Web aplikacija za dostavu paketa - poslužiteljski dio

Roguljić, Luka

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:641490

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Stručni studij

WEB APLIKACIJA ZA DOSTAVU PAKETA - POSLUŽITELJSKI DIO

Završni rad

Luka Roguljić

Osijek , 2016.

Sveučilište Josipa Jurja Strossmayera u Osijeku

ELEKTROTEHNIČKI FAKULTET OSIJEK

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:	
Studij, smjer:	
Mat. br. studenta, godina upisa:	
Mentor:	
Sumentor:	
Predsjednik Povjerenstva:	
Član Povjerenstva:	
Naslov završnog rada:	
Primarna znanstvena grana rada:	
Sekundarna znanstvena grana (ili polje) rada:	
Zadatak završnog rada	
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:	
Studij :	
Mat. br. studenta, godina upisa:	

Ovom izjavom izjavljujem da je rad pod nazivom**:**

izrađen pod vodstvom mentora

i sumentora

mojvlastiti 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 inobrazaccija.
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, ašto je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1 Opis zadatka 1

2. OPIS SUSTAVA 2

2.1 Ispitivanje sustava 2

2.2 Tehnologije 4

2.3 Podjela sustava na serverskom dijelu 4

3. BAZA PODATAKA 5

3.1 phpMyAdmin 6

3.2 Relacijski odnos u bazi podataka 9

3.3 XAMPP 9

4. STRUKTURA 11

4.1 PHP skriptni jezik 11

4.2 Spajanje na bazu podataka i poslužitelj 11

4.3 Registracija na sustav 12

4.4 Prijava u sustav 13

4.5 Odjava iz sustava 14

4.6 Uvid u aktivnosti korisnika 14

4.7 Upis narudžbe u tablicu sa narudžbama 15

4.8 Rad sa sjednicom 16

4.9 Administracija sustava 17

5. ZAKLJUČAK 21

LITERATURA 22

SAŽETAK 23

ABSTRACT 24

ŽIVOTOPIS 25

PRILOZI 26

1. UVOD

U današnje vrijeme se sve više i više koriste Internet tehnologije za razvoj aplikacija, mjesta za pohranu podataka (*eng. Cloud)*. Velika prednost ovakvih sustava što su dostupni na svakom pametnom uređaju (tablet, pametni mobitel i sl.), ali u drugu ruku je tu i strana sigurnosti na internetu koja je uvelike manja nego na drugim sustavima.

Sljedeći razmišljanja o tim tehnologijama, došlo se do ideje za stvaranje aplikacije koju bi korisnici koristili za slanje paketa. Aplikacija bi korisnicima omogućila da pružatelj usluge dostave pokupi paket za dostavu na njihovoj adresi, i dostavi je na odredište koje je korisnik upisao u obrascu za narudžbu usluge. Aplikacija bi imala tri različita sučelja za tri različite vrste korisnika na sustavu.

U poglavlju ispod je opisan zadatak kojeg je potrebno ispuniti za potpunu funkcionalnost aplikacije.

1.1 Opis zadatka

Potrebno je opisati način dostave paketa, modelirati bazu podataka koja podržava rad web aplikacije za slanje paketa registriranih korisnika. Također je neophodno ugraditi funkcionalnost pregleda svih paketa koje je korisnik slao sa svim relevantnim podacima o narudžbama. Potrebno je opisati svaku razinu aplikacije od poslužitelja, baze podataka, korisničkog sučelja, relacija između tablica u bazi podataka.

U narednim poglavljima objasniti će se postupci izrade rada od njegove pripreme, potrebnim znanjima i vještinama za njegovo izvođenje i načini rješavanja zadataka zadanih od strane mentora.

Završni rad je rađen u suradnji sa kolegom Josipom Tisajem. Kolega Tisaj je izrađivao korisničko sučelje.

2. OPIS SUSTAVA

Sustav dostave paketa u aplikaciji je razrađen na način da se osoba, koja želi iskoristiti usluge dostave paketa, registrira u sustav za dostavu paketa. Nakon što se kreirao korisnički račun, korisnik ima mogućnost korištenja aplikacije za svoje potrebe. Aplikacija pruža korisniku opcije praćenja svoje povijesti narudžbi kako bi imao uvid u svoju aktivnost. Cijena usluge ovisi o dva čimbenika koje sam korisnik bira prilikom narudžbe usluge. Jedan od faktora je prioritet dostave paketa koji korisnik bira od dva moguća ponuđena polja, a to su 'DA' i 'NE'. Cijena je veća za opciju 'DA' jer bi ona pružala uslugu dostave u roku od 24 sata za lokalno područje u krugu od 200 kilometara , a u roku od 48h za područje cijele države. Drugi čimbenik o kojem ovisi cijena usluge je veličina kutije koju korisnik bira iz obrasca za narudžbu usluge. Ponuđene veličine kutija su (sm =>mala, md=> srednja, lg=>velika, xl=> najveća) i ovisno o odabiru vrste prioriteta i veličine kutije kreira se konačna cijena usluge koja se ispisuje korisniku nakon uspješno obavljene narudžbe. Korisnik ima opciju upisivanja u bazu podataka i korištenje jedne funkcije koja u sebi ima naredbu 'SELECT' za uvid u svoju povijest narudžbi sa svim relevantnim podacima. Administrator baze podataka ima sve mogućnosti na radu sa bazom podataka kako bi je održavao, kreirao sigurnosnu kopiju i ostale operacije koje smatra potrebnima za održavanje rada sustava. Također će se kreirati posebno sučelje za vozače.

Sama izrada sustava je bila podijeljena u tri faze. Na početku zbog potreba sustava se kreirala baza podataka u jeziku za izradu bazi podataka MySQL-u , a nakon izrade baze podataka kreiran je obrazac za registraciju i obrazac za narudžbu kako bih se testirala funkcionalnost baze podataka i otklonile eventualne greške u sustavu. Nakon izrade baze podataka izrađen je logički dio pisanjem PHP koda koji odrađuje sve ono što je potrebno da bi sustav funkcionirao u cijelosti. Bitno je napomenuti da krajnji korisnik ne vidi što to PHP skripte rade u pozadini , a one se izvršavaju određenom akcijom koju korisnik sustava pokrene.

2.1 Ispitivanje sustava

Ispitivanje funkcionalnosti sustava se provodilo usporedno sa izgradnjom sustava tako da bi se što ranije uočile pogreške u sustavu koje bi se odmah ispravljale. Prvi ispit sustava je bio taj da se u skriptu, u kojoj je kod za spajanje na poslužitelj i bazu podataka, unese naredba 'INSERT' pomoću koje se ubacuju podaci u tablicu, a ako su podaci uspješno uneseni u tablicu to znači da

je spajanje na bazu podataka uspjelo. Drugi ispit sustava sadrži ispitivanje rada obrazaca sa skriptama napisanima u PHP jeziku koje nam omogućavaju unos podataka u bazu. Treće ispitivanje je bilo testiranje dodatnih funkcionalnosti sustava. Četvrti ispit sustava je ujedno i posljednji u kojem se testira sveobuhvatni sustav s implementiranim korisničkim sučeljem (HTML5, CSS3, Bootstrap RWD radni okvir, JS). Ako je sustav zadovoljio sve potrebe on je uspješno napravljen, ali nadogradnja sustava i održavanje sustava ne završava na tom mjestu.

Prva faza testiranja:

Izrađena je baza podataka i njene pripadajuće tablice (admins, drivers, orders, users (orders-users relacija)). Nakon izrade tablice preko jednostavne skripte su se pomoću naredbe INSERT ubacivali redovi da bi se testirala valjanost sustava i spoja na bazu podataka odnosno poslužitelja.

Druga faza testiranja:

Nakon izrađenih obrazaca za slanje podataka u tablice iz baze podataka i skripti koje postavljaju neke određene uvjete upisa u bazu podataka (prijava, narudžbe) , dodaje se konekcija na skripte te pripadajuće ime skripte za odgovarajući obrazac sa odgovarajućom metodom „POST“ u obrascu. Testira se registracija korisnika i narudžba. Testiranje je bilo pokazalo male nelogičnosti u određenim postavljenim uvjetima te su ti problemi otklonjeni bez problema. Iz ovog slučaja pojavljivanja problema se može zaključiti da sustav treba testirati u samom period razvoja, a ne na kraju izrade.

Treća faza testiranja:

Testiranje funkcionalnosti sučelja za vozače, administratore i povijesti narudžbi korisnika. Testiranje je prošlo uspješno bez poteškoća i zaključak je bio da je sustav pred krajem izrade

Četvrta faza testiranja:

Posljednja faza testiranja i najvažnija jer se spaja korisničko sučelje i skripte koje su kreirane za funkcionalni rad sustava. Ova faza je imala najviše grešaka zbog različitih imena skripti na koje

se pozivao dizajner pa je bilo potrebno svaki dokument pojedinačno pregledati da se te nejednakosti isprave. Nakon ispravaka nejednakosti sustav je funkcionalan i spreman za korištenje.

2.2 Tehnologije

Tehnologije korištene u izradi web aplikacije za dostavu paketa na klijentskoj strani su: HTML, CSS, JavaScript i Bootstrap radni okvir za responzivni dizajn. Klijentska strana je sve ono što korisnik sustava vidi na zaslonu uređaja preko kojeg koristi aplikaciju. Dio izrade klijentske strane aplikacije nije bio zadatak završnog rada, ali ga je potrebno spomenuti radi shvaćanja koncepta rada. Na strani poslužitelja je korišten PHP (rekurzivni akronim i backronim za „PHP: Hypertext Preprocessor“, prije „Personal Home Page Tools“), a za izradu baze podataka korišten je MySQL sustav s otvorenim kodom za kreiranje i upravljanje bazama podataka. Za razvoj baze podataka i pokretanje PHP skripti koristio se XAMPP program koji u sebi sadrži lokalni server za baze podataka (MySQL) i Apache server. Preko XAMPP-a se simulira server preko kojeg možemo pokrenuti aplikaciju.

2.3 Podjela sustava na serverskom dijelu

Sustav na poslužiteljskom dijelu podijeljen je u četiri cjeline, a to su: prijava/registracija, početna strana, naručivanje usluge i posljednji dio je pregled korisnikove povijesti aktivnosti na samom sustavu. Svaka od ovih cjelina u sebi sadrži rad s bazom podataka. Sustavu su još nadograđene funkcionalnosti za administratore i vozače.

3. BAZAPODATAKA

Baza podataka je kreirana u MySQL jeziku, a upravljanje bazom podataka se vrši preko sustava *phpMyAdmin*, koji je sustav otvorenog koda za kreiranje baze podataka, izradu tablica i popratnih operacija koje administrator baze podataka može vršiti nad bazom sukladno sa njegovim privilegijama koje ima. Super korisnik nad bazom podataka može vršiti sve postojeće operacije , tako u ovom sustavu postoji samo jedan administrator koji je ujedno i super korisnik i on uživa sve privilegije. Baza podataka sadrži dvije tablice. Jedna tablica se zove „users“ i u tu tablicu su spremljeni svi podaci koje korisnik sustava upisuje prilikom registracije, a ti uneseni podaci se koriste u daljnjem radu sustava za prijavu u sam sustav.

```
CREATETABLE `final`.`users`
( `id` INT(11)NOTNULL AUTO_INCREMENT , `name` VARCHAR(20)NOTNULL,
`lastname` VARCHAR(20)NOTNULL, `username` VARCHAR(20)NOTNULL,
`password` VARCHAR(20)NOTNULL, `email` VARCHAR(30)NOTNULL,
PRIMARYKEY(`username`),INDEX(`id`)) ENGINE = InnoDB;
```

Isječak 3.1 iz koda za kreiranje tablica (tablica „users“)

Druga tablica se zove „orders“ i ova tablica sadrži sve podatke koje je korisnik unio prilikom narudžbe usluge aplikacije. U tablici „users“ atribut „username“ je primarni ključ tablice i kao takav on je jedinstven u toj tablici i ne može se ponoviti, a u tablici „orders“, primarni ključ je atribut „id“ koji je identifikator svake narudžbe. Ove dvije tablice su povezane preko stranog ključa, gdje je u tablici „orders“ jedan od atributa atribut „username“ iz tablice „users“ i on u tablici „orders“ nije jedinstven , može se više puta ponavljati.

```
CREATETABLE `final`.`orders`
( `order_id` INTNOTNULL AUTO_INCREMENT , `city` VARCHAR(25)NOTNULL,
`contact` VARCHAR(25)NOTNULL, `adress` VARCHAR(35)NOTNULL,
`priority` VARCHAR(2)NOTNULL, `box_size` VARCHAR(10)NOTNULL,
`name` VARCHAR(15)NOTNULL, `lname` VARCHAR(15)NOTNULL,
`del_adress` VARCHAR(25)NOTNULL, `pob` VARCHAR(10)NOTNULL,
`price` VARCHAR(10)NOTNULL, `date` DATETIMENOTNULL,
`username`        VARCHAR(20)NOTNULL,PRIMARYKEY(`order_id`),INDEX(`username`))
ENGINE = InnoDB;
```

Isječak 3.2 iz koda za kreiranje tablica (tablica „orders“)

Također kreirana je nova tablica koja nije u relaciji sa ostalim tablicama iz baze podataka koja je kreirana za funkciju ovog sustava. U toj tablici imenom „admins“ pohranjeni su korisnički podaci o administratorima koji će moći pristupati informacijama o narudžbama u posljednjih 24 sata ili svim narudžbama u dosadašnjoj povijesti. Tablica je kreirana jednostavno sa samo dva

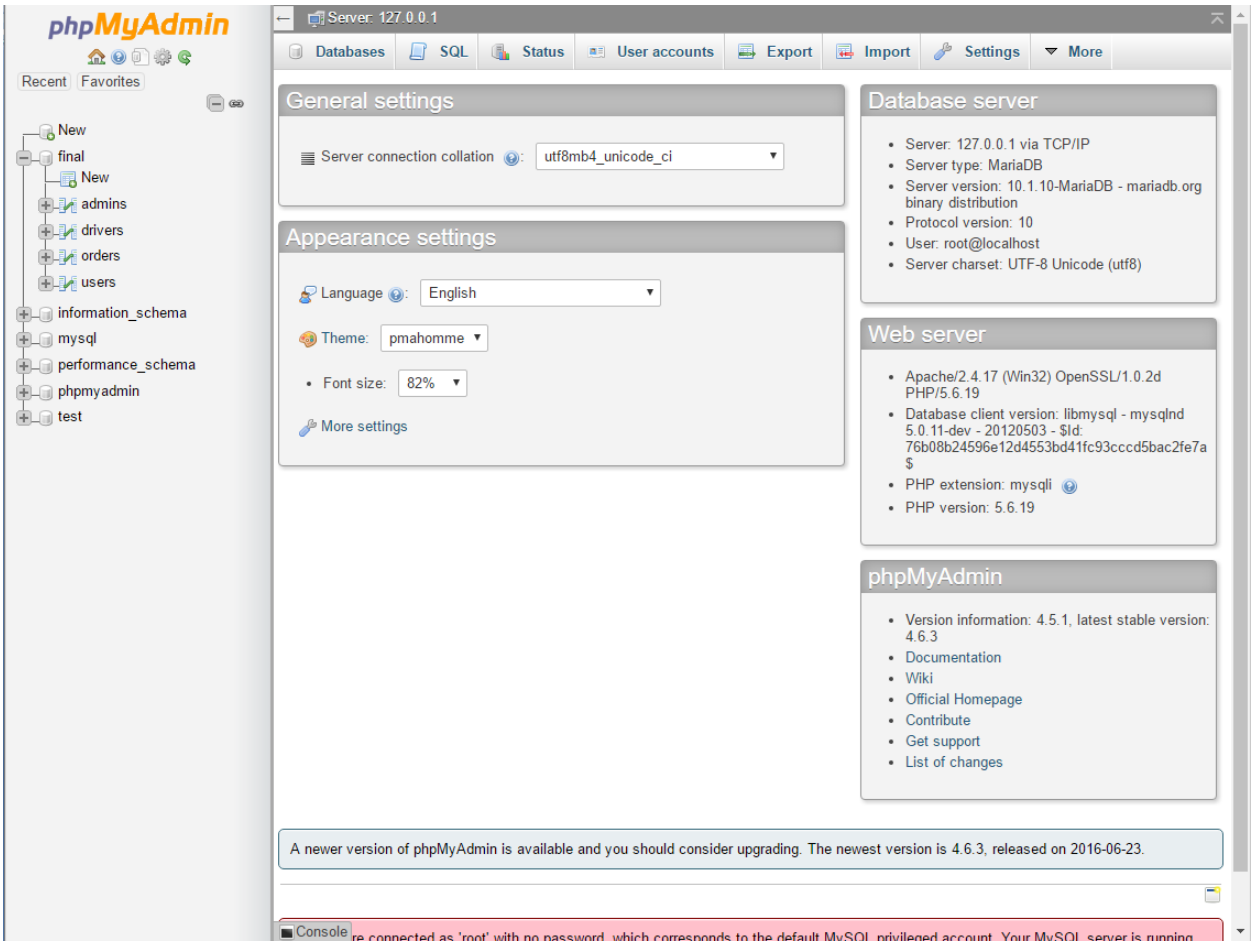
atributa , a to su „username“ kao korisničko ime administratora i „password“ atribut kao lozinka administratora.



Sl. 3.1 Program za simuliranje lokalnog servera XAMPP (apache i MySQL)

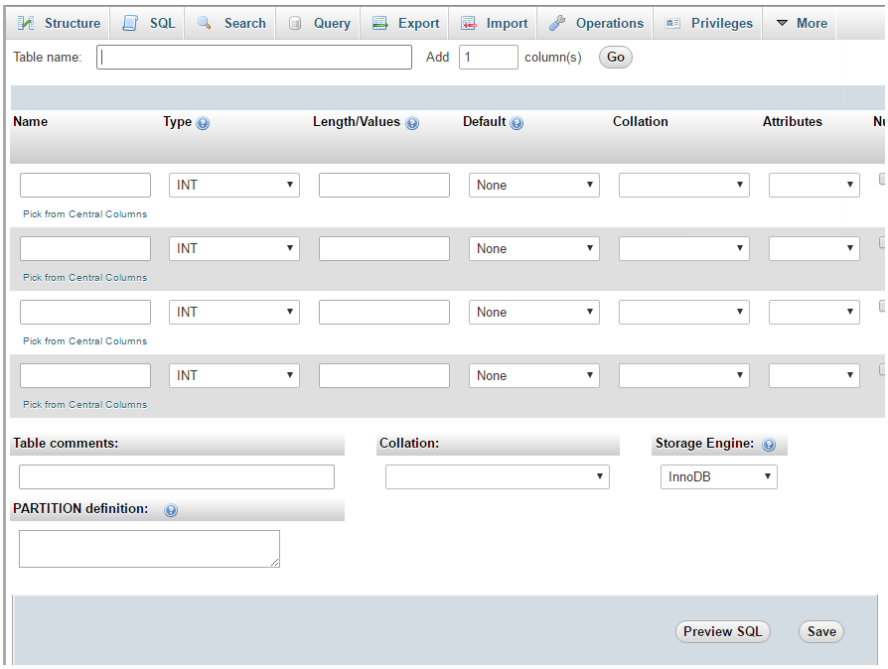
3.1 phpMyAdmin

Alat koji se koristio za upravljanje bazom podataka je *phpMyAdmin*, a *phpMyAdmin* je alat otvorenog koda i također besplatan za korištenje, a napisan je u PHP-u. Ovaj alat se koristi za lakše upravljanje bazama podataka korištenjem web pretraživača. U *phpMyAdmin-u* se mogu vršiti operacije kreiranja baze podataka, kreiranja tablica, dodavanja redova, stupaca, dodavanje odnosno oduzimanje privilegija određenim korisnicima, brisanje računa i slične operacije. Neke od značajki phpMyAdmina su: web sučelje, MySQL i MariaDB upravljanje, unos podataka iz CSV-a i SQL-a, izvoz podataka u različitim formatima (CSV, SQL, XML, PDF), administriranje više servera, stvaranje PDF grafike predloška baze podataka, stvaranje kompleksnih upita korištenjem Query by Example, globalno pretraživanje u bazi podataka ili u podskupu, transformiranje spremljenih podataka u bilo koji format korištenjem unaprijed definiranim funkcijama, stvarno vremenski grafikoni za motrenje aktivnosti MySQL servera i kompatibilnost sa različitim operacijskim sustavima.



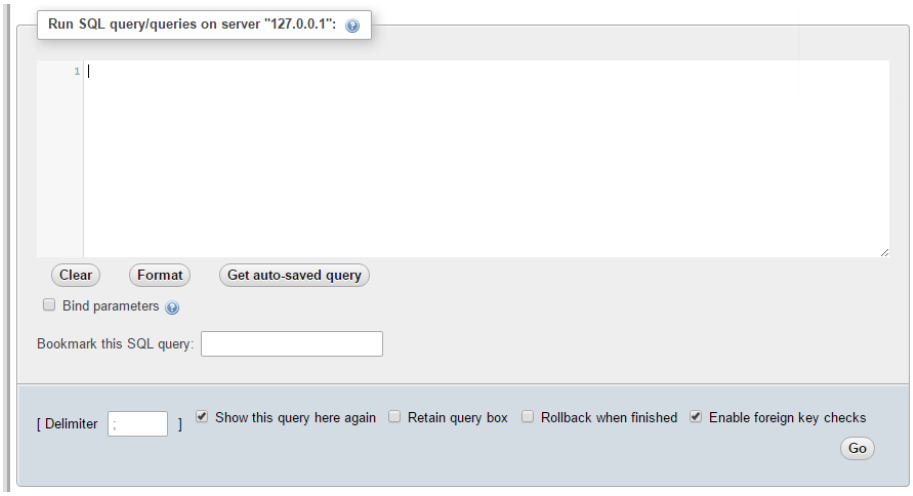
Sl. 3.1.1 Izgled *phpMyAdmin-a* u Internet pregledniku

Na priloženoj slici (Sl. 3.1.1) može se vidjeti izgled kontrolne ploče *phpMyAdmin-a*. Na lijevoj strani se nalaze dvije baze podataka koje je korisnik kreirao („game“ i „final“) , a ostale baze podataka dolaze u paketu sa programom i one su potrebne da bih se moglo pravilno simulirati lokalni server. U prozoru sa desne strane su podaci o serveru baze podataka, web serveru i linkovi na dokumente o *phpMyAdmin-u* koji se mogu koristiti kao nekakva svojevrsna pomagala. U gornjoj traci se nalaze alati pomoću kojih se upravlja bazom podataka. PhpMyAdmin se pokreće upisivanjem *localhost:8080/phpmyadmin* u traku web klijenta. Broj 8080 je port kojeg sluša Apache server. Dodatno, u kreiranju sustava su izrađene tablice za administratore i vozače. Značajka *phpMyAdmin-a* je ta što pruža kreiranje bazi podataka, tablica i relacija preko gotovog obrasca (Sl. 3.1.2).



Sl. 3.1.2 Prozor za rad preko gotovog obrasca u *phpMyAdmin*

Također pruža kreiranje istih upisivanjem MySQL upita u prozor koji je predviđen za takve upite (Sl. 3.1.3).



Sl. 3.1.3 Prozor za upisivanje SQL naredbi u *phpMyAdmin*

3.2 Relacijski odnos u bazi podataka

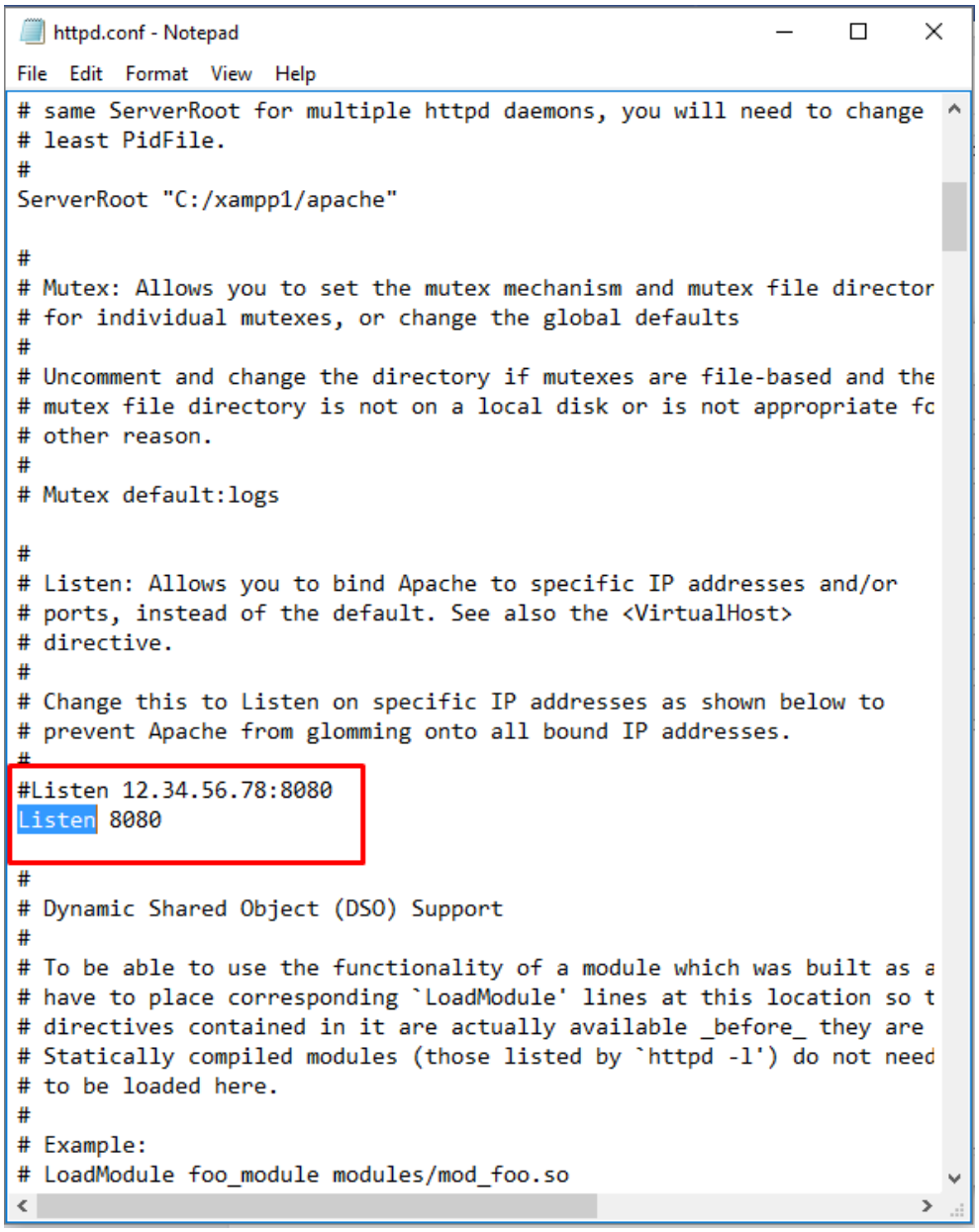
Tablice „users“ i „orders“ su u relacijskom odnosu preko stranog ključa koji povezuje te dvije tablice. Bitno je napomenuti da ako želimo stvoriti strani ključ u tablici „orders“ , treba se stvoriti jedan atribut, ne nužno istog imena kao atribut na koji će se vezati u tablici „users“, ali asocijativnog imena za lakše razumijevanje relacije, koji će biti indeksni atribut i tek onda se na taj atribut može napraviti ograničenje (*eng. CONSTRAINT*) preko kojeg se određuje ponašanje stranog ključa i njegova svojstva. U ovom slučaju u sustavu baze podataka strani ključ je poveznica na atribut „username“ preko atributa u tablici „orders“ koji je nazvan isto „username“. U MySQL-u postoji tri vrste relacija, a to su: jedan-na-jedan, jedan-na-više, više-na-više. U ovom sustavu relacija je jedan-na-više, jer jedan korisnik može imati više narudžbi, ali jedna narudžba ima samo jednog korisnika. U priloženom isječku koda prikazan je MySQL kod za kreiranje ograničenja i stvaranje relacije između dvije tablice.

```
ALTERTABLE `orders` ADDCONSTRAINT `fk_username`
FOREIGNKEY(`username`)REFERENCES `final`.`users`(`username`)
ON DELETE NO ACTION ON UPDATE CASCADE;
```

Isječak 3.2.1 iz koda za kreiranje tablica (stvaranje ograničenja)

3.3 XAMPP

XAMPP je besplatan i program otvorenog koda te je napravljen za rad na više različitih platformi. XAMPP je kratica za Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) i Pearl (P). U razvoju sustava koristile su se samo dvije opcije XAMPP-a , a to su Apache koji je simulirao poslužitelja i MySQL opcija. Program je jednostavan za korištenje i rad s njim je uvelike olakšan zadnjim poboljšanim verzijama. Jedini problem koji je nastao korištenjem programa je taj, da su se vrata (*eng. Port*) koje koristi Apache, preklapala sa vratima koje koristi *Skype*, a to je su vrata s brojem 80.



Sl. 3.3.1 Postavljanje Apache poslužitelja na nezauzeta vrata

Zbog toga problema se nije mogla pokrenuti opcija Apache. Problem je riješen tako što su bile potrebne izmjene u postavkama Apache-a tako da se promjeni broj vrata koje „sluša“ Apache (Sl. 3.3.1). XAMPP koriste programeri za razvoj aplikacija jer je praktičan za korištenje i brzo simuliranje aplikacije. XAMPP je pogodan zbog toga kada programer želi vidjeti neku promjenu u kodu, ne mora prvo preko FTP klijenta postaviti izmjenu na poslužitelja.

4. STRUKTURA

Programski kod za poslužiteljski dio je pisan isključivo u PHP-u sa kombinacijom MySQL naredbi potrebnih za rad sa bazom podataka. Podijeljen je na nekoliko dijelova radi lakšeg održavanja programskog koda i samim time razumljivost koda je puno lakša. Dijelovi su: skripta za registraciju, skripta za prijavu, skripta sjednice, skripta uvida u aktivnosti (povijest narudžbi) korisnika, skripta za spajanje na bazu podataka i poslužitelj, skripta za odjavu i skripta za upis narudžbe u bazu podataka.

4.1 PHP skriptni jezik

PHP je skriptni jezik koji se koristi za izradu dinamičnih web stranica, a čija je sintaksa zasnovana na sintaksi programskih jezika C i Pearl. Danas PHP ima svojstva objektno orijentiranog skriptnog jezika što je danas neizostavan način programiranja. PHP kao skriptni jezik, izvodi se na serverskoj strani. PHP programski kod stvara HTML stranicu na poslužitelju i onda je šalje klijentu, što znači da korisnik ne može pristupiti PHP kodu nego može samo vidjeti HTML kod. Trenutna inačica PHP-a je PHP 7, a to je ujedno i do sada najbolja verzija i smanjila je brzinu izvođenja za pola. Danas oko 82% stranica na Internetu vrti PHP. PHP kod može biti ugrađen u HTML kod ili može biti pozvan u HTML kod pomoću posebnih oznaka. PHP kod se piše između oznaka koje predstavljaju PHP kod (<?php ….. ?>), a na isti način se piše i u HTML kodu. Pozivanje PHP koda u HTML-u se izvršava pomoću naredbe „include“ (npr. include("index.php"). PHP podržava razne baze podataka i internet protokole. Da bi korisnik pokrenuo PHP skriptu treba imati ili lokalni server ili server na Internetu, pri pokretanju PHP skripte nije potreban kompajler.

4.2 Spajanje na bazu podataka i poslužitelj

Spajanje na bazu podataka i na poslužitelj je neophodno ukoliko želimo raditi s bazom podataka na sustavu. Spajanje se radi u dva koraka, od kojih je prvi korak spajanje na poslužitelj na kojem se nalazi baza podataka. Spajanje na poslužitelj je realizirano tako da se kreirala varijabla i koja je pomoću funkcije *mysqli_connect* kojoj se kao argument predaje ime poslužitelja, korisničko ime i lozinka. Ako spajanje na poslužitelj ne uspije ispisuje se greška spajanja.

```
$dbc= mysqli_connect('localhost','root','');
if(!$dbc){
die("Database connection failed: ". mysqli_error());
}
```

Isječak 4.2.1 iz skripte za spajanje na poslužitelja

Drugi korak pri kreiranju spajanja je spajanje na bazu podataka koja se odnosi na aplikaciju. Spajanje na bazu podataka se realizira tako da se ponovno kreira varijabla koja sada koristi funkciju *mysqli_select_db* koja služi kao funkcija za odabir baze podataka , a njoj se kao argumenti predaju varijabla preko koje se izvršava spajanje na poslužitelj te ime baze podataka koje je vidljivo na prikazu prozora phpMyAdmin (slika 3.4).

```
$db_selected= mysqli_select_db($dbc,"final");
if(!$db_selected){
die("Database selection failed: ". mysqli_error());
}
```

Isječak 4.2.2 iz skripte za spajanje na bazu podataka

Ako spajanje na bazu podataka ne uspije, ispisuje se pogreška pomoću *mysqli_error().* Svaka slijedeća skripta koja će raditi sa odabranom bazom podataka u skripti za spajanje mora imati skriptu za spajanje pozvanu na početku svoga koda.

4.3 Registracija na sustav

Registracija na sustav se vrši preko obrasca napisanog u HTML-u, a preko "POST" metode se šalju podaci u bazu podataka, a kao akcija, preko koje će se poslati ti podaci, se postavlja na ime skripte koja je napisana za registraciju. Prilikom registracije ima nekoliko uvjeta koji moraju biti zadovoljeni da bi se korisnik uspješno registrirao. Kao prvi uvjet postavljen je da korisničko ime poslano iz forme za registraciju mora biti pisano alfanumeričkim znakovima (Isječak 4.3.1), ako uvjet nije zadovoljen onda se ispisuje : *„Unesite Vaše korisničko ime sa alfanumeričkim znakovima [A-Z] [a-z] [0-9]“*, a ako je uvjet zadovoljen i korisničko ime napisano alfanumeričkim znakovima procedura registracije se nastavlja i vrši se provjera postojanja korisničkog imena u tablici u kojoj su upisani svi korisnici koji su prethodno registrirani.

```
if(ctype_alnum($_POST['username'])){
$query="SELECT * FROM users WHERE username = '$_POST[username]'";
$result=mysqli_query($dbc,$query);
$num=mysqli_num_rows($result);
```

Isječak 4.3.1 iz koda za provjeru postojećeg korisnika

Taj postupak se vrši tako da se MySQL naredbom odabere red u kojem je korisničko ime jednako korisničkom imenu koje je predano preko forme i onda se prebrojava broj redova gdje postoji to korisničko ime. Broj redova mora biti jednak nuli, a ako nije jednak nuli znači da to korisničko ime postoji i vrača korisnika nazad na početnu stranu da se ponovno proba registrirati. Nakon toga ako je uvjet zadovoljen također se testira uvjet postojanja e-mail adrese u sustavu, čime se ograničava prijava korisnika jedan korisnik jedan e-mail. Ako je e-mail koji je upisan u obrascu prvi koji bi se upisivao u tablicu sa korisnicima nastavlja se procedura registracije i kao zadnji uvjet je postavljen da ponovljena e-mail adresa mora biti jednaka prvotno upisanoj adresi , te ponovljena lozinka mora biti jednaka prvotno upisanoj lozinki.

```
$query2="SELECT * FROM users WHERE email = '$_POST[email]'";
$result2=mysqli_query($dbc,$query2);
$num2=mysqli_num_rows($result2);
```

Isječak 4.3.2 iz koda za provjeru email adrese

Konačno se skidaju oznake sa upisanih vrijednosti radi sprječavanja nekakvih pokušaja ometanja sustava i onda se upisuju podaci iz obrasca u tablicu sa korisnicima pomoću MySQL naredbe „INSERT INTO“.

4.4 Prijava u sustav

Prijava u sustav se također vrši preko obrasca koji ima samo tri elementa: korisničko ime, lozinku i dugme za pokretanje skripte za prijavu u sustav. U obrazac korisnik unosi korisničko ime i lozinku nakon što je to uradio korisnik, klikom na gumb za pokretanje prijave u sustav, pokreće skriptu koja provjerava postoji li red u tablici s korisnicima koji ima vrijednosti atributa „username“ i atributa „password“ jednake vrijednostima koje je unio.

```
session_start();
if($_SERVER["REQUEST_METHOD"]=="POST"){
$myusername= mysqli_real_escape_string($dbc,$_POST['username']);
$mypassword= mysqli_real_escape_string($dbc,$_POST['password']);
$sql="SELECT id FROM users WHERE
username = '$myusername' and
password = '$mypassword'";
$result= mysqli_query($dbc,$sql);
```

Isječak 4.4.1 iz skripte za prijavu u sustav

Ako se lozinka i korisničko ime podudaraju i ako je broj takvih redova jednak jedan onda se korisnik uspješno prijavio na sustav i otvara mu se stranica u kojoj ima mogućnost narudžbe,

odjave i pregleda povijesti svojih aktivnosti. Također se stvara sjednica koja omogućuje korisniku prenošenje podataka o samom njemu kroz stranicu.

4.5 Odjava iz sustava

Odjava iz sustava se radi jednostavnim korakom, a to je da korisnik pokretanjem linka za odjavu pokreče skriptu koja uništava sesiju prijavljenog korisnika i vraća ga link na početnu stranu sustava.

```
session_start();
if(session_destroy()){
header("Location: index.html");
}
```

Isječak 4.5.1 iz skripte za prijavu u sustav

4.6 Uvid u aktivnosti korisnika

Uvid u aktivnosti korisnika omogućava korisniku otvaranje povijesti svojih narudžbi sa svim relevantnim podacima o narudžbama. Funkcija, koja omogućuje korisniku pristup svojim narudžbama, je realizirana tako kada korisnik klikne na gumb za ispis povijesti narudžbi, on zapravo pokreće skriptu koja u sebi sadrži upit (*eng. Query*) napisan MySQL jezikom. Taj upit odabire sve redove iz tablice „orders“ koje imaju vrijednost atributa „username“ trenutno prijavljenog korisnika. Nakon toga se pomoću PHP koda generira tablica u koju će biti upisane svi podaci.

Povijest :

order_id	city	contact	adress	priority	box_size	name	lname	del_adress	pob	price	username
13	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1
14	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1
15	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1
16	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1
17	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1
18	Osijek	123456	Kneza Trpimira 2b	Da	xl	FERIT	ETFOS	Kneza Trpimira 2b	31000	300	testuser1

Sl. 4.6.1 Generirana tablica za pregled povijesti pomoću PHP skripte

Naravno programer može kreirati tablicu u kojoj se prikazuju samo određeni atributi i imena atributa se mogu promijeniti u imena prihvatljiva i razumljivija korisnicima.

4.7 Upis narudžbe u tablicu sa narudžbama

Upis podataka predan iz obrasca za narudžbu izvršava se jednom jednostavnom funkcijom kojoj se kao argumenti predaju vrijednosti koje je korisnik upisao i odabrao u obrascu prilikom stvaranja svoje narudžbe.

```
function
dbinput($city,$contact,$adress,$priority,$box_size,$name,$lname,$del_adress,$
pob,$price,$date,$username,$dbc){
$sql=" INSERT INTO orders(city, contact, adress, priority, box_size, name,
lname, del_adress, pob, price,date,username)
        VALUES
('$city','$contact','$adress','$priority','$box_size','$name','$lname','$del_
adress','$pob', '$price','$date','$username')";
$result=mysqli_query($dbc,$sql);
echo"Cijena usluge iznosi $price HRK.\n";
echo"<a href='slanje_paketa.php'>Povratak na stranicu</a>;";
}
```

Isječak 4.7.1 iz skripte za upis narudžbe u tablicu (funkcija upisa u tablicu „orders“)

Kako u strukturi tablice kao atribut je postavljeno i vrijeme narudžbe. Stoga je bilo potrebno kreirati varijablu u koju bi se upisalo vrijeme kada je narudžba napravljena, a varijabla koja se sprema u formatu „DATETIME“ omogućava lakše korištenje sustava ako bih se unazad svaka 24 sata radila lista sa narudžbama. Datum je formata godina/mjesec/dan sat/minute/sekunde.

```
$date=date('Y-m-d H:i:s');//DATUM UPISA
```

Isječak 4.7.2 iz koda za upis narudžbe u tablicu (format upisanog datuma DATETIME)

Potrebno je naglasiti kako sva polja obrasca trebaju biti ispunjena u protivnom se upis u bazu podataka neće izvršiti nego će se ispisati greška koja objašnjava da je korisnik pogrešno ispunio obrazac za narudžbu. Prilikom ispunjavanja forme postoje dvije varijable koje utječu na cijenu usluge , a to su veličina kutije (ima četiri opcije) i prioritet dostave (ima dvije opcije). Ovo znači da će biti ukupno osam kombinacija koje korisnik može izabrati iz ove dvije varijable. Taj problem je riješen pomoću if/elseif petlje gdje se prvo provjerava da li su sva polja postavljena pomoću isset() funkcije i !empty() funkcije koje provjeravaju da li je unos u određeno polje iz forme prazno ili nije. Ako je polje prazno on kao vrijednost vraća NULL i upis u bazu podataka

se neće izvršiti , a ako ne vrati vrijednost „NULL“ onda će se upis u bazu podataka izvršiti. Nakon što je upis izvršen, na prozoru web klijenta će se prikazati cijena usluge u HRK i poveznica za povratak. Gore priložen uvjet je poprilično velik zbog broja unosa koje korisnik unosi u formi za narudžbe.

```
if(isset($_POST['city'])&&!empty($_POST['city'])&&isset($_POST['contact'])&&!
empty($_POST['contact'])&&isset($_POST['adress'])&&!empty($_POST['adress'])&&
isset($_POST['priority'])&&!empty($_POST['priority'])&&isset($_POST['box_size
'])&&!empty($_POST['box_size'])){
if(isset($_POST['name'])&&!empty($_POST['name'])&&isset($_POST['lname'])&&
!empty($_POST['lname'])&&isset($_POST['del_adress'])&&!empty($_POST['del_adre
ss'])&&isset($_POST['pob'])&&!empty($_POST['pob'])){
if($_POST['priority']=='Da'&&$_POST['box_size']=='sm'){


dbinput($_POST['city'],$_POST['contact'],$_POST['adress'],$_POST['priority'],
$_POST['box_size'],
$_POST['name'],$_POST['lname'],$_POST['del_adress'],$_POST['pob'],$_POST['pri
ce']='150',$date,$login_session,$dbc);
}
```

Isječak 4.7.3 iz koda za upis narudžbe u tablicu (uvjet upisa)

4.8 Rad sa sjednicom

Sjednica je bitna stavka u ovoj aplikaciji, ali i inače na webu. Sjednica je način da se nekakva informacija spremi u nekakvu globalnu varijablu koja bi se mogla koristiti preko više stranica. Za razliku od kolačića, varijabla sjednice se ne sprema na korisnikovo računalo. Po unaprijed određenim postavkama sjednica se uništava kada korisnik zatvori web klijent, ili se može uništiti kada se pokrene naredba „session_destroy()“. Većina sjednica kreira jedan ključ koji je nalik na ovakav nekakav niz brojeva i slova: 765487cf34ert8dede5a562e4f3a7e12. Kada se pokrene sjednica na drugoj stranici, skenira se korisnički ključ i ako ključ je uspješno skeniran pokreće se pristup sjednici , a ako nije kreira se novi ključ.

```
session_start();
$user_check=$_SESSION['login_user'];
$ses_sql=  mysqli_query($dbc,"SELECT  username  FROM  users  where  username  =
'$user_check' ");
$row= mysqli fetch array($ses sql,MYSQLI ASSOC);
$login_session=$row['username'];
if(!isset($_SESSION['login_user'])){
header("location:login.php");
}
```

Isječak 4.8.1iz koda za kreiranje sjednice

U gore priloženom isječku iz koda za kreiranje sjednice je kreirana varijabla `$login_session` u koju se sprema vrijednost korisničkog imena koja je neophodna za rad korisnika sa funkcijama narudžbe i povijesti narudžbi jer MySQL upiti sadrže u sebi uvjet koji uspoređuje korisničko ime u tablici sa korisničkim imenom pohranjenog u sjednici. Gore je još napravljen uvjet ako sjednica nije postavljena da se korisnik preusmjeri ponovno na prijavu u sustav.

4.9 Administracija sustava

S obzirom na potrebu sustava za administracijom i upravljanjem rada sustava , kreirano je sučelje odvojeno od sučelja „običnog“ korisnika i to sučelje za administraciju je prilagođeno potrebama sustava za administriranje i održavanje rada sustava. Kreirana je i posebna tablica u bazi podataka imenom „admins“ u kojoj se preko phpMyAdmin-a unosi direktno korisnički račun administratora koji će moći onda pristupati korisničkom sučelju administratora

```
CREATETABLE `final`.`admins`
( `id` INTNOTNULL AUTO_INCREMENT ,
`username` VARCHAR(25)NOTNULL,
`password` VARCHAR(25)NOTNULL,INDEX(`id`)) ENGINE = InnoDB;
```

Isječak 4.9.1 iz koda za kreiranje tablice za administratore

Gore je priložen MySQL kod za kreiranje tablice „admins“ u bazi podataka „final“ . Tablica je kreirana da ima tri atributa od kojih je prvi atribut „id“ koji je postavljen na kao atribut kojemu je dodijeljen inkrement preko kojeg se može pratiti broj korisničkih računa administratora.
Kreirana su dva atributa koji predstavljaju korisničko ime i lozinku administratora bez kojih administrator ne može pristupati administrativnom sučelju.

Nakon kreirane tablice za administratore, nema potrebe za kreiranjem sjednice jer administrator neće raditi direktno sa sustava sa upitima za rad s bazom podataka. Administrator ima par funkcija koje mu omogućuju stvaranje tablice sa narudžbama u posljednjih 24 sata. Također administrator ima opciju da mu se kreira tablica sa svim narudžbama.

```
$sql="SELECT username AS KORISNIK, city AS GRAD, contact AS MOJ_KONTAKT,
adress AS MOJA_ADRESA, del_adress AS DESTINACIJA, pob AS PBROJ, price AS
CIJENA FROM orders ";
```

Isječak 4.9.2 iz koda za upit administratora (kreiranje tablice povijesti)

Generiranje tablice povijesti je postignuto sa kombiniranjem PHP koda i HTML oznaka za kreiranje tablice (Isječak 4.9.3).

```
$fields_num= mysqli_num_fields($result);
echo"<h1>Povijest :</h1>";
echo"<table border='1'><tr>";
// printing table headers
for($i=0;$i<$fields_num;$i++)
{$field= mysqli_fetch_field($result);
echo"<td>$field->name</td>";
}
echo"</tr>\n";
while($row= mysqli_fetch_row($result))
{echo"<tr>";
foreach($row as $cell)
echo"<td>$cell</td>";
echo"</tr>\n";
}
```

Isječak 4.9.3 iz koda za upit administratora (kreiranje tablice povijesti)

Varijabla `$fields_num`, iz gore priloženog isječka koda, prebrojava broj redova u tablici na kojoj se vrši upit i postavlja joj se vrijednost na broj redova u tablici. Nakon toga generira se tablica. Pomoću `for($i=0;$i<$fields_num;$i++)` i `mysqli_fetch_field($result)` i dohvaćaju se imena atributa, odnosno u ovom slučaju imena stupaca tablica. Na kraju ove funkcije za ispis povijesti, koriste se dvije petlje. Prva je petlja `while($row= mysqli_fetch_row($result)` koja, sve dok postoje redovi koji zadovoljavaju upit iz skripte, generira novi red pomoću `foreach` petlje, te se u svaku ćeliju upisuje vrijednost polja iz tablice „orders“.

Pristup narudžbama u posljednjih 24 sata je postignut uz minimalne izmjene u kodu za ispis ukupne povijesti narudžbi. Promjena je izvršena na upitu za bazu podataka gdje je dodan samo uvjet koji zadaje upitu da dohvati samo one redove koji su kreirani u posljednjih 24 sata.

```
$sql="SELECT username AS KORISNIK, city AS GRAD, contact AS MOJ_KONTAKT,
adress AS MOJA_ADRESA, del_adress AS DESTINACIJA, pob AS PBROJ, price AS
CIJENA FROM orders WHERE date >= now() - INTERVAL 1 DAY; „;
```

Isječak 4.9.4 iz upita za kreiranje tablice u posljednjih 24 sata

Na kraju razrade sustava uočena je potreba za kreiranjem dodatne tablice za vozače koji bi imali uvid u lokacije gdje bi kupili paket, odnosno lokacije gdje taj paket mora biti poslan, uvid u cijenu usluge, korisničko ime naručitelja i poštanski broj odredišta.

```
CREATETABLE `final`.`drivers`
( `username` VARCHAR(25)NOTNULL,
`password` VARCHAR(25)NOTNULL,
`id` INTNOTNULL AUTO_INCREMENT ,
INDEX(`id`)) ENGINE = InnoDB;
```

Isječak 4.9.5 iz MySQL koda za kreiranje tablice

Kreiranje korisničkih računa za vozače se također unosi ručno preko funkcije za ubacivanje redova u tablicu, preko phpMyAdmin-a. Ovakav način se radi iz razloga da samo administrator baze podataka može kreirati račun za vozača. Tablica „drivers“ nije ni u kakvom relacijskom odnosu sa ostalim tablicama u bazi podataka.

```
INSERTINTO `drivers`
(`username`, `password`, `id`)
VALUES('driver001','driver1',NULL),
('driver002','driver2',NULL);
```

Isječak 4.9.6 iz MySQL koda ubacivanje vozača u tablicu „drivers“

Vozači imaju samo pravo pristupa svome sučelju iz kojeg gledaju samo narudžbe korisnika sa mjestom na kojem moraju pokupiti paket i sa adresom i poštanskim brojem gdje se paket treba dostaviti. Nemaju nikakve dodatne funkcije u kojima uživaju administratori sustava. Može se reći da vozači imaju isti pristup i privilegije kao običan korisnik.

Dodavanjem sučelja za vozača i kreiranja potrebnih funkcija za ispravnost rada te razine izrada aplikacije je završena, ali njena nadogradnja je i dalje moguća. Trenutno aplikacija nije postavljena na javni poslužitelj na internetu nego se „vrti“ na lokalnom serveru i njoj pristup ima samo računalo sa XAMPP-om i svim skriptama i bazom podataka.

Kontrolna plo🞐a voza🞐a
Povratak na stranicu:

Povijest :

KORISNIK	GRAD	ADRESA	DESTINACIJA	PBROJ	CIJENA
testuser1	aa	bb	aba	12345	150
testuser1	Jane	Unknown	Unknown	1234	150

Sl. 4.9.1 Generirana tablica u sučelju vozača

5. ZAKLJUČAK

Kreirana aplikacija ima još mnogo mogućnosti za nadogradnju u budućnosti. Trenutno aplikacija radi samo na lokalnom serveru jer nije postavljena na web, tako da se može pokrenuti isključivo korištenjem programa za simulaciju lokalnog servera (u ovom slučaju XAMPP). Ako bi se aplikacija postavila na Internet to bi značilo da svaka osoba koja želi koristiti mogućnosti aplikacije mora imati samo pristup Internetu. Aplikaciju može koristiti više korisnika istovremeno, a radi se sve na jednoj bazi podataka.

LITERATURA

[1] http://www.w3schools.com/php/default.asp 05.04.2016

[2] Lukić, Ivica; Köhler, Mirko;Osnove Internet programiranja, Godina izdanja 2011

[3] http://www.tutorialspoint.com/php/ 05.04.2016

[4] http://php.net/manual/en/ 05.04.2016

[5] https://www.phpmyadmin.net/ 10.03.2016

SAŽETAK

Aplikacija za dostavu paketa putem weba može raditi na bilo kojem operacijskom sustavu, tako da je kompatibilnost ostala očuvana. Aplikacija služi prvenstveno korisnicima koji putem aplikacije naručuju od pružatelja usluge dolazak po paket, te njegovo dostavljanje na krajnje odredište. Aplikacija je jednostavna za korištenje i ne zahtjeva nekakvo napredno znanje korisnika za rad sa računalom. Za upravljanje bazom podataka koristi se phpMyAdmin i ugrađen sustav u aplikaciji kojem imaju pristup vozači i administratori.

Ključne riječi: phpMyAdmin, PHP, MySQL, HTML, sjednica, obrazac, prijava, odjava, administracija, baza podataka, XAMPP, lokalni server.

ABSTRACT

Web application for delivering packages is capable of working on any operating system, which means that compatibilty is preserved. First of all, application is used for delivering packages from one point to another . Service provider is in charge for picking up the packet from user and delivering it to the final destination. Application is very simple to use, and it doesn't require any special skills in using computer. Administrators are using phpMyAdmin for database management. Drivers and administrators are using their own interface, that is integrated in application.

Keywords: phpMyAdmin, PHP, MySQL, HTML, session, form, login, logout, administration, database, XAMPP, localhost.

ŽIVOTOPIS

Luka Roguljić rođen je 20. prosinca 1994. godine u Osijeku, a adresa prebivališta je N. Š Zrinskog 3 Ernestinovo. Osnovnu školu je završio u Ernestinovu, a srednju školu u Elektrotehničkoj i prometnoj školi u Osijeku sa zvanjem tehničar za računalstvo. Akademske godine 2013/14 upisao je stručni studij Informatike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

PRILOZI