

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**IZRADA WEB APLIKACIJE ZA ORGANIZACIJU
PRIJEVOZA ROBE**

Završni rad

Ivan Ivković

Osijek, 2019.

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. PREGLED KORIŠTENIH TEHNOLOGIJA I ALATA	2
2.1. HTML.....	2
2.2. CSS	2
2.3. PHP.....	3
2.4. MySQL.....	4
2.5. XAMPP	4
3. IZRADA APLIKACIJE.....	5
3.1. Struktura resursa web aplikacije.....	5
3.2. Semantička struktura stranica	5
3.3. Sustav prijave i registracije korisnika.....	7
3.4. Naručivanje prijevoza.....	13
3.5. Dodavanje i uklanjanje vozila	18
3.6. Postojeća rješenja	20
4. ZAKLJUČAK	21
LITERATURA.....	22
SAŽETAK.....	23
ABSTRACT	24
ŽIVOTOPIS	25
PRILOZI.....	26

1. UVOD

Prijevoz ili transport pojam je višestrukog značenja, najvećim dijelom povezan s prometom kao tercijarnom gospodarskom djelatnosti koja oduvijek igra vrlo važnu ulogu u funkcioniranju života ljudi diljem svijeta. Tako je, prema [1], transport u prvom redu definiran kao premještanje osoba, životinja, stvari, podataka i dr. pomoću prijevoznih sredstava. Drugo značenje ga opisuje jednostavno kao prijevozničku djelatnost ili prijevozništvo. U ovom završnom radu naglasak će biti na cestovnom prijevozu robe jer je upravo taj segment prijevoza u cjelini vjerojatno ključan. Razmjena dobara među ljudima postoji otkada je i ljudi. Premda se način razmjene tijekom stoljeća neprestano mijenjao, prijevoz ili prijenos dobara uvijek je postojao i pritom omogućavao nesmetano odvijanje gospodarskih procesa i opskrbljivanje različitih ljudi različitim dobrima, ovisno o potrebama. Sam način prijevoza se kroz vrijeme mijenjao, pa se postupnim razvojem tehnologija od nekada prevladavajućeg prijevoza životinjama došlo do složenih i naprednijih načina kakve poznaje suvremeni svijet: željeznički prijevoz, cestovni prijevoz, prijevoz morima i rijekama, prijevoz zrakom i dr. Kad se uzmu u obzir sve prednosti i nedostaci svakog od njih te postojeća infrastruktura, jasno je da je cestovni prijevoz najznačajniji. Njegova temeljna značajka je tzv. prijevoz „od vrata do vrata“. Za razliku od većine drugih oblika prijevoza robe, koji zbog ograničenosti infrastrukture koju koriste nisu u mogućnosti prevesti robu od samog polazišta do krajnjeg odredišta, cestovni prijevoz to uglavnom može, pa se svaki prijevoz robe, bez obzira na sredstvo, prije ili kasnije u većoj ili manjoj mjeri koristi uslugama cestovnog prijevoza. Također, cestovni je prijevoz robe ujedno i najjeftiniji, barem kada se govori o kratkim ili srednjim relacijama, a i velika konkurencija među prijevoznicima doprinosi nižim cijenama. Kad se uzmu u obzir sve navedene činjenice o cestovnom prijevozu robe, za pretpostaviti je da će prosječni korisnik, kojemu je potreban prijevoz određene robe, u prvom redu tražiti davatelje usluga prijevoza upravo u cestovnom prometu, i to na webu, ujedno s mogućnosti trenutane i jednostavne organizacije prijevoza robe.

1.1. Zadatak završnog rada

U ovom je radu prvo potrebno teorijski opisati tehnologije potrebne za izradu web aplikacije za organizaciju prijevoza robe. Zatim treba izraditi samu aplikaciju koja prijavljenim, a prethodno i registriranim, korisnicima omogućuje izbor nekog od raspoloživih teretnih vozila s upisom pojedinosti prijevoza kojeg treba obaviti, npr. polazište i odredište, vrijeme polaska, vrijeme dolaska itd. Potom novoizrađenu web aplikaciju treba usporediti s postojećim sličnim rješenjima uz naznačene prednosti i nedostatke.

2. PREGLED KORIŠTENIH TEHNOLOGIJA I ALATA

U ovom poglavlju navedene su i ukratko po potpoglavljima predstavljene osnovne značajke tehnologija i alata kojima je realizirana web aplikacija: HTML, CSS, PHP, MySQL i XAMPP.

2.1. HTML

HTML, punim nazivom *Hypertext Markup Language*, opisni je jezik koji se koristi u izradi web aplikacija za označavanje elemenata jezika. Zasniva se na predefiniranim oznakama (engl. *tags*) i pripadajućim atributima kojima se opisuju pojedini elementi HTML dokumenta, odnosno web aplikacije. Na taj se način zapravo određuje raspored sadržaja na web aplikaciji, odnosno kako će se ona prikazati u pregledniku. Osim toga, HTML nema drugih uloga jer on nije programski jezik i samim time ne može obavljati zadatke poput, primjerice, aritmetičkih operacija ili ispitivanja uvjeta [2]. HTML oznaka ima sljedeći oblik: `<ime_oznake atribut="vrijednost_atributa">Sadržaj</ime_oznake>` Ako je pojedina oznaka napisana na ispravan način, tada se u pregledniku pri gledanju web aplikacije vidi samo sadržaj te oznake, bez `<>` i `</>` kojima se samo označavaju početak i završetak oznake. Moguće je ugnježđivanje oznaka, no pritom treba biti oprezan pri njihovu zatvaranju, jer se unutarnja oznaka mora zatvoriti prije zatvaranja vanjske oznake, a ne obrnuto. Svaki HTML dokument obvezno započinje s `<html>` poslije čega se navode druge, potrebne i željene oznake, a na kraju, naravno, ide `</html>`, čime se označava kraj HTML dokumenta. Preporučljivo je još na samom početku HTML dokumenta, prije `<html>`, postaviti element `<!DOCTYPE>` koji predstavlja svojevrsnu deklaraciju tipa dokumenta.

2.2. CSS

Kao što puno ime sugerira, CSS, odnosno *Cascading Style Sheets*, jezik je koji opisuje stil HTML dokumenta, odnosno prezentaciju web aplikacija [3]. Iako se stilovi mogu opisati i u samom HTML dokumentu, preporučljivo je u tu svrhu ipak koristiti CSS, prije svega jer se razdvajanjem strukture podataka i prezentacije podataka povećava čitljivost i preglednost HTML koda, čime se olakšava održavanje i pronalazak potencijalnih pogrešaka [4]. Nadalje, CSS nudi veći izbor stilova i općenito mogućnosti kojih nema u HTML-u, poput prilagodbe prezentacije različitim uređajima u smislu različitih dimenzija zaslona.

Pojedini CSS stil opisuje se pravilom koje se sastoji od selektora i deklaracijskog bloka. Svaki selektor odnosi se na onaj HTML element na kojeg se primjenjuje određeni stil, a unutar deklaracijskog bloka navode se svojstva i pripadajuće vrijednosti. U općem obliku CSS pravilo izgleda ovako: *selektor {svojstvo: vrijednost}*.

Postoji puno vrsta selektora, a najčešće korišteni su: jednostavni selektor, klasni selektor, Id selektor, kontekstni selektor te pseudoklase. Jedna od značajki CSS-a je da se jednom definirani stilovi mogu koristiti na više različitih stranica, no prethodno je potrebno na pravilan način povezati CSS s HTML-om. To je moguće učiniti na tri načina. *Inline style* podrazumijeva dodavanje CSS deklaracije u redak HTML koda pomoću atributa *style*, a utječe na stil samo onog elementa kod kojeg je dodan. *Internal stylesheet* također se koristi unutar HTML dokumenta, na način da se unutar oznake `<head></head>` doda oznaka `<style></style>`. *External stylesheet* znači odvajanje CSS-a u posebnu datoteku s ekstenzijom `.css`, koja se potom povezuje s HTML-om pomoću oznake `<link></link>` unutar zaglavlja HTML dokumenta. Upravo treći način omogućuje stiliziranje više HTML dokumenata, pa je to jedan od razloga zašto je to i najbolji način.

2.3. PHP

PHP: Hypertext Preprocessor skriptni je jezik opće namjene prilagođen prije svega razvoju dinamičnih i interaktivnih web aplikacija [5]. Osim što je besplatan, vrlo je popularan zbog jednostavnosti te neovisnosti o operacijskom sustavu. Također, podržava i čisto proceduralno i objektno orijentirano programiranje te ima mnoge značajke koje imaju i drugi programski jezici. Varijable se, naravno, koriste za pohranjivanje vrijednosti i kod njihovog imenovanja vrijede ista pravila kao i u drugim programskim jezicima, no postoje neke važne razlike. Prvo, kod PHP-a varijable obvezno počinju znakom `$` kojim se razlikuju od „običnog“ teksta. Nadalje, kod PHP-a nije obvezno prvo deklarirati varijablu i potom joj dodijeliti vrijednost, kao i navesti tip podatka kojem ona pripada, jer će temeljem varijabli dodijeljene vrijednosti ona automatski postati tog tipa podatka. Što se tiče rukovanja memorijom, kod PHP-a se može (pomoću naredbe *unset*) i ne mora „ručno“ obrisati bilo koju varijablu jer se ona automatski briše prestankom izvršavanja PHP skripte u kojoj se koristi. Kao što je već spomenuto, kod PHP-a nije potrebno eksplicitno navoditi tipove podataka varijabli, no treba znati tipove podataka koje PHP podržava, a to su: tekstualni tip podataka, cijeli brojevi, realni brojevi, logičke varijable, nizovi i objekti.

Tu su još i petlje, funkcije, objekti, različiti operatori za obavljanje operacija nad varijablama i vrijednostima te mogućnost rada s datotekama. Ipak, najvažnije od svega je činjenica da je PHP *server-side* skriptni jezik, odnosno jezik čije se skripte izvode na poslužitelju, ne na korisnikovom računalu, te je stoga pogodan za izradu web aplikacija koje koriste veliki broj podataka [6]. Podržava rad s mnogim bazama podataka, a u ovom je radu korišten u kombinaciji s MySQL-om. PHP kod obvezno se nalazi između znakova `<?php` i `?>` koji ga omeđuju, a može biti napisan samostalno, u datoteci s ekstenzijom `.php`, ili u dokumentu u kojem se nalazi i HTML kod.

2.4. MySQL

MySQL je *open-source* sustav za upravljanje bazama podataka, s mogućnosti *online* pristupa, za što su potrebni korisničko ime i zaporka. Napisan je u C-u i C++ [7]. Zbog svoje besplatnosti, stabilnosti, brzine, postojanja inačica za sve operacijske sustave te podrške za brojne programske jezike, jedan je od najpopularnijih poslužitelja baza podataka. Vrlo često koristi se u kombinaciji s PHP-om jer su oba besplatni i otvorenog koda što ih čini dostupnima za izradu različitih web aplikacija koje koriste baze podataka. Da bi se pritom mogli čitati podatci iz baze podataka, PHP skripta treba se povezati na MySQL poslužitelj.

2.5. XAMPP

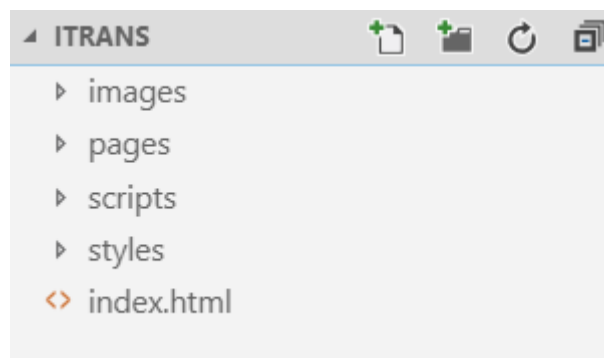
XAMPP je besplatni *open-source* višepatformski web server kojeg je razvila tvrtka Apache Friends. Prema [8], izvorno ime mu je zapravo pokrata od Cross-Platform, Apache, MySQL, PHP i Perl, no pod M se danas ipak smatra MariaDB, koja je 2015. godine zamijenila MySQL. Paket, stoga, u osnovi sadrži Apache HTTP poslužitelj, sustav za upravljanje MariaDB te interpretere za skripte pisane programskim jezicima PHP i Perl, no dostupne su i druge značajke koje uključuju phpMyAdmin, OpenSSL i dr. Službeno je XAMPP razvijan isključivo kao razvojni alat, koji web dizajnerima i programerima omogućava stvaranje lokalnog http poslužitelja na vlastitom računalu u svrhu ispitivanja rada njihovih stranica, no može se koristiti i za postavljanje web stranica na *World Wide Web* [9]. Jednostavna instalacija, kao i jednostavno korisničko sučelje u obliku upravljačke ploče, pridonose njegovoj velikoj popularnosti.

3. IZRADA APLIKACIJE

Postupak izrade web aplikacije opisan je u ovom poglavlju pri čemu je svaki korak izrade, od početka do samog kraja, opisan u posebnom potpoglavlju.

3.1. Struktura resursa web aplikacije

Svi resursi web aplikacija, odnosno stranice, slike, skripte, stilovi i dr. u pravilu se pohranjuju u jedan direktorij i unutar njega grupiraju po vrsti kako bi se povećala preglednost i time olakšalo upravljanje web aplikacijom. Strukturiranje resursa po mapama vrlo je važno i jer se time odražava datotečna struktura web aplikacije na samom poslužitelju [10].



Sl. 3.1. Struktura web aplikacije

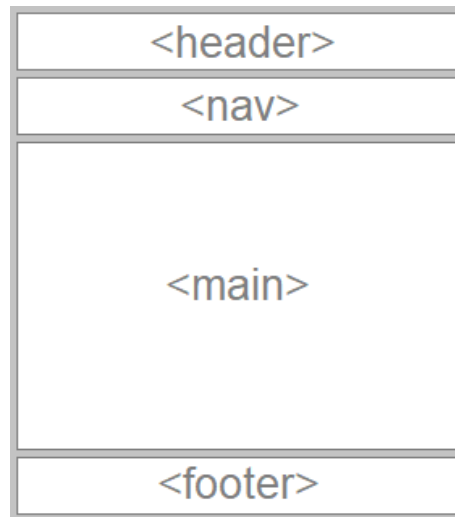
Kao što se vidi na slici 3.1., resursi web aplikacije sadržani su u direktoriju koji nosi ime web aplikacije, Itrans. Datoteka *index.html* predstavlja početnu stranicu web aplikacije. Ostale stranice, točnije HTML dokumenti, sadržani su u poddirektoriju *pages*. U poddirektoriju *images* nalaze se sve slike korištene u aplikaciji, uključujući logotip koji se nalazi u zaglavlju. Poddirektorij *scripts* obuhvaća sve PHP skripte koje se u aplikaciji koriste pri radu s bazama podataka. CSS datoteka s definiranim stiliziranjem sadržaja nalazi se u poddirektoriju *styles*. Kod povezivanja resursa unutar web aplikacije koriste se relativne putanje koje su kraće i samim time preglednije te učinkovitije u odnosu na apsolutne putanje.

3.2. Semantička struktura stranica

Jedna od glavnih pretpostavki kvalitetne web aplikacije je pravilna struktura njenih stranica, odnosno raspored semantičkih elemenata HTML dokumenta. Ne postoji točno određeni „zakon“ koji propisuje kako točno pojedina web stranica treba izgledati, no u osnovi je vrlo važna

jednostavnost strukture koja prosječnim korisnicima omogućava lako snalaženje, sa što manje nepotrebnih informacija i općenito značajki.

Na slici 3.2. prikazan je pojednostavljeni raspored HTML semantičkih elemenata na stranicama web aplikacije.



Sl. 3.2. *Uporaba semantičkih elemenata*

Elementom *<header>* opisano je zaglavlje stranice u kojem se nalaze naziv web aplikacije i logotip. U navigacijskoj traci *<nav>* nalaze se poveznice prema svim glavnim stranicama web aplikacije, a podnožje stranice predstavljeno je elementom *<footer>*, gdje se nalaze oznaka autorskog prava i godina. Do sada navedeni dijelovi pojedine stranice jednaki su za sve stranice web aplikacije. S druge strane, glavni dio pojedine stranice, predstavljen elementom *<main>*, unutar kojeg je ugniježđen jedan ili više elemenata *<article>*, svojstven je za svaku stranicu posebno. Na slici 3.3. dan je prikaz konačnog izgleda pojedine stranice web aplikacije.

Vozila

Ovdje su ispisana naša vozila koja možete naručiti. Podsjećamo da je za naručivanje vozila potrebno biti registriran i prijavljen.

Reg. oznaka	Tip	Nadogradnja	Duljina [m]	Širina [m]	Visina [m]	Nosivost [t]	
OS1786DG	poluprikolica	otvorena	13.60	2.45	2.80	25	Naruči
OS1947GV	poluprikolica	cerada	13.60	2.45	2.80	24	Naruči
OS1988TV	solo	furgon	6.20	2.45	2.80	10	Naruči
OS2017PS	solo	hladnjača	6.20	2.45	2.80	8	Naruči

© 2019 ivan.inkovic@jetfos.hr

Sl. 3.3. *Izgled stranice*

3.3. Sustav prijave i registracije korisnika

Kao što je rečeno u 1.1., web aplikacija treba prijavljenim korisnicima omogućiti narudžbu vozila. Prethodno je, naravno, potrebno se registrirati pritom navodeći podatke zatražene u registracijskom obrascu, a koji su potrebni kod izdavanja naloga za utovar i kod uspostave računa. U tu je svrhu bilo potrebno prvo stvoriti bazu podataka koja je nazvana imenom web aplikacije, Itrans. Potom je, kako je prikazano na slici 3.4., stvorena tablica *users*, u koju se spremaju podatci o korisnicima.

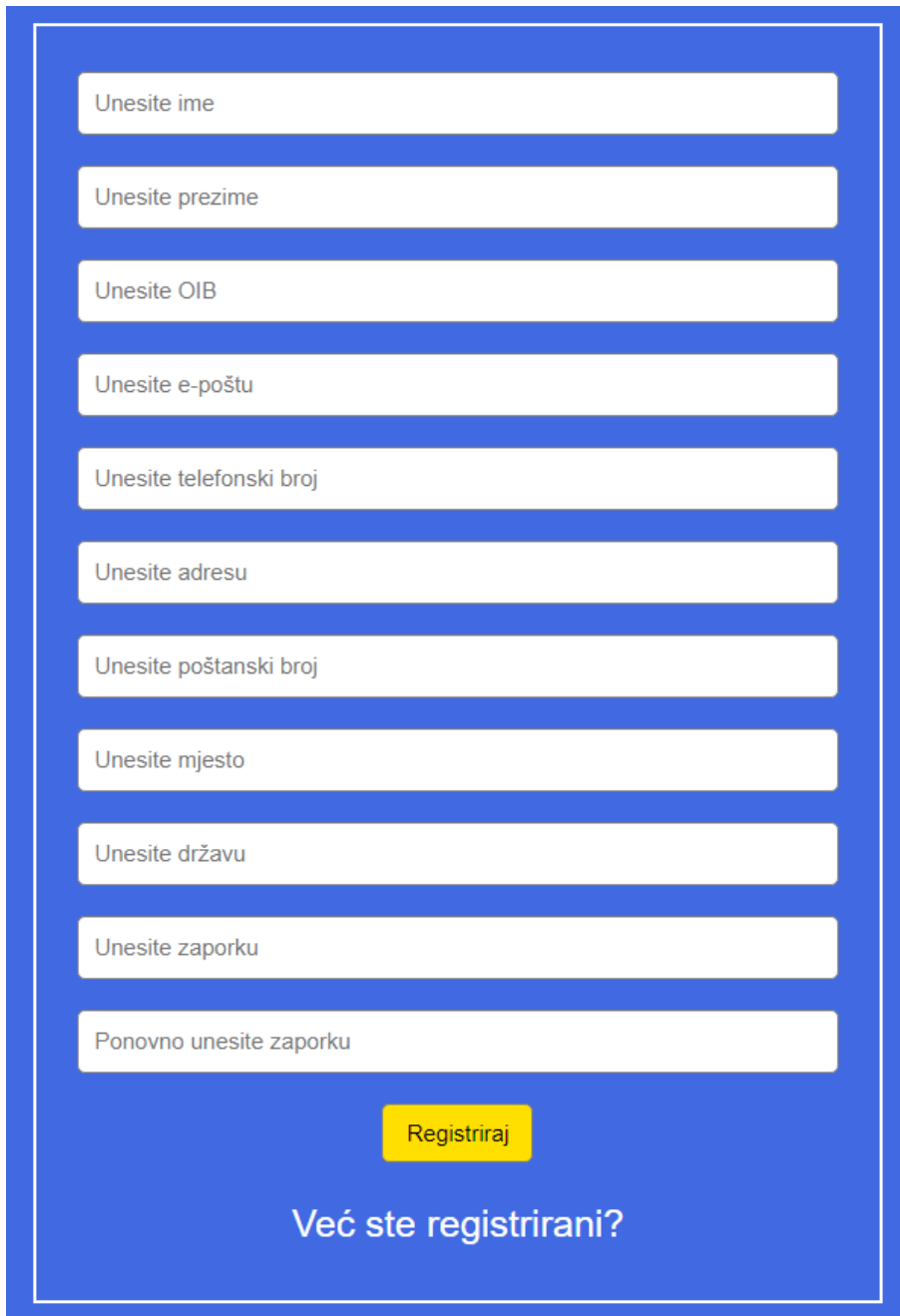
```
CREATE TABLE users(  
  id_user INT AUTO_INCREMENT,  
  email VARCHAR(50) NOT NULL,  
  password VARCHAR(50) NOT NULL,  
  user_name VARCHAR(50) NOT NULL,  
  user_surname VARCHAR(50) NOT NULL,  
  user_personal_number CHAR(11) NOT NULL,  
  user_telephone VARCHAR(50) NOT NULL,  
  user_address VARCHAR(50) NOT NULL,  
  user_postal_code INT NOT NULL,  
  user_town VARCHAR(50) NOT NULL,  
  user_country VARCHAR(50) NOT NULL,  
  user_type VARCHAR(8) NOT NULL DEFAULT 'customer',  
  CONSTRAINT users_pk PRIMARY KEY(id_user)  
);
```

Sl. 3.4. Stvaranje tablice *users*

Cjelobrojni atribut *id_user* predstavlja svojevrsni identifikator koji se automatski dodjeljuje svakom novoregistriranom korisniku. Atributi *e-mail* i *password* predstavljaju adresu e-pošte i zaporku korisničkog računa, a koji će biti potrebni kod svake prijave korisnika u sustav. Atributi *user_name*, *user_surname*, *user_personal_number*, *user_address*, *user_postal_code*, *user_town* i *user_country* pripadaju skupini korisnikovih osobnih podataka koji se koriste u dokumentiranju usluga prijevoza. Atribut *user_type* služi za razlikovanje registriranih korisnika u pogledu razine prava koje imaju pri korištenju web-aplikacije. Zamisao je postojanje dviju vrsta registriranih korisnika: *customer* i *admin*. *Customer* je registrirani korisnik koji nakon prijave ima mogućnost naručiti neko od ponuđenih vozila za obavljanje neke vrste prijevoza unosenjem podataka o prijevozu. *Admin* predstavlja administratora aplikacije, a njegove ovlasti su dodavanje novih vozila, brisanje postojećih vozila te pristup podacima o svim registriranim korisnicima, zaprimljenim narudžbama itd.

Na slici 3.4. također je vidljivo da je za podrazumijevanu vrijednost atributa *user_type* postavljena *customer*, što znači da se svakom korisniku pri registraciji automatski dodjeljuje ta razina prava, a administrator ima mogućnost promijeniti ju u bilo kojem trenutku.

Ukratko, *id_user* i *user_type* postavljaju se automatski, a ostale podatke unosi korisnik preko registracijskog obrasca, čiji je izgled prikazan slikom 3.5.



The image shows a registration form with the following fields and elements:

- Unesite ime
- Unesite prezime
- Unesite OIB
- Unesite e-poštu
- Unesite telefonski broj
- Unesite adresu
- Unesite poštanski broj
- Unesite mjesto
- Unesite državu
- Unesite zaporku
- Ponovno unesite zaporku
- Registriraj (yellow button)
- Već ste registrirani? (link)

Sl. 3.5. Registracijski obrazac

Podatci uneseni u obrascu klikom na gumb Registriraj šalju se pomoću metode *POST* na obradu koja prethodi samom umetanju unesenih podataka u tablicu *users* u bazi podataka. Primitak podataka poslanih iz obrasca metodom *POST* prikazan je slikom 3.6.

```
$email      = mysqli_real_escape_string($db, $_POST['email']);
$password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
$password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
$user_name  = mysqli_real_escape_string($db, $_POST['user_name']);
$user_surname = mysqli_real_escape_string($db, $_POST['user_surname']);
$user_personal_number = mysqli_real_escape_string($db, $_POST['user_personal_number']);
$user_telephone = mysqli_real_escape_string($db, $_POST['user_telephone']);
$user_address = mysqli_real_escape_string($db, $_POST['user_address']);
$user_postal_code = mysqli_real_escape_string($db, $_POST['user_postal_code']);
$user_town   = mysqli_real_escape_string($db, $_POST['user_town']);
$user_country = mysqli_real_escape_string($db, $_POST['user_country']);
```

Sl. 3.6. Primitak podataka poslanih iz registracijskog obrasca

Funkcija *mysqli_real_escape_string()* se, kako navodi [11], koristi za izbjegavanje posebnih znakova u stringu koji se mogu koristiti za SQL naredbe, a predaju joj se dva parametra: spoj na MySQL-ov poslužitelj, koji je ovdje spremljen u varijablu *\$db*, te primljena vrijednost.

Već je spomenuto da su korisnikovi podatci vrlo važni kod dokumentiranja usluge prijevoza. Stoga je važno osigurati popunjenost svih traženih polja u registracijskom obrascu i upozoriti korisnika u slučaju da nije to učinio. Na slici 3.7. dan je primjer provjere popunjenosti polja u obrascu za adresu korisnikove e-pošte.

```
if (empty($email))
{
    array_push($errors, "Potrebno je unijeti adresu e-pošte.");
}
```

Sl. 3.7. Provjera unesenosti adrese e-pošte

Na isti se način provjerava unesenost i svih ostalih traženih podataka. S obzirom da se adresa e-pošte koristi kod prijave korisnika u sustav, onemogućena je registracija više od jednog korisnika uporabom određene adrese elektroničke pošte, a navedeno je prikazano slikom 3.8.

```

$user_check = "SELECT * FROM users WHERE email='$email' LIMIT 1";
$result = mysqli_query($db, $user_check);
$user = mysqli_fetch_assoc($result);

if ($user)
{
    if ($user['email'] === $email)
    {
        array_push($errors, "Već je registriran korisnik s navedenim e-mailom.");
    }
}

```

Sl. 3.8. *Provjera zauzetosti adrese e-pošte*

Funkcija *mysqli_query()* izvršava upit *user_check* prema bazi podataka, a rezultat upita dohvaća se pomoću funkcije *mysqli_fetch_assoc()*. Ako u bazi podataka već postoji korisnik koji koristi adresu e-pošte koja je unesena u registracijskom obrascu, ispisat će se obavijest o zauzetosti unesene adrese.

U svakom registracijskom obrascu, pa tako i promatranom, traži se dvostruki unos željene zaporke. Dakle, na oba mjesta potrebno je unijeti potpuno istu zaporku jer će se u suprotnom pojaviti obavijest o nepodudarnosti unesenih zaporki. Provjera podudaranja unosa zaporke obavljena je jednostavnim kodom koji je prikazan slikom 3.9.

```

if ($password_1 != $password_2)
{
    array_push($errors, "Unesene zaporkе se ne podudaraju.");
}

```

Sl. 3.9. *Provjera podudarnosti unosa zaporke*

Nakon svih provjera, ali uz uvjet da nema pogrešaka, uneseni podatci napokon se zapisuju u bazu podataka, točnije u tablicu *users*, kako je prikazano slikom 3.10.

```

if (count($errors) == 0)
{
    $password = md5($password_1);
    $insert_query = "INSERT INTO users (email, password, user_name,
user_surname, user_personal_number, user_telephone, user_address,
user_postal_code, user_town, user_country) VALUES('$email', '$password',
'$user_name', '$user_surname', '$user_personal_number', '$user_telephone',
'$user_address', '$user_postal_code', '$user_town', '$user_country')";
    mysqli_query($db, $insert_query);
    header('location: login.php');
}

```

Sl. 3.10. Zapisivanje podataka u bazu podataka

Prije samog spremanja u bazu unesena zaporka se enkriptira, što je mjera opreza u slučaju neovlaštenog upada u bazu podataka. Spremanjem podataka završava postupak registracije, a korisnik se usmjerava na obrazac za prijavu, čiji je izgled prikazan slikom 3.11.

Sl. 3.11. Obrazac za prijavu

Da bi se prijavio, korisnik treba unijeti adresu e-pošte na koju se registrirao, kao i ispravnu zaporku. Podatci poslani iz obrasca za prijavu djelomično se obrađuju isto kao i kod obrade podataka poslanih iz registracijskog obrasca, odnosno na identičan način kako je bilo prikazano slikom 3.6. I ovdje je bilo potrebno osigurati potpun i ispravan unos traženih podataka, što je provjereno isto kao i kod registracije, a što je prikazano na slici 3.7.

Obavijest o pogrešci pojavljuje se kada nisu uneseni svi podatci te kada e-pošta nije unesen u svom potpunom obliku: *lokalni_dio@internetska_domena*.

Nakon potrebnih provjera i utvrđivanja ispravnosti, slijedi dohvaćanje korisničkih podataka iz tablice u bazi podataka, prikazano slikom 3.12.

```
if (count($errors) == 0)
{
    $password = md5($password);
    $query = "SELECT * FROM users WHERE email='$email' AND password ='$password' LIMIT 1";
    $results = mysqli_query($db, $query);

    if (mysqli_num_rows($results) == 1)
    {
        $logged_in_user = mysqli_fetch_assoc($results);
        $_SESSION['user'] = $logged_in_user;
        header('location: profile.php');
    }
    else
    {
        array_push($errors, "Uneseni su krivi podatci.");
    }
}
```

Sl. 3.12. Dohvaćanje podataka iz baze podataka

Unesena zaporka se enkriptira, a pomoću *mysqli_query()* izvršava se upit prema bazi podataka kojim se dohvaća korisnik. Ako korisnik s unesenim podacima nije pronađen, pojavljuje se obavijest da su uneseni krivi podatci. S druge strane, ako je sve u redu, odnosno, pronađen je točno jedan korisnik čiji podatci odgovaraju unesenim podacima, on se sprema u varijablu sesije *\$_SESSION['user']* i usmjerava na svoj korisnički profil. Od tog trenutka *customer* ima mogućnost naručivati vozila.

3.4. Naručivanje prijevoza

Izvedba mogućnosti naručivanja prijevoza zamišljena je na sljedeći način. Klikom na poveznicu Vozila korisniku se otvara istoimena stranica na kojoj se nalazi popis vozila ponuđenih za obavljanje prijevoza različitih vrsta tereta. Zapravo se radi o tablici u kojoj svaki red predstavlja jedno vozilo, a stupci označavaju neke od karakteristika vozila, poput registracijske oznake, tipa, nadogradnje te duljine, širine, visine i nosivosti. Uz desni kraj svakog od redaka tablice nalazi se gumb Naruči, koji, kad se klikne, otvara obrazac za narudžbu, predviđen za unošenje podataka.

Za početak, bilo je potrebno u sklopu baze podataka Itrans stvoriti novu tablicu, namijenjenu spremanju teretnih vozila, što je prikazano slikom 3.13.

```
CREATE TABLE vehicles(  
  id_vehicle VARCHAR(10),  
  type VARCHAR(30) NOT NULL,  
  set_up VARCHAR(30) NOT NULL,  
  length DECIMAL(4,2) NOT NULL,  
  width DECIMAL(4,2) NOT NULL,  
  height DECIMAL(4,2) NOT NULL,  
  capacity INT NOT NULL,  
  CONSTRAINT vehicles_pk PRIMARY KEY(id_vehicle)  
);
```

Sl. 3.13. Stvaranje tablice *vehicles*

Atribut *id_vehicle* primarni je ključ tablice *vehicles*, a predstavljen je registracijskom oznakom određenog teretnog vozila. Ostali atributi odnose se na već spomenuti tip vozila, nadogradnju, duljinu, širinu, visinu i nosivost.

Nakon stvaranja tablice u bazi podataka, potrebno je stvoriti i svojevrsno korisničko sučelje na stranici web aplikacije, odnosno tablični prikaz vozila s pripadajućim karakteristikama, a to je realizirano programskim isječkom sa slike 3.14.

```

<table width="300px" border="1px" cellpadding="2" cellspacing="2">
    <tr><td><b>Reg. oznaka</b></td>
    <td><b>Tip</b></td>
    <td><b>Nadogradnja</b></td>
    <td><b>Duljina [m]</b></td>
    <td><b>Širina [m]</b></td>
    <td><b>Visina [m]</b></td>
    <td><b>Nosivost [t]</b></td></tr>

<?php
$sql = "SELECT id_vehicle, type, set_up, length, width, height, capacity FROM vehicles";
$result = mysqli_query($db, $sql);

if (mysqli_num_rows($result) > 0)
{
    while($row = mysqli_fetch_assoc($result))
    {
        echo '<tr><td>'.$row["id_vehicle"].'</td>';
        echo '<td>'.$row["type"].'</td>';
        echo '<td>'.$row["set_up"].'</td>';
        echo '<td>'.$row["length"].'</td>';
        echo '<td>'.$row["width"].'</td>';
        echo '<td>'.$row["height"].'</td>';
        echo '<td>'.$row["capacity"].'</td>';
        echo '<td><a href="order.php?id_vehicle='.$row["id_vehicle"].'">Naruči</a></td>';
        '</tr>';
    }
    echo '</table>';
}
$db->close();
?>

```

Sl. 3.14. Stvaranje tabličnog prikaza i dohvaćanje podataka iz baze podataka

Cjelokupni tablični prikaz nalazi se unutar HTML oznake `<table></table>` unutar koje su ćelije predstavljene s oznakom `<td></td>`. Podatci iz baze podataka, odnosno iz tablice *vehicles*, pribavljaju se pomoću funkcije *mysqli_query()* koja izvršava upit *sql* prema bazi podataka i potom se rezultat sprema u varijablu *\$result*. U uvjetu *if* koristi se funkcija *mysqli_num_rows()* koja inače vraća broj redova dobivenih upitom, pa se njome zapravo provjerava postoje li uopće u bazi podataka podatci za prikaz. Postavljen je uvjet da taj broj treba biti veći od 0, što znači da u bazi podataka postoji barem jedno vozilo. Kada je utvrđeno postojanje jednog ili više vozila, potrebno ih je upisati u tablični prikaz, red po red. Zato se koristi petlja *while* i kao njen uvjet funkcija *mysqli_fetch_assoc()* koja dohvaća rezultat pojedinog reda kao asocijativan niz te se sadržaj reda upisuje u tablični prikaz.

U svakom redu, odnosno uz svako vozilo, postoji gumb Naruči, koji vodi prema obrascu za narudžbu upravo tog vozila u čijem se redu gumb nalazi. Vozila ponuđena u tabličnom prikazu nemaju posebne obrasce za naručivanje za svako vozilo posebno, nego se koristi mehanizam višenamjenskih stranica. On omogućuje da se, u slučajevima postojanja većeg broja stranica sličnog sadržaja, koristi zapravo samo jedna stranica, a dijelovi koji su različiti mijenjaju se ovisno o kontrolnim uvjetima. Kontrolni uvjeti prosljeđuju se preko naredbenog retka, odnosno dijela URL-a iza znaka ?, a mogu biti izvedeni pomoću kontrolne riječi ili kontrolne varijable. Ovdje se koristi kontrolna varijabla `id_vehicle` koja preko naredbenog retka predaje skripti `order.php` registracijsku oznaku odabranog vozila, omogućavajući tako naručivanje istoga. U sklopu skripte nalazi se obrazac za naručivanje, u kojeg korisnik unosi tražene podatke o samoj narudžbi. Ti se podatci trebaju negdje spremiti, što znači da je potrebno stvoriti novu tablicu u bazi podataka, a taj postupak prikazan je slikom 3.15.

```
CREATE TABLE orders(  
  id_order INT AUTO_INCREMENT,  
  departure_date DATETIME NOT NULL,  
  departure_place VARCHAR(30) NOT NULL,  
  arrival_date DATETIME NOT NULL,  
  arrival_place VARCHAR(30) NOT NULL,  
  order_info VARCHAR(100) NOT NULL,  
  id_user INT,  
  id_vehicle VARCHAR(10),  
  CONSTRAINT orders_pk PRIMARY KEY(id_order),  
  CONSTRAINT orders_fk_vehicles FOREIGN KEY(id_vehicle) REFERENCES vehicles(id_vehicle),  
  CONSTRAINT orders_fk_users FOREIGN KEY(id_user) REFERENCES users(id_user)  
);
```

Sl. 3.15. Stvaranje tablice `orders`

Kako je vidljivo, tablica ima primarni ključ `id_order` kojim se svakoj narudžbi kao oznaka automatski dodjeljuje jedinstveni broj. Također, ova tablica razlikuje se od prethodne dvije po postojanju stranih ključeva koji su ovdje nužni kako bi se narudžba povezala s korisnikom koji ju obavlja kao i vozilom koje se naručuje.

Neposredno prije slike 3.15. objašnjeno je da se registracijska oznaka vozila šalje u obrazac za narudžbu preko kontrolne varijable. U obrascu, točnije u skripti `order.php`, potrebno je na neki način primiti i prihvatiti poslanu registracijsku oznaku. Kako je to učinjeno, prikazano je slikom 3.16. gdje se nalazi izvedba obrasca za narudžbu u skripti `order.php`.

```

<form method="post" action="order.php">
  <div><?php echo $_GET["id_vehicle"];?></div>
  <input type="date" class="input" name="departure_date" placeholder="Unesite nadnevak polaska">
  <input type="text" class="input" name="departure_place" placeholder="Unesite mjesto polaska">
  <input type="date" class="input" name="arrival_date" placeholder="Unesite nadnevak dolaska">
  <input type="text" class="input" name="arrival_place" placeholder="Unesite mjesto dolaska">
  <input type="text" class="input" name="order_info" placeholder="Unesite dodatne informacije o
prijevozu">
  <input type="hidden" name="id_vehicle" value="<?php echo $_GET["id_vehicle"]; ?>">
  <button type="submit" class="button" name="order_btn">Naruči</button>
</form>

```

Sl. 3.16. Izvedba obrasca za narudžbu

Za dohvaćanje registracijske oznake u skripti koristi se metoda *GET*, a varijabla u kojoj je spremljena registracijska oznaka je globalna varijabla *\$_GET* koja se ovdje poziva dva puta. Prvi poziv služi isključivo za ispis registracijske oznake na vrhu obrasca za narudžbu, kako bi korisnik i dalje vidio registracijsku oznaku vozila kojeg naručuje. Drugi poziv je skriven i njegov se učinak ne vidi u samom obrascu, no koristan je jer će kasnije omogućiti da se registracijska oznaka dohvati pomoću *POST*-a i spremi u bazu podataka zajedno s drugim podatcima unesenim u obrascu, što je prikazano kodom sa slike 3.17.

```

$id_vehicle = $_POST['id_vehicle'] ?? '';
$departure_date = $_POST['departure_date'] ?? '';
$departure_place = $_POST['departure_place'] ?? '';
$arrival_date = $_POST['arrival_date'] ?? '';
$arrival_place = $_POST['arrival_place'] ?? '';
$order_info = $_POST['order_info'] ?? '';
$id_user = $_SESSION['user']['id_user'];
$sql_form = "INSERT INTO orders(id_vehicle, departure_date, departure_place,
arrival_date, arrival_place, order_info, id_user) VALUES('$id_vehicle',
'$departure_date', '$departure_place', '$arrival_date', '$arrival_place',
'$order_info', '$id_user')";
if (mysqli_query($db, $sql_form))
{
  echo "Uspješno ste naručili prijevoz.";
}
else
{
  echo "Dogodila se pogreška: " . $sql_form . " . mysqli_error($db);
}

```

Sl. 3.17. Dohvaćanje i spremanje podataka narudžbe u tablicu orders

Podatci su iz obrasca poslani metodom *POST* te sadržani u globalnoj varijabli *\$_POST*. S obzirom da nije moguće istovremeno koristiti *POST* i *GET*, a *POST* je i sigurniji u odnosu na *GET*, prethodno spomenuti skriveni poziv *GET*-a u obrascu sada dobiva puni smisao i moguće je pristupiti unosu narudžbe u bazu podataka. Potrebno je još samo dohvatiti oznaku korisnika koji obavlja narudžbu, a to je učinjeno korištenjem globalne varijable *\$_SESSION* koja sadrži varijable sesije. Podatci narudžbe sada se mogu spremirati u tablicu *orders* te se, ako je sve u redu, korisniku ispisuje obavijest o uspješnoj narudžbi. S druge strane, ako narudžba nije bila uspješna, pojavit će se poruka o pogrješci.

Stranica s tabličnim prikazom vozila ponuđenih za obavljanje prijevoza dostupna je svim posjetiteljima web aplikacije, uključujući i neprijavljene posjetitelje. U tom slučaju teoretski bilo tko može otvoriti obrazac za narudžbu, popuniti ga i poslati, no bez bilo kakve oznake koja jedinstveno obilježava korisnika i omogućuje pristup njegovim podacima potrebnima za organizaciju prijevoza, prijevoz nije moguće organizirati, pa i onda kad su svi drugi podaci u obrascu uneseni na odgovarajući način. Kako bi se to spriječilo, potrebno je stvoriti mehanizam koji onemogućuje pristup obrascu za anonimne i neprijavljene posjetitelje. U tu je svrhu stvorena funkcija koja provjerava je li posjetitelj, odnosno korisnik, uopće prijavljen, a njena implementacija dana je slikom 3.18.

```
function isLoggedIn()
{
    if (isset($_SESSION['user']))
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Sl. 3.18. Funkcija *isLoggedIn()*

Jednostavnom provjerom postojanja sesije provjerava se je li korisnik prijavljen te funkcija, ovisno o ishodu provjere, vraća vrijednost *true* (ako je korisnik prijavljen) ili *false* (ako korisnik nije prijavljen). Primjer korištenja funkcije dan je slikom 3.19.

```
if (!isLoggedIn())
{
    header('location: login.php');
}
```

Sl. 3.19. *Provjera prijavljenosti korisnika pozivom funkcije isLoggedIn()*

Prikazani isječak postavljen je na vrh skripte order.php, a izvršava se kada funkcija vraća rezultat *false*, odnosno kada korisnik nije prijavljen. U tom se slučaju on preusmjerava na obrazac za prijavu.

3.5. Dodavanje i uklanjanje vozila

Prethodno poglavlje odnosilo se na značajku web aplikacije dostupnu običnom korisniku, *customeru*, a to je naručivanje vozila za obavljanje prijevoza. Ovo poglavlje bavi se dijelom web aplikacije namijenjenom isključivo ovlaštenoj osobi, odnosno *adminu*, čija je zadaća po potrebi dodati ili ukloniti vozila iz baze podataka. To mu treba biti omogućeno na samom sučelju web aplikacije, kako ne bi morao ulaziti u samu bazu podataka i ondje djelovati. Za početak je potrebno u kod prikazan slikom 3.14. dodati sadržaj kako je prikazano slikom 3.20.

```
if (mysqli_num_rows($result) > 0)
{
    while($row = mysqli_fetch_assoc($result))
    {
        //...
        echo '<td><a href="order.php?id_vehicle='.$row["id_vehicle"].'">Naruči</a></td>';
        if(isAdmin())
        {
            echo '<td><a href="delete.php?id_vehicle='.$row["id_vehicle"].'">Obriši</a></td>';
        }
        '</tr>';
    }
}
echo '</table>';
if(isAdmin())
{
    echo '<td><a href="add.php">Dodaj vozilo</a></td>';
}
```

Sl. 3.20. *Gumbi za dodavanje i uklanjanje vozila*

Gumb za brisanje, odnosno za uklanjanje određenog vozila u bazi podataka, nalazi se u onom redu tabličnog prikaza u kojem su prikazani podatci tog vozila. Kraće rečeno, nalazi se desno od gumba za naručivanje i funkcionira na sličan način: pomoću kontrolne varijable šalje se registracijska oznaka u skriptu zaduženu za obavljanje zadatka. Ispod tablice nalazi se jedan gumb za dodavanje vozila. Primjetno je da se oba novo dodana gumba nalaze unutar *if*-ova koji kao uvjet koriste funkciju *isAdmin()*, čija je definicija dana slikom 3.21.

```
function isAdmin()  
{  
    if (isset($_SESSION['user']) && $_SESSION['user']['user_type'] == 'admin' )  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

Sl. 3.21. Funkcija *isAdmin()*

Jednostavno, u funkciji se provjerava prijavljenost korisnika i vrsta računa. Ako je korisnik prijavljen i ako je *admin*, funkcija vraća *true*, a u suprotnom vraća *false*.

Dakle, ako je korisnik *admin* prikazat će mu se gumbi za uklanjanje vozila i gumb za dodavanje vozila, a neprijavljen korisnik i *customer* ih uopće neće vidjeti ni u kojem slučaju.

Dodavanje novog vozila funkcionira na sličan način kao dodavanje novog korisnika, odnosno registracija. Ispunjava se obrazac za dodavanje vozila i pritom se provjerava popunjenost svakog polja kao i ispravnost unosa, da bi se na kraju svi uneseni podatci zapisali u tablicu *vehicles*.

Skripta za uklanjanje vozila prikazana je slikom 3.22.

```
<?php  
    include "connect.php";  
    $id_vehicle = $_GET['id_vehicle'];  
    $sql = "DELETE from vehicles WHERE id_vehicle='$id_vehicle'";  
    mysqli_query($db, $sql);  
?>
```

Sl. 3.22. Uklanjanje vozila iz baze podataka

3.6. Postojeća rješenja

Timocom je vodeći europski pružatelj IT-usluga na području logistike. Njegov rad zasniva se na poslovanju s različitim prijevozničkim tvrtkama prije svega kroz ponude tereta, utovarnog prostora kao i skladišnog poslovanja, ali i GPS praćenje vozila. Zbog multifunkcionalnosti često je korišten kod različitih pružatelja usluga prijevoza roba, kojima je zapravo namijenjen i prema kojima je najvećim dijelom orijentiran. S druge strane, široki spektar usluga koje ova aplikacija pruža nerijetko otežava snalaženje korisnicima, osobito onim novijim, a tome doprinosi i nužno kombinirano korištenje desktop i web aplikacije, koje su, ipak, vrlo privlačnog i suvremenog izgleda. Što se tiče organizacije prijevoza tereta, korisnicima, odnosno nalogoprimcima, na raspolaganju je tablica u koju nalogodavci prethodno trebaju, što je detaljnije moguće, unijeti potrebne podatke poput polazišta, odredišta, vrste i količine tereta itd. Klikom na određeni redak tablice, potencijalnom nalogoprimcu otvaraju se podatci o nalogodavcu, a dodatni parametri prijevoza utvrđuju se telefonskim putem ili nekim drugim načinom komunikacije. S obzirom na veliku ponudu tereta kojeg je potrebno prevesti s različitih polazišta do različitih odredišta, korisnici na raspolaganju imaju i filter koji se nalazi iznad tablice, a omogućuje prikaz samo onih ponuda koje se odnose na polazište i odredište određeno filterom. Nalogodavci i nalogoprimci su registrirani i prijavljeni korisnici, a članarina, odnosno pristup aplikaciji, naplaćuje se svakog mjeseca.

LKW Walter tvrtka je čija je osnovna djelatnost organizacija transporta utovara u cestovnom i kombiniranom (cestovno-željezničkom i međuobalnom) prijevozu. Organizacija prijevoza obavlja se pomoću višenamjenske web aplikacije jednostavnog izgleda, što olakšava korištenje aplikacije za sve vrste korisnika, pa tako primjerice već početna stranica vrlo pregledno diferencira naručitelje prijevoza i transportne partnere. Ova aplikacija također je u prvom redu orijentirana prema prijevoznicima kojima omogućuje pretragu utovara tereta, prijavu praznog kamiona, odnosno raspoloživog teretnog prostora, karticu za gorivo itd. Pretraga tereta funkcionira na način da korisnik sam unosi željeno polazište i željeno odredište da bi mu se potom otvorio popis ponuda određenih unesenim parametrima. Klik na jednu od izlistanih ponuda otvara prozor u kojem se nalaze opširniji podatci o ponuđenom teretu, dionici koju je potrebno prijeći od polazišta do odredišta kao i kontakt podatci nalogodavca. Registracija i uporaba aplikacije u potpunosti je besplatna.

4. ZAKLJUČAK

Cilj završnog rada bio je izraditi web aplikaciju za organizaciju prijevoza robe u cestovnom prometu kao najčešćem načinu transporta roba između polazišta i odredišta. Tijekom izrade korištene su najosnovnije tehnologije koje su trebale omogućiti uspješnu realizaciju sustava registracije i prijave korisnika, uz razlikovanje određenih skupina korisnika. Potom je, ovisno o korisničkoj skupini, bilo potrebno omogućiti naručivanje prijevoza robe točno određenim vozilom za jednu skupinu korisnika te upravljanje podacima o svim vozilima i korisnicima za drugu skupinu korisnika, uključujući unos novih vozila u bazu podataka preko sučelja aplikacije. Sve navedeno uspješno je realizirano spomenutim tehnologijama i alatima. Aplikacija završnog rada je, u usporedbi s postojećim aplikacijama namijenjenim pitanjima logistike, izgledom i količinom funkcionalnosti vrlo jednostavna, što je i bio cilj. Naime, postojeća rješenja uglavnom su orijentirana prema pružateljima usluga prijevoza i samim time teže dostupna i manje prihvatljiva prosječnim korisnicima, kojima nisu potrebne druge funkcionalnosti osim jednostavne narudžbe određenog vozila za obavljanje određenog prijevoza.

LITERATURA

- [1] Hrvatska enciklopedija, *Transport*, <http://www.enciklopedija.hr/Natuknica.aspx?ID=62056>, pristupljeno 17. svibnja 2019.
- [2] I. Lukić, M. Köhler, *Osnove Internet programiranja*, Elektrotehnički fakultet Osijek, Osijek, 2011.
- [3] W3Schools, *CSS Tutorial*, <https://www.w3schools.com/css/default.asp>, pristupljeno 17. svibnja 2019.
- [4] W3C, *HTML & CSS*, <https://www.w3.org/standards/webdesign/htmlcss>, pristupljeno 17. svibnja 2019.
- [5] PHP, *What is PHP?*, <https://www.php.net/manual/en/intro-what-is.php>, pristupljeno 20. svibnja 2019.
- [6] GeeksforGeeks, *PHP Tutorials*, <https://www.geeksforgeeks.org/php/>, pristupljeno 20. svibnja 2019.
- [7] DB-Engines, *MySQL System Properties*, <https://db-engines.com/en/system/MySQL>, pristupljeno 20. svibnja 2019.
- [8] TechnoSIP Inc., *TechnoSIP Review – XAMPP*, <https://www.technosip.com/toolsreviews/technosip-review-xampp/>, pristupljeno 21. svibnja 2019.
- [9] Global Health Data Management, *XAMPP*, <https://globalhealthdatamanagement.tghn.org/community/blogs/post/5122/2013/07/xampp/>, pristupljeno 21. svibnja 2019.
- [10] Mozilla, *Dealing with files*, https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files, pristupljeno 22. svibnja 2019.
- [11] W3Schools, *PHP mysqli_real_escape_string() Function*, https://www.w3schools.com/php/func_mysqli_real_escape_string.asp, pristupljeno 15. lipnja 2019.

SAŽETAK

Tijekom razvoja web aplikacije korišteni su HTML, CSS, PHP, MySQL i XAMPP, koji su prvo teoretski opisani i objašnjeni, a potom se, primjenjujući stečena znanja, pristupilo izradi aplikacije. Prvo je stvorena datotečna struktura web aplikacije i semantička struktura sastavnih stranica web aplikacije. Zatim je izrađen sustav registracije i prijave korisnika, uz diferenciranje vrsta korisnika i sprječavanje neovlaštenog pristupa. Slijedio je rad s bazom podataka, odnosno s dohvaćanjem i zapisivanjem podataka o vozilima i narudžbama. Na kraju je novoizrađena web aplikacija uspoređena s postojećim aplikacijama slične namjene.

Ključne riječi: baza podataka, korisnik, narudžba, struktura, web aplikacija

ABSTRACT

Development of a web application for freight organization

During the web application development, HTML, CSS, PHP, MySQL, and XAMPP were used. Firstly, they were described and elaborated on from the theoretical point of view and then, with the usage of acquired knowledge, the application design process started. The first step was to create the web application data structure and semantic structure of web application component pages. Secondly, the user registration and login system were created, including the differentiation of user types and the prevention of unauthorized access. The next step in working with the database was fetching and writing data about vehicles and orders. Finally, the newly created web application was compared to existing applications of similar purposes.

Keywords: database, user, order, structure, web application

ŽIVOTOPIS

Ivan Ivković rođen je 12. prosinca 1996. u Osijeku. Od 2003. do 2011. pohađao je Osnovnu školu Višnjevac, nakon koje je upisao Isusovačku klasičnu gimnaziju s pravom javnosti u Osijeku. Ondje je 2015. godine maturirao i potom upisao preddiplomski studij Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

PRILOZI

CD:

1. Završni rad „Izrada web aplikacije za organizaciju prijevoza robe.docx“
2. Završni rad „Izrada web aplikacije za organizaciju prijevoza robe.pdf“
3. Izvorni kod web aplikacije