ili podržati postojeći. Registracija traži od korisnika osnovne informacije kao što su ime, prezime, email te lozinku kao što je prikazano na slici 6.2.

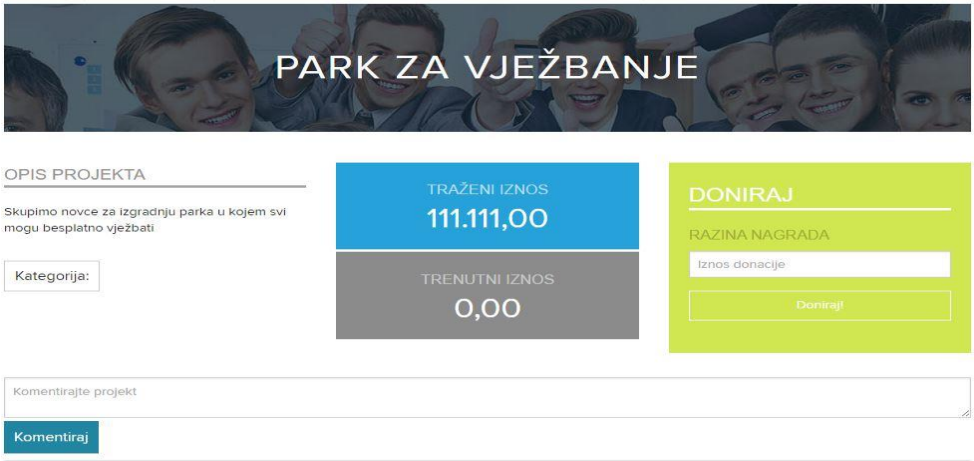
Sl. 6.2. *Registracija*

Nakon obavljene registracije i prijave na stranicu, linkom iz navigacije Aktivni projekti korisnika se prebacuje na stranicu trenutno aktivnih projekata gdje je također moguće pretraživati projekte u tražilici. Na stranici su vidljivi naziv projekta, traženi iznos i trenutni iznos donacija.



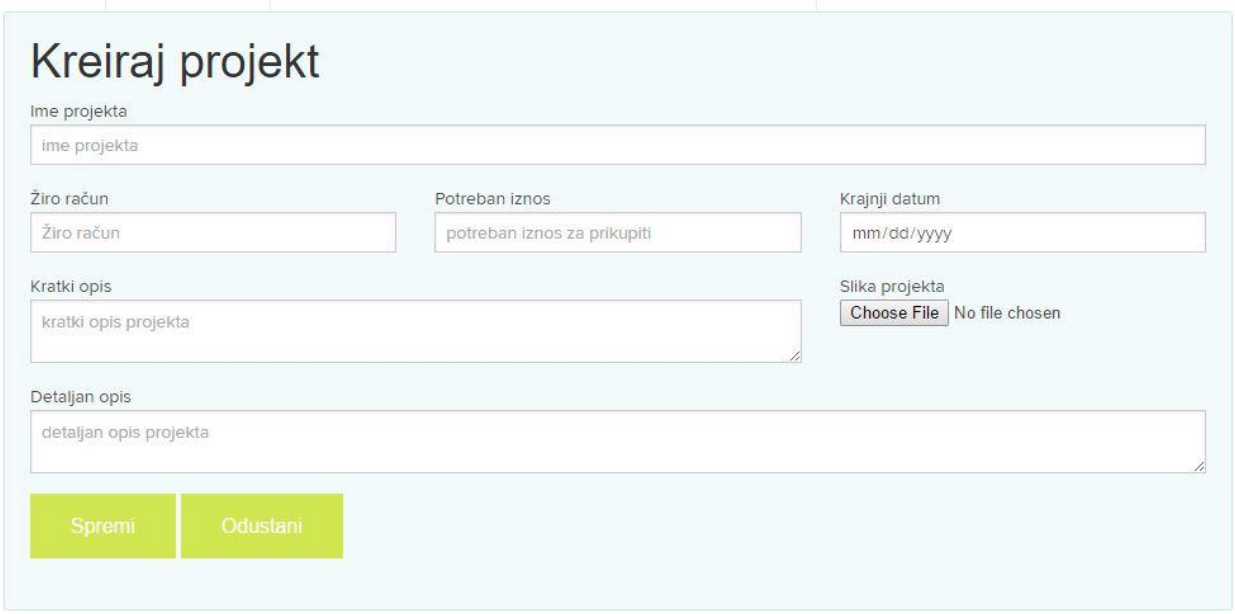
Sl. 6.3. *Aktivni projekti*

U koliko se korisnik odluči donirati određenom projektu, klikom na dugme Doniraj korisnika se prebacuje na stranicu u kojoj ima detaljniji opis projekta i polje za unos donacije. Također moguće je i komentirati aktivni projekt.



Sl. 6.4. *Donacija projektu*

Korisnik također može kreirati svoj projekt, na početnoj stranici se nalazi dugme Kreiraj projekt koje korisnika odvodi na stranicu za pravljenje novog projekta. Potrebno je unijeti ime projekta, žiro račun, traženi iznos, datum do kojeg je trajanje projekta predviđeno, kratki i detaljni opis. Nakon kreiranja, projekt se nalazi na stranici aktivni projekti te drugi korisnici mogu vršiti donacije.



Sl. 6.5. *Kreiranje novog projekta*

7. ZAKLJUČAK

U ovom završnom radu napravljena je web stranica čija je tema bila crowdfunding. Bilo je potrebno koristiti određene tehnologije kao što su PHP, HTML, SASS, MySQL, XAMPP te se upoznati sa PayPal Sandbox virtualnim okruženjem koje omogućuje pregled virtualne transakcije. Na početnoj stranici nalazi se dio za prijavu i registraciju korisnika, dio za pretraživanje postojećih projekata te dio za pregled aktivnih projekata. Kako bi korisnik napravio donaciju, podržao postojeći ili kreirao vlastiti projekt mora obaviti registraciju. Pod registracijom je potrebno unijeti osnovne podatke kao što su ime, prezime, email adresu te lozinku. Nakon registracije, na stranici podrži projekte, korisnik ima uvid u sve trenutno aktivne projekte i njihovo financijsko stanje, odnosno koliki im je iznos potreban za uspjeh te iznos koji trenutno imaju. Na stranici kreiraj projekt korisnik mora unijeti ime projekta, potreban iznos koji želi prikupiti, datum do kojeg će projekt biti aktivan odnosno krajnji rok za prikupljanje novca, žiro račun, opis projekta i sliku po mogućnosti. Tijekom izrade ovog završnog rada nailazi se na razne probleme, koje je moguće riješiti pretraživanjem na internetu ili konzultacijom s mentorom i drugim kolegama.

LITERATURA

[1] *Wiki* o PHP-u, https://hr.wikipedia.org/wiki/PHP, lipanj 2016.

[2] Uvod u PHP, http://php.com.hr/77, lipanj 2016.

[3] *Wiki* o HTML-u, https://hr.wikipedia.org/wiki/HTML, lipanj 2016.

[4] *Wiki* o MySQL-u, https://hr.wikipedia.org/wiki/MySQL, lipanj 2016.

[5] PHP petlje, http://php.com.hr/88, lipanj 2016.

[6] W3schools, stranica za online učenje, http://www.w3schools.com/, lipanj 2016.

[7] PayPal stranica za *developere*, https://developer.paypal.com/ , kolovoz 2016.

[8] Službena stranica Sass-a, http://sass-lang.com/documentation/, kolovoz 2016.

SAŽETAK

U radu je objašnjena ideja i izrada stranice za grupno financiranje. Objašnjene su tehnologije koje su korištene pri izradi ovakve stranice. Dan je izgled baze podataka koja je bila ključna pri izradi stranice. Pomoću PHP-a i MySQL-a odrađena je logika i manipulacija bazom podataka na stranici, a za dizajn je korišten HTML i Sass. Na stranici se korisnik može registrirati, prijaviti, dodavati projekte, komentirati, kreirati vlastite projekte i donirati projektu po želji. PayPal sandbox korišten je pri stvaranju virtualnog okruženja koje omogućava praćenje transakcija virtualnom novcem odnosno simulira prave transakcije koje bi se koristile kada bi projekt bio pušten u proizvodnju.

Ključne riječi: Grupno financiranje, PayPal, PHP/MySQL, Sass, sandbox

ABSTRACT

Developing Website for Crowdfunding

The paper explains idea and creation of the web page for crowdfunding. Paper also explains technologies which are used in the creation of website. Design and structure of the database is also shown, which was crucial in building web page. PHP and MySQL worked out the logic and manipulation of the database on the website, HTML and Sass were used for design. On this page users can register, log in, add projects, comment, create their own projects and donate to the project. PayPal sandbox is used to create a virtual environment that allows you to track transactions. Sandbox is using virtual money to simulate real transaction.

Keywords: Crowdfunding, PayPal, PHP/MySQL, Sass, sandbox

ŽIVOTOPIS

Luka Mašanović rođen je 15.4.1994. g. u Osijeku. Nakon završene osnovne škole Frana Krste Frankopana u Osijeku, upisuje Elektrotehničku i prometnu školu Osijek, smjer tehničar za računalstvo u Osijeku te maturira 2013.g. Iste godine upisuje sveučilišni studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, smjer računarstvo. Govori i piše engleski jezik. Poznaje rad na računalu, odlično se služi sa MS Office alatima kao što su Word, Excel, Power Point. Poznaje razne web tehnologije PHP, HTML, CSS, MySQL. U slobodno vrijeme zanima se za web development, web dizajn i nove tehnologije.

Potpis:

_____

PRILOZI

Na CD-u priloženom uz Završni rad nalaze se:

Završni rad.docx

Završni rad.pdf

Direktorij web stranica: datoteka u kojoj se nalaze svi dokumenti vezani za web stranicu.