Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom

Tarnaj, Marko

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:200:854910

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Stručni studij

INTERNET APLIKACIJA ZA UPRAVLJANJE OBITELJSKIM POLJOPRIVREDNIM GOSPODARSTVOM

Završni rad

Marko Tarnaj

Osijek, 2016.

ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na stručnom studiju

Osijek,

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na stručnom studiju

Ime i prezime studenta:	Marko Tarnaj
Studij, smjer:	Stručni studij, informatika
Mat. br. studenta, godina upisa:	0165051891, 2013.
Mentor:	Doc.dr.sc. Josip Balen
Sumentor:	
Predsjednik Povjerenstva:	
Član Povjerenstva:	
Naslov završnog rada:	Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom
Primarna znanstvena grana rada:	
Sekundarna znanstvena grana (ili polje) rada:	
Zadatak završnog rada	U teorijskom dijelu rada potrebno je proučiti i opisati tehnologije za izradu Internet aplikacija (HTML, CSS, JavaScript, PHP). U praktičnom dijelu rada potrebno je izraditi Internet aplikaciju za upravljanje obiteljskim poljoprivrednim gospodarstvom. Internet aplikacija treba imati bazu podataka (proizvoda) i mogućnost Internet narudžbe i kupovine proizvoda.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: Postignuti rezultati u odnosu na složenost zadatka: Jasnoća pismenog izražavanja: Razina samostalnosti:

Potpis sumentora:	Potpis mentora:

Dostaviti:

1. Studentska služba

U Osijeku, godine Potpis predsjednika Odbora:

ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

IZJAVA O ORIGINALNOSTI RADA

Osijek,

Ime i prezime studenta:	Marko Tarnaj
Studij :	Stručni studij informatike
Mat. br. studenta, godina upisa:	0165051891, 2013.

Ovom izjavom izjavljujem da je rad pod nazivom**:** Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom

izrađen pod vodstvom mentora Doc.dr.sc. Josipa Balena

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
2. PREGLED KORIŠTENIH TEHNOLOGIJA 2
2.1. HTML opisni jezik 2
2.2. CSS opisni jezik 2
2.3. JavaScript programski jezik 3
2.4. PHP programski jezik 4
2.5. Baza podataka i SQL programski jezik 4
3. RAZVOJ APLIKACIJE ZA UPRAVLJANJE OPG–OM 7
3.1. Korisnički zahtjevi 7
3.2. Razvoj Internet stranice – zajednički elementi 7
3.3. Razvoj Internet stranice – sadržaj 12
3.4. Razvoj Internet aplikacije 20
4. ZAKLJUČAK 26
LITERATURA 27
SAŽETAK 28
ABSTRACT 28
ŽIVOTOPIS 29

1. UVOD

Ulaskom Republike Hrvatske u Europsku Uniju otvorila su se vrata mnogim gospodarstvima da iskoriste fondove iz Europske Unije kako bi modernizirali i promovirali svoje gospodarstvo. U svrhu reklamiranja i prodaju gospodarskih proizvoda izrađena je "Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom" (u daljnjem tekstu OPG). Internet aplikacija će korisnicima pružiti jednostavan način za upravljanje svojim gospodarstvom na internetu. Korisnik aplikacije moći će:

- Prezentirati svoje gospodarstvo posjetiteljima na Internetu kroz razne medije,
- Prezentirati svoje gospodarske proizvode, te imati mogućnost upravljanja proizvodima (stvaranje novih, brisanje i uređivanje postojećih)
- Omogućiti posjetiteljima Internet narudžbu i kupovinu

Cilj izrade završnog rada je prikazati stečena znanja iz područja Internet programiranja i baza podataka. Jezici korišteni za izradu Internet stranica su HTML i CSS, a za funkcionalnost određenih elemenata korišteni su JavaScript i PHP programski jezici. Za upravljanje i rad s bazom podataka potrebno je poznavanje SQL programskog jezika. U teorijskom dijelu su navedene i opisane tehnologije korištene za izradu Internet aplikacije, kao i sam proces izrade. Za izradu čitavog rada koristit će se probni sadržaj (*engl. sample content*), koji je nakon prezentiranja subjekt promjene.

Poglavlje 2 opisuje korištene programske jezike i alate za izradu i testiranje aplikacije. Poglavlje 3 pruža pregled koraka razvoja Internet aplikacije i Internet stranice na temelju pretpostavljenih korisničkih zahtjeva.

1.1. Zadatak završnog rada

U teorijskom dijelu rada potrebno je proučiti i opisati tehnologije za izradu Internet aplikacija (HTML, CSS, JavaScript, PHP). U praktičnom dijelu rada potrebno je izraditi Internet aplikaciju za upravljanje obiteljskim poljoprivrednim gospodarstvom. Internet aplikacija treba imati bazu podataka (proizvoda) i mogućnost Internet narudžbe i kupovine proizvoda.

2. PREGLED KORIŠTENIH TEHNOLOGIJA

2.1. HTML opisni jezik

HTML (*engl. HyperText Markup Language*) nije programski jezik, već opisni jezik (*engl. markup language*) i zbog tog svojstva ne može obavljati funkcije, izvršavati zadatke, vrtiti programske petlje, niti obavljati računske operacije. Također, važno je napomenuti da postoje određene oznake koje se koriste kako bi se obični tekst razlikovao od opisnih riječi. Oznake govore Internet pregledniku kako rasporediti sadržaj na Internet stranici i kako ona treba izgledati [1]. Nadalje, uloga HTML-a je da predstavlja kostur Internet stranice, a kasnije se određenim alatima, kao što su CSS i JavaScript, na kostur dodaju novi elementi kako bi stranica na kraju imala cjelovitu strukturu.

Pošto je HTML jezik označavanja, jedini alati potrebni za izradu Internet stranice su tekstualni uređivač i Internet preglednik. Internet preglednik će interpretirati ono što je napisano u tekstualnom dokumentu, a zadatak Internet dizajnera je urediti tekstualni dokument po nekoj ideji koju ima. Pošto na trenutnom tržištu postoji nekoliko različitih popularnih Internet preglednika, važno je stranicu testirati na više Internet preglednika, jer nisu svi jednaki i neće svi jednako interpretirati ono što je napisano u tekstualnom dokumentu. Još od 2011. godine najpopularnija dva preglednika su Google Chrome i Mozilla Firefox. Rezultati statističkih istraživanja prikazuju kako više od 85% korisnika na Internetu koristi ili Chrome ili Firefox [2]. Zbog navedenih razloga, Internet dizajneru je cilj svoju stranicu učiniti funkcionalnu na barem dva najpopularnija Internet preglednika.

2.2. CSS opisni jezik

CSS (*engl. Cascading Style Sheets*) je također opisni jezik. Iako HTML ima mogućnost uređivanja i stiliziranja elemenata, popularnost razvoja Internet stranica zahtjevala je nove standarde i nove mogućnosti u dizajnu. Upravo zato se razvio CSS koji uvelike proširuje mogućnosti u Internet dizajnu. CSS kôd sastoji se od tri glavna elementa[3]:

- Selektori,
- Svojstva,
- Vrijednosti.

Selektorima se referenciraju određeni elementi HTML stranice, svojstvima opisujemo pojedine atribute kao što su margine, veličina fonta, boja i sl., a vrijednostima definiramo veličine koju pojedino svojstvo može poprimiti.

CSS kôd moguće je povezati sa HTML dokumentom na nekoliko načina [4]:

1. Umetanje u zaglavlje HTML dokumenta,
2. Dodavanje unutar linije HTML koda,
3. Povezivanje s vanjskim dokumentom,
4. Umetanje vanjskog dokumenta.

U praksi se najobičnije koristi povezivanje s vanjskim dokumentom. To znači da se čitav CSS kôd piše u zasebnu tekstualnu datoteku, a zatim se u HTML dokumentu upiše jedna linija kôda koja povezuje dva dokumenta. Ova metoda pruža najveću preglednost i najlakše snalaženje u kôdu [5].

2.3. JavaScript programski jezik

JavaScript je objektno orijentirani programski jezik koji se izvodi na klijentskom računalu (*engl. client side*). Kao i HTML i CSS, JavaScript je već unaprijed podržan od strane Internet preglednika [6]. JavaScript je skriptni jezik, što znači da ga nije potrebno prevesti prije izvođenja (*engl. compile*), već će Internet preglednik čitati skriptu liniju po liniju i prema tome ju izvršiti na stranici. Svrha korištenja JavaScript-a je omogućavanje interaktivnosti između više elemenata stranice. Također uvodi određeni stupanj dinamičnosti na Internet stranicu, kako bi se posjetiteljima stranice pružila veća doza interaktivnosti sa stranicom [7].

Kao i povezivanje CSS-a, JavaScript kôd je također moguće implementirati na nekoliko načina [8]:

- Spremiti ga unutar samog HTML dokumenta,
- Upisati ga u polje za adresu u Internet pregledniku,
- Spremiti ga u posebnu JavaScript datoteku.

Najpopularnija metoda je spremanje JavaScript kôda u zasebnu datoteku, jer pruža veću preglednost i snalaženje u okruženju. Unutar HTML dokumenta važno je pomoću <*script*> oznake

obavijestiti Internet preglednik da slijedi skriptni jezik. Unutar <*script*> oznaka piše se JavaScript kôd.

2.4. PHP programski jezik

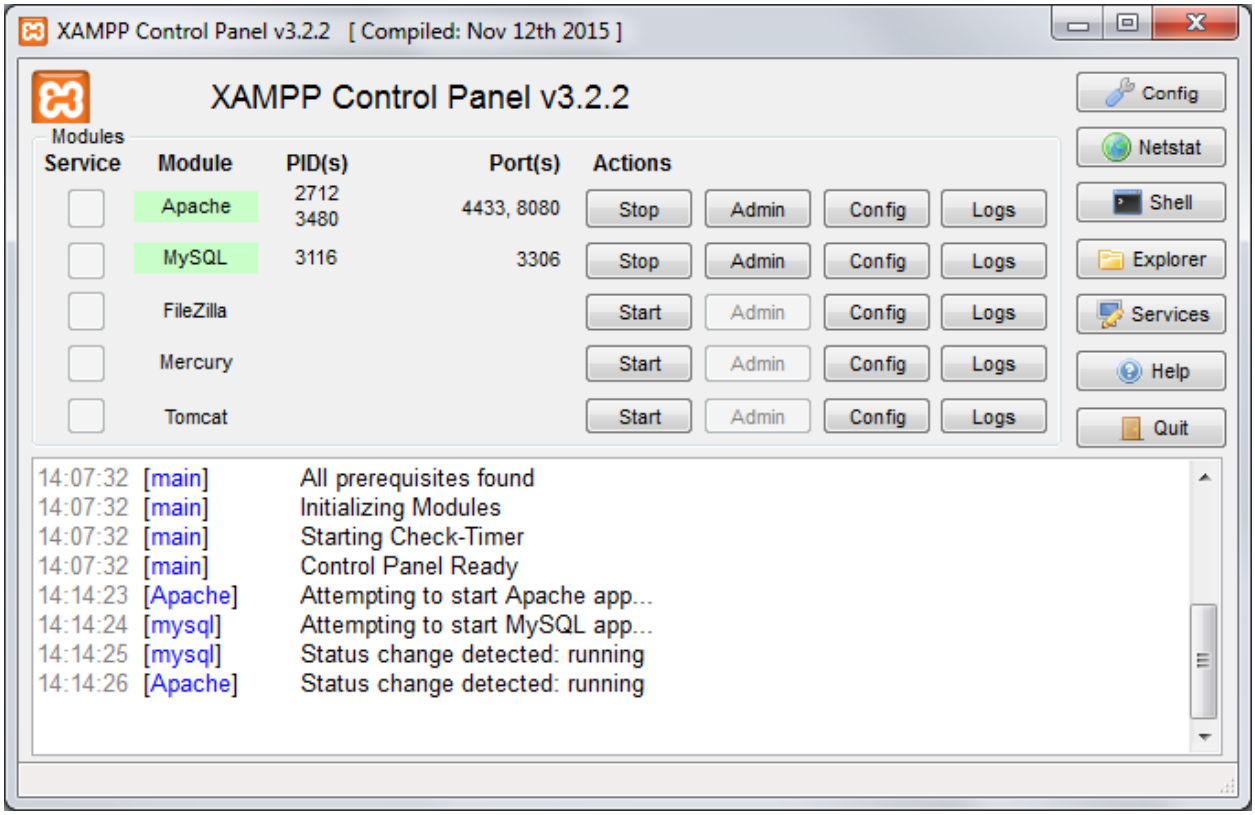
PHP (*engl. Hypertext Preprocessor*) je skriptni programski jezik otvorenog koda za dinamičko generiranje HTML kôda [9]. Za razliku od JavaScript-a, skripte se izvode na serveru (*engl. server side*) umjesto na klijentskom računalu. Drugim riječima, PHP je skriptni programski jezik pomoću kojeg možete kreirati HTML stranicu na serveru prije nego što je ona poslana klijentu popunjenu dinamičkim sadržajem. Ovim načinom generiranja sadržaja klijent ne može vidjeti kôd (skriptu) koji je generirao sadržaj koji gleda, već ima pristup samo čistom HTML kôdu. Pošto je PHP otvorenog kôda, to znači da svi imaju mogućnost skinuti izvorne PHP kôdove pisane u C programskom jeziku i, ukoliko ih razumije, može ih mijenjati po svojoj volji te dodavati nove funkcije PHP-u [10].

PHP je jedna od najnaprednijih i najkorištenijih skriptnih tehnologija koje se izvode na serveru (*engl. server side*). Najvažnija uloga PHP-a je bogatstvo funkcija za manipuliranje mnogo različitih tipova sadržaja. Od manipuliranja grafikom (png, jpg, flash,…) do učitavanja .NET modula i rada sa XML-om. Ono što PHP stavlja još više ispred ostalih Internet skriptnih tehnologija je njegova podrška za baratanje širokom paletom baza podataka. Podržava sve popularnije baze podataka kao što su MySQL, PostgreSQL, dBase, Oracle, ODBC [11].

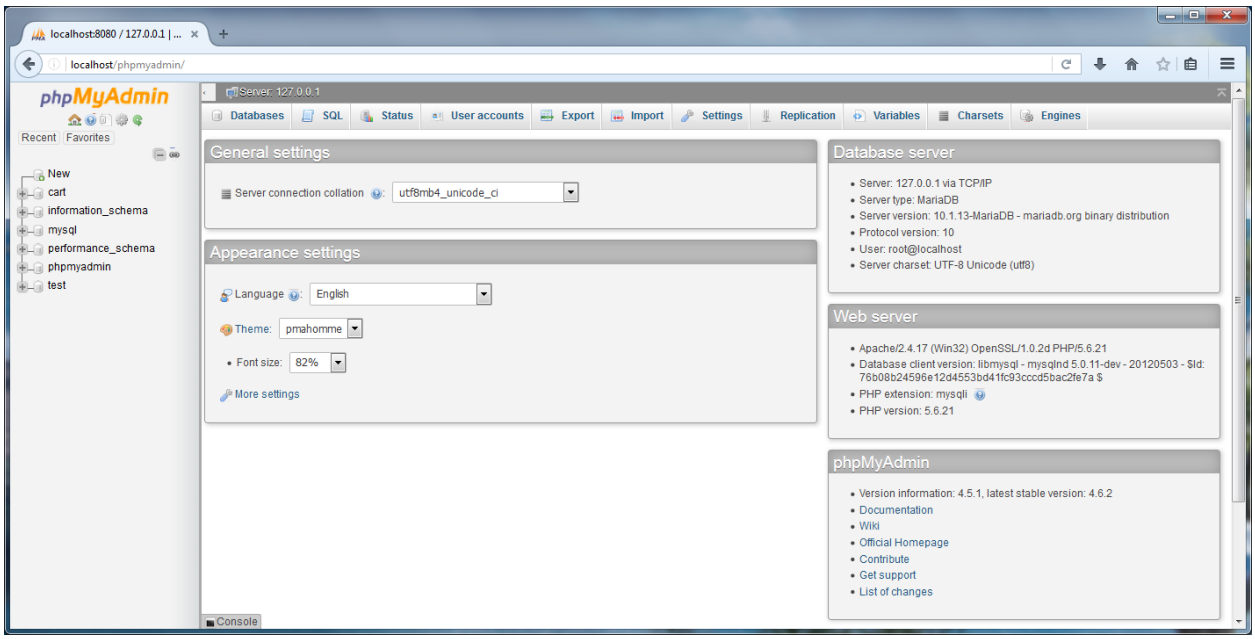
2.5. Baza podataka i SQL programski jezik

Za potrebu izrade aplikacije, potrebna je baza podataka koja sadrži tablicu gospodarskih proizvoda koje vlasnik obiteljskog poljoprivrednog gospodarstva želi prodavati. Pri izradi i testiranju aplikacije, veoma popularan alat je XAMPP. To je besplatan program otvorenog kôda koji omogućava korisniku kreiranje lokalne baze na svom računalu. Iako samo korisnik računala ima pristup toj bazi, XAMPP uvelike pomaže programeru uvidjeti greške u programu kako bi ih mogao ispraviti prije nego bazu podataka napravi na pravom serveru [12].

Nakon pokretanja XAMPP programa, potrebno je pokrenuti Apache i MySQL servise (slika 2.1.) kako bi mogli pristupiti kontrolnoj ploči baze podataka. Kontrolnoj ploči se pristupi tako da se u adresu Internet pretraživača upiše *localhost/phpmyadmin* (slika 2.2.).



Slika 2.1. – XAMPP kontrolna ploča



Slika 2.2. – Kontrolna ploča baze podataka

Za rad u bazi podataka potrebno je poznavanje SQL (*engl. Structured Query Language*) programskog jezika. On daje mogućnosti kreiranja tablica u bazi, umetanja podataka u tablice, brisanje i uređivanje podataka, i mnogo drugih korisnih naredbi [13]. Iako je kontrolna ploča dizajnirana tako da za rad u bazi nije potrebno poznavanje programskog jezika SQL jer se sve naredbe mogu uraditi preko korisničkog sučelja, na dnu stranice se nalazi konzola u koju se mogu direktno unositi SQL upiti i naredbe, te je svakako poželjno poznavanje SQL programskog jezika za rad s bazom.

3. RAZVOJ APLIKACIJE ZA UPRAVLJANJE OPG–OM

3.1. Korisnički zahtjevi

Prije početka razvoja aplikacije, potrebno je razmotriti tko će koristiti aplikaciju. Korisnici se mogu podijeliti u dvije grupe. Prvu grupu čini jedan korisnik koji upravlja proizvodima, idealno vlasnik obiteljskog poljoprivrednog gospodarstva, ili osoba zadužena za upravljanje prodajom gospodarskih proizvoda. Za tu osobu potrebno je napraviti aplikaciju koja će koristiti PHP programski jezik, te izgraditi sučelje koje će osobi omogućiti uređivanje proizvoda, dodavanje novih i brisanje postojećih proizvoda. Prije toga potrebno je u bazi podataka napraviti tablicu.

Druga grupa korisnika su posjetitelji stranice. Stranicu treba dizajnirati prema njihovim pretpostavljenim potrebama. Internet stranica treba sadržavati navigacijsku traku, informacije o OPG-u kao logo, naziv OPG-a, adresa, telefon i e-mail adresa. Navigacijska traka vodi na sljedeće stranice: "Početna", "Kupite naše proizvode" i "Kontaktirajte nas". Na početnoj stranici korisno je imati pomičnu galeriju slika vezanih uz područje rada OPG-a, kako bi posjetitelji odmah znali čime se bavi OPG. Nakon toga slijedi tekst koji informira posjetitelja čime se točno OPG bavi. Posjetitelj nakog toga može vidjeti koji točno proizvodi se nude. Klikom na pojedini proizvod biti će preusmjeren na stranicu koja sadrži košaricu za kupovinu. Na toj stranici su također ponuđeni svi proizvodi, te se klikom na pojedine proizvode oni ubacuju u košaricu. Košarica treba biti povezana sa PayPal servisom koji omogućuje trgovanje predmetima preko Interneta. Posljednja potrebna stranica je kontakt, gdje posjetitelji mogu poslati e-mail vlasniku stranice. Potrebna polja za slanje su njihovo ime, e-mail i komentar. Također, stranica u zaglavlju sadrži poveznice na društvene mreže ukoliko ih vlasnik OPG-a koristi.

3.2. Razvoj Internet stranice – zajednički elementi

Na slici 3.1. može se vidjeti dizajn Internet stranice. Postoje određeni elementi koji će biti zajednički, dok će se prostor za sadržaj razlikovati za "Početna", "Kupite naše proizvode" i "Kontaktirajte nas". Na slici je vidljivo da zajednički elementi uključuju navigacijsku traku, poglavlje koje sadrži osnovne informacije o OPG-u, i zaglavlje namijenjeno za poveznice na društvene mreže.

Slika 3.1. – Izgled Internet stranice

Navigacijska traka (slika 3.2.) služi za navigaciju Internet stranicom. Sadržaj navigacijske trake izrađen je pomoću nenumerirane liste, kako je vidljivo u programskom kôdu 3.1. Kratice koje postoje na traci su:

- "Početna" – Početna stranica koja sadrži JavaScript klizač slika, informacije o OPG-u smještene u klasu *#info* i ponudu proizvoda,
- "Kupite naše proizvode" – stranica koja je namijenjena za Internet kupovinu, sadrži košaricu i ponudu,
- "Kontaktirajte nas" – stranica koja sadrži formu, namijenjena posjetiteljima stranice da mogu svoje mišljenje o stranici, proizvodima, ili bilo kakvim upitima poslati OPG-u,
- "O nama" – kratica koja se referencira na klasu *#info* na početnoj stranici.

Slika 3.2. – Navigacijska traka

```
<ul class="menu">
  <li><a href="http://localhost:8080/pocetna.php">Početna</a></li>
  <li><a href="http://localhost:8080/kupite.php">Kupite naše
proizvode</a></li>
  <li><a href="http://localhost:8080/kontakt.php">Kontaktirajte
nas</a></li>
  <li style="float:right"><a href="#info">O nama</a></li>
</ul>
```

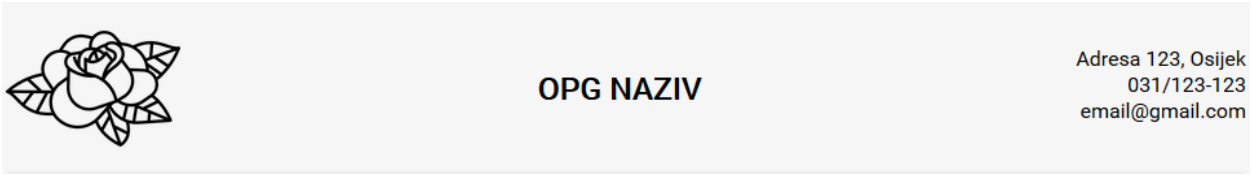
Programski kôd 3.1. – HTML kôd navigacijske trake

Kako bi cijela navigacijska traka bila u jednom redu, potrebno je napisati *display: block*, u suprotnom će lista biti u 4 reda, zbog 4 elementa unutar liste. CSS kôd za stiliziranje navigacijske trake vidljiv je u programskom kôdu 3.2. Korišteni je klasni selektor *ul.menu* što znači da će samo nenumerirana lista unutar *.menu* klase biti promijenjena [14].

```
ul.menu {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #67B23F;
    box-shadow: 0px -500px 0px 500px #67B23F}
ul li {
    float: left;
    border-right: 1px solid white}
li:first-child {
    border-left: 1px solid white}
ul li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none}
li a:hover {
    background-color: #235922}
    border-right: 1px solid white}
li:first-child {
    border-left: 1px solid white}
ul li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none}
li a:hover {
    background-color: #235922}
```

Programski kôd 3.2. – CSS kôd navigacijske trake

U poglavlju (*engl. header*) je potrebno imati informacije o stranici, tj. bitne informacije o OPG-u kao što su logo, naziv, i kontaktne informacije. Izgled poglavlja vidljiv je na slici 3.3. Na primjeru su korištene probne slike, naziv i kontaktne informacije.



Slika 3.3. – Izgled poglavlja

Cijelo zaglavlje smješteno je u jednu klasu *#top*, a svaki element smješten je u zasebne podklase. Naziv podklasa ujedno i opisuje poziciju sadržaja unutar klase. Tako npr. sadržaj klase *.column-center* smješten je u sredinu roditeljske klase. HTML dio kôda vidljiv je u programskom kôdu 3.3.

```
<div id="top">
  <div class="column-center">
    <h2>OPG NAZIV</h2>
  </div>
  <div class="column-left">
    <img src="images/logo.png">
  </div>
  <div class="column-right">
    <p>Adresa 123, Osijek</p><br>
    <p>031/123-123</p><br>
    <p>email@gmail.com</p>
  </div>
</div>
```

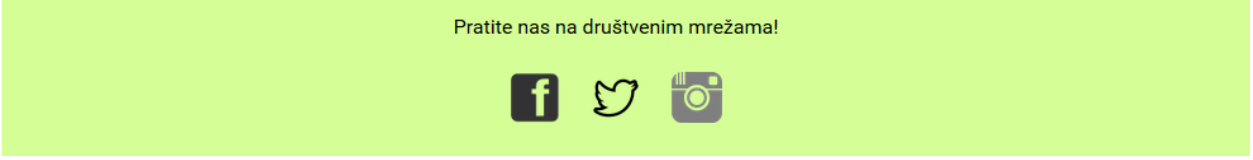
Programski kôd 3.3. – HTML kôd poglavlja

CSS dio kôda vidljiv je u programskom kôdu 3.4. Poglavlje je potrebno podijeliti u 3 dijela, stoga se koristi naredba *width: 33%* za svaki dio. Nadalje, potrebno je roditeljskoj klasi *#top* deklarirati *position: relative;*, a elementima *h2* i *p* deklarirati *position: absolute*, jer će se time *absolute* elementi referencirati samo na roditeljsku *relative* klasu [15].

```
#top {
    height: 133px;
    position: relative;
    border-width: 2px;
    border-bottom-width: 2px;
    border-bottom-color: black;
    border-bottom-style: solid}
.column-left {
    float: left;
    width: 33%;}
.column-left img {
    height: 135px;
    width: 135px}
.column-right {
    float: right;
    width: 33%}
.column-center {
    display: inline-block;
    width: 33%}
div.top {
    height: 10em;
    display: flex;
    align-items: center;
    justify-content: center}
div.column-center {
    height: 133px;
    position: relative}
div.column-center h2 {
    margin: 0;
    position: absolute;
    top: 50%;
    left: 50%;
    margin-right: -50%;
    transform: translate(-50%, -50%)}
div.column-right {
    height: 133px;
    position: relative}
div.column-right p {
    position: absolute;
    right: 0;
    margin-top: 34px}
```

Programski kôd 3.4. – CSS kôd poglavlja

Zaglavlje (*engl. footer*) je smješteno na dnu stranice, te sadrži slike kao poveznice na Internet stranice društvenih mreža. U primjeru su postavljene 3 društvene mreže: *Facebook*, *Twitter* i *Instagram*. Dizajn zaglavlja vidljiv je na slici 3.4.



Slika 3.4. – Izgled zaglavlja

Zaglavlje je smješteno u klasu *#footer*, a poveznice na društvene mreže napravljene su pomoću slika unutar <*img*> oznaka. HTML dio kôda vidljiv je u programskom kôdu 3.5., a CSS dio kôda vidljiv je u programskom kôdu 3.6.

```
<div id="footer">
  <p>Pratite nas na društvenim mrežama!</p>
  <p>
    <a href="https://www.facebook.com/" title="Facebook"><img
src="images/fb.png"></a>
    <a href="https://twitter.com/?lang=en" title="Twitter"><img
src="images/tw.png"></a>
    <a href="https://www.instagram.com/?hl=en" title="Instagram"><img
src="images/ig.png"></a>
  </p>
</div>
```

Programski kôd 3.5. – HTML kôd zaglavlja

```
#footer {
    text-align: center;
    color: black;
    background: #D6FF96;
    height: 130px;
    line-height: 20px}
#footer img {
    height: 40px;
    width: 40px;
    padding: 10px}
```

Programski kôd 3.6. – CSS kôd zaglavlja

3.3. Razvoj Internet stranice – sadržaj

Sadržaj Internet stranica razlikuje se za svaku kraticu. "Početna" stranica sadrži JavaScript klizač slika. Namijena klizača je prikazati posjetiteljima nekoliko slika kako bi se upoznali sa radom OPG-a u samo nekoliko trenutaka. Izvorni kôd za klizač slika preuzet je sa Internet stranice [16]. Primjer tranzicijskog učinka vidljiv je na slici 3.5.

Slika 3.5. – Primjer trnazicijskog učinka

Kako bi klizač uspješno funkcionirao potrebno je unutar <head> oznaka povezati HTML dokument sa JavaScript dokumentom, te je potrebano slike staviti unutar klase *#slider* (programski kod 3.7.).

```
<head>
  <link href="js-image-slider.css" rel="stylesheet" type="text/css" />
  <script src="js-image-slider.js" type="text/javascript"></script>
</head>
<div id="sliderFrame">
  <div id="slider">
    <img src="images/image-slider-1.jpg" />
    <img src="images/image-slider-2.jpg" />
    <img src="images/image-slider-3.jpg" />
    <img src="images/image-slider-4.jpg" />
  </div>
</div>
```

Programski kôd 3.7. – HTML kôd JavaScript klizača slika

U CSS dijelu kôda (programski kôd 3.8.) *width* i *height* oznake potrebno je promijeniti na istu veličinu slika. Druga opcija je slikama urediti visinu i širinu da odgovara okviru klizača. Najvažniji atribut za klizač slika je *transform*, i njegova vrijednost *translate3d(0,0,0)*. Ta vrijednost označava pomak na x, y, z osi nove slike klizača u odnosu na početnu. Vrijednosti (0,0,0) označavaju da se nove slike neće pomaknuti u odnosu na poziciju prve slike.

```
#slider {
    width: 960px;
    height: 540px;
    transform: translate3d(0, 0, 0);
    box-shadow: 0px 1px 5px #999999 }
```

Programski kôd 3.8. – CSS kôd klase *#slider*

JavaScript dio kôda sadrži unaprijed napisane funkcije za animacije i tranzicije slika.. Izvorni kôd pruža mogućnosti podešavanja tranzicijskih učinaka, kako je vidljivo u programskom kôdu 3.9. Neke od mogućih promjena su vrijeme između tranzicija *pauseTime*, biranje početne slike *startSlide*, trajanje tranzicijskog učinka *transitionTime*, podijela slike prilikom tranzicijskog efekta *slices* i *boxes*, itd. [17].

```
var sliderOptions = { sliderId: "slider", startSlide: 0, effectRandom:
false, pauseTime: 2600,  transitionTime: 500, slices: 12, boxes: 8,
hoverPause: 1, autoAdvance: true };
```

Programski kôd 3.9. – JavaScript kôd za izmjenjivanje tranzicijskih učinaka

Nakon JavaScript klizača slika slijedi klasa *#info* koja je namijenjena za tekstualni prikaz informacija o OPG-u (slika 3.6.). U primjeru je korišten probni tekst, iako je namijenjen za bilo kakve medije koje mogu posjetiteljima opisati čime se OPG bavi, kao što su zvučni mediji, video prikazi (npr. *YouTube*), brojne slike i sl.

O NAMA

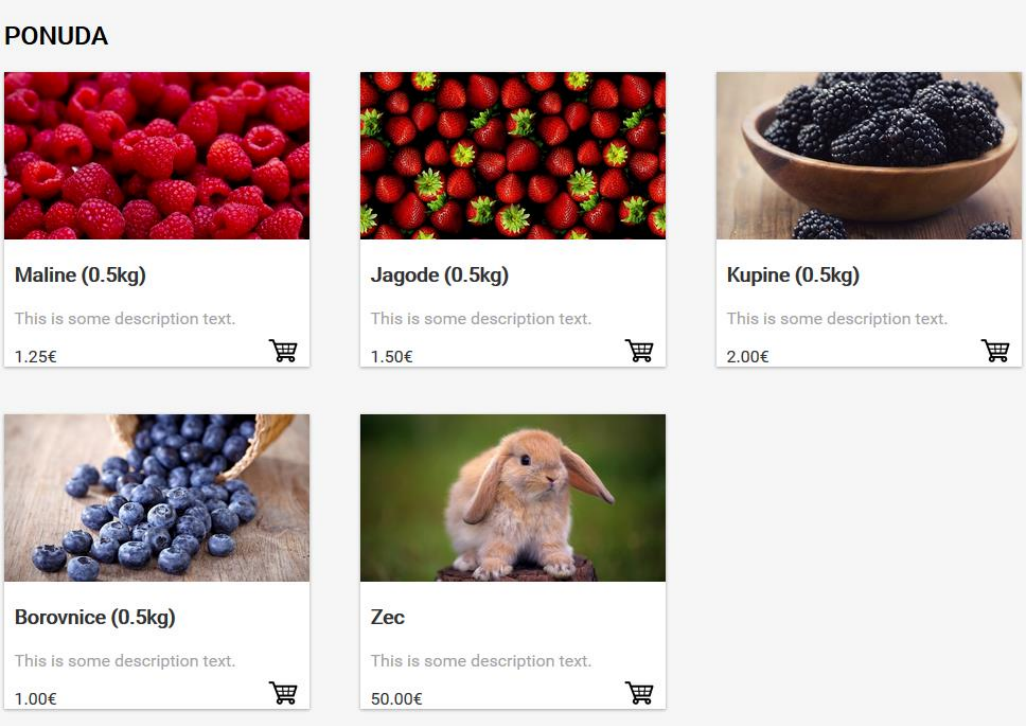
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras non lorem posuere, blandit urna sed, ultricies ex. Pellentesque eget lorem in sem condimentum malesuada vitae et enim. Nullam et fringilla ipsum. Aenean et massa ut est rutrum fringilla eget nec metus. Nulla pulvinar ex vel laoreet dictum. Fusce hendrerit venenatis eros, sed aliquet mauris rutrum in. Duis faucibus enim id justo vestibulum, eu feugiat ipsum tincidunt. Aenean ut ex posuere, tempor ligula quis, imperdiet quam. Integer id diam non urna rhoncus aliquam nec ac orci. Nunc nibh sem, ullamcorper eget ligula at, ultricies mattis magna. Proin laoreet sapien eget sagittis elementum. Morbi pellentesque erat ligula, ac ultricies purus elementum eu. Sed nec ullamcorper ante. Pellentesque rhoncus nibh ligula, lobortis hendrerit odio facilisis vitae. Mauris non consequat elit. Maecenas vitae erat mi.

Sed quam enim, mattis in metus ac, faucibus tincidunt ipsum. Proin nec risus nec nulla faucibus consequat. Fusce nec lacinia arcu. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eleifend tellus ante, venenatis volutpat tortor aliquam quis. Praesent elementum justo non bibendum pellentesque. Etiam ac ullamcorper sapien, sit amet ultricies urna. Nunc eu rhoncus elit. Aliquam eu diam facilisis, pretium turpis maximus, scelerisque tortor. In hac habitasse platea dictumst. Nullam sollicitudin tortor sit amet est tincidunt, a dictum nunc viverra. In efficitur egestas odio sit amet congue. Vestibulum ac mauris vitae dolor faucibus sagittis. Donec commodo, nisi et molestie ultricies, diam orci dapibus mauris, at tincidunt lacus tellus vitae lacus.

Integer commodo, diam et eleifend egestas, lectus dui finibus purus, id commodo mauris nisi sed nisl. Nulla sit amet lacinia mi, et fermentum libero. Quisque pharetra aliquet congue. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis porttitor ipsum nec scelerisque commodo. Ut ut ligula id risus venenatis cursus. Praesent turpis ligula, mattis id aliquet at, consequat nec ante. Aenean purus nibh, dapibus sed fringilla nec, vehicula a tortor. Vivamus consequat congue risus sit amet suscipit. Donec ut mi id ipsum sagittis rutrum. Nam maximus sapien at est volutpat imperdiet.

Slika 3.6. – "O nama"

Nakon informacija o OPG-u slijedi ponuda proizvoda koja je generirana PHP kôdom i povezana je s bazom podataka. Izgled ponude sa nekoliko primjera proizvoda prikazan je na slici 3.7. Korištene su probne slike sa interneta i probne cijene.



Slika 3.7. – Ponuda proizvoda

U HTML dijelu kôda, proizvodi se pozivaju iz baze preko PHP funkcije *<?php products(); ?>* . PHP dio kôda koji opisuje što *<?php products(); ?>* radi vidljiv je u programskom kôdu 3.10. Funkcija prvo naredbom *$get* dohvaća podatke iz baze podataka. *If* uvjet provjerava postoje li uopće podaci u bazi. Ukoliko ne postoje, ispisat će se "*U bazi ne postoje proizvodi za prikazati*". Ukoliko postoje, *while* petlja će ispisivati svaki od proizvoda jedan po jedan.

```
function products () {
  $get=mysql_query('SELECT id, name, description, price, image FROM
products WHERE quantity>0 ORDER BY id DESC');
  if (mysql_num_rows($get)==0) {
    echo "U bazi ne postoje proizvodi za prikazati";  }
  else {
    while ($get_row=mysql_fetch_assoc($get)) {
      echo '<div class="product"><div class="inner"><img style="object-fit:
cover; max-width:100%; max-height:100%" src="images/'.$get_row['image'].'"
alt=""><h3>'.$get_row['name'].'</h3><p>'.$get_row['description'].'</p><p
style="color:#333333;">'.number_format($get_row['price'], 2).'€ <a
href="cart.php?add='.$get_row['id'].'"><img style="height:30px; width:30px;
margin-top:-10px; float:right" src="images/c.png"></a></p></div></div>';
    } } }
```

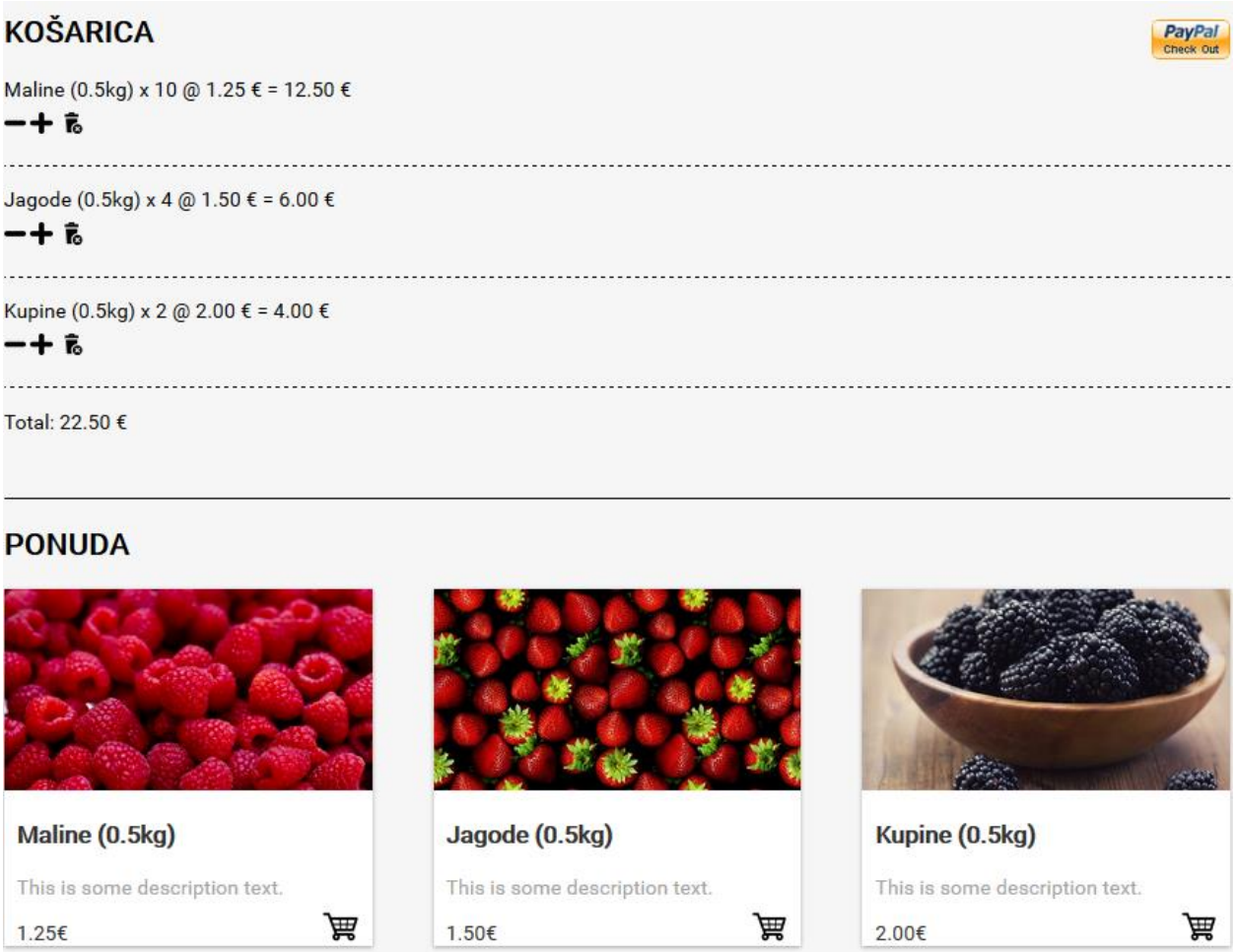
Programski kôd 3.10. – *products()* funkcija

Pri ispisu proizvoda iz baze podataka korištene su CSS klase *#product* i *#inner*. Pomoću CSS kôda proizvodi su prikazani u kutijama, te prelaskom miša preko pojedine kutije stvara se sjena pomoću naredbi box-*shadow* i *transition* u klasi *.product .inner*. Također su omogućeni tranzicijski efekti radi veće interaktivnosti s kutijama i boljeg korisničkog iskustva u klasi *.product .inner:hover*. Kao dodatan alaz za dizajniranje kutija korišten je *Box-Shadow* alat [18]. CSS kôd vidljiv je u programskom kôdu 3.11.

```
.product {
    width: 30%;
    float: left;
    margin-right: 5%}
.product:nth-child(3n) {
    margin-right: 0}
.product .inner {
    margin-bottom: 45px;
    background: white;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0,
0.24);
    transition: all 0.3s cubic-bezier(.25, .8, .25, 1);
    position: relative}
.product h3 {
    padding-left: 10px;
    padding-right: 10px;
    text-align: justify;
    color: #333333}
.product p {
    padding-left: 10px;
    padding-right: 10px;
    text-align: justify;
    color: #A4A4A4}
.product .inner:hover {
    box-shadow: 0 14px 28px rgba(0, 0, 0, 0.25), 0 10px 10px rgba(0, 0, 0,
0.22)}
```

Programski kôd 3.11. – CSS kôd za kutije proizvoda

Posjetitelji klikom na sličicu košarice na pojedinom proizvodu biti će preusmjereni na stranicu "Kupite naše proizvode". U košaricu će se automatski ubaciti proizvod sa početne stranice, ali će svi proizvodi također biti prikazani ispod košarice. Posjetitelji mogu ubacivati dodatne proizvode, a košarica računa ukupnu cijenu svih proizvoda. Ispod svakog proizvoda u košarici moguće je dodati više istih proizvoda, oduzeti proizvode, ili u potpunosti obrisati proizvod iz košarice. Izgled košarice vidljiv je na slici 3.8. Stranica je također povezana sa Internet servisom za kupovinu i trgovanje PayPal. Za to je potrebno prema PayPal pravilima koristiti specifične varijable [19].



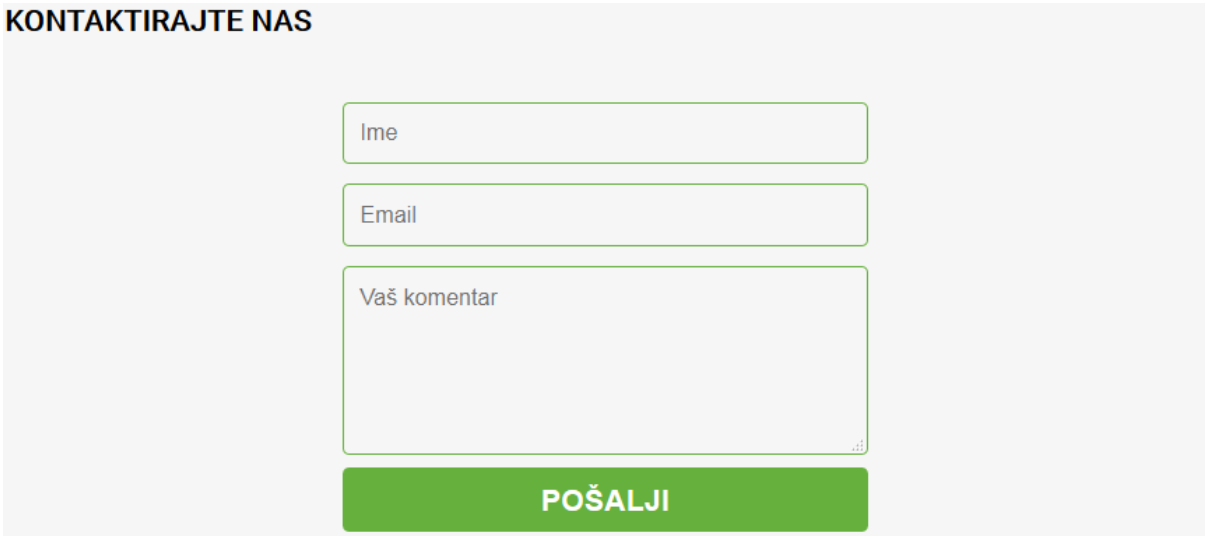
Slika 3.8. – Košarica i ponuda

PHP funkcija *products()* već je prikazana, stoga u programskom kôdu 3.12. vidljiva je PHP funkcija *cart()* koja služi za računanje ukupne cijene proizvoda. Nju se poziva pisanjem linije koda *<?php cart(); ?>* u HTML dokumentu. Za omogućavanje rada košarice, koristi se PHP sesija (*engl. session*). Sesije se koriste za pamćenje korisničkih informacija na stranici, a zasebne su za svakog korisnika [20]. Time košarica ima mogućnost pamćenja koliko je proizvoda posjetitelj stavio u košaricu, te ima mogućnost dodavati i/ili oduzimati pojedine proizvode. Funkcija *cart()* računa cijenu svakog pojedinog proizvoda na temelju količine proizvoda, te ima mogućnost zbrajanja ukupne cijene svih proizvoda iz košarice. Također, funkcija ne dozvoljava unošenje više proizvoda u košaricu nego što je količina proizvoda u bazi podataka. Time se osigurava da se prodavač ne zatekne u neugodnoj situaciji prodavanja više proizvoda nego što posjeduje, a kupac se osigurava od neuspješnog trgovanja.

```
function cart() {
  $total=0;
  foreach($_SESSION as $name=> $value) {
    if ($value>0) {
      if (substr($name, 0, 5)=='cart_') {
        $id=substr($name, 5, strlen($name)-5);
          $get=mysql_query('SELECT id, name, price FROM products WHERE
id='.mysql_real_escape_string((int)$id));
          while ($get_row=mysql_fetch_assoc($get)) {
            $sub=$get_row['price']*$value;
            echo '<p style="  border-width: 1px; border-bottom-width: 1px;
border-bottom-color: black; border-bottom-style:
dashed;">'.$get_row['name'].' x '.$value.' @
'.number_format($get_row['price'], 2).' € = '.number_format($sub, 2).' €
<br><a href="cart.php?remove='.$id.'"><img style="height:17px; width:17px;
margin-right:4px; margin-top:8px" src="images/-.png"></a><a
href="cart.php?add='.$id.'"><img style="height:17px; width:17px; margin-
right:4px; margin-top:8px" src="images/+.png"></a>  <a
href="cart.php?delete='.$id.'"><img style="height:17px;width:17px; margin-
right:8px; margin-top:4px" src="images/x.png"></a><br><br></p>';
        }}$total +=$sub;}}
  if ($total==0) {
    echo "<P>Košarica je prazna.</P>";}
  else {
    echo '<p>Total: '.number_format($total, 2).' €</p>';
    ?> <p> <form action="https://www.paypal.com/cgi-bin/Internetscr"
method="post"> <input type="hidden" name="cmd" value="_cart"> <input
type="hidden" name="upload" value="1"> <input type="hidden" name="business"
value="email_tvrtke@email.com"> <?php paypal_items();
    ?> <input type="hidden" name="currency_code" value="EUR"> <input
type="hidden" name="amount" value="<?php echo $total; ?>"> <input
style="position: absolute; margin-top:-44px; top:0; right:0;" type="image"
src="http://www.paypal.com/en_US/i/btn/x-click-but03.gif"  name="submit"
alt="Make payments with PayPal - it's fast, free and secure!"> </form> </p>
<?php }}
```

Programski kôd 3.12. – *cart()* funkcija

Sljedeća važna funkcija Internet stranice je omogućiti posjetiteljima da kontaktiraju OPG pomoću kontaktne forme koja zahtjeva posjetitelja da pomoću svog imena i e-mail adrese pošalje komentar. Povratne informacije posjetitelja uvijek su korisne na bilo kojoj Internet stranici, i uvijek je korisno imati tu mogućnost, jer omogućuje vlasniku da se prilagodi tržištu ukoliko posjetitelji to od njega zahtjevaju. Izgled kontakne forme vidljiv je na slici 3.9.



Slika 3.9. – Kontaktna forma

HTML dio kôda vidljiv je u programskom kôdu 3.13. Dvije moguće korištene metoda su *GET* i *POST*, međutim *POST* metoda se koristi za zaštitu informacija, te pruža veću sigurnost pri slanju podataka [21].

```
<form action="mail.php" method="POST">
  <input name="name" type="text" class="feedback-input" placeholder="Ime"
/>
  <input name="email" type="text" class="feedback-input"
placeholder="Email" />
  <textarea name="text" class="feedback-input" placeholder="Vaš
komentar"></textarea>
  <input type="submit" value="POŠALJI" />
</form>
```

Programski kôd 3.13. – HTML forma

Kako bi se ispunjena forma poslala, potrebno je koristiti PHP funkciju za slanje pošte. Unutar dokumenta mail.php potrebno je definirati sve varijable vezane uz slanje pošte, te ih povezati sa varijablama unutar kontaktne forme. Sadržaj *mail.php* vidljiv je u programskom kôdu 3.14.

```
<?php $name=$_POST['name'];
$email=$_POST['email'];
$text=$_POST['text'];
$formcontent="From: $name \n Message: $text";
$recipient="OPG@gmail.com";
$subject="Contact Form";
$mailheader="From: $email \r\n";
mail($recipient, $subject, $formcontent, $mailheader) or die("Error!"); ?>
```

Programski kôd 3.14. – Funkcija za slanje pošte *mail.php*

3.4. Razvoj Internet aplikacije

Nakon dizajniranja stranice za posjetitelje, potrebno je izraditi Internet aplikaciju koja omogućuje korisniku, tj. vlasniku Internet stranice da može upravljati proizvodima koje će prodavati na Internet stranici. Za izradu aplikacije potrebno je napisati PHP kôd koji se sastoji od sljedećih elemenata:

- Pregled postojećih proizvoda u bazi,
- Uređivanje postojećih proizvoda u bazi,
- Brisanje postojećih proizvoda iz baze,
- Dodavanje novih proizvoda u bazu.

Prije svega potrebno je izraditi tablicu u bazi podataka. Ime tablice je *products*, a sadrži sljedeće stupce:

- *id* – tipa INT(11). Id je također i primarni ključ tablice, te je podešen da pri unosu novih proizvoda bude automatski povećan u odnosu na prijašni proizvod. To osigurava da svaki proizvod ima svoj unikatni primarni ključ,
- *name* – tipa VARCHAR(50). Ovaj stupac predviđen je za ime proizvoda,
- *description* – tipa TEXT. Ovaj stupac predviđen je za opis proizvoda,
- *price* – tipa FLOAT, predstavlja cijenu proizvoda,
- *shipping* – tipa FLOAT, predstavlja cijenu dostave,
- *quantity* – tipa INT(11), predstavlja količinu proizvoda. Ovaj parametar koristi da posjetitelji stranice ne mogu zatražiti u košaricu više proizvoda nego što je dostupno,
- *image* – tipa TEXT, služi za pohranjivanje imena slika u bazu. Slike nije praktično spremati u bazu zato što zauzimaju previše prostora. Elegantnije rješenje je pohranjivanje slika u datoteku, a u bazu pohraniti putanju slike.

Nadalje, za uspješan rad aplikacije potrebno je napisati kôd za spoj na bazu. Taj kôd mora bit uključen (*engl. incude*) na svim drugim stranicama. Programski kôd za spoj na bazu vidljiv je u programskom kôdu 3.15. Pošto se radi na lokalnoj bazi podataka koristeći XAMPP, korisnički račun je 'root' bez ikakve šifre.

```php
<?php $server='localhost'; $user='root'; $pass=''; $db='cart';
$connection=mysql_connect($server, $user, $pass) or die ("Could not connect
to server ... \n" . mysql_error ());
mysql_select_db($db) or die ("Could not connect to database ... \n" .
mysql_error ()); ?>
```

Programski kôd 3.15. – Spoj na bazu *connect-db.php*

Za prikaz podataka iz tablice korišten je programski kôd 3.16. Pomoću *echo* naredbe moguće je generirati HTML elemente kao npr. tablicu, te polja tablice ispuniti podacima iz baze podataka. Naredba *$result=mysql_query("SELECT * FROM products")* deklarira varijablu *result* i daje joj vrijednosti podataka iz baze podataka, a potom petlja *while($row=mysql_fetch_array($result))* ispisuje sve elemente svakog reda iz tablice iz baze podataka, dok se svi redovi ne ispišu. Izgled aplikacije vidljiv je na slici 3.10.

```php
<?php include('connect-db.php');
  $result=mysql_query("SELECT * FROM products") or die(mysql_error());
  echo "<div id='view'><table>";
  echo "<tr> <th>ID proizvoda</th> <th>Ime</th> <th>Opis proizvoda</th>
<th>Cijena</th> <th>Dostava</th> <th>Količina</th> <th>Slika</th> <th></th>
<th></th></tr>";
  while($row=mysql_fetch_array( $result)) {
    echo '<tr> <td>' . $row['id'] . '</td>';
    echo '<td>' . $row['name'] . '</td>';
    echo '<td>' . $row['description'] . '</td>';
    echo '<td>' . $row['price'] . '</td>';
    echo '<td>' . $row['shipping'] . '</td>';
    echo '<td>' . $row['quantity'] . '</td>';
    echo '<td>' . $row['image'] . '</td>';
    echo '<td><a href="edit.php?id=' . $row['id'] . '">Uredi</a></td>';
    echo '<td><a href="delete.php?id=' . $row['id'] .
'">Obriši</a></td></tr>';}
  echo "</table></div>";?> <br><p><a href="new.php">Dodaj proizvod</a></p>
```

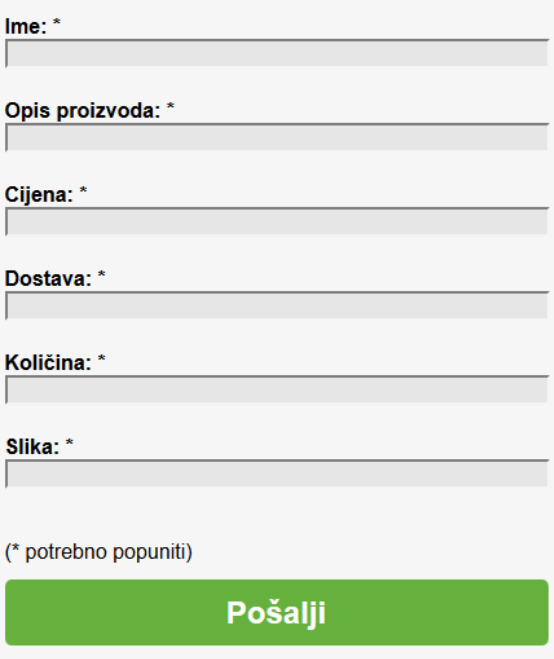
Programski kôd 3.16. – Ispis podataka iz baze podataka

ID proizvoda	Ime	Opis proizvoda	Cijena	Dostava	Količina	Slika		
1	Zec	This is some description text.	50	1	1	item_1.jpg	Uredi	Obriši
2	Borovnice	This is some description text.	1	1	10	item_2.jpg	Uredi	Obriši
3	Kupine	This is some description text.	2	1	20	item_3.jpg	Uredi	Obriši
4	Jagode	This is some description text.	1.5	1	20	item_4.jpg	Uredi	Obriši
10	Maline	This is some description text.	1.25	1	10	item_5.jpg	Uredi	Obriši

Dodaj proizvod

Slika 3.10. – Prikaz podataka iz tablice

Klikom na "Dodaj proizvod“ korisnik će biti preusmjeren na stranicu gdje ima mogućnost dodavanja novih proizvoda, kao što je vidljivo na slici 3.11. Korisnik nema mogućnost unošenje *id* varijable, jer ona se automatski postavlja sama, te se razlikuje za svaki proizvod.



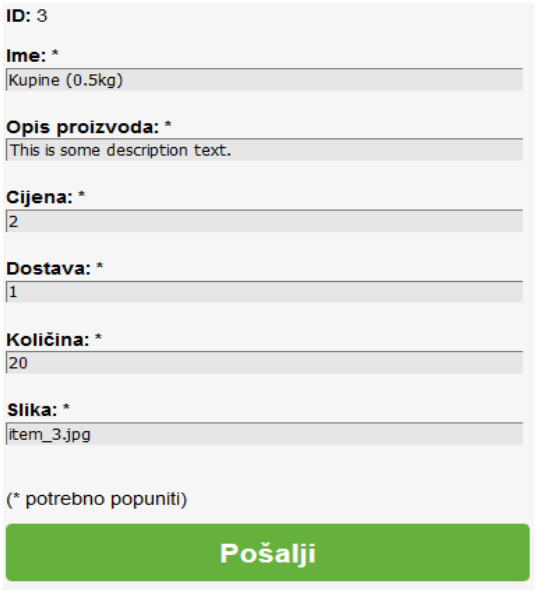
Slika 3.11. – Unošenje proizvoda

U programskom kôdu 3.17. napisano je kako je omogućeno unošenje novih proizvoda u bazu preko aplikacije. Prvo je potrebno stvoriti formu koja omogućava korisniku unos podataka o proizvodu. Zatim naredba *include('connect-db.php');* uključuje kôd za spoj na bazu podataka. Zatim se kreiraju varijable i pridodaju im se vrijednosti iz forme. Ukoliko sva potrebna polja nisu ispunjena, pozvat će se greška da nisu sva polja u formi ispunjena. Ukoliko su sva polja ispunjena, izvršit će se *INSERT* naredba koja će podatke unijeti u bazu podataka. Naredba *header("Location: view.php");* nekon izvršenja akcije nad bazom podataka vraća korisnika na stranicu za prikaz svih podataka iz tablice.

```
<?php function renderForm($form1, $form2, $form3, $form4, $form5, $form6,
$error) { ?> <html> <body> <?php
if ($error !='') { echo $error; } ?>
    <form action="" method="post">
      <strong>Ime: *</strong> <input type="text" name="name" value="<?php
echo $form1; ?>" /><br/>
      <strong>Opis proizvoda: *</strong> <input type="text"
name="description" value="<?php echo $form2; ?>" /><br/>
      <strong>Cijena: *</strong> <input type="text" name="price"
value="<?php echo $form3; ?>" /><br/>
      <strong>Dostava: *</strong> <input type="text" name="shipping"
value="<?php echo $form4; ?>" /><br/>
      <strong>Količina: *</strong> <input type="text" name="quantity"
value="<?php echo $form5; ?>" /><br/>
      <strong>Slika: *</strong> <input type="text" name="image"
value="<?php echo $form6; ?>" /><br/> <p>(* potrebno popuniti)</p>
      <input type="submit" name="submit" value="Pošalji">
    </form> </body> </html> <?php }
include('connect-db.php');
if (isset($_POST['submit'])) {
  $name=mysql_real_escape_string(htmlspecialchars($_POST['name']));
  $description=mysql_real_escape_string(htmlspecialchars($_POST
['description']));
  $price=mysql_real_escape_string(htmlspecialchars($_POST['price']));
  $shipping=mysql_real_escape_string(htmlspecialchars($_POST
['shipping']));
  $quantity=mysql_real_escape_string(htmlspecialchars($_POST
['quantity']));
  $image=mysql_real_escape_string(htmlspecialchars($_POST['image']));
  if ($name=='' || $description=='' || $price=='' || $shipping=='' ||
$quantity=='' || $image=='') {
    $error='GREŠKA: Molimo Vas da ispunite sva potrebna polja!';
    renderForm($name, $description, $price, $shipping, $quantity, $image,
$error); }
  else {
  mysql_query("INSERT products SET name='$name',
description='$description', price='$price', shipping='$shipping',
quantity='$quantity', image='$image'") or die(mysql_error());
          header("Location: view.php"); } }
else { renderForm('', '', '', '', '', '', ''); } ?>
```

Programski kôd 3.17. – Unošenje proizvoda

Nakon funkcije unošenja novih proizvoda, potrebno je omogućiti uređivanje postojećih proizvoda. Uređivanje se obavlja na temelju primarnog ključa pojedinog proizvoda iz baze. Na slici 3.12. je vidljivo da uređivanje proizvoda dohvaća primarni ključ proizvoda, te prikazuje već postojeće vrijednosti atributa i daje mogućnost izmjene vrijednosti atributa.



Slika 3.12. – Uređivanje proizvoda

Kao i kod dodavanja novih proizvoda, za uređivanje je potrebno stvoriti formu za unošenje proizvoda. Pošto je programski kôd za stvaranje forme gotovo jednak kao i kod unošenja novog proizvoda, u programskom kôdu 3.18. je izostavljen taj dio. Međutim, potrebno je formu ispuniti podacima iz baze, te klikom na pošalji, umjesto INSERT naredbe izvršit će se UPDATE naredba za ažuriranje podataka iz baze. Naredba *SELECT * FROM products WHERE id=$id"* osigurava prikaz samo onog proizvoda na koji je korisnik kliknuo, te se samo njega uređuje i mijenjaju njegove varijable.

```
    else {
      mysql_query("UPDATE products SET name='$name',
description='$description', price='$price', shipping='$shipping',
quantity='$quantity', image='$image' WHERE id='$id'") or
die(mysql_error());
      header("Location: view.php"); } }
    else { echo 'Error!'; } }
else {
  if (isset($_GET['id']) && is_numeric($_GET['id']) && $_GET['id'] > 0) {
    $id=$_GET['id'];
    $result=mysql_query("SELECT * FROM products WHERE id=$id") or
die(mysql_error());
    $row=mysql_fetch_array($result);
    if($row) {
      $name=$row['name']; $description=$row['description'];
price=$row['price']; $shipping=$row['shipping'];
$quantity=$row['quantity']; $image=$row['image'];
      renderForm($id, $name, $description, $price, $shipping, $quantity,
$image, ''); }
    else { echo "Greška!"; } }
  else { echo 'Greška!'; } } ?>
```

Programski kôd 3.18. – Uređivanje proizvoda

Posljednja potrebna funkcija je mogućnost brisanja proizvoda iz baze podataka. Kada korisnik na prikazu svih podataka iz baze klikne na "Obriši", pozvat će se kôd za brisanje podataka iz tablice. Naredba *DELETE FROM products WHERE id=$id* osigurava da se obriše samo onaj redak na koji je korisnik kliknuo. Cijeli kôd za brisanje vidljiv je u programskom kôdu 3.19.

```
<?php include('connect-db.php');
if (isset($_GET['id']) && is_numeric($_GET['id'])) {
  $id=$_GET['id'];
  $result=mysql_query("DELETE FROM products WHERE id=$id") or
die(mysql_error());
  header("Location: view.php"); }
else { header("Location: view.php"); } ?>
```

Programski kôd 3.19. – Brisanje proizvoda

4. ZAKLJUČAK

"Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom" korisniku aplikacije pruža jednostavno sučelje za promoviranje svog gospodarstva na Internetu, kao i jednostavno upravljanje proizvodima koje želi prodati. Pošto se u današnje vrijeme sve više gospodarstava odlučuje trgovanje preko Interneta, ovom aplikacijom kroz nekoliko jednostavnih koraka se to može uraditi. Korisnik aplikacije ima mogućnost dodavanja proizvoda, kao i brisanje i uređivanje istih, a posjetiteljima stranice će biti prikazani proizvodi koje je korisnik stavio u bazu putem aplikacije. Posjetiteljima je na raspolaganju biranje proizvoda na stranici koristeći košaricu za kupovinu, a košarica je povezana sa servisom za trgovanje PayPal.

Za izradu Internet stranice potrebno je poznavanje HTML i CSS opisnih jezika. Za omogućavanje funkcija klizača slika, košarice, prikaza proizvoda na stranici, te izradu aplikacije za upravljanje proizvodima, potrebno je poznavanje JavaScript i PHP programskih jezika. Za komunikaciju s bazom podataka također je potrebno poznavanje SQL programskog jezika. Za izradu i testiranje baze podataka korišten je XAMPP programski alat. Čitavu Internet stranicu i aplikaciju moguće je pisati u bilo kojem uređivači tekstualnih dokumenata, dok je za pristup bazi podataka potreban neki od Internet pretraživača.

LITERATURA

[1] HTLM Source, *What is HTML*, http://www.yourhtmlsource.com/starthere/whatishtml.html, svibanj 2016.

[2] W3Schools, *Browser Statistics*, http://www.w3schools.com/browsers/browsers_stats.asp, lipanj 2016.

[3] Wikipedia, *Cascading Style Sheets*, https://en.wikipedia.org/wiki/Cascading_Style_Sheets, svibanj 2016.

[4] TechTarget, *CSS*, http://searchsoa.techtarget.com/definition/cascading-style-sheet-CSS, svibanj 2016.

[5] W3, *HTML&CSS*, https://www.w3.org/standards/webdesign/htmlcss, svibanj 2016.

[6] AboutTech, *JavaScript*, http://javascript.about.com/od/reference/p/javascript.htm, lipanj 2016.

[7] TutorialsPoint, *JavaScript Overview*, http://www.tutorialspoint.com/javascript/javascript_overview.htm, lipanj 2016.

[8] Wikipedia, *JavaScript*, https://en.wikipedia.org/wiki/JavaScript, lipanj 2016.

[9] PHP, *What is PHP*, http://php.net/manual/en/intro-whatis.php, lipanj 2016.

[10] PHP, *What can PHP do*, http://php.net/manual/en/intro-whatcando.php, lipanj 2016.

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

[12] Apache Friends, *XAMPP*, https://www.apachefriends.org/index.html, svibanj 2016.

[13] W3Schools, *SQL*, http://www.w3schools.com/sql/sql_intro.asp, svibanj 2016.

[14] W3Schools, *CSS Syntax*, http://www.w3schools.com/css/css_syntax.asp, svibanj 2016.

[15] CSS-Tricks, *Absolute Positioning Inside Relative Positioning*, https://css-tricks.com/absolute-positioning-inside-relative-positioning/, svibanj 2016.

[16] Menucool, *Image Slider*, http://www.menucool.com/javascript-image-slider, svibanj 2016.

[17] Menucool. *JavaScript Image Slider*, http://www.menucool.com/slider/javascript-image-slider-demo1, svibanj 2016.

[18] CSSmatic, *Box Shadow*, http://www.cssmatic.com/box-shadow, svibanj 2016.

[19] PayPal, *Adding PayPal Checkout to Your 3rd-party Shopping Cart*, https://www.paypal.com/cgi-bin/webscr?cmd=p/pdn/howto_checkout-outside, lipanj 2016.

[20] W3Schools, *PHP Sessions*, http://www.w3schools.com/php/php_sessions.asp, svibanj 2016.

[21] W3Schools, *HTML Forms*, http://www.w3schools.com/html/html_forms.asp, svibanj 2016.

SAŽETAK

U završnom radu razvijena je "Internet aplikacija za upravljanje obiteljskim poljoprivrednim gospodarstvom", koja korisnicima pruža jednostavan način za promoviranje svog gospodarstva i upravljanjem proizvoda. Uz aplikaciju dolazi i Internet stranica na kojoj se posjetitelji mogu upoznati sa područjem rada gospodarstva, te obavljati Internet kupovinu i trgovanje pomoću Internet servisa PayPal. U teorijskom dijelu rada opisane su korištene tehnologije, te je opisan proces razvoja stranice i aplikacije. Praktični dio rada sastoji se od dizajniranja i izrade Internet stranice i baze podataka, a potom izrade aplikacije i povezivanje sa Internet stranicom. Korišteni su opisni jezici HTML i CSS, skriptni programski jezici JavaScript i PHP, te SQL programski jezik za relacijske modele.

Ključne riječi: obiteljsko poljoprivredno gospodarstvo, Internet aplikacija, Internet stranica, Internet trgovina

ABSTRACT

In this thesis an "Internet application for managing family agriculture economy" was developed, which provides users with an easy way to promote their business and provides simple product management. Along with the application comes a website where visitors can get acquainted with the area of operation of the economy , and perform Internet shopping and Internet trading using the PayPal service. The theoretical part describes the technologies used and describes the process of developing the website and the applications. Practical part of the work consists of designing and developing the website and the database, followed by developing the application and connect it to the website. HTML and CSS were the markup languages used, JavaScript and PHP were the scripting programming languages used, and SQL programming language was used for relational models.

Keywords: family agriculture economy, Internet application, website, Internet shopping

ŽIVOTOPIS

Marko Tarnaj rođen je 3. prosinca 1992. godine u Osijeku. Od 1999. do 2007. godine pohađa Osnovnu školu Grigora Viteza u Osijeku. Tijekom 2006. i 2008. godine provodi ljeta u Kanadi usavršavajući znanje engleskog jezika i napredujući komunikacijske vještine. Godine 2007. upisuje III. gimnaziju Osijek koju završava 2011. polaganjem državne mature. Iste godine upisuje Elektrotehnički fakultet Osijek na Sveučilištu Josipa Juraja Strossmayera u Osijeku na preddiplomski smjer računarstva. Godine 2013. prebacuje se na stručni smjer informatike na istom fakultetu. Godine 2014. pohađa edukaciju u Edunova školi za informatiku i menadžment u Osijeku, čime stječe zvanje "Voditelj izrade i provedbe projekata financiranih iz EU fondova".

Marko Tarnaj