

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Nikola Varšić

ON-LINE TRGOVINA ZA OPENERP

ZAVRŠNI RAD

Varaždin, 2014.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Nikola Varšić

Matični broj: 39234/10-R

Studij: Informacijski sustavi

ON-LINE TRGOVINA ZA OPENERP

ZAVRŠNI RAD

Mentor:

Doc.dr.sc. Markus Schatten

Varaždin, rujan 2014.

Sadržaj

1. Uvod	1
2. OpenERP	2
2.1. Instalacija	4
3. Korištene tehnologije.....	6
3.1. HTML.....	6
3.2. CSS.....	7
3.3. JavaScript	8
3.3.1. jQuery	9
3.4. PHP.....	9
3.5. PostgreSQL	9
4. Krićka obrada korištene tehnologije.....	10
5. Opis implementacije	12
5.1. Struktura on-line trgovine	12
5.2. Funkcionalnosti on-line trgovine	13
6. Primjeri korištenja	24
7. Zaključak	30
8. Literatura	31
9. Prilog	32

1. Uvod

Svakodnevnim korištenjem interneta pruža se mogućnost za sve većim unapređenjem i proširenjem poslovanja. Jedna od tih mogućnosti je i on-line prodaja. Prodaja putem interneta je najprofitabilniji oblik trgovine jer ima niske troškove poslovanja. Nije potreban poslovni prostor, već samo internet stranica, nije potrebna osoba koja radi na blagajni, nema radnog vremena, tj. kupovanje je moguće 24 sata na dan te je proizvod dostupan svim korisnicima bez obzira gdje se nalazili. Promijene cijena su trenutne kao i sve informacije koje on-line trgovina pruža. On-line trgovine postaju sve popularnije jer korisnici ne moraju odlaziti u trgovinu da bi kupili proizvode, već samo naprave nekoliko klikova i proizvod je na njihovim vratima pa time štede svoje vrijeme, a i novac jer nema troškova odlaska u trgovinu. Također, lako je usporediti cijene proizvoda u različitim on-line trgovinama pa se može kupiti tamo gdje je najjeftinije.

U ovom radu napravljena je web stranica za on-line trgovinu koja koristi OpenERP sustav za dodavanje, mijenjanje, brisanje proizvoda i njihovih kategorija te kreiranje prodajnih naloga, računa, otpremnica i srodnih dokumenata za evidentiranje prodaje. Web stranica omogućava pregledavanje proizvoda po kategorijama, dodavanje proizvoda u košaricu i pretraživanje proizvoda.

2. OpenERP

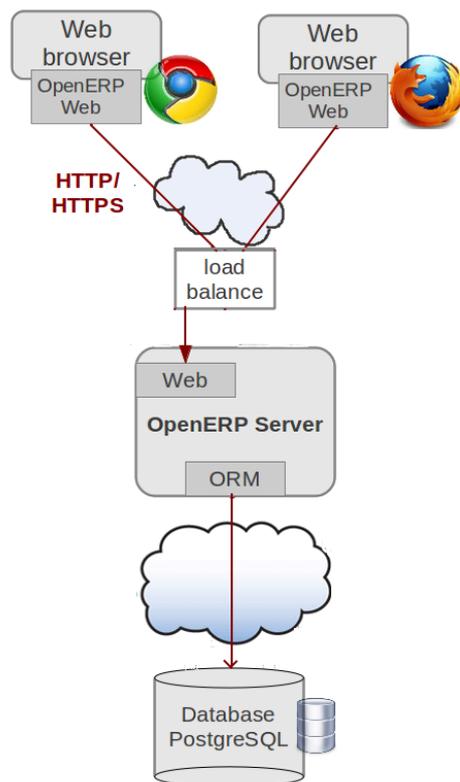
Planiranje resursa poduzeća (eng. Enterprise Resource Planning - ERP) je skup poslovnih aplikacija koje su integrirane zajedno kako bi pomogle poduzeću u prikupljanju, upravljanju i izvješćivanju informacija za osnovne poslovne procese. Te se poslovne aplikacije obično nazivaju moduli te se oni mogu samostalno instalirati i konfigurirati, ovisno o specifičnim potrebama poslovanja (Moss, 2013).

OpenERP sustav je ERP sustav koji je besplatan i slobodan je za korištenje, odnosno open source. Najbolji je sustav takve vrste i omogućava 7 načina prikaza podataka kao što navodi Infodom.hr (2014):

Odoo se generalno smatra jednim od najboljih ERP rješenja u širem smislu. Uz standardne ERP module Odoo sadrži brojne druge funkcionalnosti poput Upravljanja projektima i Upravljanja znanjem i jedino je rješenje koje standardno omogućuje 7 različitih pogleda na podatke – forma za unos, lista, kanban, kalendar, gantogram, grafički i potpuni GIS pogled.

Trenutno postoji više od 4150 dostupnih modula koji se mogu preuzeti i slobodno koristiti. Svaki mjesec izađe oko pedeset novih modula. Dnevno ima preko tisuću instalacija što ga čini brojem jedan po broju instalacija menadžment softvera. Razvoju i ispravljanju OpenERP sustava doprinosi više od dvije tisuće programera diljem svijeta. Uz tisuće manjih i srednjih kompanija, koriste ga i velike organizacije poput Danone, Francuske pošte i vlade Portugala.

OpenERP sustav napravljen je u programskom jeziku Python. Sastoji se od troslojne arhitekture dizajnirane prema MVC (eng. Model-View-Controller) dizajnu: Model - PostgreSQL baza, View - Web browser, Controller - OpenERP server (Slika 1.). Svrha MVC arhitekture je da se odvoji vizualni dio od poslovnih logike čime se u velikoj mjeri poboljšava preglednost koda i olakšava održavanje i nadogradnja aplikacije (The Architecture of OpenERP, 2014).



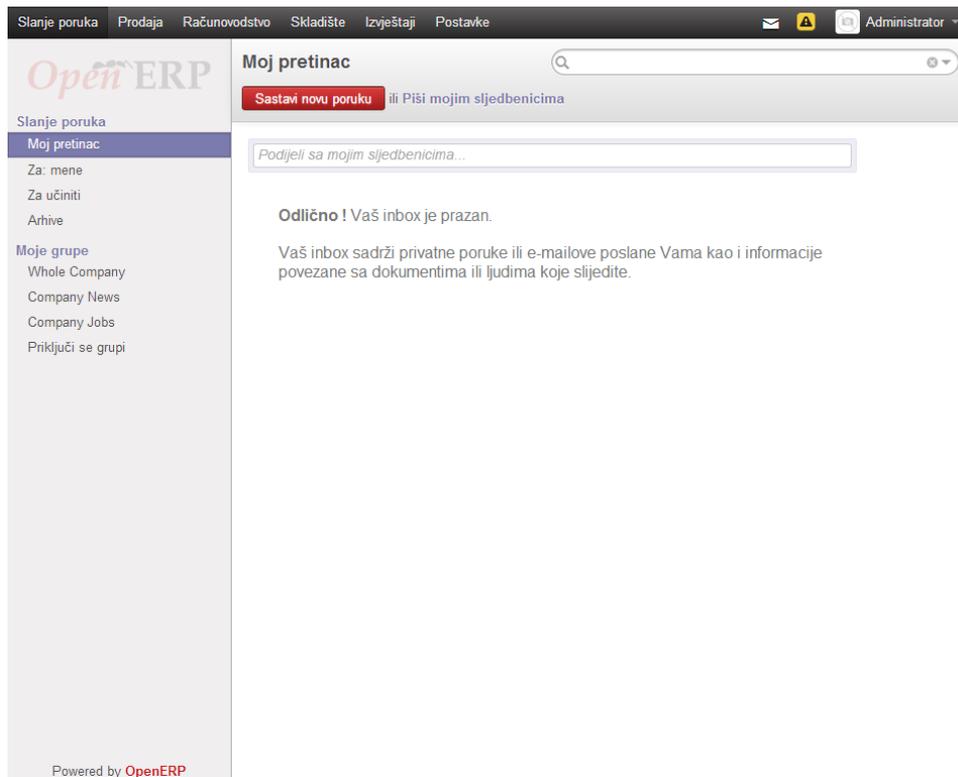
Slika 1. OpenERP arhitektura (Izvor:

https://doc.odoo.com/7.0/book/1/1_1_Inst_Config/1_1_Inst_Config_architecture/)

OpenERP sustav je neovisan o platformi jer se izvršava u web pregledniku pa nije bitno kakav operacijski sustav korisnik koristi. Podržani preglednici su: Google Chrome, Internet Explorer, Firefox i Safari.

U svibnju ove godine OpenERP je doživio debrendiranje pa je naziv promijenjen u Odoo. Razlog tome je što su se počeli baviti i izgradnjom web stranica, e-trgovinom, POS aplikacija, CMS sustavom i ostalim stvarima koje nisu usko vezane uz ERP. U nastavku ću koristiti naziv OpenERP sustav zbog toga jer je to naziv ovog rada.

Verzija koja je korištena u ovom radu je OpenERP 7.0.



Slika 2. Izgled OpenERP sustava

2.1. Instalacija

Koraci instalacije:

- Skidanje OpenERP sustava s <https://www.odoo.com/page/download>
- Instalacija OpenERP sustava zajedno sa PostgreSQL sustavom za upravljanje bazama podataka
- Kreiranje baze
- Instalacija modula:
 - U gornjem horizontalnom meniju odaberemo Postavke
 - U lijevom vertikalnom meniju odaberemo Instalacija modula
 - U popisu modula odaberemo modul koji želimo instalirati i kliknemo na instaliraj

Funkcionalnosti OpenERP sustava su podijeljene u module čime se izbjegava instaliranje nepotrebnih funkcionalnosti. U verziji OpenERP 7.0 postoje 1982 modula koja je moguće instalirati. Kompletan popis modula može se naći na sljedećem linku: <https://www.odoo.com/apps> .

OpenERP dolazi s nekoliko predinstaliranih modula koji su potrebni za osnovnu funkcionalnost sustava. U ovom radu su instalirani sljedeći moduli: upravljanje prodajom (eng. Sales Management) i upravljanje skladištem (eng. Warehouse Management).

3. Korištene tehnologije

U izradi web stranice koriste se dvije vrste tehnologija. Jedna se izvršava na poslužiteljskom računalu pa se nazivaju poslužiteljske (eng. Server-side) tehnologije. U ovom radu korišten je jezik PHP koji se izvršava na poslužitelju. Druga tehnologija se izvršava u web pregledniku korisnika pa se naziva klijentska (eng. Client-side) tehnologija. Na klijentskoj strani korištene su sljedeće tehnologije: HTML, CSS i JavaScript.

3.1. HTML

HTML (eng. Hyper-Text Markup Language) je jezik za oblikovanje sadržaja dokumenta. Za tu namjenu koriste se HTML elementi koji izgledaju u obliku oznaka (eng. tags). Sadržaj se piše unutar dviju oznaka – početna i završna (<element> sadržaj </element>). U HTML dokument mogu se ugraditi i skripte nekih drugih jezika, npr. JavaScript i PHP (Kermek, 2014).

Struktura HTML dokumenta:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Naslov </title>
  </head>
  <body>
    <p> Sadržaj </p>
    <!-- Komentar -->
  </body>
</html>
```

Kod 1. Struktura HTML dokumenta

Prva linija označava verziju HTML dokumenta, u ovom slučaju HTML5. Cijeli HTML dokument nalazi se unutar <html> oznake i dijeli se na dva dijela, zaglavlje (eng. head) i tijelo (eng. body). U zaglavlju se nalaze podaci o samom dokumentu, dok se u tijelu nalazi sadržaj koji se prikazuje korisniku.

Oznaka	Opis
<code><h1> </h1> ... <h6> </h6></code>	Definira naslove
<code><p> </p></code>	Definira odlomak
<code><table> </table></code>	Definira tablicu
<code><tr> </tr></code>	Redak tablice
<code><td> </td></code>	Ćelija tablice
<code><div> </div></code>	Definira područje
<code><a> </code>	Definira poveznicu
<code><form> </form></code>	Definira formu za unos podataka
<code></code>	Definira sliku
<code>
</code>	Prijelaz u novi redak
<code><!-- --></code>	Definira komentar
<code><pre> </pre></code>	Sadržaj je formatiran kako je napisan
<code><code> </code></code>	Tekst ugrađenog koda

Tablica 1. Popis nekih HTML oznaka

3.2. CSS

CSS (Cascading Style Sheets) je jezik za vizualno opisivanje web stranica uključujući boje, izgled i fontove. Omogućava prilagođavanje stranice različitim vrstama uređajima, kao što su monitori, tableti, mobiteli ili printeri. CSS je neovisan o HTML-u i može se koristiti s bilo kojim drugim XML-baziranim jezikom oznaka. U početku su HTML i CSS bili u jednom dokumentom, ali kako je HTML dokument postajao sve složeniji bilo je sve teže čitati i pisati kod. S ciljem boljeg razumijevanja HTML i CSS koda, jezici su se razdvojili u dva zasebna dokumenta (W3schools, 2014).

CSS je moguće integrirati u HTML na dva načina:

- direktnim pisanjem u sam HTML dokument (ne preporuča se)
 - po potrebi, u istoj liniji (eng. inline)
 - jednom u dokumentu
- odvajanjem CSS koda u zasebnu datoteku, te njegovo uključivanje u HTML dokument.

```
<link rel="stylesheet" href="css/jquery-ui.min.css" type="text/css">
```

Kod 2. Uključivanje CSS-a u HTML kod

Postoji više načina korištenja CSS uputa (Kermek, 2014):

- implicitno (prema oznaci)
- eksplicitno (prema class atributu)
- jednoznačno (prema id atributu)
- pseudo-klasa (za dodavanje specijalnih efekata)

```
body
{
  margin: 0;
  background-color: #E8EAEFF;
}
.kosaricaMiniProizvodi
{
  height: 50px;
  width: 90%;
  display: block;
  margin: 5px auto;
}
#meni
{
  background-color: red;
  color: white;
}
#meni:hover
{
  background-color: white;
  color: red;
}
```

Kod 3. Primjer korištenja CSS uputa

3.3. JavaScript

JavaScript je skriptni jezik koji se izvršava u web pregledniku korisnika. Koristi se za bolju interakciju korisnika sa web aplikacijom. Neovisan je o platformi pa se može izvršavati u bilo kojem pregledniku.

3.3.1. jQuery

jQuery je JavaScript biblioteka koja olakšava izradu web stranica koje koriste JavaScript. Pisanje koda pomoću jQuery-a je puno brže i jednostavnije jer za istu funkcionalnost napisanu pomoću jQuery-a u jednoj liniji koda potrebno je od 10 do 20 linija koda JavaScript-a.

Sa stranice <http://jquery.com/download/> potrebno je skinuti jednu od dviju verzija jQuery-a: verzija 1.x i 2.x. Razlika je u tome što verzija 1.x podržava preglednik Internet Explorer verzije 6, 7 i 8, dok verzija 2.x podržava preglednik Internet Explorer verzije 9 i više.

Uključivanje jQuery-a u HTML kod:

```
<script type="text/javascript" src="js/jquery-1.11.1.min.js"></script>
```

Kod 4. Uključivanje jQuery-a u HTML kod

Korišteni jQuery dodaci:

- jQueryUI
- Flex slider
- Fancy box

Korištena verzija u ovom radu je jQuery 1.11.1.

3.4. PHP

PHP (eng. Hypertext Preprocessor) je programski jezik koji se izvršava na strani poslužitelja. Namijenjen je izradi aplikacija koje se izvršavaju na Windows ili UNIX poslužiteljima. Podržava proceduralno i objektno programiranje. Najčešće se koristi za izradu dinamičkih web aplikacija (Herrington J D, 2007).

Korištena verzija u ovom radu je PHP 5.4.12.

3.5. PostgreSQL

PostgreSQL je objektno-relacijski sustav za upravljanje bazama podataka koji zadovoljava ANSI-SQL:2008 standard. Radi gotovo na svim platformama. PostgreSQL je besplatan sustav i otvorenog je koda (eng.open source). Ima punu podršku za vanjske ključeve, spajanja, poglede, trigere i pohranjene procedure (Postgresql.org, 2014).

Korištena verzija u ovom radu je PostgreSQL 9.2.

4. Kritička obrada korištene tehnologije

Izradi on-line trgovine za OpenERP sustav moglo se pristupiti na dva načina. Prvi način je da se napravi stranica koja će s OpenERP sustavom komunicirati preko XML-RPC servisa. Drugi način je da se direktno spajamo na bazu OpenERP sustava. U ovom radu korišten je drugi način zbog toga jer je korištena tehnologija PHP koja podržava XML-RPC servis, ali nema dobrog konektora, odnosno ORM modela za komunikaciju sa OpenERP sustavom. Počeo sam raditi na prvi način, ali imao sam problem prilikom dohvaćanja količine proizvoda na zalih. Nekoliko dana sam pokušavao riješiti taj problem, ali sam odustao i prešao na drugi način. Kod XML-RPC servisa za „select“ naredbu potrebno je raditi dva upita. Prvi upit je da pronade skup vrijednosti prema nekom uvjetu i vrati samo identifikator tih vrijednosti, a drugi upit je da prema tome identifikatoru vrati ostale vrijednosti tog zapisa.

Drugi način je malo kompliciraniji i dugotrajniji jer zahtjeva poznavanje strukture baze podataka i poslovnu logiku OpenERP sustava. Prilikom kupnje proizvoda u OpenERP sustavu postoji nekoliko koraka koje je potrebno napraviti:

1. Izrada prodajnog naloga
2. Potvrda prodaje
3. Kreiranje fakture i nacrt račun
4. Potvrda nacrt račun
5. Registriranje plaćanja i kreiranje otpremnice
6. Otpremanje

Za prvih pet koraka potrebno je napraviti upite koji će simulirati navedene korake. Ima gotovo 30 upita za te korake što baš i nije malo. Prilikom prolaženja kroz ovih 5 koraka u OpenERP sustavu, na PostgreSQL server monitoru zabilježio sam preko 300 upita koji imaju istu namjenu kao onih 30 koje sam napisao. Zadnji korak nije potrebno implementirati jer taj korak radi osoba koja otprema proizvode.

Za dodavanje proizvoda u košaricu korišten je AJAX poziv. To je dobro jer nema osvježavanja stranice i stvara se prividnost trenutnog odgovora. Prilikom dodavanja automatski se promijeni količina proizvoda i ukupna vrijednost košarice. Mini košarica se također mijenja prilikom svake promjene i promijeni se boja pozadine kako bi se dobio uvid promjene. Promjenom stanja u mini košarici također se mijenja i stanje košarice.

Prije unosa slike u sustav bilo bi dobro sve slike pretvoriti u istu rezoluciju kako ne bi bilo problema prilikom prikazivanja istih jer se u CSS-u još ne mogu elementi poravnati po vertikali pa ako je neka slika manja od elementa onda je ona pozicionira na vrhu elementa, a donji dio ostaje prazan.

U OpenERP sustavu postoje 3 veličine slike, normalna, srednja i mala. OpenERP automatski pretvara sliku u tri navedene veličine. Nedostatak je što se može unijeti samo jedna slika po proizvodu. To se može riješiti dodavanjem jedne tablice (1:M) koja će se referencirati na tablicu proizvoda. Međutim potreban je mehanizam koji će pretvoriti sliku u bytea format jer se slike u bazi nalaze u tom formatu. Kod kategorija uopće nema mogućnosti dodavanja slika. Ovaj problem sam riješio na način da se svaki put uzima slučajna slika proizvoda koji se nalazi u toj kategoriji. Ovaj problem se također može riješiti kao i za slike, dodavanjem nove tablice.

Za straničenje proizvoda također je korišten AJAX poziv. Nije baš jednostavno to napraviti jer se elementi dinamički dodaju pa treba raditi izračune u PHP-u i u javascript-u.

Prilikom kreiranja ili uređivanja nekog dokumenta u OpenERP sustavu kreira se obavijest koju dobivaju sve osobe koje koriste OpenERP sustav. U ovom radu nakon obavljene kupnje upisuju se i izmjenjuju podaci pet dokumenata nekoliko puta pa sam odlučio da to nije potrebno napraviti.

Prije same kupnje potrebno je unijeti podatke o kupcu kako bi znali kome prodajemo i kako bi mogli provjeriti identitet kupca. U ovom radu moguće je unijeti samo jednu adresu kupca jer OpenERP nema mogućnost spremanja više adresa za istu osobu. Ovaj problem je moguće riješiti na način da se doda još jedna tablica u kojoj će se nalaziti adresa za dostavu i identifikator osobe.

U Open ERP sustavu ne postoji mogućnost za spremanje podataka klijenta za prijavu, odnosno registraciju. Ovaj problem je riješen na način da su pronađena polja u tablici „res_partner“ koja se ne koriste za korisnika koji obavlja kupnju preko on-line trgovine. U navedena polja su uneseni korisničko ime, lozinka i kod za aktivaciju korisnika.

Nedostatak korištenog način je što postoji samo jedno korisničko ime za spajanje na bazu podataka. Navedeno korisničko ime sve ovlasti nad bazom podataka pa je moguće „hakirati“ bazu, odnosno mijenjati i brisati podatke. Ovaj problem se može riješiti tako da se kreira još jedno korisničko ime koje će moći samo upisivati i dohvaćati podatke iz potrebnih tablica.

5. Opis implementacije

Web aplikacija je izrađena na način da postoji jedna glavna datoteka s osnovnom strukturom HTML-a, ostale datoteke se uključuju u tu datoteku po potrebi.

5.1. Struktura on-line trgovine

Glavna datoteka ove web aplikacije je index.php (Kod 5.). Unutar tijela HTML dokumenta navedene datoteke nalaze se tri glavna područja: zaglavlje (eng. header), glavni dio (eng. main) i podnožje (eng. footer). U zaglavlju se nalazi naslov stranice, brzi pregled košarice, horizontalni meni, te tražilica. Glavni dio sadrži dvije sekcije (eng. section) od kojih je jedna za vertikalni popis kategorija, dok se unutar druge sekcije nalazi glavni sadržaj. Podnožje se sastoji od popisa linkova iz horizontalnog menija, kontakta i uvjeta korištenja.

```
<!DOCTYPE html>
<html>
  <head>
    <title>On-line trgovina</title>
  </head>
  <body>
    <header>
      <div id="zaglavlje">...</div>
      <nav>...</nav>
    </header>
    <main>
      <section id="kategorije">...</section>
      <section id="sadrzaj">...</section>
    </main>
    <footer>...</footer>
  </body>
</html>
```

Kod 5. HTML struktura index.php dokumenta

U tu datoteku se pomoću kontrolne strukture „require_once“ uključuju sve ostale datoteke koje su potrebne za rad sustava (Kod 6).

```
<?php
  require_once './zaglavlje.php';
?>
```

Kod 6. Uključivanje datoteke zaglavlje.php

5.2. Funkcionalnosti on-line trgovine

Kao što je rečeno u prethodnom potpoglavlju, glavna datoteka ima osnovnu strukturu i u nju se uključuju ostale datoteke prema potrebi. Prilikom zahtjeva za uključivanje datoteke šalje se parametar „p“ pomoću kojeg se određuje koja datoteka se uključuje (Kod 7.).

```
if(isset($_GET['p']))
{
    if($_GET['p'] == "kos")
        require_once './kosarica.php';
    else if($_GET['p'] == "nap")
        require_once './narudzba.php';
    else if($_GET['p'] == "oNa")
        require_once './oNama.php';
    else if($_GET['p'] == "kon")
        require_once './kontakt.php';
    else if($_GET['p'] == "pom")
        require_once './pomoc.php';
    else if($_GET['p'] == "pre")
        require_once './pretrazivanje.php';
}
```

Kod 7. Komunikacija datoteka preko „get“ metode

Spajanje na bazu

Klasa baza.class.php sadrži funkcije za spajanje na bazu i prekidanje veze s bazom (Kod 8.). Pošto se podaci za spajanje nalaze u jednoj datoteci, prilikom promjene podataka za spajanje potrebno je samo u toj datoteci promijeniti podatke.

```
function spajanjeNaBazu ()
{
    $baza = pg_connect ("host=localhost port=5432 dbname=vj5 user=openpg
password=openpgpwd");
    if (!$baza)
    {
        exit();
    }
    return $baza;
}

function prekidVeze ($veza)
{
    $veza->close();
}
```

Kod 8. Funkcija „spajanjeNaBazu“

Navedena klasa sadrži također i dvije funkcije za upite. Potrebne su dvije funkcije jer SELECT vraća skup vrijednosti, dok INSERT i UPDATE vraćaju samo jednu vrijednost

(koliko je redova izmijenjeno). Funkcija `upitSelect($upit)` dobiva tekst upita i, ako je broj redova veći od 0, vraća rezultat upita. Funkcija `upitOstali($upit)` također dobiva tekst upita i, ako je upit ispravan, vraća broj redova koji su promijenjeni, u protivnome vraća null.

```
function upitSelect($upit)
{
    $veza = self::spajanjeNaBazu();
    $rezultat = pg_query($veza, $upit);
    if(pg_num_rows($rezultat) == 0)
        return 0;
    else
        return $rezultat;
    self::prekidVeze($veza);
}

function upitOstali($upit)
{
    $veza = self::spajanjeNaBazu();
    $rezultat = pg_query($veza, $upit);
    if($rezultat)
        return $rezultat;
    else
        return null;
    self::prekidVeze($veza);
}
```

Kod 9. „Select“ upit i ostali upiti

Upit za kategorije

Kategorije se nalaze u tablici `product_category` koja je povezana sama sa sobom što omogućava da se u jednoj tablici nalaze sve kategorije i njihove potkategorije. Prvi upit dohvaća sve kategorije koje imaju potkategorije dok drugi upit dohvaća sve potkategorije prema svojoj kategoriji.

```

$upit = "select id, name from product_category where id in (select
parent_id from product_category) AND type = 'normal' order by id";
$resultat = $baza->upitSelect($upit);
echo "<ul>".
    "<li class='kategorijeNaslov'>Kategorije</li>";
while($red = pg_fetch_assoc($resultat))
{
    echo "<li class='kategorija' id='{ $red['id'] }'>{ $red['name'] }</li>";
    $upit2 = "select id, name from product_category where parent_id =
{ $red['id'] } AND type = 'normal'";
    $resultat2 = $baza->upitSelect($upit2);
    echo "<ul class='podkategorija'>";
    while($red2 = pg_fetch_assoc($resultat2))
    {
        echo "<li id='{ $red2['id'] }'><a
href='index.php?idKat={ $red2['id'] }.'>{ $red2['name'] }</a></li>";
    }
    echo "</ul>";
}
echo "</ul>";

```

Kod 10. Ispis kategorija i potkategorija

Upiti za količinu proizvoda na zalihi

OpenERP nema nigdje zapisano koliko nekog proizvoda ima na zalihi. U tablici „stock_move“ zapisana su sva kretanja proizvoda u poduzeću. Najčešće su to ulazak i izlazak proizvoda. Zbog toga su potrebna dva upita jer jedan služi za ulazne proizvode, a drugi za izlazne proizvode. (Kod 11.) Stanje zalihe se dobije ako se izlazni proizvodi oduzmu od ulaznih.

```

$zalihaUlaz = 0;
$upit = "select sum(product_qty) as kolicina, product_id from
stock_move where location_id NOT IN (12) and location_dest_id IN (12)
and state IN ('done') AND product_id = $id group by product_id";
$resultat = $baza->upitSelect($upit);
if($resultat)
{
    $red = pg_fetch_array($resultat);
    $zalihaUlaz = $red['kolicina'];
}
$zalihaIzlaz = 0;
$upit = "select sum(product_qty), product_id from stock_move where
location_id IN (12) and location_dest_id NOT IN (12) and state in
('done') and product_id = $id group by product_id";
$resultat = $baza->upitSelect($upit);
if($resultat)
{
    $red = pg_fetch_array($resultat);
    $zalihaIzlaz = $red['kolicina'];
}
$zaliha = $zalihaUlaz - $zalihaIzlaz;

```

Kod 11. Izračun zalihe

Upit detalji proizvoda

Svi važniji detalji o proizvodu nalaze se u dvije tablice, `product_template` i `product_product`. Detalji se filtriraju prema dobivenom ID-u proizvoda (Kod 12.).

Važno je spomenuti da je slika spremljena u bazu u bytea formatu. Prilikom odabira stupca potrebno je napisati da se bytea format pretvori u base64 format. Kod dohvaćanja stupca u PHP-u potrebno je dešifrirati base64 format. Kod HTML elementa `` potrebno je napisati „`data:image/png;base64;dešifrirani base64 format`“ za atribut `src`.

```
$idPr = $_GET['idPr'];
$upit = "select pt.id, pt.name, pt.list_price, encode(pp. "age,
'base64') AS slika, encode(pp.image, 'base64') AS slika_srednja,
pt.warranty, description_sale from product_template pt join
product_product pp on(product_tmpl_id = pt.id) where pt.id = $idPr"
$resultat = $baza->upitSelect($upit);
if($resultat)
{
    while($red = pg_fetch_assoc($resultat))
    {
        $slika = base64_decode($red['slika']);
        echo "<img src='data:image/png;base64,$slika'>";
    }
}
```

Kod 12. Upit za detalje proizvoda

Upit tražilica

Nakon unesene riječi u input polje potrebno je pritisnuti tipku „enter“ ili kliknuti na sliku povećala kako bi pretraživanje počelo. Pretraživanje se obavlja samo nad poljima koja su znakovnog tipa.

```
$rijec = $_GET['rijec'];
$upit = "select pt.id, pt.name, pt.list_price, encode(pp.image_medium,
'base64') AS slika from product_template pt join product_product pp
on(product_tmpl_id = pt.id) where pt.name ~* '$rijec' OR pt.description
~* '$rijec'";
```

Kod 13. Upit za traženje riječi

Dodavanje proizvoda u košaricu

Kod 14. prikazuje neimenovanu funkciju koja će se izvršiti svaki put kad se klikne na HTML element čija je vrijednost atributa class jednaka „dodaj“.

```
$(".dodaj").click(function()
{
    id = $(this).attr("id");
    kolicina = "plus";
    kosarica(id, kolicina);
    kosaricaMini(id, kolicina);
});
```

Kod 14. jQuery Dodavanje proizvoda

Izvršavanjem gore navede funkcije također će se izvršiti i funkcija „kosarica“ (Kod 15.) koja prima dva parametra: prvi je identifikator proizvoda, dok je drugi količina za koju se stanje košarice mora promijeniti. Kod pregleda košarice moguće je smanjiti, povećati ili unijeti proizvoljnu količinu proizvoda u košarici. Ako je količina = „plus“ onda to znači da dodajemo 1 jedinicu proizvoda u košaricu, ako je količina = „minus“ onda smanjujemo količinu proizvoda za jedan, a ako je količina neki broj x veći od 1 onda dodajemo x jedinica proizvoda. Unutar funkcije „kosarica“ izvršiti će se AJAX (eng. Asynchronous JavaScript and XML) poziv koji prima parametre: url – adresa datoteke na koju se šalje zahtjev, type – vrsta zahtjeva (post/get), dataType – tip podataka i data – vrijednosti podataka. U ovom slučaju poziva se datoteka „proizvodiUKosarici.php“.

```

function kosarica(id, kolicina)
{
    $.ajax(
    {
        url: "proizvodiUKosarici.php",
        type: "post",
        dataType: "json",
        data: {id: id, kolicina: kolicina},
        success: function(data)
        {
            if(data == -2)
                alert("Greška - AJAX: -2 ");
            else if(data == -1)
                kosaricaMini();
            else if(data >= 0)
            {
                smanjiKolicinuKosarice(id, data);
                kosaricaMini();
            }
        },
        error: function()
        {
            alert("Greska - AJAX: 1");
        }
    });
}

```

Kod 15. JQuery košarica

Prilikom svakog poziva datoteke „proizvodiUKosarici.php“ šalju se dva parametra, ID proizvoda i količina, ali uvijek samo za jedan proizvod. U varijabli „\$index“ (Kod. 16) nalazi se pozicija proizvoda u sesiji. Iz sesije je dohvaćena vrijednost količine trenutnog proizvoda i dodijeljena je varijabli „\$kolicinaKosarica“. Ranije je spomenuto da se šalju vrijednosti količine „plus“, „minus“ ili neki pozitivni broj. Plus i minus je potrebno pretvoriti u 1, odnosno -1. Navedeni brojevi se nalaze u varijabli „\$kolicinaUInt“. Zbrajanjem varijabli „\$kolicinaKosarica“ i „\$kolicinaUInt“ dobije se buduća vrijednost košarice. Ako je taj zbroj manji ili jednak stanju zalihe onda se količina u košarici smanjuje ili povećava za 1, odnosno postavlja u određenu vrijednost, ovisno o poslanom parametru. U slučaju da je taj zbroj veći onda se u košaricu postavlja najveća količina koja je dostupna na zalihi. Ako je poslana količina = 1 ili ako se u košarici nalazi jedan proizvod, a potrebno je smanjiti količinu, onda se pomoću funkcije „array_splice()“ pomiču sve vrijednosti iz sesije za jedno mjesto u lijevo (na manji index) nakon indexa na kojem se nalazi trenutni proizvod. Na kraju se funkcijom „json_encode()“ vraća stanje zalihe ili -1, ovisno o slučaju.

```

if(($kolicinaKosarica+$kolicinaUInt) <= $zaliha)
{
    if($kolicinaKosarica > 0)
    {
        if($kolicina == "plus")
            $_SESSION['kolicina'][$index]++;
        else if($kolicina == "minus")
        {
            if($kolicinaKosarica == 1)
            {
                array_splice($_SESSION['id'], $index, 1);
                array_splice($_SESSION['kolicina'], $index, 1);
            }
            else
                $_SESSION['kolicina'][$index]--;
        }
        else if($kolicina > 1)
            $_SESSION['kolicina'][$index] = $kolicina;
        else if($kolicina == 0)
        {
            array_splice($_SESSION['id'], $index, 1);
            array_splice($_SESSION['kolicina'], $index, 1);
        }
    }
    else
    {
        if($kolicina == "plus")
        {
            $_SESSION['id'][] = $id;
            $_SESSION['kolicina'][] = 1;
        }
        else if($kolicina > 1)
        {
            $_SESSION['id'][] = $id;
            $_SESSION['kolicina'][] = $kolicina;
        }
    }
    echo json_encode(-1);
}
else
{
    $_SESSION['kolicina'][$index] = intval($zaliha);
}
}

```

Kod 16. Proizvodi u košarici

Nakon uspješnog poziva i odgovora provjerava se vraćena vrijednost. Za vraćenu pozitivnu vrijednost potrebno je promijeniti stanje košarice što radi funkcija „smanjiKolicinuKosarice“ (Kod 17.).

```

function smanjiKolicinuKosarice(id, kolicina)
{
    $(".kosaricaProizvodi").each(function()
    {
        element = $(this).find(".kosaricaProizvodiKolicina").children("a");
        if(element.attr("id") == id)
        {
            element.next("input").val(kolicina);
        }
    });
    kosaricaUkupno();
};

```

Kod 17. Smanjivanje količine u košarici

Također je i potrebno pozvati funkciju „kosaricaMini()“ (Kod 18.) koja služi za tekući prikaz proizvoda u košarici. Ona sadrži AJAX poziv prema datoteci „proizvodiDetaljiJQ.php“. U navedenoj datoteci se dohvaćaju detalji za sve proizvode koji se nalaze u košarici. Nakon uspješnog dohvaćanja prvo se brišu postojeći proizvodi u mini košarici pomoću metode „remove()“. Zatim se u varijablu proizvod dodaju HTML elementi sa dohvaćenim vrijednostima za izgled mini košarice. Na kraju se pomoću „append()“ metode dodaje varijabla proizvod ostavljenom elementu mini košarice.

```

function kosaricaMini ()
{
$.ajax(
{
url: "proizvodiDetaljiJQ.php",
type: "post",
dataType: "json",
data: {detalji: "da"},
success: function(data)
{
$('.kosaricaMiniProizvodi').remove();
$('#kosaricaMiniPregled').remove();
proizvod = "";
i = 0;
$(data).each(function(z, v)
{
i++;
slika = "data:image/png;base64,";
slika+= v.slika;
proizvod+= "<div class='kosaricaMiniProizvodi'>"+
"<div class='kosaricaMiniProizvodiSlika'>"+
"<img src='"+slika+"' alt='slika'>"+
"</div>"+
"<div class='kosaricaMiniProizvodiDetalji'>"+
"<div id='"+v.id+"'
class='kosaricaMiniProizvodiDetaljiNaziv'>"+
"<a
href='index.php?idPr="+v.id+"'>"+v.naziv+"</a>"+
"</div>"+
"<div
class='kosaricaMiniProizvodiDetaljiCijena'>"+
v.cijena+" kn x "+v.kolicina+
"</div>"+
"<div
class='kosaricaMiniProizvodiDetaljiIzbrisi'>"+
"<img src='img/x1.png' alt='X'> "+
"</div>"+
"</div>"+
"</div>";
});
$('#kosaricaMiniKolicina').text(i);
$('.kosaricaMiniPozadina').append(proizvod);
});
}
}

```

Kod 18. Osvježavanje proizvoda u mini košarici

Podaci za prodajni nalog i ostale tablice

Kao što je u prethodnom poglavlju rečeno potrebno je više od 30 upita kako bi se unijeli svi potrebni podaci za evidentiranje prodaje. Trebalo bi dvadesetak stranica da se to sve objasni pa ću objasniti samo za unos prodajnog naloga i unos stavki prodajnog naloga.

Prodajni nalog ima svoje ime koje je u obliku „SOxxx“, gdje x-evi predstavljaju broječanu vrijednost od 001 do 999. Navedena broječana vrijednost nije identična identifikatoru prodajnog naloga pa je potrebno iz sekvence uzeti idući broj (popis sekvenci od strane OpenERP sustava nalazi se u tablici „ir_sequence“). Kad smo saznali idući broj onda je potrebno formatirati broj da poprimi oblik „SOxxx“. To je napravljeno pomoću uvjetnog grananja. Za unos prodajnog naloga važno je još saznati identifikator kupca (dobiven je nakon unosa podataka o kupcu), identifikator prodavača (pošto se radi o on-line trgovini, odabran je identifikator administratora, 1) te datum i vrijeme. Ovo su vrijednosti koje se mijenjaju, sve ostalo je uvijek isto. Ako je upit ispravan vraćen je identifikator prodajnog naloga.

```

$upit = "SELECT nextval('ir_sequence_021') AS id";
$resultat = $baza->upitSelect($upit);
if($resultat)
{
    $red = pg_fetch_array($resultat);
    $idProdajnogNaloga = $red['id'];
}
$duzina = strlen($idProdajnogNaloga);
if($duzina == 1)
    $imeProdajnogNaloga = "SO00".$idProdajnogNaloga;
else if($duzina == 2)
    $imeProdajnogNaloga = "SO0".$idProdajnogNaloga;
else if($duzina > 2)
    $imeProdajnogNaloga = "SO".$idProdajnogNaloga;
$upit1 = "INSERT INTO sale_order (picking_policy, order_policy,
partner_invoice_id, invoice_quantity, date_order, partner_id, user_id,
name, partner_shipping_id, shipped, state, shop_id, pricelist_id,
create_uid, create_date, write_uid, write_date, date_confirm, company_id)
VALUES ('direct', 'manual', {$idPartnera}, 'order', '{$datum}',
{$idPartnera}, {$idKorisnika}, '{$imeProdajnogNaloga}', {$idPartnera},
'f', 'progress', 1, 1, {$idKorisnika}, '{$datumVrijeme}', {$idKorisnika},
 '{$datumVrijeme}', '{$datum}', 1) RETURNING id";
$resultat1 = $baza->upitOstali($upit1);
if($resultat1)
{
    $red1 = pg_fetch_array($resultat1);
    $idProdajnogNaloga = $red1['id'];
}

```

Kod 19. Unos prodajnog naloga

Vraćeni identifikator služi kako bi se stavke prodajnog naloga povezale sa prodajnim nalogom. Za stavke prodajnog naloga potrebno je saznati ime i cijenu svakog proizvoda u košarici. To ćemo napraviti pomoću upita na tablicu „product_template“ koja sadrži te podatke. Rezultati upita će se upisati u polja „\$imeProizvoda“ i „\$cijena“. Upit će se izvršiti onoliko puta koliko proizvoda ima u košarici (u varijabli „\$brojProizvoda“ nalazi se broj proizvoda u

sesiji). Nakon toga unijet će se podaci o stavkama u tablicu „sale_order_line“ toliko puta koliko ima proizvoda u košarici.

```
$brojProizvoda = count($_SESSION['id']);
$timeProizvoda = array();
$cijena = array();
for($i = 0; $i < $brojProizvoda; $i++)
{
    $upit = "SELECT name, list_price FROM product_template WHERE id =
{$idProizvoda[$i]}";
    $rezultat = $baza->upitSelect($upit);
    if($rezultat)
    {
        $red = pg_fetch_array($rezultat);
        $timeProizvoda[] = $red['name'];
        $cijena[] = $red['list_price'];
    }
}
$idStavkeProdajnogNaloga = array();
for($i = 0; $i < $brojProizvoda; $i++)
{
    $upit2 = "INSERT INTO sale_order_line (product_uos_qty, product_uom,
sequence, order_id, price_unit, product_uom_qty, discount, product_id,
name, delay, state, th_weight, invoiced, type, create_uid,
create_date, write_uid, write_date, company_id, order_partner_id,
salesman_id) VALUES ({$kolicina[$i]}, 1, 10, {$idProdajnogNaloga},
{$cijena[$i]}, {$kolicina[$i]}, 0, {$idProizvoda[$i]},
'{$timeProizvoda[$i]}', 0, 'confirmed', 0, 'f', 'make_to_stock',
{$idKorisnika}, '{$datumVrijeme}', {$idKorisnika}, '{$datumVrijeme}',
1, {$idPartnera}, {$idKorisnika}) RETURNING id";
    $rezultat2 = $baza->upitOstali($upit2);
    if($rezultat2)
    {
        $red2 = pg_fetch_array($rezultat2);
        $idStavkeProdajnogNaloga[$i] = $red2['id'];
    }
}
```

Kod 20. Unos prodajnog naloga

6. Primjeri korištenja

U ovom poglavlju će biti prikazana i opisana stranica zajedno s procesom kupnje koji je implementiran u ovom radu.

U gornjem desnom kutu se nalazi mini košarica u kojoj se na brzi način može vidjeti koliko i kojeg proizvoda ima u košarici. Pored mini košarice s lijeve strane nalaze se linkovi za prijavu i registraciju. Prilikom registracije potrebno je unijeti sve tražene podatke. Nakon registracije na mail stiže aktivacijski link preko kojeg se aktivira korisnički račun. Kad se obavi prijava moguće je vidjeti i promijeniti podatke korisnika. Također je moguće vidjeti i spremljene proizvode u listu želja. Odjaviti se može klikom na tekst „Odjava“ koji se nalazi pored mini košarice s lijeve strane. Ispod toga nalazi se horizontalni meni u kojem se nalaze linkovi na detalje o nama, na upit preko e-maila i na pomoć. S lijeve strane ekrana nalazi se popis kategorija koje na klik prikažu svoje potkategorije. Klikom na potkategoriju ispisuju se proizvodi u toj potkategoriji (Slika 3.). Naslovna stranica također sadrži nekoliko proizvoda koji stalno rotiraju iz lijeve strane u desnu. Klikom na sliku proizvoda otvara se stranica sa detaljima proizvoda. Klikom na gumb dodaj, proizvod se dodaje u košaricu.



Slika 2. Naslovna stranica

Iznad popisa proizvoda nalazi se mogućnost odabira koliko proizvoda želite prikazati. Također je moguće sortirati proizvode po cijena, od najmanje prema najvećoj ili od najveće prema najmanjoj, i po abecedi, rastuće ili padajuće. Klikom na gumb dodaj, proizvod se dodaje u košaricu. Ispod popisa proizvoda nalazi se odabir stranica ako ima više proizvoda za prikazati. Klikom na naziv proizvoda ili na opširnije otvaraju se detalji proizvoda (Slika 4.).



Slika 3. Popis proizvoda

S lijeve strane se nalazi slika koja se otvara na klik miša. Pored slike nalazi se naziv proizvoda, cijena, dostupnost, jamstvo te gumb za dodavanje proizvoda u košaricu. Klikom na taj gumb dodaje se proizvod u košaricu. Ispod gumba „Dodaj u košaricu“ nalazi se gumb „Dodaj u listu želja“ koji na klik miša dodaje proizvod u listu želja. Ispod slike nalazi se opis proizvoda, odnosno njegove karakteristike. Ispod opisa nalazi se popis sličnih proizvoda koji se nalaze u istoj kategoriji kao i glavni proizvod.

ON-LINE TRGOVINA ZA OPENERP

Početna O nama Kontakt Pomoć Moji podaci Lista želja Traži ...

Kategorije

- Računala
- Periferija
- Komponente
- Multimedija
- Programi
- Mobiteli
- Mrežna oprema

Besplatna dostava

Dostava u roku 48 sati

Lenovo C260 AiO

Dostupno: 5 kom

Jamstvo: 12 mjeseci

Cijena: 3399.00 kn

[Dodaj u košaricu](#)

[Dodaj u listu želja](#)

Opis proizvoda

- Zaslón: 19.5" 1600 x 900 HD+
- Format ekrana: 16:9
- Procesor: Intel Celeron J1900 Quad Core 2.0GHz / 2.41GHz (2MB cache)
- Radna memorija: 1x 4GB 1600MHz DDR3 (proširivo do 16GB)
- Grafički sustav: nVidia GeForce 800A 1GB
- Tvrdi disk: 500GB SATA 7200rpm
- Optički uređaj: DVDRW DL
- Mrežna kartica: da (Gb LAN)

Slika 4. Detalji proizvoda

Klikom na mini košaricu otvara nam se košarica u kojoj se nalaze svi proizvodi koje smo dodali (Slika 5.). U stupcu količina nalazi se broj proizvoda u košarici. Moguće je upisati željenu količinu proizvoda unosom broja u bijelo polje. Također je moguće povećati ili smanjiti količinu za jedan klikom na plus, odnosno minus. Ako se upiše količina veća od one na zalihi, postavlja se maksimalna količina koja postoji na zalihi. U stupcu ukloni nalazi se „x“ koji na klik briše proizvod iz košarice. Klikom na gumb naplata otvara se obrazac za ispunjavanje podataka o narudžbi (Slika 6.).

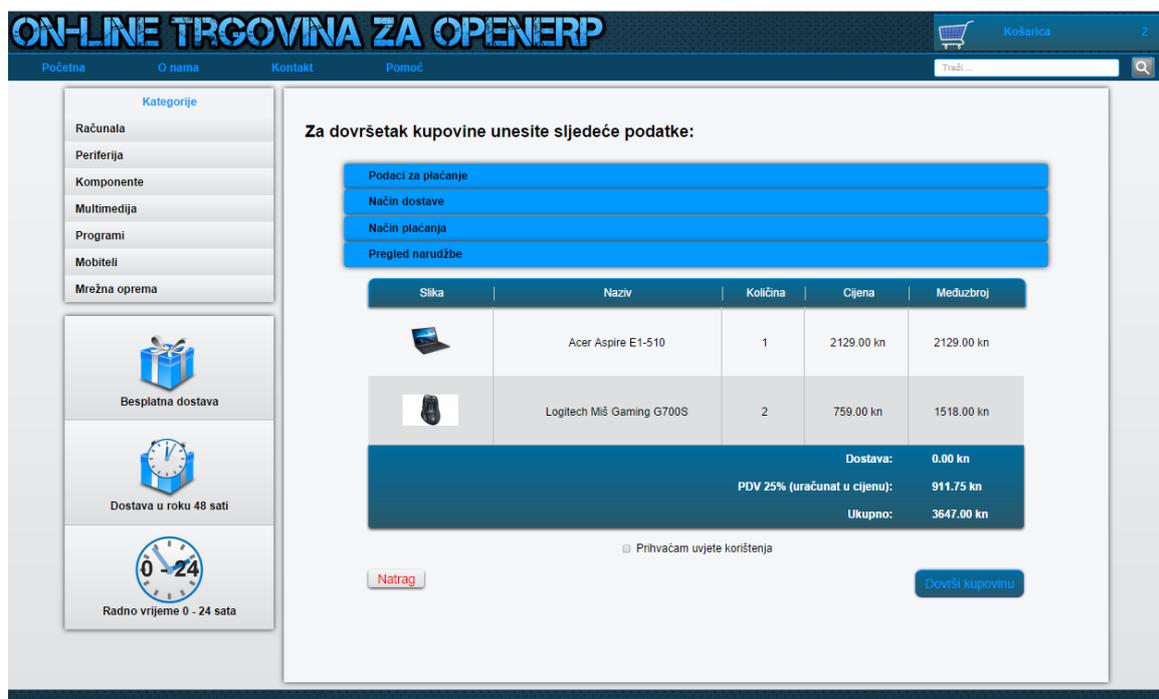


Slika 5. Košarica

Najprije je odabrati način kupovine. Ako se odabere prijava onda se potrebno prijaviti i u sljedećem koraku se automatski popune polja za dostavu. Ako se odabere kao gost onda je u sljedećem potrebno unijeti tražene podatke. Nije moguće nastaviti ako svi podaci nisu ispravno popunjeni. Način dostave je posredstvom „Hrvatske pošte“. Moguće je platiti pouzecom ili preko PayPal računa. Na kraju imamo pregled narudžbe (Slika 7.) i prihvaćanja uvjeta korištenja te ako je sve uredi onda se završava kupnja klikom na gumb „završi kupovinu“.



Slika 6. Unos podataka za dovršetak kupovine



Slika 7. Pregled narudžbe

Nakon uspješno dovršene kupovine u OpenERP sustavu se generira otpremnica koju je potrebno potvrditi prilikom otpremanja proizvoda (Slika 8.).

7. Zaključak

U današnje vrijeme internet je jako raširen i puno korisnika koristi internet svakodnevno. Statistike iz 2013. godine pokazuju da 64 posto Hrvata starijih od 15 godina koristi Internet, od čega 36 posto korisnika kupuje preko interneta (Preko interneta kupuje trećina korisnika, 2013).

OpenERP sustav je najbolji ERP sustav koji je otvorenog koda. Preko milijun korisnika svakodnevno koristi taj sustav. OpenERP sustav možda i nije najbolje rješenje za on-line trgovinu ako bi se upotrebljavao samo u tu svrhu jer ima puno tablica koje usporavaju rad sustava, ali ako se već koristi u poduzeću onda je i poželjno da se koristi kako bi se izbjegli problemi u sinkronizaciji i redundanciji podataka. Odabrani način izrade zahtjeva puno znanja i vremena za shvaćanje i korištenje baze podataka.

U nadolazećoj verziji, OpenERP će imati čarobnjak za izradu stranica (eng. website buider) pomoću kojeg će se stranice moći kreirati na način da se mišem dodaju elementi za prikaz (eng. drag-n-drop). Također će imati modul za e-trgovinu (eng. e-commerce) koji će zajedno sa čarobnjakom za izradu stranica omogućiti da se on-line trgovina izradi u nekoliko sati.

8. Literatura

- [1] Fancybox. (2014). Fancy box – jquery dodatak. Preuzeto 22. kolovoza 2014. s <http://fancybox.net>
- [2] Herrington J D (2007) PHP trikovi. Zagreb: Miš.
- [3] Infodom. (2014). Odoo. <http://www.infodom.hr/default.aspx?id=587>
- [4] Kermek D (2014) HTML jezik. Materijali s predavanja, Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin.
- [5] Lider. (2013). Preko interneta kupuje trećina korisnika u Hrvatskoj. Preuzeto 1. rujna 2014. s <http://liderpress.hr/tehnopolis/preko-interneta-kupuje-trecina-korisnika/>
- [6] Moss G (2013) Working with OpenERP. Online pregled dostupnih stranica. Preuzeto 21. kolovoza 2014. s <http://books.google.hr/books?id=i-0tAgAAQBAJ&printsec=frontcover&hl=hr#v=onepage&q&f=false>
- [7] Odoo. (2014). Odoo/OpenERP. Preuzeto 15. travnja 2014. s <http://www.odoo.com>
- [8] Odoo. (2014). The Architecture of OpenERP. Preuzeto 3. rujna 2014. s https://doc.odoo.com/7.0/book/1/1_1_Inst_Config/1_1_Inst_Config_architecture/
- [9] OpenERPBook.com. (2014). OpenERp Book. Online knjiga. Preuzeto 3. rujna 2014. s <http://openerpbook.com/>
- [10] Php.net. (2014). PHP: Hypertext Preprocessor. Preuzeto 18. kolovoza 2014. s <http://php.net/>
- [11] Taylor M (2009). OpenERP: an introduction to Open Source ERP. Preuzeto 28. kolovoza 2014. s <http://www.techworld.com/blogs/open-for-business/openerp-an-introduction-to-open-source-erp-3538884/>
- [12] The jQuery Foundation. (2014). jQuery user interface. Preuzeto 20. kolovoza 2014. s <http://jqueryui.com>
- [13] W3schools (2014). Cascading Style Sheets. . Preuzeto 19. kolovoza 2014. s http://www.w3schools.com/css/css_intro.asp
- [14] Woothemes (2014). Flex slider – jquery dodatak. Preuzeto 23. kolovoza 2014. s <http://flexslider.woothemes.com>

9. Prilog

Uz ovaj rad je priložen CD na kojem se nalazi kompletan kod aplikacije zajedno s bazom podataka.