Aplikacija za web prodaju i vođenje poslovanja

Težak, Helena

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:110:995711

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-09**

Repository / Repozitorij:

Polytechnic of Međimurje in Čakovec Repository - Polytechnic of Međimurje Undergraduate and Graduate Theses Repository

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

HELENA TEŽAK

APLIKACIJA ZA WEB PRODAJU I VOĐENJE POSLOVANJA

ZAVRŠNI RAD

ČAKOVEC, 2022

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

HELENA TEŽAK

APLIKACIJA ZA WEB PRODAJU I VOĐENJE POSLOVANJA

APPLICATION FOR WEB SALES AND BUSINESS MANAGEMENT

ZAVRŠNI RAD

Mentor / ica:
Dr. Sc. Sanja Brekalo, prof. v.š.

ČAKOVEC, 2022.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 1. veljače 2022.

država:	**Republika Hrvatska**
Predmet:	**PHP programiranje-izborni**
Grana:	**2.09.06 programsko inženjerstvo**

ZAVRŠNI ZADATAK br. 2021-RAC-R-73

Pristupnik:	**Helena Težak (0313023781)**
Studij:	redovni preddiplomski stručni studij Računarstvo
Smjer:	Programsko inženjerstvo

Zadatak: **Aplikacija za web prodaju i vođenje poslovanja**

Opis zadatka:

Aplikacija se izrađuje u PHP-u te koristi bazu podataka za zapisivanje podataka koji se skupljaju od strane administratora i korisnika. Aplikacija se izrađuje na primjeru web dućana koji se bavi prodajom kućnih ljubimaca, opreme te hrane za iste. Za aplikaciju se predviđa da administratori unose svu robu koja se prodaje u sustav te tako prikazuju mrežno robu koju je moguće naručiti online preko dostave ili prikazuju ponudu u dućanu. Pozadinski dio aplikacije omogućuje izdavanje računa i vođenje skladišta dućana. Administratori nakon naručivanja robe izdaju račune za online prodaju automatski ili sami izrađuju račune za prodaju u trgovini. Klijenti u aplikaciji mogu pregledavati vlastite narudžbe i ispisivati račune u PDF formatu.

Zadatak uručen pristupniku: 1. veljače 2022.
Rok za predaju rada: 20. rujna 2022.

Mentor:

dr. sc. Sanja Brekalo, prof. v. š.

Predsjednik povjerenstva za završni ispit:

ZAHVALA

Ovim bih se putem htjela zahvaliti svojoj mentorici dr. sc. Sanji Brekalo, prof. v. š., na brojnim stručnim savjetima, strpljenju i što je uvijek bila na raspolaganju kada sam se susretala s nejasnoćama u vezi završnog rada.

Zahvaljujem se i ostalim profesorima na suradnji, pruženom znanju i što su uvijek bili na raspolaganju za moja pitanja.

Također se zahvaljujem svojim roditeljima na pruženoj podršci i strpljenju u periodu obavljanja stručnog studija.

Helena Težak

SAŽETAK

Tema ovog završnog rada je izrada web aplikacije za vođenje poslovanja i web prodaje na primjeru web trgovine koja se bavi prodajom kućnih ljubimaca, opreme te kvalitetne hrane za ljubimce. *Aplikacija* je izrađena u *PHP* programskom jeziku te koristi MySQL bazu podataka za *zapisivanje podataka* kao što su proizvodi, kategorije proizvoda, poruke, upiti, korisnici, košarica, lista želja, narudžbe i podaci o plaćanju i dostavi, a administratori i korisnici navedene podatke tada putem aplikacije skupljaju tako što se zapisuju u bazu. Administratori u sustav unose sve artikle koji su u prodaji te tim putem mrežno prikazuju robu koju je moguće naručiti mrežno preko dostave ili koja je dostupna u samom dućanu.

Pozadinski dio aplikacije omogućuje *izdavanje računa* i *vođenje skladišta* dućana, što uključuje dodavanje proizvoda te uređivanje i uklanjanje proizvoda s web aplikacije. Administrator nakon što korisnik naruči robu može promijeniti status narudžbe koji je vidljiv korisniku te automatski izdati račun za mrežnu prodaju ili ih sami izrađuju za prodaju u trgovini, a svaki račun sadrži pečat i potpis.

Administrator ima mogućnost blokiranja i brisanja pojedinih korisnika, kao i mogućnost odobravanja korisnika nakon njihove registracije. Što se tiče upita, nakon što korisnik pošalje upit, administrator može odabrati hoće li on biti vidljiv ostalim korisnicima ili će biti sakriven, a u slučaju da je upit neprimjeren, administrator ga može obrisati. Također postoji i mogućnost ispisivanja PDF dokumenata o popisu korisnika u bazi, popisu proizvoda (kućni ljubimci, oprema, hrana), popisu svih narudžbi i slično.

Korisnici u aplikaciji mogu naručivati dostupne proizvode, pregledavati proizvode u košarici, dodavati proizvode u svoju listu želja te pregledavati vlastite narudžbe i ispisivati račune pojedinih narudžbi u PDF formatu. Dodatna mogućnost koju imaju registrirani korisnici je slanje upita / poruka administratoru u vezi narudžbe ili u vezi proizvoda.

Ključne riječi: *aplikacija, PHP, phpMyAdmin, zapisivanje podataka, izdavanje računa, vođenje skladišta*

SADRŽAJ

1. UVOD 7
2. Materijali i metode 8
3. DIJAGRAMI 10
   3.1 Dijagram slučajeva korištenja 10
   3.2 Dijagram rada aplikacije 12
   3.3. E.R model 13
4. BAZA PODATAKA I FUNKCIONALNOSTI 15
   4.1. Tablice 16
   4.2. Sesije 18
   4.3. Funkcionalnost 20
      4.3.1 Prijava korisnika 24
      4.3.2 Registracija korisnika 27
      4.3.3 Dodavanje novih proizvoda 30
      4.3.4 Pregled proizvoda 31
      4.3.5 Dodavanje proizvoda u košaricu i na listu želja 34
      4.3.6 Uređivanje proizvoda, korisnika i narudžba 35
      4.3.7 Brisanje proizvoda, upita, poruka, narudžbi i korisnika 37
      4.3.8 Filtriranje i pretraživanje proizvoda 38
      4.3.9 Postavljanje narudžbe 38
5. ZAKLJUČAK 42
6. LITERATURA 43
7. POPIS SLIKA / FOTOGRAFIJA 44
8. POPIS KODOVA 45

1. UVOD

U današnje vrijeme broj korisnika u mrežnim trgovinama neprekidno raste, posebno proteklih nekoliko godina zbog pojave epidemije koronavirusa. Istraživanje je pokazalo da 55 posto kupaca odabire kupnju putem interneta. Kao glavni razlog internetske trgovine navode taj da im je na internetu lakše usporediti cijene i na taj način odabrati jeftiniji proizvod nego u dućanu. Još jedan važan razlog je i brži pronalazak traženih proizvoda zbog mogućnosti pretraživanja uz odabir filtera. **[1]**

Upravo tom činjenicom potaknuta je izrada aplikacije čiji je cilj upravljanje podacima u bazi jedne web trgovine i web prodaja. Glavne funkcionalnosti aplikacije su košarica, proces narudžbe (primatelj, adresa dostave, odabir načina plaćanja), PDF račun, upiti, paginacija proizvoda i forme za upravljanje proizvodima, korisnicima, upitima i narudžbama koje su vidljive administratoru.

Za pisanje i uređivanje koda koristio se Visual Studio Code koji ima podršku za većinu programskih jezika, kao i za PHP koji se koristi u radu. **[2]** PHP (engl. *Hypertext Preprocessor*) je skriptni programski jezik namijenjen za razvoj aplikacija pomoću kojeg bi aplikacija nad MySQL bazom podataka trebala izvršavati osnovne CRUD metode (engl. *Create, Read, Update, Delete*).

Podacima na web poslužitelju pristupa se preko RDBMS sustava (engl. *Relational Database Management System*), a podacima se upravlja preko PHP skripte u aplikaciji. Za izradu baze koristio se phpMyAdmin, grafičko sučelje razvijeno u PHP-u koje omogućuje stvaranje baze izravno na web poslužitelju. **[3]**

Dizajn web stranice izrađen je kombiniranjem HTML koda (engl. *HyperText Markup Language)*, CSS stila (engl. *Cascading Sytel Sheets*) i pomoću biblioteke za kreiranje animacija *Animate.CSS*.

2. Materijali i metode

Aplikacija je podijeljena na dva dijela od kojih je prvi namijenjen gostima te ne zahtijeva registraciju i prijavu u sustav, dok drugi dio zahtijeva prijavu i on je namijenjen registriranim korisnicima.

Neprijavljeni korisnici ili gosti imaju mogućnost pregledavanja proizvoda po kategoriji, filtriranje po vrsti ljubimca i pregled detalja proizvoda. Neprijavljeni korisnici ne mogu stavljati proizvode u košaricu, naručivati proizvode niti komunicirati s administratorom.

Drugi dio aplikacije namijenjen je registriranim korisnicima koji se moraju prijaviti u sustav kako bi imali mogućnost korištenja različitih opcija aplikacije. Registrirani korisnici mogu pregledavati proizvode te ih stavljati u košaricu ili u listu želja, mogu naručivati proizvode i pregledavati vlastite narudžbe s mogućnošću preuzimanja računa određene narudžbe.

Pri naručivanju proizvoda otvara se prozor gdje korisnici moraju ispuniti podatke za dostavu proizvoda. Registrirani korisnici imaju uvid u dostupnost narudžbe, a mogu naručiti proizvode koji su dostupni *online* i u poslovnici. Dodatne mogućnosti za registrirane korisnike su pisanje upita na stranici i kontaktiranje administratora pomoću forme ukoliko trebaju informacije vezane za narudžbu i slično.

Registrirani korisnici, kao i na mnogim drugim aplikacijama, mogu uređivati svoj profil, odnosno podatke poput korisničkog imena, broja telefona i adrese elektroničke pošte, a imaju i opciju mijenjanja lozinke.

Najveća prava u aplikaciji ima administrator, kojemu se nudi mnogo više mogućnosti od korisnika, kao na primjer pretraživanje i uvid u sve registrirane korisnike i njihove podatke, dok zbog zaštite korisničkih podataka korisnici nemaju uvid u navedeno.

Administratoru je omogućeno dodavanje novih proizvoda u aplikaciju gdje odabire njihovu kategoriju i dostupnost, odobravanje korisnika nakon njihove registracije, ali i blokiranje ukoliko su u aplikaciji pisali nešto neprimjereno; pregled svih narudžbi, upita i kategorija. Administrator može dodati nove kategorije, pri čemu se filtar kod pregledavanja proizvoda također poveća za dodanu kategoriju.

Administratoru je omogućeno primanje obavijesti o novim porukama, upitima i novim registriranim korisnicima, čije profile mora odobriti da bi se mogli prijaviti u sustav. Kada mu stigne novi upit, može ga postaviti kao često pitanje na stranici, što tada postaje vidljivo ostalim korisnicima, dok ga također može i sakriti ili samo označiti kao pročitano, pri čemu se miče obavijest o novom upitu.

Poruke šalju korisnici vezano za informacije o narudžbi, prigovore, neispravnost pošiljke, otkazivanje narudžbe i slično. Poruke se sastoje od jedne poruke korisnika i jedne poruke administratora, a kada korisnik dobije odgovor od administratora, na početnoj mu se stranici prikaže obavijest nakon što se prijavi. Tada može otvoriti poruku i označiti ju kao pročitano, pri čemu administrator vidi da je poruka pročitana.

U radu se opisuju sve navedene funkcionalnosti, a najlakše se prikazuju dijagramom korištenja (engl. *Use Case diagram*).

3. DIJAGRAMI

U nastavku rada prikazat će se dva dijagrama koja prikazuju kako su podijeljene uloge u aplikaciji te se iz njih može vidjeti kako aplikacija funkcionira. Prikazat će se i treći dijagram u kojemu se vidi struktura baze podataka i veze između tablica.

3.1 Dijagram slučajeva korištenja

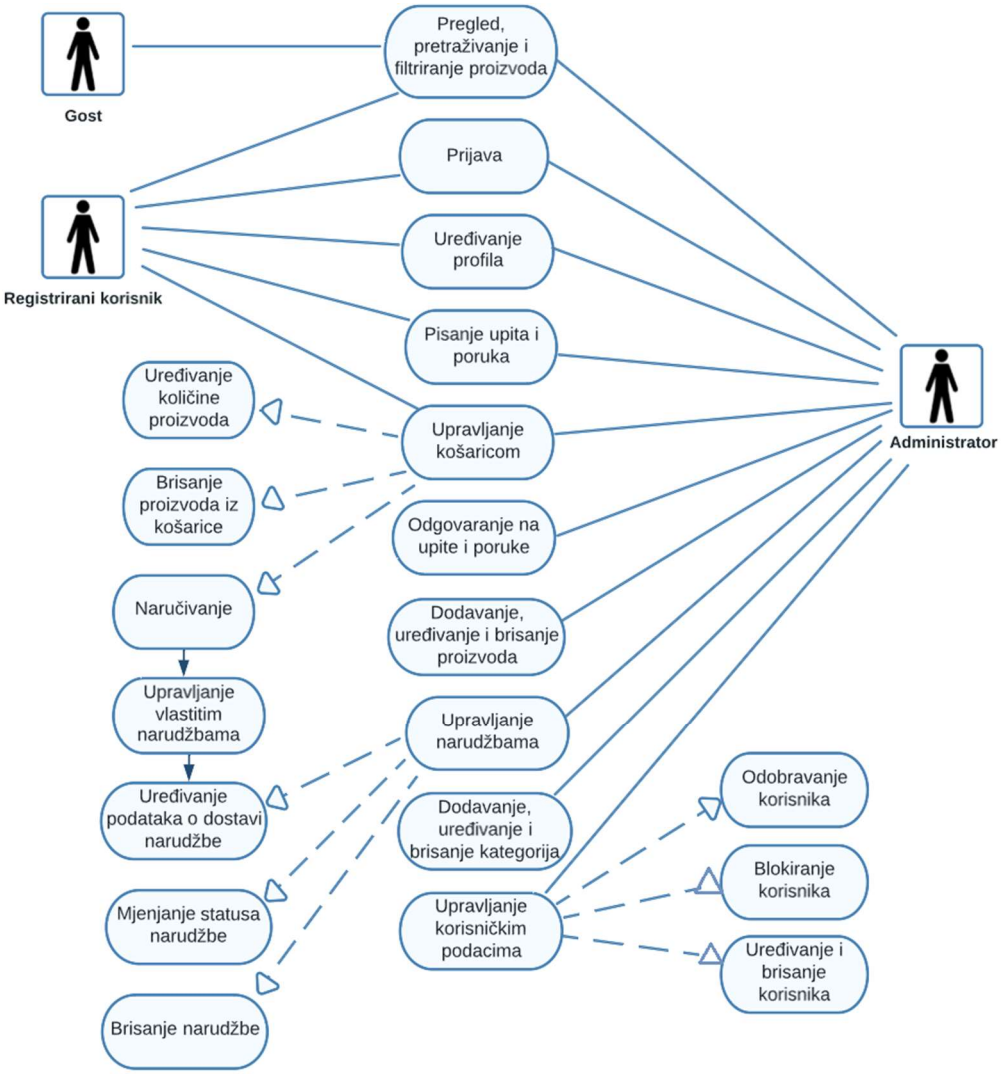
Dijagram slučajeva korištenja (engl. *Use Case Diagram*) općenito služi za vizualizaciju rada neke aplikacije ili procesa te opisuje funkcionalno ponašanje procesa s nečim što se naziva akter, a što u ovom slučaju predstavljaju gost, registrirani korisnik i administrator. **[4]**

Gost je neprijavljeni korisnik te mu je kao takvome pristup aplikaciji ograničen, točnije on može vršiti sljedeće aktivnosti: pregled proizvoda, filtriranje proizvoda i pretraživanje proizvoda na aplikaciji. Ostale mogućnosti u aplikaciji su mu onemogućene, za razliku od korisnika koji u odnosu na gosta, ima veći pristup aplikaciji.

Korisnik je onaj koji je odradio proces registracije i koji je već prijavljen u aplikaciju, a on može vršiti sljedeće: prijava, pregled proizvoda, pretraživanje proizvoda, filtriranje proizvoda, stavljanje proizvoda u košaricu u kojoj može uređivati količinu proizvoda koju naručuje i brisati proizvode iz nje naručivanje proizvoda, uređivanje podataka o dostavi vlastite narudžbe, kontaktiranje administratora i pisanje upita.

Administrator ima potpuni uvid u registrirane korisnike, proizvode i narudžbe te nad njima ima potpunu kontrolu. On može mijenjati status neke narudžbe izravno na tablici svih narudžbi ili na posebnom zaslonu za mijenjanje statusa.

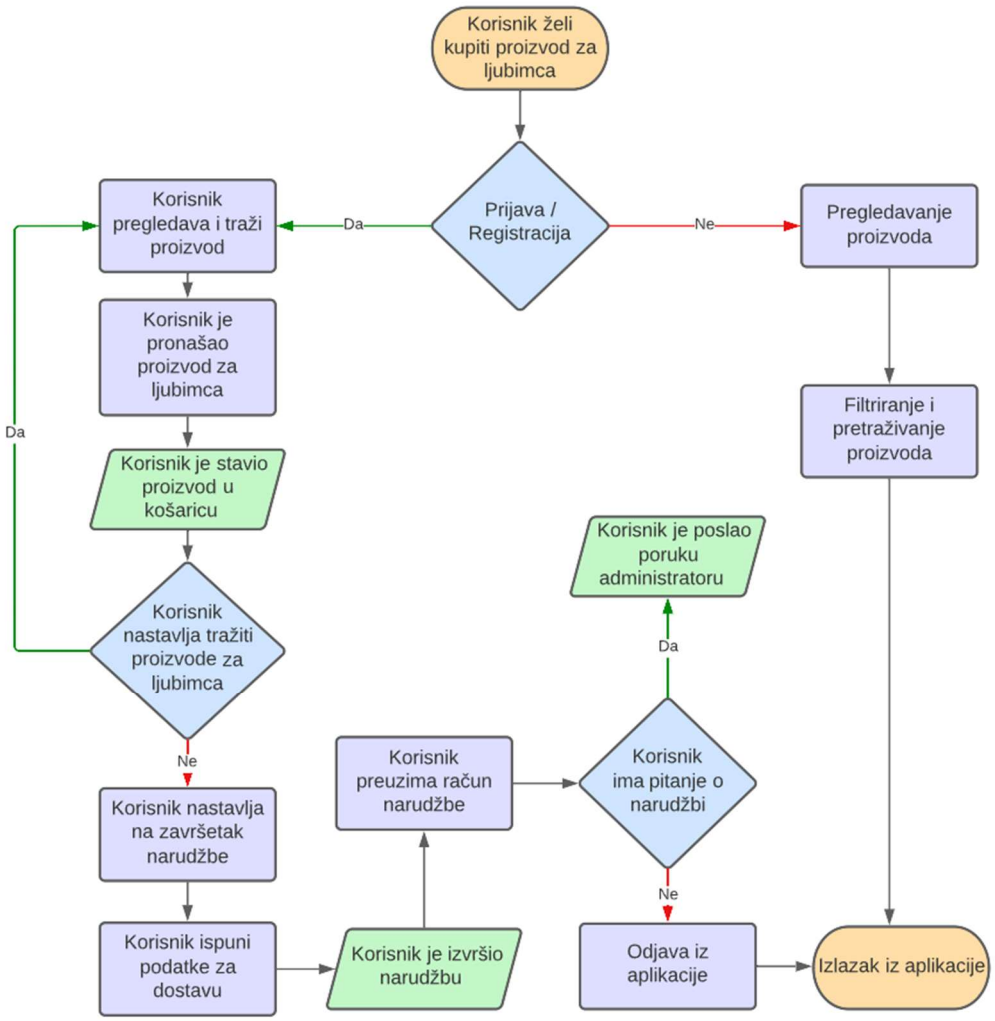
Slika 1 prikazuje dijagram slučajeva korištenja s troje aktera i s funkcionalnostima koje svaki akter ima u aplikaciji.



Slika 1. Use Case diagram
Izvor: izrada autora

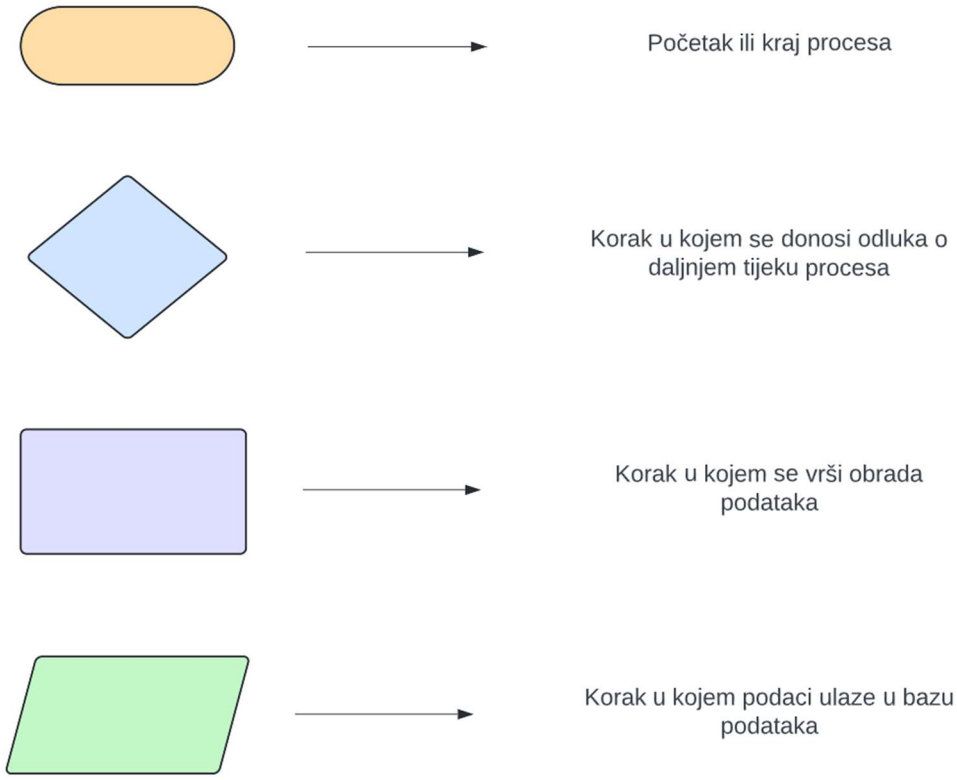
3.2 Dijagram rada aplikacije

Dijagram rada aplikacije ili dijagram toka prikazan je na Slika 2. To je dijagram u kojemu se prikazuju koraci nekog bitnog procesa. **[5]** U ovom slučaju prikazuje se najvažniji proces ovog rada, a to je proces kupovine proizvoda za ljubimce.



Slika 2. Dijagram rada aplikacije
Izvor: izrada autora

Svaki oblik na dijagramu rada aplikacije predstavlja drugu vrstu koraka u procesu, no mora se napomenuti da svaki oblik nema svoje određeno značenje, već to određuje osoba koja stvara dijagram. U ovom slučaju, da bi se bolje razumio dijagram i njegovi nacrtani procesi, Slika 3 objašnjava značenja korištenih znakova.

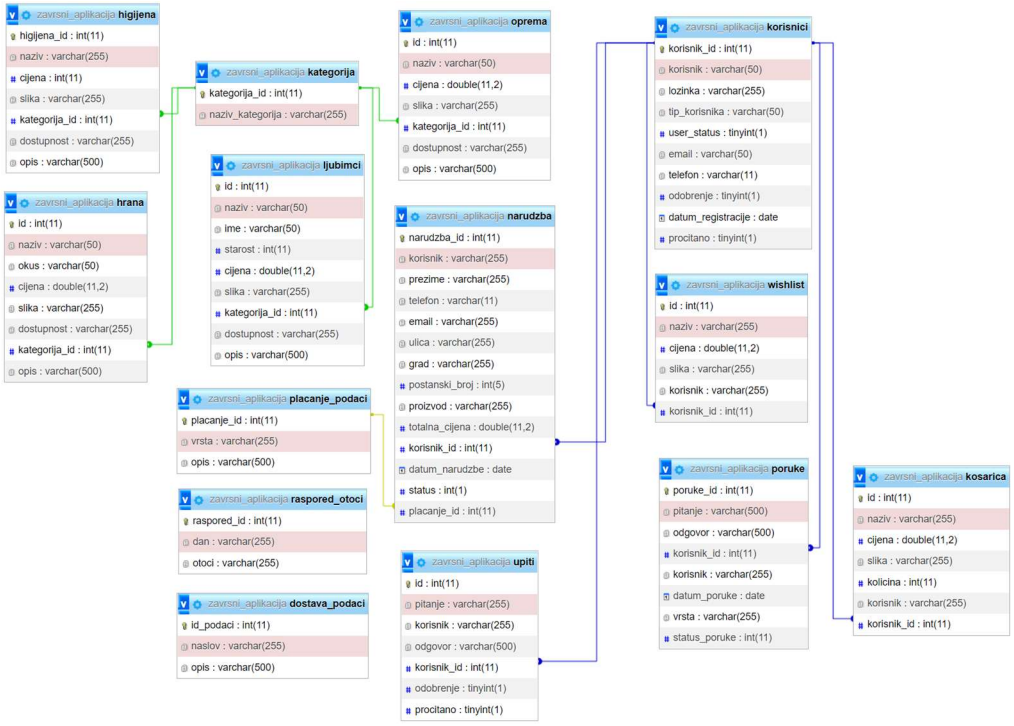


Slika 3. Objašnjenje korištenih znakova
Izvor: izrada autora

3.3.

E.R model

E.R (engl. *Entity Relationship model*) je dijagram koji prikazuje strukturu baze podataka, atribute i veze između tablica. Slika 4. prikazuje E.R model izrađene aplikacije.



Slika 4. E.R model
Izvor: izrada autora

Za jednostavno upravljanje MySQL bazom podataka i za izradu ovog dijagrama koristio se phpMyAdmin koji je automatski dobiven prilikom instalacije XAMPP web servera. PhpMyAdmin omogućuje jednostavno stvaranje, premještanje i kopiranje bilo koje baze podataka, stvaranje i uređivanje tablica i polja te upravljanje ključevima. Prilikom pokretanja XAMPP-a, potrebno je uključiti Apache web server i MySQL bazu podataka kako bi se moglo pristupiti aplikaciji.

4. BAZA PODATAKA I FUNKCIONALNOSTI

Za sve mogućnosti koje ova aplikacija pruža bilo je potrebno kreirati bazu podataka s tablicama u koje se spremaju podaci kao što su korisnici, narudžbe, proizvodi i slično. U datoteci „baza.php“ kreirana je konekcija (Kod 1.) za povezivanje aplikacije s bazom . Slika 5. prikazuje popis tablica koje su se koristile u izradi aplikacije.

Primjer 1:

```
<?php
    $konekcija=mysqli_connect ('localhost', 'root', '', 'zavrsni_aplikacija');

    if(mysqli_connect_errno())
    {
        echo "Greška s povezivanjem na bazu " . mysqli_connect_errno();
        die();
    }
?>
```

Kod 1. Kreiranje konekcije s bazom
Izvor: autor

Table	Action	Rows	Type	Collation
dostava_podaci	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci
higijena	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_unicode_ci
hrana	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8_unicode_ci
kategorija	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8_unicode_ci
korisnici	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_unicode_ci
kosarica	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci
ljubimci	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_unicode_ci
narudzba	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_croatian_ci
oprema	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8_unicode_ci
placanje_podaci	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci
poruke	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_unicode_ci
raspored_otoci	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_unicode_ci
upiti	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci
wishlist	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci
14 tables	Sum	74	InnoDB	utf8mb4_general_ci

Slika 5. Popis svih tablica
Izvor: izrada autora

4.1. Tablice

Glavne tablice u aplikaciji su korisnici, kosarica, narudzba, oprema, hrana, ljubimci i higijena. Za izradu aplikacije i upravljanje podacima iz baze podataka koristile su se DML naredbe (engl. *Data Manipulation Language*), a pod njih se ubrajaju INSERT, UPDATE, SELECT i DELETE naredbe.

Tablica *korisnici* sprema sve korisnike koji su izvršili proces registracije i njima se tada u bazu pod *odobrenje* sprema broj 1, što određuje je li korisnik potvrđen od strane administratora. Administrator najprije treba odobriti korisnika kako bi se mogao prijaviti u aplikaciju. Navedena tablica ima primarni ključ *korisnik_id*, koji je postavljen na AI (engl. *Auto Increment*) i preko kojeg su neke tablice vezane za ovu tablicu. Atribut *tip_korisnika* postavljen je svakom korisniku na *korisnik*, osim administratoru kojemu je postavljen na *admin* i pomoću tog se atributa određuju ograničenja na stranici. Prilikom registracije lozinka se iz korisničkog unosa sprema u bazu pomoću funkcije *password_hash*, koja stvara novu kriptiranu lozinku i prikazana je na Kod 2. Prilikom prijave u aplikaciju vrši se provjera podudara li se kriptirana lozinka s unosom korisnika pomoću funkcije *password_verify* koja je prikazana na Kod 3.

Primjer 2:

```
$hash = password_hash($lozinka, PASSWORD_DEFAULT);
```

Kod 2. Funkcija koja stvara novu *hash* lozinku
Izvor: autor

Primjer 3:

```
password_verify($lozinka, $rezultat['lozinka'])
```

Kod 3. Funkcija za provjeru lozinke
Izvor: autor

Tablica *kosarica* sprema naziv, cijenu, sliku i količinu proizvoda koji korisnik stavi u košaricu. Količina je na početku postavljena na broj 1, a korisnik kasnije može mijenjati količinu proizvoda, što je složeno pomoću *UPDATE* naredbe koja se koristi za izmjenu podataka u tablicama.

Navedena tablica ima jedan strani ključ (engl. *Foreign key*), *korisnik_id* pomoću kojega je tablica povezana s tablicom *korisnici* te dohvaća korisnika koji je stavio proizvod u košaricu i sprema njegov *korisnik_id* i korisničko ime (atribut *korisnik*) u tablicu. Atributi *korisnik* i *korisnik_id* su u ovom slučaju sesijske varijable koje su kreirane prilikom prijave.

Nakon što korisnik stavi u košaricu sve što planira kupiti, nastavlja dalje na završetak narudžbe gdje ispunjava podatke za dostavu kao što su primatelj narudžbe, adresa za dostavu i način plaćanja, dok su ostali podaci poput naručitelja, telefona i adrese elektroničke pošte spremljeni pomoću sesije. Kada korisnik završi narudžbu, ona se sprema u tablicu *narudzba*. Navedena tablica ima dva strana ključa, *korisnik_id* koji je već spomenut i *placanje_id* pomoću kojeg se dohvaćaju svi načini plaćanja koja postoje na stranici iz tablice *placanje_podaci*. Tablicom *placanje_podaci* može upravljati samo administrator, a na aplikaciji može dodavati i nove opcije plaćanja koje se tada vide u narudžbi.

Tablice *ljubimci*, *oprema*, *higijena* i *hrana* su slične, samo imaju različit broj atributa, a *id* svake tablice je postavljen na AI. Atribut *cijena* je u svakoj tablici tip podatka *double*, kako bi se cijena na aplikaciji prikazala u decimalnim brojevima. Sve tablice imaju strani ključ *kategorija_id* pomoću kojeg dohvaćaju kategorije. Kada administrator dodaje novi proizvod, mora odabrati jednu od kategorija.

Na sličan način napravljene su i tablice *upiti* i *poruke*, svaka od njih ima atribute *pitanje*, *odgovor*, *korisnik* i *korisnik_id*, gdje je *korisnik_id* ponovno strani ključ koji povezuje ove tablice s tablicom *korisnici*.

4.2. Sesije

Sesije omogućuju spremanje podataka u varijable koje se mogu koristiti kroz više stranica. Pohranjuju korisničke informacije i sadrže informacije o jednom korisniku koje ostaju zabilježene kroz skripte i one traju sve dok se ne zatvori preglednik. **[6]** Kod 5. prikazuje dohvaćanje varijabli sesije. Sesija se pokreće funkcijom *session_start()* koja se mora pisati na početku svakog dokumenta koji koristi sesiju, dok funkcija *session_destroy()* uništava *session ID*, prilikom čega ne uništava podatke koji su spremljeni u sesiji.

Primjer 4:

```
$_SESSION['korisnik']=$korisnik;
$_SESSION['korisnik_id'] = $rezultat['korisnik_id'];
$_SESSION['telefon'] = $rezultat['telefon'];
$_SESSION['email'] = $rezultat['email'];
```

Kod 4. Kreirane varijable sesije u login.php
Izvor: autor

Primjer 5:

```
$korisnik = $_SESSION['korisnik'];
$korisnik_id = $_SESSION['korisnik_id'];
$telefon = $_SESSION['telefon'];
$email = $_SESSION['email'];
```

Kod 5. Dohvaćanje varijabla sesije u narudzba.php
Izvor: autor

Primjer 6:

```
<?php
    session_start();

    //ponistavanje svih varijabli
    $_SESSION = array();

    //unistavanje cookia
    if(isset($_COOKIE[session_name()])) {
        setcookie(session_name(), '', time()-36000, '/');
    }

    //unistavanje sesije
    session_destroy();

    //redirekcija na stranicu prijave
    header("Location: login.php");
?>
```

Kod 6. Primjer korištenja session_start() i session_destroy()
Izvor: autor

Na Kod 6 prikazan je kod za odjavu iz aplikacije, funkcija *session_destroy* uništava sesiju, a korisnika koji se odjavio iz aplikacije pomoću zaglavlja preusmjerava na stranicu za prijavu.

4.3. Funkcionalnost

Aplikacija se sastoji od različitih stranica koje nude različite funkcionalnosti korisnicima i administratoru. Početna stranica gostima nudi mogućnost prijave, registraciju, odabir proizvoda po kategoriji i pretraživanje proizvoda.

Navigacijska traka je kreirana u datoteci „navbar.php“ te se pomoću naredbe *require_once* navigacijska traka uključuje u „pocetna.php“. *Require_once* provjerava je li datoteka već uključena kako ne bi došlo do konflikata s funkcijama i varijablama.

U datoteci za kreiranje navigacijske trake postavljeno je koje opcije može vidjeti administrator, koje opcije može vidjeti korisnik te koje opcije može vidjeti neprijavljeni korisnik na navigacijskoj traci pomoću koda iz datoteke „tipovi_korisnika.php“ koja je uključena u datoteku s navigacijskom trakom.

```php
<?php
function Admin()
{
    if(isset($_SESSION['korisnik']) && $_SESSION['tip_korisnika'] == 'admin')
    {
        return true;
    } else
    {
        return false;
    }
}

function User()
{
    if(isset($_SESSION['korisnik']) && $_SESSION['tip_korisnika'] == 'korisnik')
    {
        return true;
    } else
    {
        return false;
    }
}
?>
```

Slika 6. Tipovi korisnika
Izvor: izrada autora

Slika 6. prikazuje provjeru tipa korisnika pomoću funkcija. Funkcije *Admin* i *User* provjeravaju je li prijavljeni korisnik administrator ili obični korisnik te ukoliko

je uvjet u *if* naredbi zadovoljen, ona vraća *TRUE* i navedeni tip korisnika može vidjeti određene opcije na aplikaciji.

Naslov „PET SHOP“ na početnoj stranici animiran je pomoću Animate.css biblioteke. To je biblioteka animacija koja se može koristiti u različitim preglednicima i koristi se za isticanje, početne stranice i za usmjeravanje pažnje. **[7]**

Administratoru se na navigacijskoj traci pored korisničkog imena prikazuje broj obavijesti, a taj je broj napravljen pomoću *COUNT()* funkcije koja vraća broj redaka koji odgovaraju određenom kriteriju. Slika 7. prikazuje izradu tih brojeva.

Slično tome je kartica na početnoj stranici gdje piše koliko je novaca zarađeno, a čiji je broj izračunat funkcijom *SUM()*. Funkcija *SUM()* vraća ukupni zbroj numeričkog stupca, što je prikazano na Kod 7.

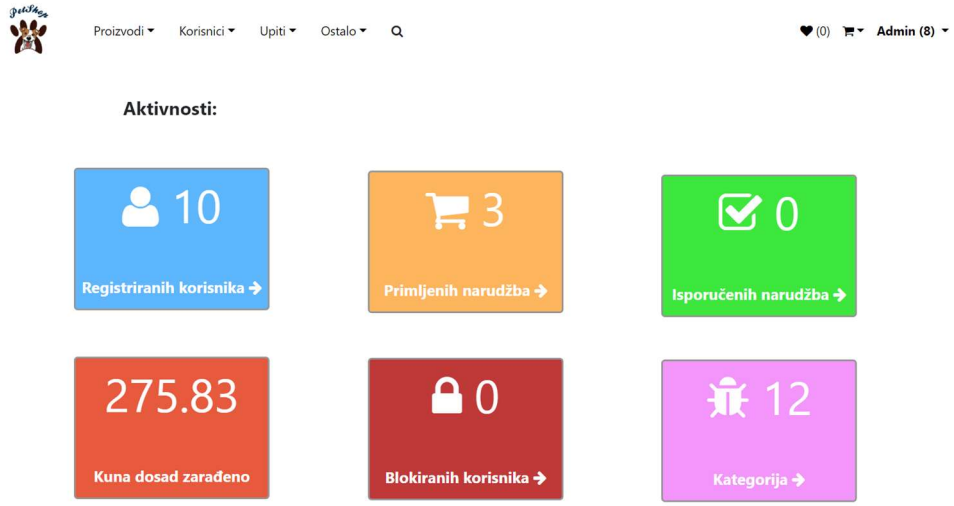
```
$korisnik_id = $_SESSION['korisnik_id'];

$query="SELECT COUNT(*) AS count FROM kosarica WHERE korisnik_id='$korisnik_id';";
$redci = $konekcija->query($query);
while ($row=$redci->fetch_assoc()){
    $output = $row['count'];
}

$query1="SELECT COUNT(*) AS count FROM wishlist WHERE korisnik_id='$korisnik_id';";
$redci = $konekcija->query($query1);
while ($row=$redci->fetch_assoc()){
    $output1 = $row['count'];
}
```

Slika 7. Korištenje COUNT() funkcije
Izvor: izrada Autora

Primjer 7:

```
 if(Admin()){

        $query6="SELECT SUM(totalna_cijena) AS sum FROM
narudzba;";
        $redci = $konekcija->query($query6);
        while ($row=$redci->fetch_assoc()){
            $output3 = $row['sum'];
        }
}
```

Kod 7. Korištenje SUM() funkcije za zbroj stupca
Izvor: autor



Slika 8. Izgled početne stranice kod administratora
Izvor: izrada autora

Na Slika 8 može se vidjeti da pored korisničkog imena „Admin“ piše broj obavijesti, što se kreiralo pomoću *COUNT()* funkcije, a na isti su način kreirane kartice različite boje na početnoj stranici za administratora.

Početna stranica za korisnika drugačija je od početne stranice za administratora. Korisnik nema uvid u kartice i nema toliko opcija u navigacijskoj traci kao administrator (Slika 9).

Slika 9. Izgled početne stranice za korisnika
Izvor: izrada autora

4.3.1 Prijava korisnika

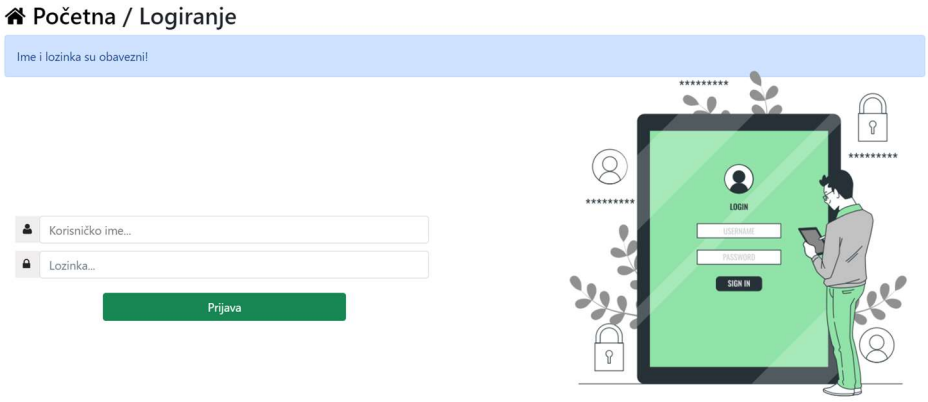
Za prijavu korisnika koristi se *SQL* naredba *SELECT*, pripremljena izjava koja se koristi za zaštitu od SQL napada i *bind_param* funkcija koja prima vrijednosti i određuje tipove varijabli, čime se također smanjuje rizik SQL napada (Kod 8.). U SQL-u se umeće upitnik ondje gdje se želi zamijeniti cijeli *integer*, *string*, *double* ili *blob* vrijednost. Pripremljena se izjava sastoji od pripreme (engl. *Prepare*) i izvođenja (engl. *Execute*). U fazi pripreme šalje se uzorak izjave koja se izvršava, a izjava se priprema tako da se na mjesta na koja će doći varijable stavi upitnik. Nakon pripreme slijedi izvođenje preko *execute* metode koja vraća *TRUE* ili *FALSE* izvršavanja izjave. U fazi izvođenja vežu se parametri i šalju poslužitelju nakon čega poslužitelj izvršava naredbu s vezanim vrijednostima.

Primjer 8:

```
$query = "SELECT * FROM korisnici WHERE korisnik=? LIMIT 1";
$query = $konekcija -> prepare($query);
$query -> bind_param('s', $korisnik);
if($query -> execute()){
     kod, validacija…
}
```

Kod 8. SQL naredba za prijavu korisnika
Izrada: autor

Prilikom prijave korisnik mora upisati točno korisničko ime i lozinku, a ukoliko je nešto krivo upisao, ispisuju se poruke validacije (Slika 10.).



Slika 10. Validacija
Izvor: izrada autora

Takva validacija kreira se pomoću funkcije *empty* koja provjerava je li varijabla prazna ili ne. Unutar *if* naredbe postavio se uvjet da ukoliko su varijable *korisnik* i *lozinka* prazni, ispisuje se poruka validacije, što je prikazano na Kod 9.

Primjer 9:

```
if(empty($korisnik) && empty($lozinka)){
   echo '<div class="alert alert-primary">Ime i lozinka su
   obavezni!</div>';
}elseif(empty($lozinka)){
   echo '<div class="alert alert-danger">Lozinka je
   obavezna!</div>';
}…
```

Kod 9. Validacija kod prijave
Izrada: autor

Ono što slijedi kod prijave je provjera ispravnosti upisane lozinke već spomenutom i objašnjenom funkcijom *password_verify*. Kod 10. prikazuje primjer provjere upisane lozinke.

Primjer 10:

```
if(password_verify($lozinka, $rezultat['lozinka'])){
  if($rezultat['user_status'] == 0){
      if($rezultat['odobrenje'] == 0){
        //varijable sesije
        } else{
      echo '<div>Administrator nije odobrio račun!div>';
        }
} else{
      echo '<div>Račun Vam je blokiran!div>';
}
}else {
      echo '<div class="alert alert-danger">Pogrešna
      lozinka, pokušajte ponovno!</div>';
}
```

Kod 10. Provjera lozinke
Izvor: autor

Unutar *if* naredbe postavlja se uvjet da ako su lozinka i korisničko ime ispravni, kreiraju se sesijske varijable koje pohranjuju korisničke informacije i traju sve dok se ne korisnik ne odjavi. Prilikom prijave također se provjerava status korisnika, točnije je li blokiran i je li odobren. Ukoliko nije odobren ili je blokiran, ne može se prijaviti i na ekranu mu se ispisuje odgovarajuća poruka. Oba parametra *user_status* i *odobrenje* tip su podatka *boolean* te kada je korisnik blokiran, *user_status* je postavljen na broj 1, a isto vrijedi i kada administrator još nije odobio korisnika.

4.3.2 Registracija korisnika

Pri registraciji je kreirano novo polje za validaciju (engl. *Array*) te se pomoću toga ispod svakog unosa ispisuje poruka (Kod 11.). Za upisivanje podataka koristila se funkcija *trim()* koja uklanja razmake.

Primjer 11:

```
$input_error = array();
if (empty($korisnik)) {
     $input_error['korisnik'] = "Korisničko ime je obavezno";
}else{
     $korisnik = trim($_POST['korisnik']);
}

if (empty($lozinka)) {
     $input_error['lozinka'] = "Lozinka je obavezna";
}

if (!empty($lozinka)) {
   if ($c_lozinka!==$lozinka) {
       $input_error['notmatch']="Lozinke se ne podudaraju!";
    }
}
```

Kod 11. Validacija kod registracije
Izvor: autor

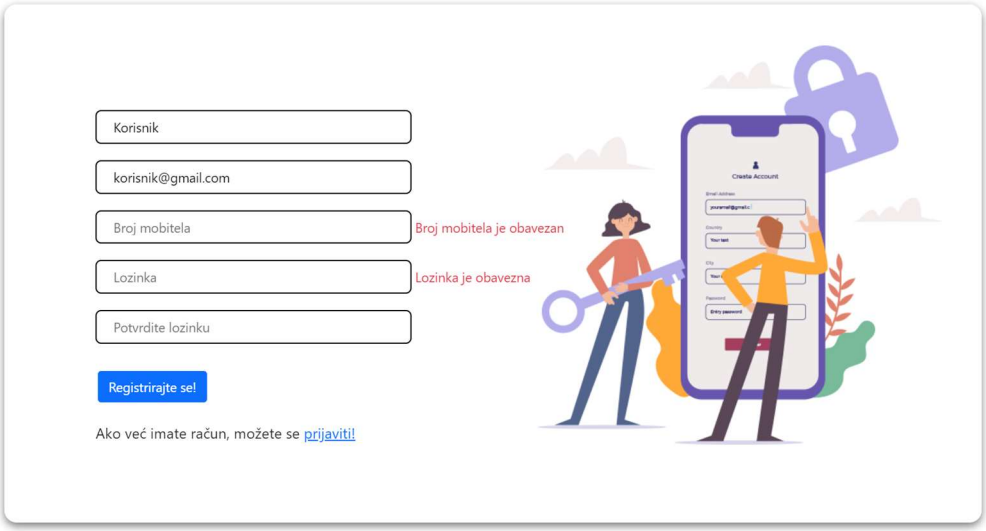
Pomoću *input_error()* određuje se što se ispisuje ukoliko je jedan od navedenih parametara prazan prilikom ispunjavanja forme za registraciju.

Primjer 12:

```
<input type="text" class="form-control2" value="<?=
isset($korisnik)?
$korisnik:'' ?>" name="korisnik" placeholder="Korisničko ime" />

<?= isset($input_error['korisnik'])? '<label class="text-
danger">'.$input_error['korisnik'].'</label><br>':'';  ?>
```

Kod 12. Kod za ispisivanje poruke validacije
Izvor: izrada autora

Da bi se poruka validacije ispisala pored unosa, stavljena je funkcija *isset* koja provjerava je li varijabla postavljena i da nije *NULL* (Kod 12.). Ukoliko varijabla postoji i nije *NULL*, ova funkcija vraća *TRUE*, a u suprotnom vraća *FALSE* te se ispisuje poruka da je potrebno unijeti određeni podatak, kao što je prikazano na Slika 11.



Slika 11. Izgled validacije na stranici za registraciju
Izvor: izrada autora

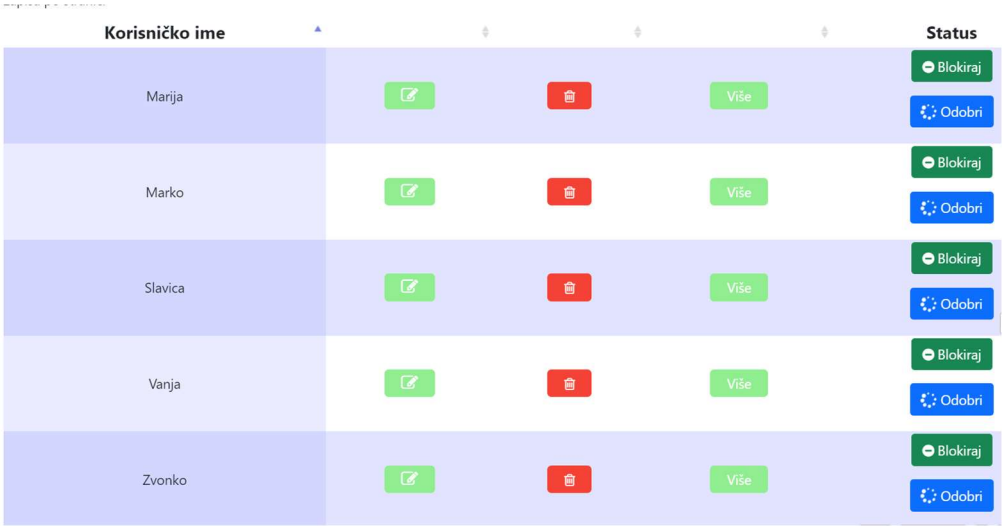
Ovakav način validacije složen je na cijeloj aplikaciji kod dodavanja novih proizvoda, kategorija, pisanja upita i popunjavanja forme za narudžbu (Kod 13.).

Primjer 13:

```
<input type="text" value="<?= isset($korisnik)? $korisnik:''
?>" name="korisnik"/>
```

Kod 13. Ispisivanje vrijednosti u input elementu
Izvor: izrada autora

Nakon registracije administrator mora odobriti korisnika, a to može učiniti na tablici gdje se nalazi prikaz svih korisnika. Slika 12 prikazuje tablicu sa svim korisnicima.



Slika 12. Prikaz svih korisnika
Izvor: izrada autora

4.3.3 Dodavanje novih proizvoda

Kao što su napravljene forma i validacija za registraciju, na isti način su napravljene forme za dodavanje hrane, ljubimaca, opreme i proizvode za higijenu, no jedina razlika je ta što kod dodavanja proizvoda administrator mora odabrati dostupnost i kategoriju proizvoda, u suprotnom se ispisuje poruka validacije.

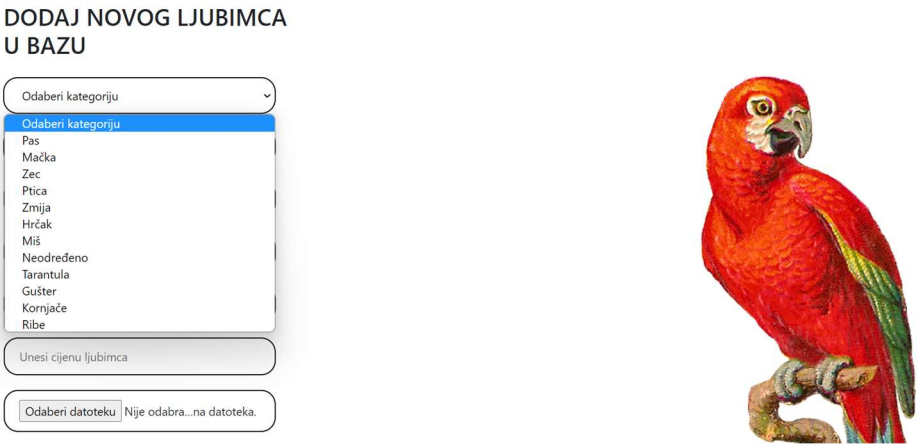
Ukoliko je atribut *dostupnost* različit od „Nedostupno“, *SQL* upit se izvršava i proizvod se upisuje u određenu tablicu u bazu podataka, ako je i ostatak forme dobro ispunjen.

Primjer 14:

```
<select name="kategorija_id">
       <option value="9">Odaberi kategoriju</option>
       <?php
          $query='SELECT * FROM kategorija';
          $redci = mysqli_query($konekcija, $query);
          while ($row=$redci->fetch_assoc()){
             echo"<option value=" .
$row['kategorija_id'] . ">" . $row['naziv_kategorija'] .
"</option>";
                        }
       ?>
</select>
```

Kod 14. Odabir kategorije kod dodavanja proizvoda
Izvor: autor

Nakon što su se odabrale sve kategorije iz tablice, unutar *value* u *option* elementu se ispisuju svi nazivi kategorija, što prikazuje Slika 13.

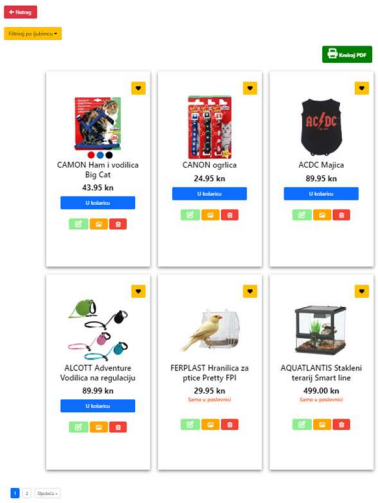


Slika 13. Ispis svih kategorija za odabir
Izvor: Izrada Autora

Ukoliko administrator ne odabere kategoriju, ona ostaje postavljena na broju 9, gdje je naziv kategorije „Neodređeno“. Proizvod se upisuje u bazu ukoliko je *kategorija_id* različita od broja 9 i ukoliko je ostatak forme ispravno ispunjen (Kod 14.).

Gumb za slanje podataka ima atribut *name* postavljen na „dodajLjubimca“ i to se poziva u naredbi *if*, a svi se podaci dobivaju iz obrasca i provjeravaju je li obrazac ispravno ispunjen nakon čega se podaci prosljeđuju preko POST metode.

4.3.4 Pregled proizvoda



Slika 14. Izgled stranice za pregled proizvoda
Izvor: izrada autora

Na stranicama za pregled hrane, ljubimaca, opreme i higijene postoji mogućnost pregledavanja proizvoda, stavljanja proizvoda u košaricu i na listu želja, filtriranje po kategoriji životinje, a administrator ima mogućnost uređivanja podataka o proizvodu, uređivanje slike proizvoda, brisanje proizvoda i preuzimanje PDF dokumenta o popisu proizvoda. Slika 14. prikazuje izgled stranice za pregled proizvoda.

Klikom na pojedini proizvod korisnika se preusmjerava na stranicu detalja proizvoda, gdje se može vidjeti i kategorija proizvoda koja je inače u zasebnoj tablici, pa se to izradilo pomoću *INNER JOIN-a* (Slika 15). *INNER JOIN* stvara novu tablicu kombinirajući retke i dohvaćajući podatke koji se pojavljuju u obje tablice (Kod 15.).

Primjer 15:

```
$query = "SELECT hrana.*, kategorija.naziv_kategorija
FROM hrana INNER JOIN kategorija ON hrana.kategorija_id
 = kategorija.kategorija_id WHERE hrana.id=$id";
```

Kod 15. Primjer INNER JOIN upita
Izvor: autor

U prikazanom upitu dohvaćaju se svi podaci iz tablice *hrana* i naziv kategorije iz tablice *kategorija*, a navedene su tablice povezane preko primarnog ključa tablice *kategorija*, a to je *kategorija_id*.

Slika 15. Izgled detalja proizvoda
Izvor: izrada autora

4.3.5 Dodavanje proizvoda u košaricu i na listu želja

Primjer 16:

```
if(isset($_POST['add_to_cart'])){
    $naziv = $_POST['naziv'];
    $cijena = $_POST['cijena'];
    $slika = $_POST['slika'];
    $korisnik = $_SESSION['korisnik'];
    $korisnik_id = $_SESSION['korisnik_id'];
    $kolicina = 1;

    $select_cart = mysqli_query($konekcija, "SELECT * FROM
    `kosarica` WHERE naziv='$naziv' AND korisnik_id =
    '$korisnik_id'");

    if(mysqli_num_rows($select_cart) > 0){
       echo '<div class="alert alert-warning">Proizvod je
       već u košarici!</div>';
       }else{
         $insert_product = mysqli_query($konekcija,
         "INSERT
         INTO `kosarica`(naziv, cijena, slika, kolicina,
         korisnik, korisnik_id) VALUES('$naziv',
         '$cijena', '$slika', '$kolicina', '$korisnik',
         '$korisnik_id')");
         echo '<div class="alert alert-success">Dodano u
         košaricu!</div>';
    }
}
```

Kod 16. Kod za dodavanje proizvoda u košaricu
Izvor: autor

Kod 16. prikazuje upit za dodavanje proizvoda u košaricu i upit za provjeru postoji li proizvod već u košarici. Za dodavanje u košaricu uzimaju se naziv, slika i cijena proizvoda, a kako bi korisnik vidio samo svoje proizvode u košarici, pomoću varijabli sesije se dohvaćaju njegovo korisničko ime i *korisnik_id*.

U slučaju da korisnik želi staviti isti proizvod u svoju košaricu, pomoću *SQL* upita provjerilo se ima li taj korisnik već određeni proizvod u košarici te ukoliko ima, ispisuje se poruka „Proizvod je već u košarici“. Na isti način je napravljeno dodavanje proizvoda u listu želja.

4.3.6 Uređivanje proizvoda, korisnika i narudžba

Uređivanje proizvoda u aplikaciji podijeljeno je na tri forme: formu za uređivanje podataka o proizvodu kao što su naziv, cijena i opis, formu za uređivanje dostupnosti i formu za uređivanje kategorije.

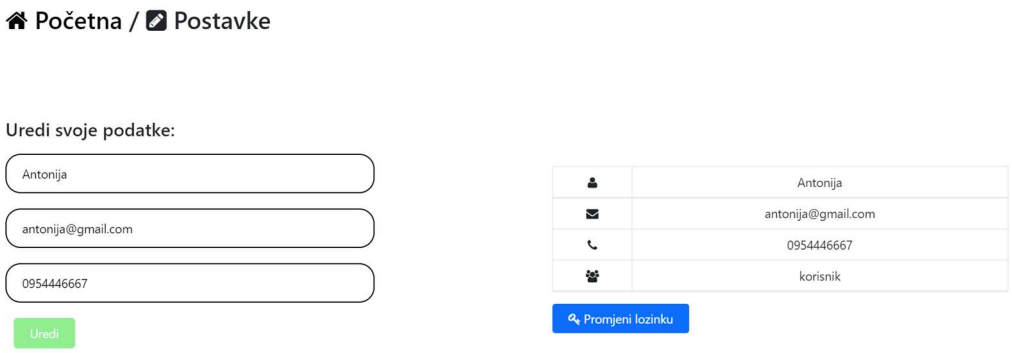
Na početku svakog dokumenta za uređivanje u aplikaciji je $_GET super globalna varijabla koja sadrži vrijednosti proslijeđene GET metode, u ovom slučaju sadrži *id* proizvoda.

```php
if(Admin()){
    if(isset($_POST['urediLjubimca'])){
        $naziv = $_POST['naziv'];
        $ime = $_POST['ime'];
        $starost = $_POST['starost'];
        $cijena = $_POST['cijena'];
        $opis = $_POST['opis'];

        if(empty($naziv) || empty($ime) || empty($starost) || empty($cijena) || empty($opis)){
            echo '<div class="alert alert-danger">Molimo ispunite sva polja</div>';
        } else{
            $update = $konekcija->prepare('UPDATE ljubimci SET naziv=?, ime=?, starost=?, cijena=?, opis=? WHERE id=?');
            $update -> bind_param('ssssss', $naziv, $ime, $starost, $cijena, $opis, $id);

            if($update->execute()){
                echo '<div class="alert alert-success">Proizvod je uspješno uređen!</div>';
                echo'<a type="button" href="ljubimci.php" class="btn btn-warning px-3 me-2">
                <i class="icon-arrow-left"></i>
                Vrati se na popis
                </a>';
            } else{
                echo '<div class="alert alert-danger">Neuspješno dodavanje u bazu!</div>';
            }
        }
    }
```

Slika 16. Uređivanje proizvoda
Izvor: izrada autora

Za uređivanje se izvršila provjera je li neko od polja prazno. U tom slučaju aplikacija ispisuje poruku „Molimo ispunite sva polja“, što se može vidjeti na Slika 16.

Na isti je način kreirano uređivanje korisnika, no tamo administrator može korisniku promijeniti samo korisničko ime, s obzirom da su ostalo privatni podaci korisnika koje sam korisnik može promijeniti u postavkama. Postavke su također napravljene na isti način kao uređivanje proizvoda i korisnika, ali tu stranicu može vidjeti samo korisnik koji je prijavljen.

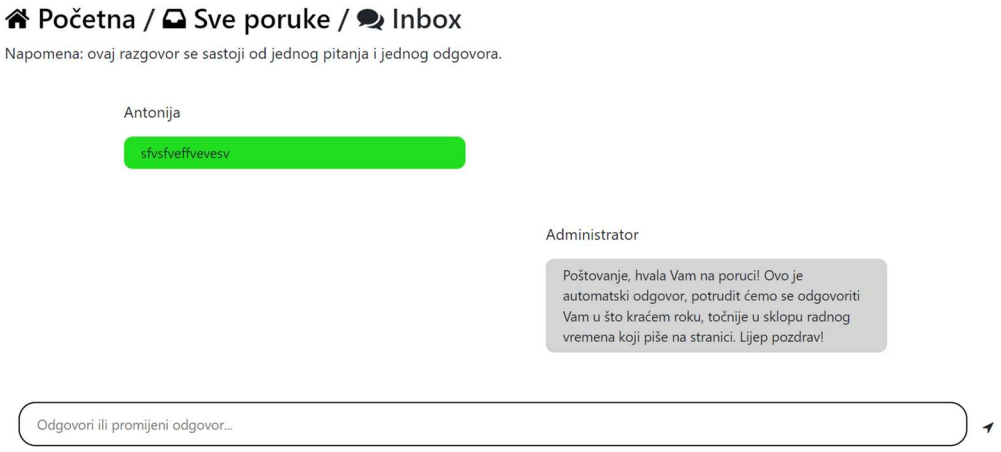


Slika 17. Stranica za uređivanje profila
Izvor: izrada autora

Korisnik može mijenjati svoje korisničko ime, broj mobitela i e-mail, a ispod tablice s osobnim podacima postoji gumb koji vodi na stranicu za promjenu lozinke, gdje se također uređuje pomoću *SQL* naredbe *UPDATE* i pomoću funkcije *password_hash* koja postavlja novu kriptiranu lozinku (Slika 17.).

Administrator može uređivati podatke o dostavi narudžbe ukoliko ga korisnik to zamoli preko poruke.

Pomoću *SQL* naredbe *UPDATE* kreirano je i odgovaranje na poruke i upite. Kada dobije novu poruku ili upit, administrator može na njih odgovoriti ili promijeniti svoj prethodni odgovor. Ukoliko administrator još nije odgovorio na poruku, korisnik na poruku dobije automatski odgovor koji se odmah sprema u bazu, a kada administrator odgovori, to se promjeni. Slika 18. prikazuje poruke između korisnika i administratora.



Slika 18. Primjer automatskog odgovora
Izvor: izrada autora

4.3.7 Brisanje proizvoda, upita, poruka, narudžbi i korisnika

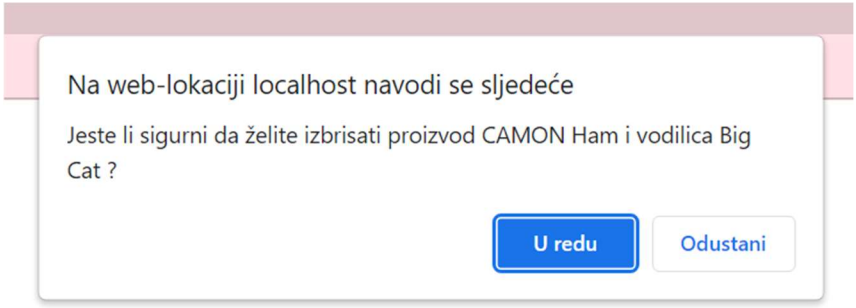
Brisanje se na stranici izvelo pomoću *SQL* naredbe *DELETE* i pomoću funkcije *bind_param*, što je prikazano na Kod 17.

Primjer 17:

```
if(isset($_GET['obrisi'])){
    $obrisi_id = $_GET['obrisi'];
     $query = $konekcija -> prepare('DELETE FROM hrana
     WHERE id=?');
    $query -> bind_param('s', $obrisi_id);

    if($query->execute()){
       echo '<div id="demo" class="alert alert-
       success">
       Proizvod je uspješno izbrisan!</div>';
    } else{
        echo '<div class="alert alert-danger">
       Proizvod neuspješno izbrisan!</div>';
    }
};
```

Kod 17. Brisanje proizvoda
Izvor: autor

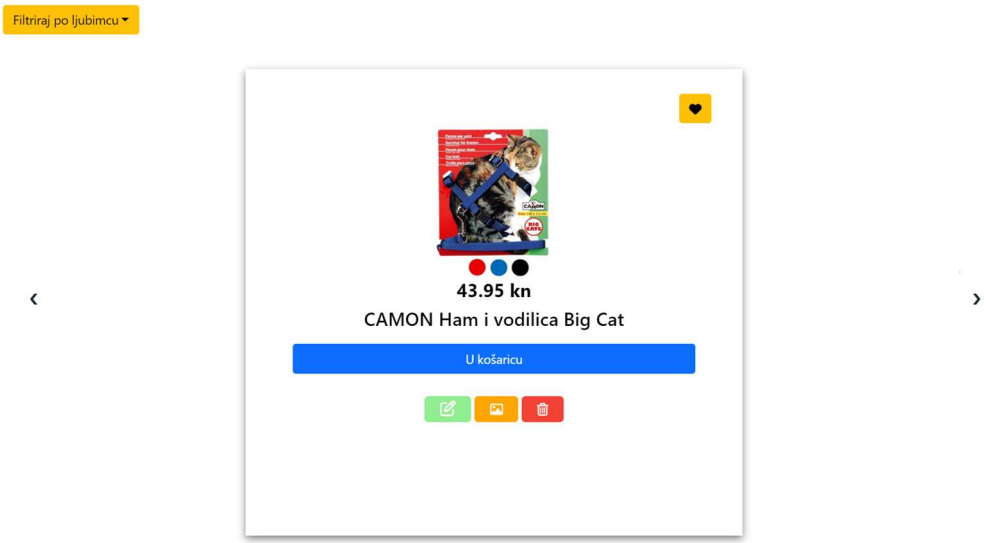
Kako bi administrator mogao potvrditi da želi obrisati određeni proizvod, kreirana je *JavaScript* metoda „confirm“ koja prikazuje dijaloški okvir s porukom, gumbom „U redu“ i gumbom „Odustani“, što se može vidjeti na Slika 19. Ukoliko je administrator kliknuo „U redu“, metoda vraća *TRUE*, u suprotnom vraća *FALSE* i brisanje proizvoda se otkazuje.



Slika 19. Okvir za potvrdu brisanja
Izvor: izrada autora

4.3.8 Filtriranje i pretraživanje proizvoda

U aplikaciji se može pretraživati po kategorijama ljubimca pomoću *dropdown-a.* Sa *SQL* naredbom *SELECT* prikazale su se sve kategorije unutar filtera te korisnik pomoću označivanja checkbox-a određuje koje kategorije želi vidjeti. Ukoliko korisnik ništa ne filtrira, proizvodi se prikazuju normalno, a ukoliko je korisnik odabrao kategorije, proizvodi se prikazuju tako da korisnik može vidjeti svaki proizvod zasebno sa strelicama koje vode ka sljedećem proizvodu, kao što je prikazano na Slika 20, a niz odabranih proizvoda ispisuje se pomoću *foreach* petlje.



Slika 20. Izgled filtriranih proizvoda
Izvor: izrada autora

4.3.9 Postavljanje narudžbe

Za završetak narudžbe, pomoću *SELECT* naredbe na stranici se dohvaćaju svi proizvodi koji su u košarici prijavljenog korisnika s količinom određenog proizvoda i sveukupnom cijenom. Korisnik mora ispuniti formu s traženim podacima kao što su adresa dostave narudžbe, primatelj i način plaćanja. Validacija forme je složena kao u cijeloj aplikaciji.

Totalna cijena dobiva se množenjem cijena proizvoda i njihove količine. Podaci naručitelja, odnosno prijavljenog korisnika, spremljeni su pomoću sesije i prikazuju se kada se narudžba završi, dok podatke o primatelju mora upisati naručitelj. Nakon završetka narudžbe, korisniku se prikažu detalji narudžbe i može preuzeti PDF račun

gdje se vidi broj narudžbe, datum, kontakt naručitelja, adresa i naziv primatelja, naručeni proizvodi, totalna cijena, način plaćanja i pečat.

Kada se narudžba spremi u bazu podataka, proizvodi se brišu iz košarice te se ispisuju detalji narudžbe. Upisivanje narudžbe u bazu podataka napravljeno je pomoću *bind_param()* funkcije i pripremljene izjave, kao što je prikazano na Kod 18.

Primjer 18:

```
$query = $konekcija->prepare('INSERT INTO narudzba(korisnik,
primatelj, telefon, email, ulica, grad, postanski_broj,
proizvod,totalna_cijena, korisnik_id, status, placanje_id)
VALUES (?,?,?,?,?,?,?,?,?,?,?,?)');

$query->bind_param('sssssssssss', $korisnik, $primatelj,
$telefon, $email, $ulica, $grad, $postanski_broj,
$proizvod, $ukupna_cijena, $korisnik_id, $status,
$placanje_id);

if ($query->execute()) {
$delete_cart = $konekcija->prepare("DELETE FROM `kosarica`
WHERE korisnik_id = ?");
   $delete_cart->execute([$korisnik_id]);
   //kod za prikaz detalja proizvoda
}else{
   echo '<div>Neuspješna kupnja!</div>';
}
```

Kod 18. Brisanje proizvoda iz košarice nakon narudžbe
Izvor: autor

Slika 21. Izgled košarice
Izvor: izrada autora

Korisnik prije narudžbe u košarici ima mogućnost mijenjanja količine proizvoda, brisanja pojedinog proizvoda iz košarice, a može i isprazniti cijelu košaricu. Izgled košarice prikazan je na Slika 21.

Nakon što dođe na formu za završavanje narudžbe, korisnik mora upisati tražene podatke, u suprotnom se ispisuju poruke validacije kao na Slika 22.

Slika 22. Forma za završetak narudžbe
Izvor: izrada autora

Slika 23. Ispis detalja nakon završetka narudžbe
Izvor: izrada autora

Nakon što korisnik izvrši narudžbu, forma za popunjavanje podataka o dostavi se miče i prikazuju se detalji narudžbe (Slika 23.). Narudžba je vidljiva administratoru u tablici gdje, ukoliko je potrebno, može uređivati podatke dostave, brisati ju i mijenjati status narudžbe (Slika 24.).

Slika 24. Izgled ekrana za upravljanje narudžbama
Izvor: izrada autora

5. ZAKLJUČAK

Pomoću PHP programskog jezika, Apache i MySQL baze podataka omogućen je razvoj aplikacije prikazane u ovom završnom radu. Pri izradi mrežnih aplikacija moguće je korištenje programskih okruženja kao što su biblioteke za lakše razvijanje i uljepšavanje izgleda aplikacije. Primjer takve biblioteke je Animate.CSS koja omogućuje animacije te FPDF klasa koja omogućuje izradu PDF dokumenata.

Za upravljanje podacima unutar aplikacije najznačajnije su bile DML naredbe koje se koriste pri umetanju podataka u tablicu, dohvat podataka iz tablice, uređivanju podataka i brisanju podataka iz tablice. Koristili su se i *INNER JOIN-ovi* za spajanje više tablica u jednoj datoteci.

Kako bi se korisnički podaci zaštitili od SQL napada koriste se pripremljene izjave i *bind_params* funkcija, a za zaštitu lozinke koristi se funkcija *password_hash.*

Za oblikovanje i upravljanje bazom podataka koristi se phpMyAdmin korisničko sučelje, a za dizajn stranice korišten je HTML kod i CSS stil.

Upotrebom svih navedenih programskih alata i funkcija za zaštitu podataka omogućena je izrada aplikacije s osnovnim funkcionalnostima koje upravljaju MySQL bazom podataka.

6. LITERATURA

[1] - © 2022. HANZA MEDIA d.o.o – Novo istraživanje otkriva kako hrvatski građani doživljavaju online kupovinu (11.02.2022.)

https://novac.jutarnji.hr/novac/aktualno/novo-istrazivanje-otkriva-kako-hrvatski-gradani-dozivljavaju-online-kupovinu-15118251

[2] - © 2001 – 2022 The PHP Group – PHP: HyperText Preprocessor (11.02.2022.)

https://www.php.net/

[3] – HDONWEB – phpMyAdmin kreiranje MySQL baze podataka (12.02.2022.)

https://www.hdonweb.com/programiranje/phpmyadmin-kreiranje-mysql-baze-podataka

[4] - © 2022 SlideServe – UML Modeli i dijagrami (25.08.2022.)

https://www.slideserve.com/hyacinth-robertson/uml-modeli-i-dijagrami

[5] – © Microsoft 2022 (25.08.2022.)

https://support.microsoft.com/hr-hr/office/stvaranje-osnovnog-dijagrama-tijeka-e207d975-4a51-4bfa-a356-eeec314bd276

[6] – Copyright 1999 – 2022 by Refsnes Dana (27.08.2022.)

https://www.w3schools.com/php/php_sessions.asp

[7] – Animate.css (28.08.2022.)

https://animate.style/

7. POPIS SLIKA / FOTOGRAFIJA

Slika 1. Use Case diagram 11
Slika 2. Dijagram rada aplikacije 12
Slika 3. Objašnjenje korištenih znakova 13
Slika 4. E.R model 14
Slika 5. Popis svih tablica 15
Slika 6. Tipovi korisnika 20
Slika 7. Korištenje COUNT() funkcije 21
Slika 8. Izgled početne stranice kod administratora 23
Slika 9. Izgled početne stranice za korisnika 23
Slika 10. Validacija 25
Slika 11. Izgled validacije na stranici za registraciju 28
Slika 12. Prikaz svih korisnika 29
Slika 13. Ispis svih kategorija za odabir 31
Slika 14. Izgled stranice za pregled proizvoda 31
Slika 15. Izgled detalja proizvoda 33
Slika 16. Uređivanje proizvoda 35
Slika 17. Stranica za uređivanje profila 36
Slika 18. Primjer automatskog odgovora 36
Slika 19. Okvir za potvrdu brisanja 37
Slika 20. Izgled filtriranih proizvoda 38
Slika 21. Izgled košarice 40
Slika 22. Forma za završetak narudžbe 40
Slika 23. Ispis detalja nakon završetka narudžbe 41
Slika 24. Izgled ekrana za upravljanje narudžbama 41

8. POPIS KODOVA

Kod 1. Kreiranje konekcije s bazom 15
Kod 2. Funkcija koja stvara novu *hash* lozinku 16
Kod 3. Funkcija za provjeru lozinke 16
Kod 4. Kreirane varijable sesije u login.php 18
Kod 5. Dohvaćanje varijabla sesije u narudzba.php 18
Kod 6. Primjer korištenja session_start() i session_destroy() 19
Kod 7. Korištenje SUM() funkcije za zbroj stupca 22
Kod 8. SQL naredba za prijavu korisnika 24
Kod 9. Validacija kod prijave 25
Kod 10. Provjera lozinke 26
Kod 11. Validacija kod registracije 27
Kod 12. Kod za ispisivanje poruke validacije 28
Kod 13. Ispisivanje vrijednosti u input elementu 29
Kod 14. Odabir kategorije kod dodavanja proizvoda 30
Kod 15. Primjer INNER JOIN upita 32
Kod 16. Kod za dodavanje proizvoda u košaricu 34
Kod 17. Brisanje proizvoda 37
Kod 18. Brisanje proizvoda iz košarice nakon narudžbe 39