Application programming interface (API) za aplikaciju za planiranje izleta

Strugačevac, David

Undergraduate thesis / Završni rad

2022

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:200:047909

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Stručni studij računarstvo

APPLICATION PROGRAMMING INTERFACE (API) ZA APLIKACIJU ZA PLANIRANJE IZLETA

Završni rad

David Strugačevac

Osijek, 2022

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 01.09.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	David Strugačevac
Studij, smjer:	Preddiplomski stručni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR 4694, 26.07.2018.
OIB Pristupnika:	73974240123
Mentor:	Prof. dr. sc. Dominika Crnjac Milić
Sumentor:	Prof. dr. sc. Krešimir Nenadić
Sumentor iz tvrtke:	Helena Stjepanović
Predsjednik Povjerenstva:	Izv.prof.dr.sc. Zdravko Krpić
Član Povjerenstva 1:	Prof. dr. sc. Dominika Crnjac-Milić
Član Povjerenstva 2:	Doc. dr. sc. Tomislav Galba
Naslov završnog rada:	Application programming interface (API) za aplikaciju za planiranje izleta
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Zadatak rada je izraditi REST API za aplikaciju za planiranje izleta. Pomoću API-ja mora biti moguće izraditi mobilnu ili web aplikaciju. Predvidjeti tri korisnička profila s obzirom na funkcionalnosti i pristup podacima: organizator, vodič, korisnik. Korisnik ima mogućnost pregleda i rezervacije izleta. Vodič je osoba zadužena za izlet i vidi samo vlastite izlete i vrijeme kada se održavaju te broj prijavljenih. Organizator vidi sve izlete, kreira nove i uređuje postojeće izlete, definira rutu i plan puta (vrijeme i atrakcije), dodjeljuje vodiča te se brine o rezervacijama. Potrebno je osmisliti sučelje mobilne ili web aplikacije. Još je potrebno napraviti bazu podataka za pohranu informacija. Za funkcionalnosti mobilne ili web aplikacije potrebno je napraviti API. Tehnologija za izradu je skriptni jezik PHP i MVC obrazac Laravel. U radu je potrebno ukazati na korisnost ovakvog proizvoda s organizacijskog aspekta, te opisati korištene tehnologije pri izradi rada. Kao dodatak radu je potrebno dokumentirati API. Mentorica: prof.dr.sc. Dominika Crnjac Milić (FERIT); Sumentori: izv.prof.dr.sc. Krešimir Nenadić (FERIT) i Helena Stjepanović (Factory / Plava tvornica d.o.o.)) (izrada rada u suradnji s tvrtkom)
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	01.09.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 11.09.2022.

Ime i prezime studenta:	David Strugačevac
Studij:	Preddiplomski stručni studij Računarstvo
Mat. br. studenta, godina upisa:	AR 4694, 26.07.2018.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom**: Application programming interface (API) za aplikaciju za planiranje izleta**

izrađen pod vodstvom mentora Prof. dr. sc. Dominika Crnjac Milić

i sumentora Prof. dr. sc. Krešimir Nenadić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. SLIČNE WEB APLIKACIJE 2

2.1.Brod Tours.....2

2.2.Kuna Putovanja.....4

2.3.Starline Agencija.....4

2.4.Ora Tours.....4

2.5.Arriva Travel.....5

3. TEHNOLOGIJE I PROGRAMSKI JEZICI KORIŠTENI ZA IZRADU RADA 6

3.1. Rest API...…….6

3.2. PROGRAMSKI JEZIK PHP….…….6

3.3. Programsko okruženje Laravel……..…….7

3.4. Platforma Internet poslužitelja XAMPP..…… ……..……...7

3.5. Strukturirani programski jezik SQL..……...……. ……...8

3.6. Visual Studio Code……...9

3.7. Lunacy…….10

3.8. Računalni program za testiranje aplikacije Postman……..……...11

4. PROCES IZRADE APLIKACIJE 13

4.1. Dizajn aplikacije……...……..……....13

4.1.1. Početna stranica……..……...……..13

4.1.2. Registracija korisnika…….……..……14

4.1.3. Prijava korisnika…….……..……14

4.1.4. Ponuda izleta….……..……..……...15

4.1.5. Detalji izleta……...……...……17

4.2. Funkcionalnost aplikacije……...……...19

4.2.1. Baza podataka……..…….……..19

4.2.2. Prijava i registracija ……...…….……...20

4.2.3. Odjava……. ……...…….……..……...21

4.2.4. Prikaz izleta……...……...…….……...21

4.2.5. Dodavanje, uređivanje i brisanje izleta……...…….…….....21

4.2.6. Rezervacija mjesta za izlet…...……...…….……...……....22

5.ZAKLJUČAK..... 23

LITERATURA 24

SAŽETAK..... 25

ABSTRACT 26

ŽIVOTOPIS..... 27

1. UVOD

U današnjem svijetu jako puno ljudi putuje u razne dijelove svijeta iz poslovnih ili privatnih razloga. Jedan dio populacije putuje individualno u vlastitom aranžmanu, dok drugi dio pretražuje putničke agencije kako bi putovali na određena odredišta. Većina ljudi se susrela s načinom rada putničkih agencija. One imaju portfelj određenih izleta koje nude i za koje su se specijalizirale. Također, vodiči koji za njih rade specijaliziraju se za vođenje posjetitelja na određenim izletima. Educiraju se i pripremaju izlaganje o određenim sadržajima vezanim za destinaciju koju izletnicima nastoje prikazati u najboljem i najzanimljivijem načinu. Također putnička agencija pruža osiguran prijevoz i smještaj te je to dodatna pogodnost, uz vodiča i cijelu organizaciju izleta, kako bi se ljudi odlučili na izlet putem agencije. Zbog velikog broja korisnika te zbog uštede novčanih sredstava, odnosno smanjenja broja zaposlenih, većina putničkih agencija ima svoju web aplikaciju. Aplikacija uvelike koristi korisnicima kako bi mogli vidjeti plan puta, destinaciju, cijenu, slike, smještaj i ostale bitne informacije vezane uz određeni izlet. Najvažnije je da korisnici mogu rezervirati određeni broj mjesta iz svog doma jer danas većina ljudi ima kod kuće pametni mobilni uređaj ili računalo pomoću kojeg mogu pristupiti raznim aplikacijama, pa tako i aplikacijama putničkih agencija. Rezultat završnog rada je aplikacija za planiranje izleta s relacijskom bazom podataka te *REST API(engl. Representational state transfer Application programming interface)* koji će omogućiti komunikaciju mobilne/web aplikacije s bazom podataka. Zbog toga što je zadatak rada izrada *REST API*-a, postoji samo pozadinski dio aplikacije(*backend)*, a korisničko sučelje (frontend) izrađeno je u programu *Lunacy* čije su slike prikazane u nastavku ovoga rada. U drugom poglavlju opisana su rješenja i aplikacije koje su slične aplikaciji ovog rada. Alati koji su korišteni u već spomenutom pozadinskom dijelu aplikacije opisani su u trećem poglavlju ovoga rada. Četvrto poglavlje govori o cijelom procesu izrade aplikacije te o tome što sve funkcionalnosti aplikacije.

1.1. Zadatak završnog rada

Zadatak rada je izraditi REST API za aplikaciju za planiranje izleta. Pomoću API-ja mora biti moguće izraditi mobilnu ili web aplikaciju. Predvidjeti tri korisnička profila s obzirom na funkcionalnosti i pristup podacima: organizator, vodič, korisnik. Korisnik ima mogućnost pregleda i rezervacije izleta. Vodič je osoba zadužena za izlet i vidi samo vlastite izlete i vrijeme kada se održavaju te broj prijavljenih. Organizator vidi sve izlete, kreira nove i uređuje postojeće izlete, definira rutu i plan puta (vrijeme i atrakcije), dodjeljuje vodiča te se brine o rezervacijama. Potrebno je osmisliti sučelje mobilne ili web aplikacije. Još je potrebno napraviti bazu podataka

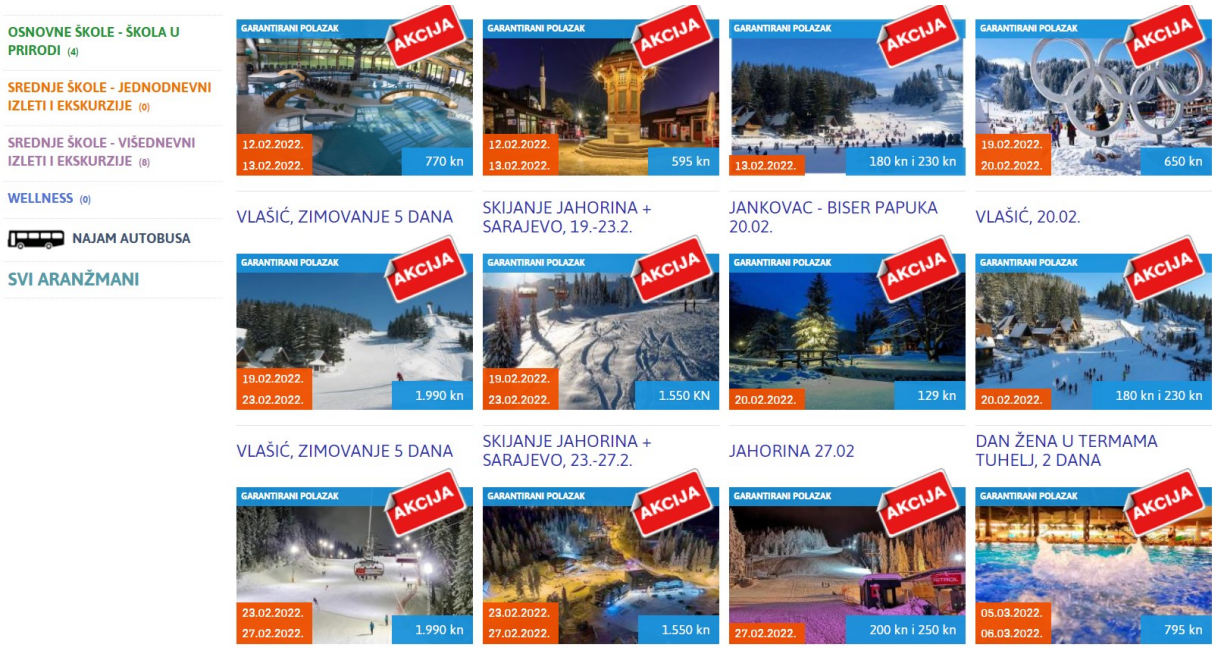
za pohranu informacija. Za funkcionalnosti mobilne ili web aplikacije potrebno je napraviti API. Tehnologija za izradu je skriptni jezik PHP i MVC obrazac Laravel. U radu je potrebno ukazati na korisnost ovakvog proizvoda s organizacijskog aspekta, te opisati korištene tehnologije pri izradi rada. Kao dodatak radu je potrebno dokumentirati API.

2. SLIČNE WEB APLIKACIJE

Na internetu postoji velik broj web aplikacija putničkih agencija putem koje korisnici mogu vidjeti cijelu ponudu izleta te rezervirati svoja mjesta za izlet. Razne aplikacije većinom se razlikuju samo u dizajnu, odnosno izgledu aplikacije. Mogućnosti su na svim aplikacijama gotovo identične s tim da se teži na uvođenju novih stvari kako bi što više zainteresirali korisnike.

2.1. Brod Tours

Jedan od primjera aplikacije je web aplikacija **Brod tours** [1] koja većinom nudi putovanja u susjedne države, ali u ponudi se može naći i putovanje u druge Europske države. Na početnoj stranici nalazi se ponuda svih putovanja koje agencija nudi te korisnik odabire gdje želi putovati.

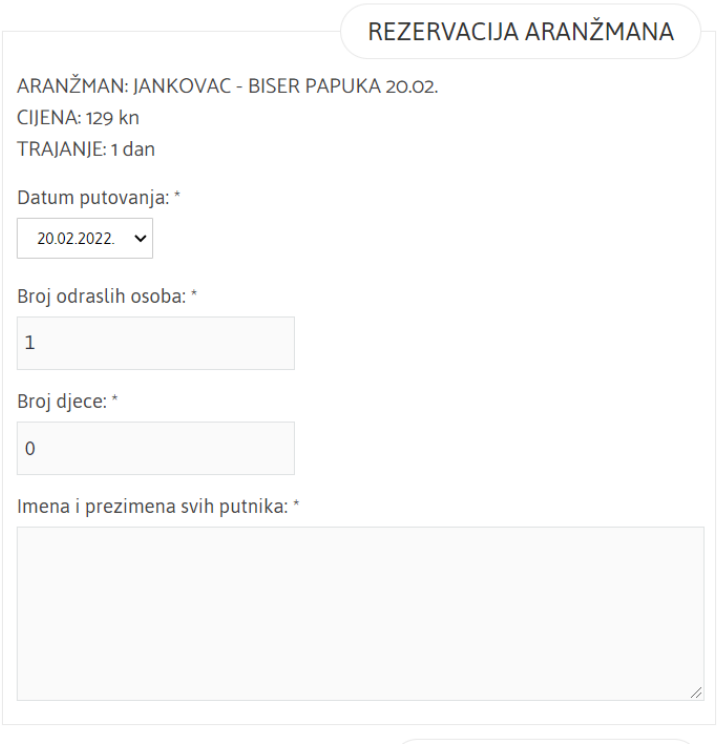


Sl. 2.1. *Prikaz početne stranice putničke agencije* ***Brod tours****[1]*

Nakon odabiranja putovanja korisniku se prikažu detalji putovanja te opcija rezervacije. Rezervacija se može obaviti telefonskim putem, putem elektroničke pošte te popunjavanjem online obrasca za rezervaciju izleta gdje korisnik unosi osobne informacije, broj putnika, kontakt broj te datum putovanja ukoliko postoji ta mogućnost.

Sl. 2.2. *Prikaz detalja putovanja i mogućnost rezervacije*[1]

Ukoliko korisnik želi obaviti rezervaciju putem web obrasca kliknut će na gumb „REZERVACIJA“ te će mu se nakon toga otvoriti obrazac koji treba popuniti ukoliko želi dovršiti rezervaciju. Zbog toga što korisnik unosi osobne podatke, agencija je dužna te podatke čuvati i ne smije ih dijeliti. Obrazac za rezervaciju zahtjeva od korisnika točne podatke jer se prije početka putovanja ti podaci pregledavaju i ukoliko nisu točni korisnik neće moći krenuti na putovanje.



Sl. 2.3. *Prikaz jednog dijela obrasca za rezervaciju*[1]

Baš kao što i navedeni primjer funkcioniranja aplikacija za rezervaciju izleta funkcionira, tako izgledaju i funkcioniraju i ostale aplikacije iste namjene jer je to najjednostavniji i najbrži način pružanja usluge korisnicima i zarade novca agencijama.

2.2. Kuna putovanja

Kuna putovanja je turistička agencija nove dimenzije na ovim tržištima i nalazi se u Osijeku. Kuna putovanja također ima svoju web aplikaciju preko koje korisnici mogu vidjeti cijelu ponudu izleta i rezervirati mjesta za izlet.

2.3. Starline agencija

Starline agencija zadužena za putovanja također se nalazi u Osijeku i većinom nudi putovanja u susjedne države. Starline također ima web aplikaciju s ponudom izleta.

2.4. Ora Tours

Još jedna putnička agencija u nizu čije je sjedište u Osijeku. Nedavno je agencija bila zatvorena zbog pandemije COVID-19.

2.5. Arriva Travel

Arriva Travel ističe se od navedenih agencija po tome što ova agencija nudi putovanja po raznim dijelovima Europe. Ne nude sve agencije putovanja po raznim dijelovima Europe iz razloga što su ta putovanja zahtijevaju više novčanih sredstava te upravo zbog toga mora biti veći broj putnika, a kada bi svaka agencija u ponudi imala putovanja po Europi bilo bi teško pronaći zadovoljavajući broj putnika koji bi omogućili agenciji zaradu.

3. TEHNOLOGIJE I PROGRAMSKI JEZICI KORIŠTENI ZA IZRADU RADA

U ovom poglavlju opisane su tehnologije i programski jezici korišteni za izradu aplikacije. Programi koji su korišteni za izradu pozadinskog dijela aplikacije ovog rada su *SQL (*engl. ***Structured Query Language****), PHP (*engl. ***Hypertext Preprocessor****)* i *Laravel*. Za pomoć u korištenju navedenih tehnologija koriste se alati poput *Visual Studio Code* i *XAMPP*. U računalnom programu *Lunacy* izrađeno je korisničko sučelje (engl. *frontned*).

3.1. Rest API

REST se može definirati kao arhitekturni stil koji se nalazi povrh niza principa među računalnim sustavima koji olakšavaju međusobnu komunikaciju. API označava sučelje za programiranje aplikacija koje je skup definicija i protokola za izgradnju i integraciju aplikacijskog softvera.

REST API komunicira putem HTTP zahtjeva za izvođenje operacija nad bazom podataka kao što su pisanje, čitanje, ažuriranje i brisanje zapisa unutar resursa. Na primjer, API koristi zahtjev GET za dohvaćanje zapisa, POST za stvaranje zapisa, PUT za ažuriranje zapisa i DELETE za BRISANJE za brisanje jednog zapisa. Sve *HTTP* metode mogu se koristiti u API pozivima. Dobro dizajnirani REST API sličan je Internet mjestu koje se izvodi u Internet pregledniku s ugrađenom HTTP funkcijom. [2]

3.2. Programski jezik PHP

PHP (engl. Hipertext Preprocesor) je široko korišten jezik općekorisnih skripti otvorenog koda koji je posebno pogodan za razvoj web stranica i može se ugraditi u HTML(*engl. HyperText Markup Language*). HTML je standardni jezik označavanja za stvaranje web stranica. PHP se koristiti na svim većim operativnim sustavima (Linux, Windows i Mac OS X). PHP se koristi za rad s bazom podataka, odnosno da bi povezala bazu podataka sa samom aplikacijom. Jedna od namjena korištenja PHP-a je funkcionalnost prijave i registracije korisnika preko baze podataka. PHP također podržava mnoge od najkorištenijih poslužitelja, što znači da će programeri imati luksuz izbora ako ga odluče koristiti. Nedvojbeno, najbolja stvar u PHP-u je činjenica da je lako pokupiti kompletne novosti, a istovremeno imati sve napredne značajke koje bi zaseban profesionalac mogao tražiti. [3] PHP programski jezik ima mogućnost komuniciranja s bazom podataka, odnosno može unositi, dohvaćati, mijenjati i brisati podatke iz baze podataka. Postoji puno načina upravljanjem podacima s baze podataka, a PHP ima jedan od najjednostavnijih

pristupa upravljanja podacima. Sintaksa prilikom upravljanja nad podacima podsjeća na sintaksu upravljanja nad podacima direktno iz MYSQL poziva. [4]

3.3. PHP okruženje Laravel

Laravel je besplatan i otvoren PHP okvir koji pruža niz gotovih biblioteka i alata za izgradnju modernih PHP aplikacija. Laravel se velikim dijelom temelji na arhitekturi MVC-a(*engl. Model-View-Controller*). MVC je razvojni model koji odvaja aplikaciju u tri glavne logičke komponente: model, prikaz i kontroler. Laravel je stvoren kako bi se inženjerima olakšalo programiranje PHP projekata. Potpunim sustavom kojim se iskorištavaju njegove ugrađene značajke te različitim kompatibilnim paketima i proširenjima, Laravel je posljednjih godina brzo povećao svoju popularnost, pri čemu su ga brojni programeri usvojili kao svoj okvir za izbor pojednostavnjenog razvojnog procesa. Laravel podržava velik broj gotovih funkcionalnosti za pristup bazama podataka. Uz pomoć Laravel-a programeri mogu stvoriti nove modele, kontrolere i druge komponente aplikacije, što ubrzava cjelokupni razvoj aplikacija. [5] [6]

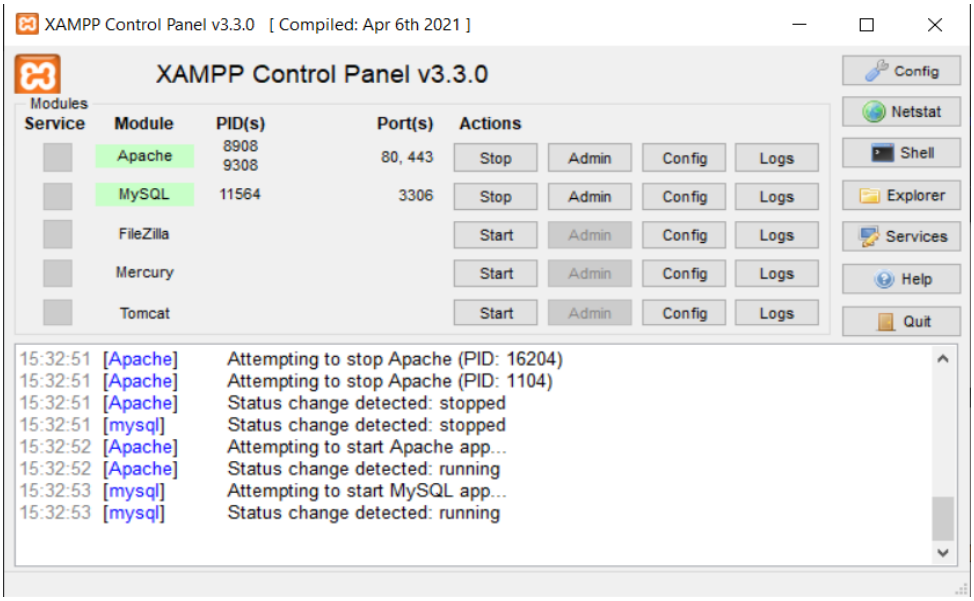
Pomoću Laravela mogu se kreirati i nove baze podataka s novim tablicama koje su napisane u modelu objekta u Laravelu te se to naziva *Code First* pristup. *Code First* pristup označava kreiranje modela koji se koriste u projektu prije kreiranja baze podataka. Laravel ima veći broj već gotovih funkcija i alate koji izuzetno olakšavaju posao programerima kako ne bi morali pisati kod od samog početka. U ovom radu korištene su funkcije za autentifikaciju, rad s bazom podataka te biblioteka za slanje elektroničke pošte.[7]

Laravel pruža mnoštvo jednostavnih mogućnosti koje već ima u sebi, a jedan od njih korišten je i u ovome radu. Riječ je o slanju adrese elektroničke pošte sa servera na adresu elektroničke pošte korisnika. To je potrebno u ovome radu u slučaju ako korisnik zaboravi svoju zaporku i ne može pristupiti računu, odnosno prijaviti se u sustav. Za slanje elektroničke pošte korišten je paket Laravel-a API za elektroničku poštu preko kojeg korisnici dobivaju elektroničku poštu.

Laravel pruža jednostavan API elektroničke pošte koji pokreće popularna *SwiftMailer* biblioteka. Laravel i *SwiftMailer* pružaju programe za slanje e-pošte putem *Mailguna*, poštanske oznake, Amazona SES-a i pošte, što omogućuje brz početak slanja e-pošte putem lokalne usluge ili usluge sa servera. [8] *Mailgun* je usluga dostave e-pošte bazirana na API-ju za upravljanje elektroničkom poštom.

3.4. Platforma Internet poslužitelja XAMPP

XAMPP je skup alata koji emuliraju rad nekoliko vrsta poslužitelja. Sastoji se od emulatora HTTP poslužitelja, SQL baze podataka i prevoditelja za različite skriptne programske jezike kao što je PHP. Dostupan je na jedanaest jezika i podržan je različitim platformama kao što su Windows i Linux. XAMPP omogućuje korištenje SQL, PHP, *phpMyAdmin* koji se koriste pri izradi internet stranice. *PhpMyAdmin* besplatni je GUI *(engl.* ***Graphical User Interface)*** skripta pisan u PHP jeziku za izvođenje administracije zajedničke baze podataka *MariaDB*. Sve operacije nad bazom podataka mogu se izvršiti preko grafičkog korisničkog sučelja koje pruža *PhpMyAdmin*.[9]

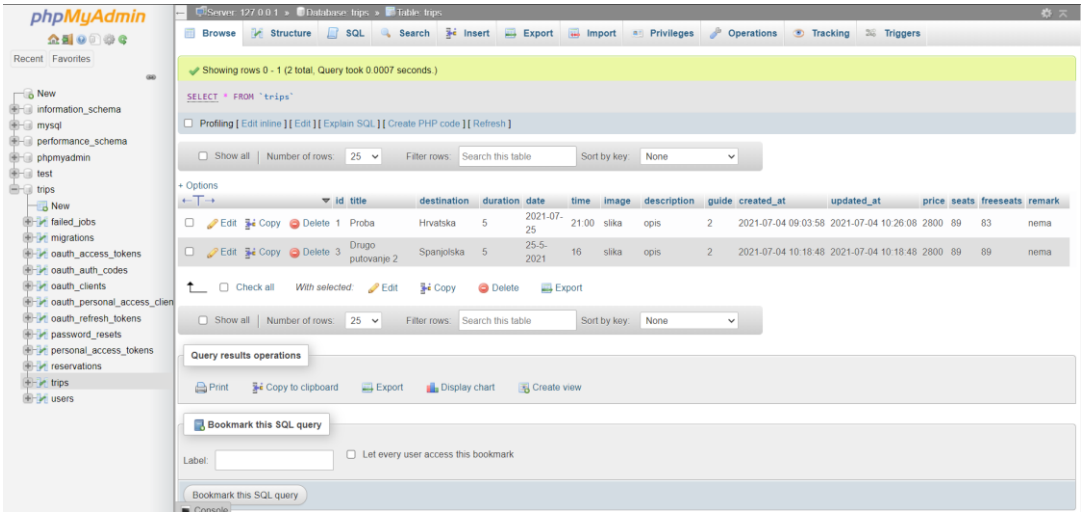


Sl. 3.1. *Prikaz kontrolne ploče programa XAMPP*

3.5. Strukturirani programski jezik SQL

SQL znači strukturirani jezik za upite nad bazom podataka (*engl. Structured Query Language*). SQL se koristi za komunikaciju s bazom podataka. Prema ANSI (engl. *American National Standard Institute*), riječ je o standardnom jeziku za relacijske sustave za upravljanje bazama podataka. SQL se koristi za izvođenje zadataka kao što su ažuriranje podataka u bazi podataka ili dohvaćanje podataka iz baze podataka. Neki od uobičajenih sustava za upravljanje relacijskim bazama podataka koji koriste SQL su: Oracle, Sybase, Microsoft SQL Server, Access, Ingres itd. Iako većina baza podataka koristi SQL, većina njih također ima vlastita dodatna proširenja koja se obično koriste samo na njihovom sustavu. Međutim, standardne SQL naredbe kao što su „Select", „Insert", „Update", „Delete", „Create" i „Drop" mogu se koristiti za postizanje gotovo svega što je potrebno učiniti s bazom podataka. [10] Ono što je ključno za bazu podataka je to da svaka

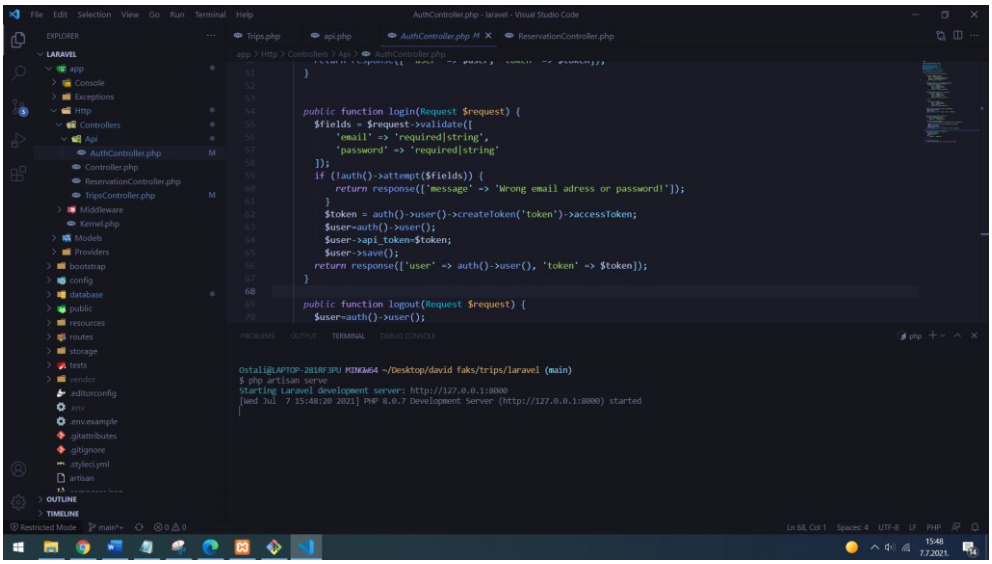
tablica ima primarni ključ preko kojeg se razlikuju podaci, odnosno taj ključ je unikatan za određenu informaciju te se vrijednost tog ključa ne smije ponavljati. Osim primarnog ključa koji služi za razlikovanje podataka iz iste tablice, postoji i strani ključ. Strani ključ označava vezu jedne tablice s drugom tablicom. Uobičajeno je da primarni ključ jedne tablice bude jednak stranom ključu druge tablice. Uz navedene ključeve koji označavaju veze između tablica postoje i druge opcije u SQL-u koje pomažu ograničiti unos netočnih podataka i slično. Na primjer, opcija NOTNULL sprječava da određeno polje bude prazno, odnosno bez vrijednosti, opcija UNIQUE označava da to polje treba biti unikatno, odnosno da se vrijednost tog polja ne bi trebalo i ne može ponavljati te upravo to svojstvo ima primarni ključ. Za spremanje zaporke i drugih osjetljivih podataka koje treba osigurati od neovlaštenog pristupa koriste se algoritmi za enkripciju podataka nakon čega se sprema u bazu podataka upravo taj enkriptirani podatak. Enkripcija podataka je sigurnosna metoda kod koje su informacije kodirane i kojima se može pristupiti ili dešifrirati samo s ispravnim ključem za dešifriranje. Enkripcija se ostvaruje putem jednog od nekoliko različitih algoritama koji vraća vrijednost koja se može dekriptirati ispravnim ključem za dešifriranje. Svaka od različitih mogućnosti šifriranja daje vam drugačiju razinu sigurnosti. Kao što je već spomenuto, što je veća razina šifriranja koju korisnik koristi, to je opterećenje mikroprocesora na Microsoft SQL poslužitelju veće.[11]



Sl. 3.2. *Prikaz sadržaja SQL tablice korištenjem programa XAMPP*

3.6. Razvojni programski alat *Visual Studio Code*

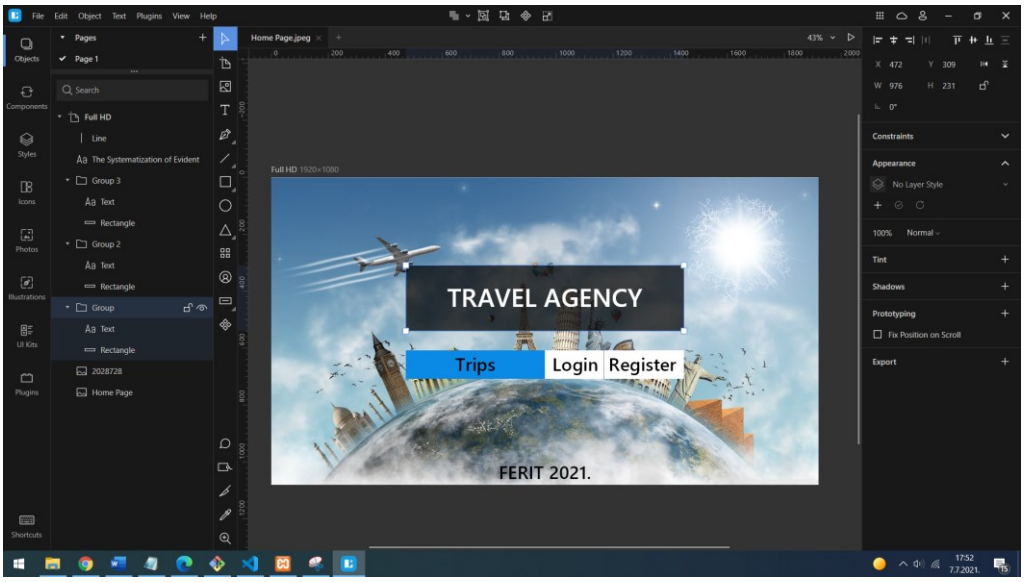
Visual Studio Code je jednostavan, ali moćan uređivač izvornog koda za računalo i dostupan je za Windows, MacOS i Linux. Dolazi s ugrađenom podrškom za JavaScript, TypeScript i Node.js te ima bogata proširenja za druge programske jezike (kao što su C++, C #, Java, Python, PHP). Većina programera koristi Visual Studio Code jer je jednostavan za korištenje, pregledno prikazuje cijeli strukturu projekta, ima puno alata koji automatski rasporede linije koda ili dodaju točku zarez ukoliko se piše u programskom jeziku koji to zahtjeva, a najvažnije je to što podržava razne programske jezike. Navedeni alati programeru omogućuju podršku za ispravljanje grešaka, dovršetak linije koda, isticanje sintakse i ostale razne mogućnosti.



Sl. 3.3. *Prikaz programa Visual Studio Code*

3.7. Lunacy

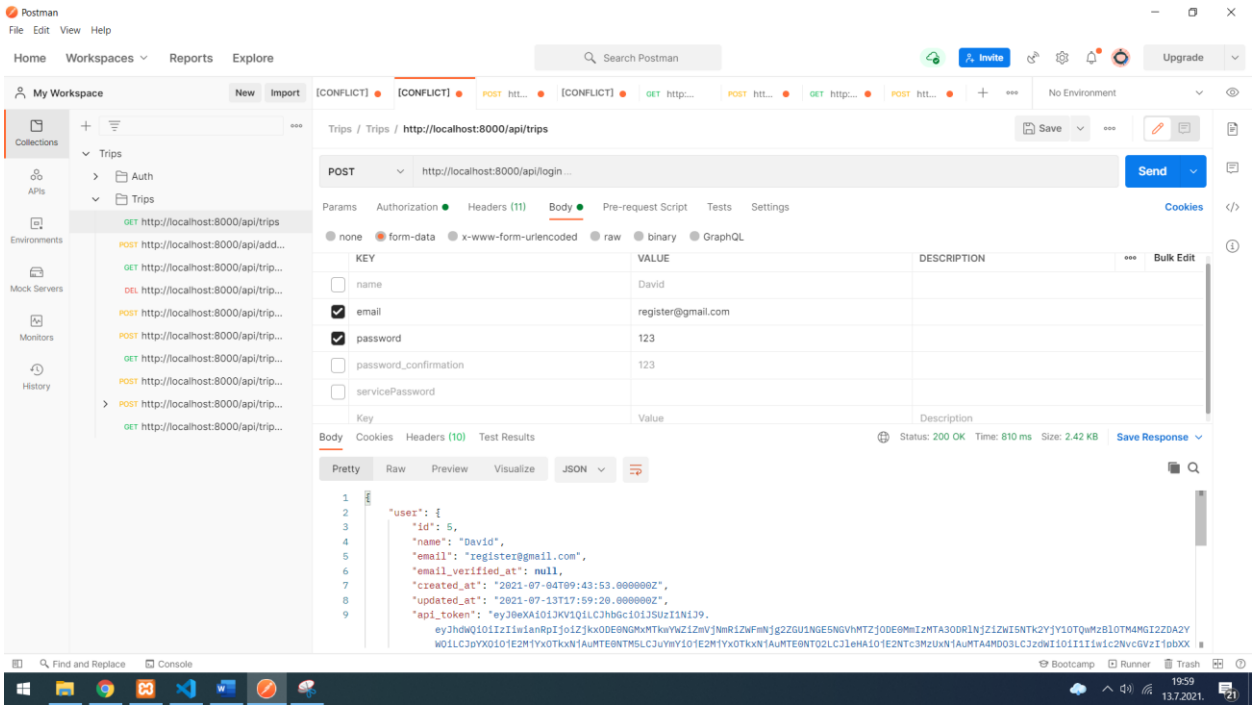
Lunacy je aplikacija za izradu grafičkog dizajna s punom podrškom za *sketch* datoteke. *Sketch* datoteka je vektorska grafička datoteka koju je kreirao *sketch*, program grafičkog dizajna. Postoji nekoliko aplikacija na tržištu koje omogućuju otvaranje *sketch* datoteke, ali niti jedna od njih ne može jamčiti najmanje 80% renderiranje točnost. Lunacy jamči 99%. Bez obzira na format verzije. Nijedna od tih aplikacija ne dopušta uređivanje projekta i spremanje u *sketch* datoteku. [12] Rad u Lunacy-u jednostavan je zbog toga što nudi razne mogućnosti dizajniranja, ima razne oblike koji mogu poslužiti za izradu pojedine komponente stranice te zbog toga što ima jednostavnu programsku traku na kojoj se nalazi sve što korisnik treba prilikom dizajniranja aplikacije.



Sl. 3.4. *Prikaz programa Lunacy*

3.8. Računalni program za testiranje aplikacije Postman

Upravo zbog toga što je zadatak ovog rada bio napraviti Rest Api potreban je računalni program za testiranje aplikacije. Postman je najpoznatiji program za slanje podataka i zahtjeva za određene rute kako bi se testirala funkcionalnost cijele aplikacije. Postman ima mogućnost slanja zahtjeva za Get, Post, Delete, Put i brojne druge metode, a navedene metode su najčešće i najvažnije. Bez takvog računalnog programa nije moguće testirati ispravnost aplikacije ako nije napravljen i prikaz same aplikacije što zahtjeva dodatan i nimalo lak posao. Postman je API klijent koji programerima olakšava stvaranje, dijeljenje, testiranje i dokumentiranje API-ja. To se postiže tako što se korisnicima omogućuje stvaranje i spremanje jednostavnih i složenih HTTP zahtjeva, kao i čitanje njihovih odgovora.[13] Postman ima razne mogućnosti koje uvelike olakšavaju programerima diljem svijeta rad s testiranjem aplikacije i općenito međusobnu komunikaciju zbog toga što svaki programer može napraviti svoju zbirku i podijeliti ju s drugima. To će omogućiti drugim programerima kako bi mogli vidjeti tko je što napravio, imali bi sve potrebne podatke za svaki dio aplikacije te bi znali kako pojedini dio aplikacije radi. Postman ima mogućnost generiranja API dokumentacije što uvelike olakšava posao programerima jer ne moraju ručno pisati dokumentaciju koja može biti jako složena ukoliko je u pitanju malo zahtjevnija aplikacija.



Sl. 3.5. *Prikaz programa Postman*

4. PROCES IZRADE APLIKACIJE

Pri izradi mobilne ili web aplikacije potrebno je osmisliti dizajn same aplikacije kako bi programeri koji rade na funkcionalnosti aplikacije znali koje sve podatke na pojedinim rutama aplikacija zahtjeva te kakvi se podaci unose sa strane korisnika. U ovom radu nije bilo potrebno „isprogramirati“ dizajn aplikacije već samo izraditi u grafičkom programu kako bi otprilike izgledalo korisničko sučelje. Kako bi se omogućila registracija i prijava korisnika, postavljanje izleta te rezerviranje mjesta potrebno je napraviti bazu podataka kako bi se spremili i prikazivali razni podaci o korisniku, izletu ili rezervaciji.

4.1. Dizajn aplikacije

Dizajn aplikacije predstavlja izgled aplikacije, odnosno pogled na aplikaciju s korisničke i administrativne strane. Kako je već navedeno, Lunacy je program koji je korišten za izradu dizajna u ovome radu.

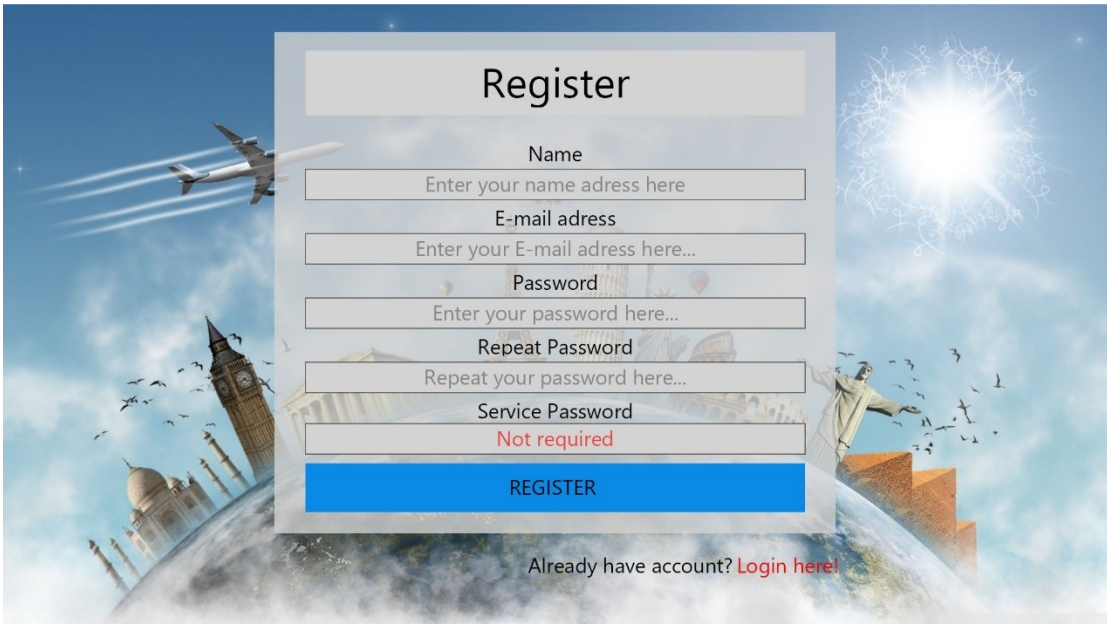
4.1.1. Početna stranica

Početna stranica predstavlja stranicu(sliku) koja je prikazana kada korisnik uđe u aplikaciju. Početna stranica sadrži pozadinsku sliku, ime putničke agencije i tri poveznice. Klikom na svaku poveznicu otvara se nova stranica(slika) s različitim sadržajem.

Sl. 4.1. *Prikaz početne stranice*

4.1.2. Registracija korisnika

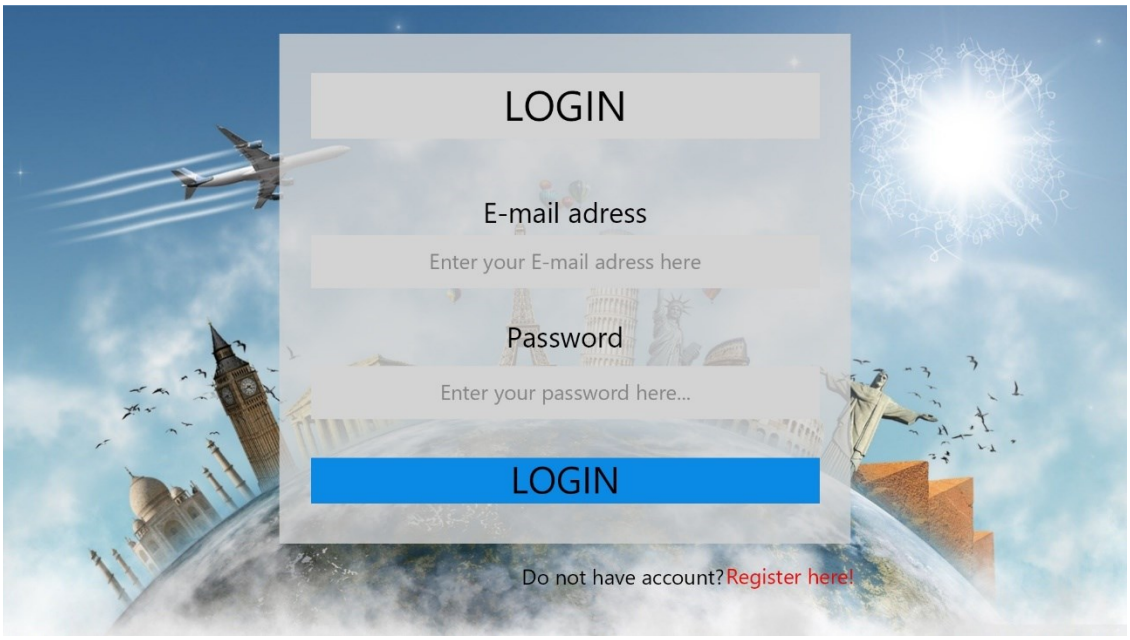
Klikom na poveznicu „Register“ na početnoj stranici prikazat će se stranica s poljima koje je potrebno popuniti kako bi se registrirao (napravio) novi korisnik i upisao u bazu podataka s korisnicima. Polja zahtijevaju određene podatke korisnika zbog zaštite korisničkog računa. Prilikom registracije polje za zaporku tvrtke („Service password“) nije obavezno jer korisniku to nije potrebno već je potrebno samo pri registraciji novog vodiča ili organizatora. Ukoliko se novi vodič ili organizator žele registrirati oni će morati upisati zaporku tvrtke te će im automatski biti dodijeljena pozicija ovisno o šifri koju su unijeli. Na stranici postoji i poveznica za prijavu korisnika kako bi korisnik lakše i brže otvorio stranicu za prijavu ukoliko već ima korisnički račun(profil).



Sl. 4.2. *Prikaz stranice za registraciju*

4.1.3. Prijava korisnika

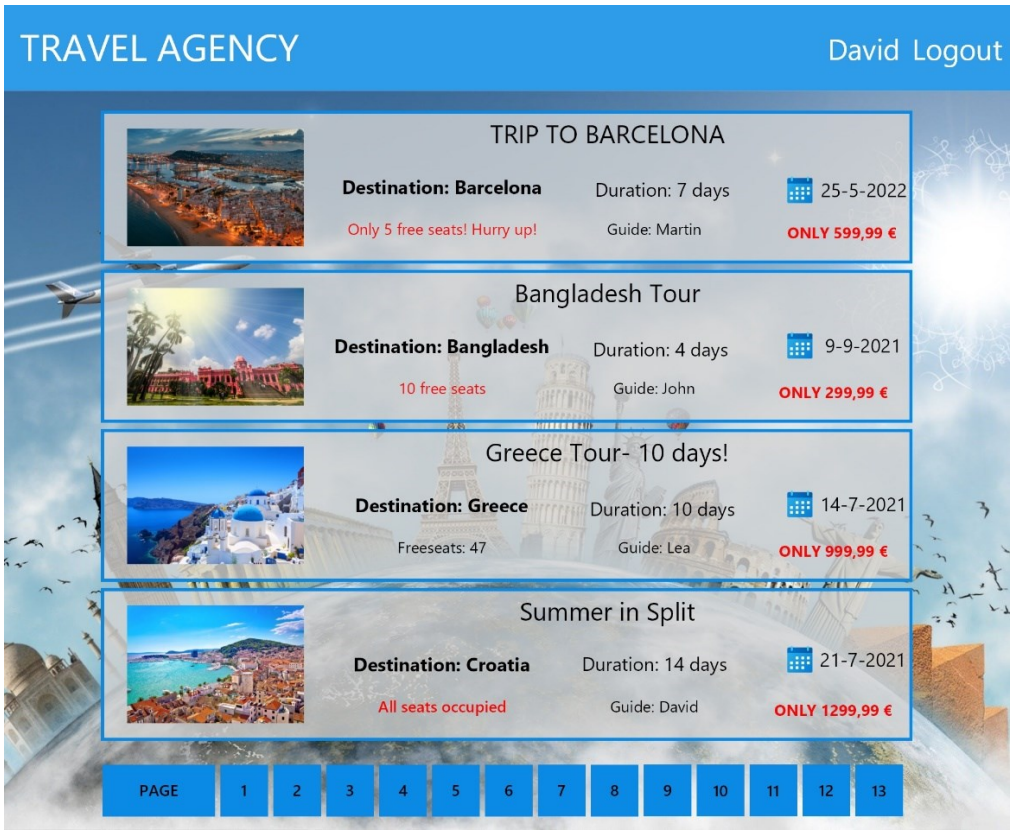
Korisnicima koji imaju svoje korisničke račune potrebna je stranica za prijavu u aplikaciju kako bi mogli vidjeti cijelu ponudu izleta te rezervirati mjesta ukoliko žele. Prilikom prijave korisnici moraju znati svoju elektroničku poštu adresu i zaporku koju su upisali prilikom registracije te se tako osigurava autentičnost korisnika. Stranica sadrži poveznicu na stranicu za registraciju korisnika ukoliko korisnik nema svoj korisnički račun.



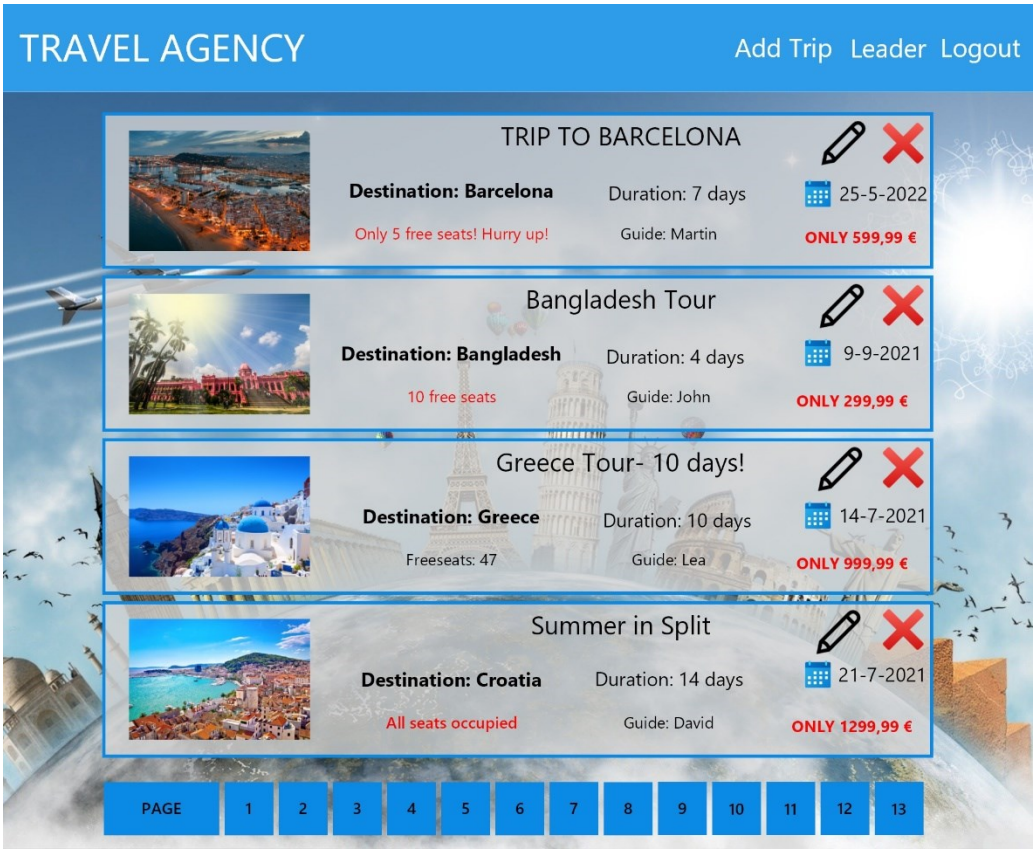
Sl. 4.3. *Prikaz stranice za prijavu*

4.1.4. Ponuda izleta

Nakon što se korisnik registrirao ili prijavio u sustav, klikom na poveznicu „Trips“ s početne stranice, prikazat će se lista svih nadolazećih izleta koji se nalaze u bazi podataka aplikacije. Uz izlete, korisnicima će biti prikazano njihovo ime te će vidjeti poveznicu za odjavu. Za svako izlet korisniku će biti prikazani i najvažniji podaci uz sliku izleta, a to znači da korisnik može vidjeti ime, cijenu, datum, trajanje, destinaciju, vodiča i broj slobodnih mjesta za svako izlet. Slikom 3.4. prikazuje se stranica s listom izleta koju vidi korisnik. Isti takav prikaz izleta vidi i vodič, ali on ne vidi sve izlete već samo izlete na kojima je on vodič. Slikom 4.5. prikazuje se stranica s listom izleta koju vidi organizator. Organizator ima mogućnost uređivanja i brisanja izleta stoga on ima dodatne komponente kod svakog izleta. Organizator također može dodavati nove izlete i upravo zbog tog ima dodatnu poveznicu („Add Trip“). Klikom na poveznicu „Add Trip“ organizatoru će se otvoriti stranica s poljima za dodavanje izleta ista kao i kod registracije samo s drugim poljima, a to znači kako će organizator morati upisati naziv izleta, destinaciju, trajanje, datum, vodiča, vrijeme, opis i napomenu. Isto tako će mu biti prikazani podaci prilikom uređivanja izleta s već postojećim podacima, ali imat će mogućnost promijeniti ih.



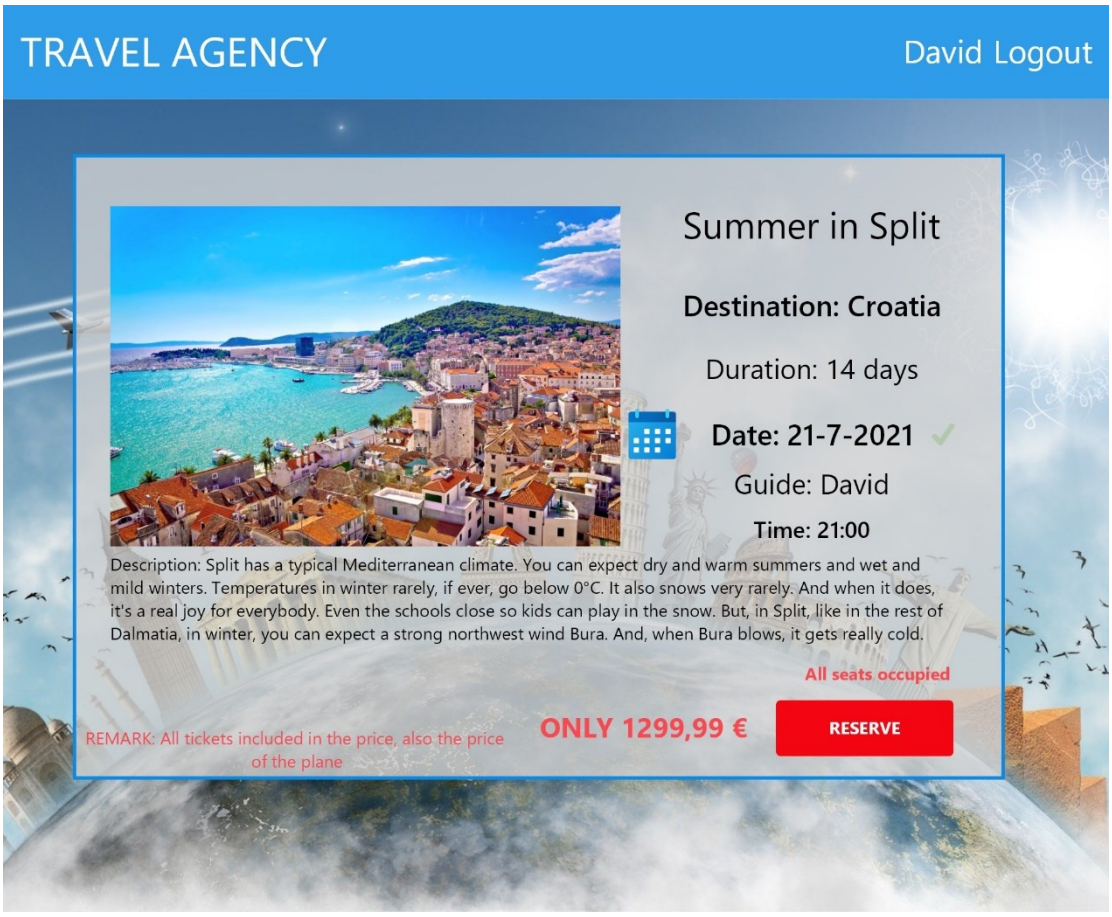
Sl. 4.4. *Prikaz ponude izleta korisniku*



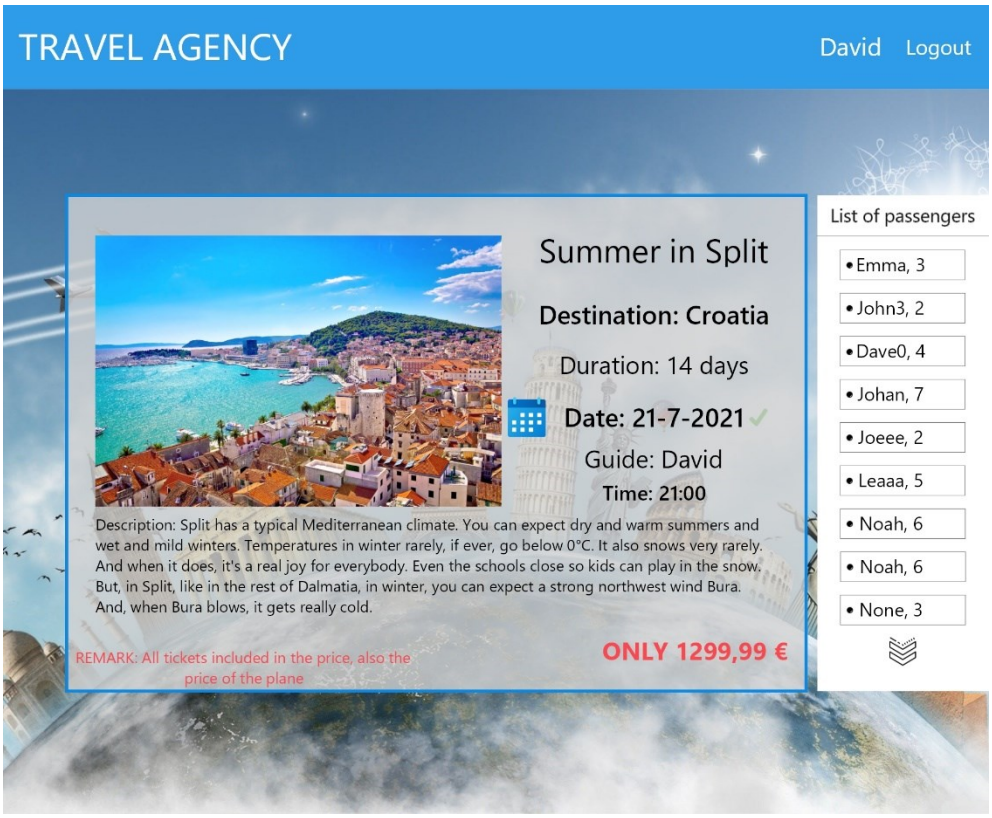
Sl. 4.5. *Prikaz ponude izleta organizatoru*

4.1.5. Detalji izleta

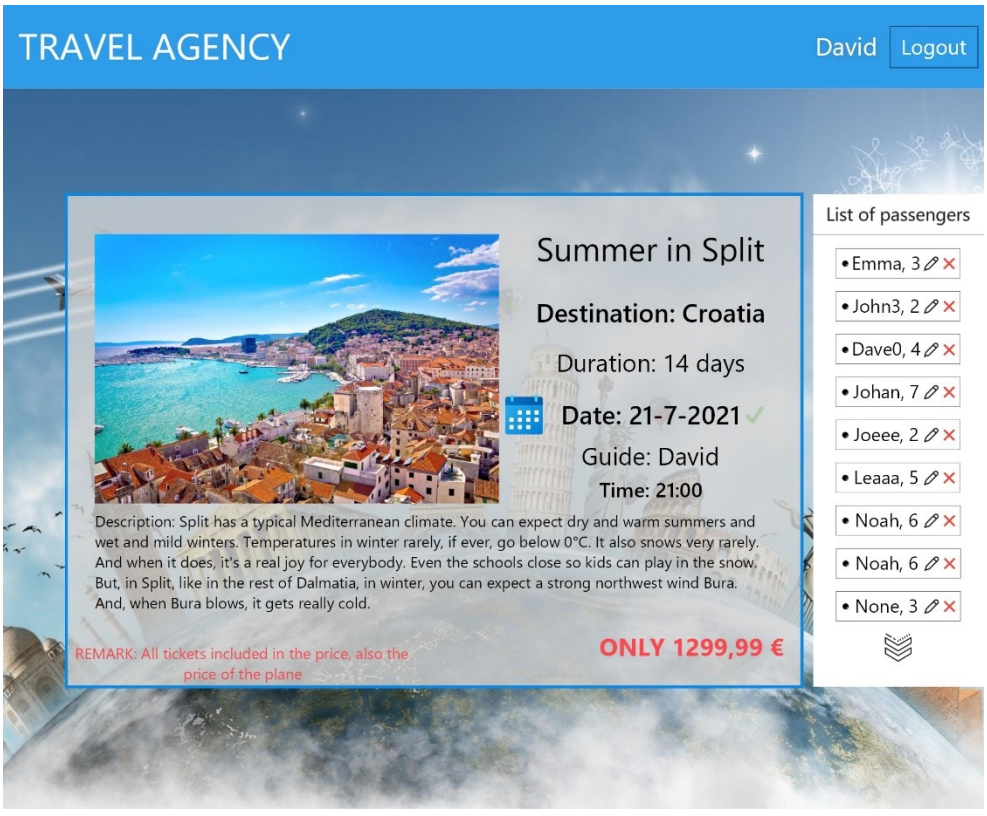
Klikom na izlet korisniku će biti prikazana stranica s detaljima o tom izletu. Korisnik će imati mogućnost rezervacije mjesta. Klikom korisnika na gumb za rezervaciju mjesta otvoriti će se novi prozor s poljem koje će zahtijevati unos broja mjesta. Zbog toga što postoje tri vrste korisničkih profila postoje i tri prikaza detalja o izletu. Slikom 4.6. prikazuje se ono što korisnik vidi, slika 4.7. nema mogućnost rezervacije i prikazuje listu putnika koji su potrebni vodiču te slika 4.8. prikazuje listu putnika s mogućnošću uređivanja i brisanja od strane organizatora. Klikom na uređivanje izleta organizatoru će se otvoriti novi prozor s imenom putnika i brojem mjesta koje je putnik rezervirao te će moći mijenjati broj rezerviranih mjesta.



Sl. 4.6. *Prikaz izleta korisniku*



Sl. 4.7. *Prikaz izleta vodiču*



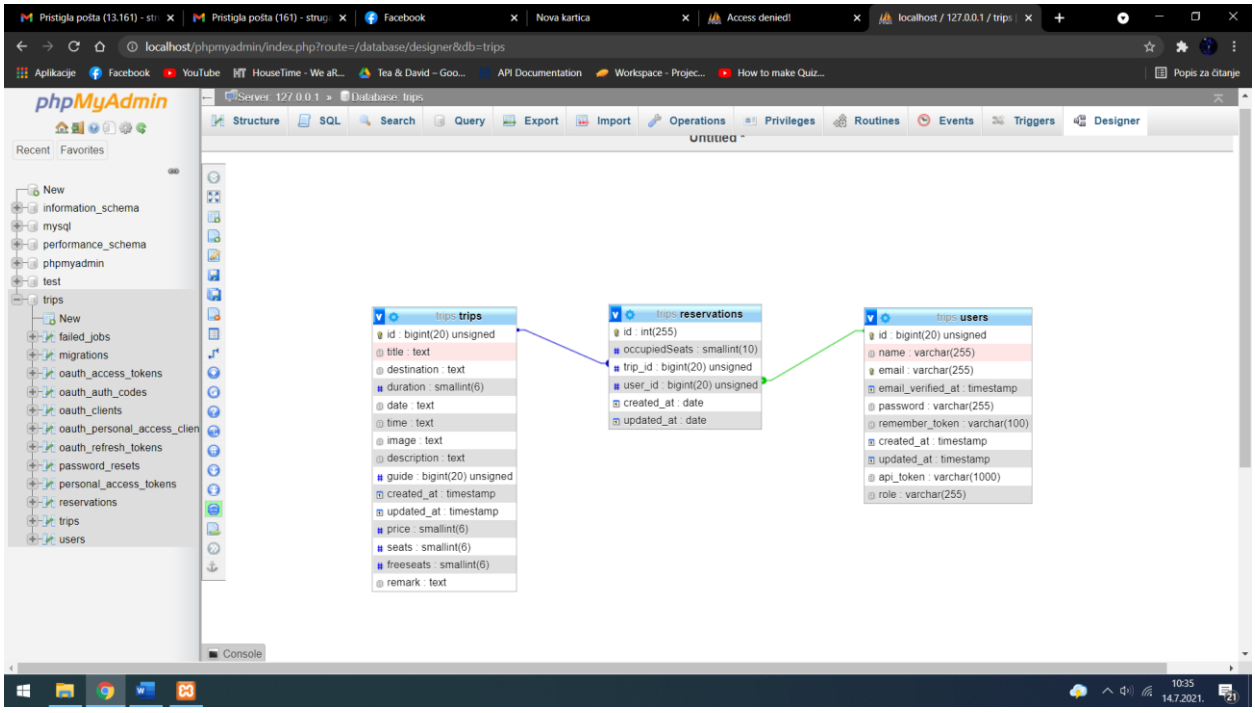
Sl. 4.8. *Prikaz izleta organizatoru*

4.2. Funkcionalnost aplikacije

Nakon što su programeri upoznati s dizajnom aplikacije moraju obrazložiti svu funkcionalnost aplikacije. Kada svi programeri znaju kako aplikacija treba funkcionirati tada prelaze u fazu sastavljanja baze podataka. Dizajn i baza podataka čine strukturu projekta jer se dalje sve odvija prema dizajnu i bazi podataka. Nakon definiranja strukture projekta programeri mogu početi s pisanjem koda, odnosno programiranjem. Koristeći Laravel ovoj aplikaciji korisnicima je omogućena prijava i registracija, pregled svih nadolazećih izleta, pregled svih važnih detalja svih izleta, rezervacija mjesta te brisanje postojeće rezervacije. Organizatoru je omogućeno dodavanje, uređivanje i brisanje izleta i rezervacija, a vodiču je omogućen pregled njegovih izleta.

4.2.1. Baza podataka

Bez baze podataka gotovo je nemoguće napraviti ovu aplikaciju zbog toga što se kontinuirano izleti izmjenjuju izleti izmjenjuju na način da se brišu izleti koji su prošli i dodaju novi nadolazeći izleti, zbog dolaska novih i odlaska starih korisnika te zbog rezervacija. Baza podataka služi kako bi se svi podaci koji su potrebni za funkcionalnost aplikacije pohranili te koristili tamo gdje su potrebni. Bazi podataka ima pristup jedino administrator same aplikacije jer njemu nije u interesu brisati ili uređivati podatke koji se nalaze u bazi podataka. Slikom 4.9. prikazuje se relacijska veza između tablica korištenih u ovome radu. Rezervacije sadrže strani ključ korisnika i izleta kako bi se točno znalo za koje putovanje je određen korisnik rezervirao mjesta.



Sl. 4.9. Prikaz baze podataka preko ER dijagrama

4.2.2. Prijava i registracija

Prijava i registracija u gotovo svim aplikacijama postoje zbog toga kako bi se korisnici (ali i zaposlenici) identificirali. Identifikacija je jedna od najvažnijih dijelova svakog sustava zbog krađe ili iskorištavanja tuđih podataka, primjerice bankovni sustavi. U ovoj aplikaciji identifikacija je važna upravo zbog toga kako bi rezervirana mjesta bila zaista uručena korisniku koji je obavio rezervaciju. Prilikom registracije korisnik upisuje svoje ime, adresu elektroničke pošte i zaporku koju mora upisati dva puta zbog sigurnosti jer se može dogoditi pogrešan unos znakova na tipkovnici. Prilikom unosa korisnik ne može vidjeti unesenu zaporku već samo točkice koje zamjenjuju slova, stoga ne može biti siguran da je unesena zaporka ujedno i željena zaporka korisnika te je upravo iz tog razloga potrebno ponoviti unos zaporke. Prilikom registracije postoji polje za unos zaporke tvrtke, ova zaporka obavezna je samo ako se registrira novi vodič ili novi organizator koji su zaposlenici tvrtke koja je vlasnik aplikacije. Ukoliko zaporka odgovara zaporki za voditelja, korisnik će biti registriran kao voditelj, ukoliko je zaporka jednaka zaporki za vodiča korisnik će biti registriran kao vodič. Ukoliko se registrira novi korisnik koji nije član tvrtke njemu se jednostavno dodjeljuje titula gosta te on može pregledavati izlete i obaviti rezervaciju. Prilikom prijave i registracije stvara se niz slučajnih znakova koji Laravel koristi kako bi uspješno identificirao korisnika. Zaporka se u bazu podataka sprema kao šifrirani podatak iz kojeg je nemoguće doznati o kojoj se zaporki zapravo radi. Za šifriranje zaporke većinom se koriste paketi programskog okruženja u kojem se programira aplikacija. To je dobra praksa radi same korisnikove sigurnosti, ali i sigurnosti same tvrtke koja je vlasnik aplikacije.

Ukoliko je korisnik napravio već svoj račun, on se mora ponovno prijaviti u sustav. Prilikom prijave potrebni su adresa elektroničke pošte i zaporka. Podaci uneseni prilikom prijave provjeravaju se s podacima u bazi podataka te ukoliko uneseni podaci nisu točni korisniku se pojavi greška te mora pokušati ponovo.

Nakon prijave u sustav korisniku se dodjeljuje token koji služi za autentifikaciju korisnika. Token za autentifikaciju koristi se u gotovo svim aplikacijama u kojima se traži od korisnika da se prijave ili registriraju. Token omogućava aplikaciji ograničavanje pristupa aplikaciji i podacima korisnicima koji ne bi trebali imati pristup tim podacima.

4.2.3. Odjava

Nakon što je korisnik identificiran u sustavu, on ima mogućnost odjave. Prilikom odjave korisniku se briše token koji služi za identifikaciju korisnika te je nakon toga on odjavljen iz sustava i ima mogućnost prijave ili registracije.

4.2.4. Prikaz izleta

Nakon prijave ili registracije korisnik ima mogućnost uvida u izlete i rezervaciju mjesta. Pregled izleta biti će prikazan baš kao što je već prethodno prikazano, a podaci će se dohvaćati iz baze podataka. Ukoliko ne postoji niti jedan izlet zapisan u bazi podataka korisnik će dobiti poruku kako ne postoji niti jedan izlet. Kod prikaza izleta važno je napomenuti kako je potrebno provjeriti poziciju prijavljenog korisnika jer vodič smije vidjeti samo svoje izlete dok svi ostali korisnici mogu vidjeti sve izlete.

Ako korisnik klikne na bilo koji ponuđeni izlet otvorit će mu se detaljan prikaz izleta i pružit će mu se mogućnost rezervacije. Koristeći Laravel, svaki izlet dohvaća se preko njegovog „Id“ jer to predstavlja identifikacijski broj izleta te je on jedinstven što znači kako ne mogu dva izleta imati isti „Id“. Kao što je već prethodno rečeno, postoje tri vrste korisnika i upravo zbog toga postoje i različiti pogledi na izlete. Organizator i vodič imaju uvid u detalje izleta i listu rezervacija dok običan korisnik može vidjeti samo detalje putovanja.

4.2.5. Dodavanje, uređivanje i brisanje izleta

Kao što je već spomenuto, organizator se brine za izlete i upravo zbog toga ima mogućnost dodavanja, uređivanja i brisanja izleta. Kod svih tih funkcija prvo se provjerava da li je prijavljeni korisnik ujedno i organizator. Ostalim korisnicima neće biti prikazane mogućnosti upravljanja nad izletima, no korisnik može samostalno u preglednik upisati rutu koja predstavlja neku od tih funkcija te može lako izvršiti određenu funkciju nad izletom. Nakon toga provjeravaju se podaci koji su uneseni sa strane organizatora te se pojavljuje greška ukoliko su neki od podataka pogrešni. Kod dodavanja izleta postoje dvije provjere uz već navedenu provjeru pozicije prijavljenog korisnika. Drugom provjerom kontrolira se da li je dodijeljen vodič na putovanju doista i postoji, ukoliko ne postoji organizator će vidjeti grešku te imati mogućnost ponovnog dodjeljivanja vodiča. Uređivanje izleta je kompleksnije od dodavanja jer prilikom uređivanja izleta potrebno je vodit brigu o tome da li izlet uopće postoji, poziciji korisnika, broju rezerviranih mjesta i dodjeli vodiča

ukoliko se taj podataka mijenja. Brisanje izleta je najjednostavnija funkcija jer je potrebno provjeriti samo da li je korisnik ujedno i organizator te da li izlet koji se želi obrisati postoji.

4.2.6. Rezervacija mjesta za izlet

Rezerviranje mjesta najvažniji je dio ove aplikacije jer je to ujedno i tema ovog rada. Kako bi korisnik mogao rezervirati mjesta za izlet potrebna mu je prijava u aplikaciji kako bi ga sustav identificirao i uspješno zabilježio njegovu rezervaciju. Sama rezervacija korisnika je jednostavna jer korisnik mora samo odabrati putovanje i broj koliko mjesta želi rezervirati. Rezervacija nije moguća za vodiče i organizatore, korisnik ne može rezervirati više od 8 mjesta te korisnik ne može dva puta izvršiti rezervaciju istog izleta. Ukoliko korisnik pokuša izvršiti rezervaciju kada su sva mjesta već rezervirana prikazat će mu se greška kao i kod pokušaja rezervacije više mjesta nego što je slobodno. Prilikom svake rezervacije u bazu podataka upisuje se broj mjesta koliko je korisnik rezervirao te se oduzima od broja slobodnih mjesta tog izleta.

Korisnici mogu i obrisati svoju rezervaciju ukoliko to žele, no samo ako su već prethodno rezervirali proizvoljan broj mjesta za odabrani izlet. Kod brisanja rezervacije brišu se svi podaci vezani uz tu rezervaciju te se uzima broj rezerviranih mjesta koje je korisnik imao rezervirano i dodaje se na trenutni broj slobodnih mjesta tog izleta kako bi se prikazao stvaran broj slobodnih mjesta.

Korisnicima nije više ništa potrebno u vezi rezervacija jer im je rezerviranje i brisanje njihovih rezervacija dovoljno i nema potrebe za druge radnje nad rezervacijama dok organizatoru treba ipak malo više mogućnosti jer on vodi brigu oko svih rezervacija te se brine da sve ide po planu i da ne bude nikakvih problema. Organizator ima mogućnost uređivanja postojećih rezervacija kao i njihovo brisanje. Ukoliko korisnik želi promijeniti broj rezerviranih mjesta, to može učiniti tako da obavijesti organizatora koji će to obaviti. Organizator ima mogućnost brisanja rezervacija ukoliko se dogodi nekakav problem ili slično. Kod uređivanja rezervacije također se provjerava ima li ima dovoljno slobodnih mjesta ako je korisnik zatražio povećanje mjesta te se odmah i smanjuje broj slobodnih mjesta izleta, vrijedi i obrnuto.

5. ZAKLJUČAK

Aplikacija za rezervaciju izleta služi korisniku kako bi najlakšim i najjednostavnijim putem mogao rezervirati mjesta za izlet iz udobnosti svog doma. Isto tako koristi i zaposlenicima tvrtke koji posjeduju ovu aplikaciju kako bi brzo i lako mogli obavljati svoje poslove, odnosno kako bi organizator vodio brigu oko izleta i rezervacija, a vodiči oko plana putovanja i samog održavanja izleta koje oni održavaju. Aplikacija je transparentna kako za korisnika tako i za zaposlenike zbog svoje jednostavnosti i mogućnosti uvida u izlete i rezervacije. U procesu izrade same aplikacije u PHP programskom jeziku i Laravel okruženju uvidjele su se pozitivne praktične stvari takvog programiranja, te sama povezanost između baze podataka i aplikacije. Dobre strane takvog programiranja su da je jednostavno i lako učenje tehnike programiranja uz prethodno znanje nekih programski jezika i rad s bazom podataka. Pri završetku aplikacije otvorile su se nove mogućnosti nadogradnje aplikacije kako bi ova aplikacija što više nudila korisnicima i pojednostavnila im korištenje iste. Komponentu za nadogradnju aplikacije ovog rada može se navesti ostavljanje recenzija i poruka od strane korisnika za pojedino putovanje ili za pojedinog vodiča, a ne bi odmoglo ni to da korisnici mogu ostaviti općeniti dojam na rad agencije i njezinih zaposlenika. Trenutno se može reći da je korisnik ograničen u vezi svog računa jer nema mogućnost mijenjanja informacija o računu. U svrhu daljnjeg istraživanja ne bi bilo loše uključiti i to da korisnici sami predlože odredišta putovanja te da koristeći anketu agencija provjeri koliko je korisnika zainteresirano za predloženo putovanje jer će agencija privući više korisnika, a samim time i ostvariti svoj cilj.

LITERATURA

[1] Brod Tours, dostupno na: https://www.brod-tours.hr/o-nama/ ,(Pristupljeno dana: 15.07.2021.)
[2] REST API, dostupno na: https://www.ibm.com/cloud/learn/rest-apis
(Pristupljeno dana: 15.07.2021.)
[3] What is PHP?, dostupno na: https://codeinstitute.net/blog/what-is-php-programming
(Pristupljeno dana: 15.07.2021.)
[4] 5 Ways To use row database queries in Laravel, dostupno na:
https://blog.quickadminpanel.com/5-ways-to-use-raw-database-queries-in-laravel/
(Pristupljeno dana: 23.01.2022.)
[5] What is Laravel?, dostupno na: https://www.digitalocean.com/community/tutorials/what-is-laravel (Pristupljeno dana: 15.07.2021.)
[6] Why Laravel? , dostupno na: https://laravel.com/ ,
(Pristupljeno dana: 15.07.2021)
[7] Stack Overflow Contributors, Learning Laravel, dostupno na:
https://www.researchgate.net/publication/334304361_Exploring_Object_Oriented_and_Database_Programming_Concepts_of_PHP_-_Harnessing_the_Power_of_XAMPP
(Pristupljeno dana: 15.07.2021)
[8] Laravel Mail, dostupno na: https://laravel.com/docs/8.x/mail
(Pristupljeno dana: 15.07.2021)
[9] Dr.Poornima, G.Naik, Girish R. Naik, Exploring Object Orientedand Database:
Programming Concepts in PHP, dostupno na:
https://www.researchgate.net/profile/Poornima_Naik3/publication/334304361_Exploring_Object_Oriented_and_Database_Programming_Concepts_of_PHP_-_Harnessing_the_Power_of_XAMPP/links/5d4c16d5299bf1995b6f7874/Exploring-Object-Oriented-and-Database-Programming-Concepts-of-PHP-Harnessing-the-Power-of-XAMPP.pdf
(Pristupljeno dana: 23.01.2022.)

[10] What is SQL?, dostupno na: http://www.sqlcourse.com/intro.html
(Pristupljeno dana: 23.01.2022.)
[11] Hashing versus Encryption, dostupno na: https://www.sqlshack.com/the-hashbytes-function-in-t-sql/ (Pristupljeno dana: 23.01.2022.)
[12] About Lunacy, dostupno na: https://docs.icons8.com/about (Pristupljeno dana: 23.01.2022.)
[13] What is Postman?, dostupno na: https://www.blazemeter.com/blog/how-use-postman-manage-and-execute-your-apis (Pristupljeno dana: 23.01.2022.)

SAŽETAK

Web aplikacija čija je svrha olakšati korisnicima pristup ponudi koju agencija nudi, pregled svih detalja ponude i mogućnost rezervacije iz udobnosti svog doma. Postoje tri moguće uloge u aplikaciji. Voditelj može kreirati i upravljati ponudom putovanja, vodič može pregledati svoja putovanja na kojima je on zadužen kao vodič i krajnji korisnik koji ima mogućnost pregleda ponude putničke agencije i ima pristup rezervaciji mjesta za izlet. Voditelj se također brine o rezervacijama, što znači da on može brisati i uređivati rezervacije umjesto korisnika ukoliko korisnik zatraži takvo nešto ili ukoliko se dogodi problem.

Ključne riječi: putnička agencija, putovanje, Laravel, rezervacija, web aplikacija

ABSTRACT

Title: APPLICATION PROGRAM APPLICATION INTERFACE (API) FOR EXCURSION PLANNING APPLICATION

Web application whose purpose is to make it easier for users to access the offer offered by the agency, to review all the details of the offer and to reserve from the comfort of their home. There are three possible roles in the application. The tour manager can create and manage a travel offer, guide can view his travels where he is in charge as a guide and end user who has the possibility to review travel agency's offer and has access to reservation of a place for excursion. The manager also takes care of the reservations, which means that it can delete and edit the reservations instead of the user if the user requests such a thing or if a problem occurs.

Keywords: travel agency, travel, Laravel, reservation, web application

ŽIVOTOPIS

David Strugačevac rođen je u Osijeku 29.01.1999. godine. Osnovnu školu Osnovna Škola Valpovo pohađao je i završio u Valpovu. Nakon završene Osnovne škole svoje školovanje nastavlja u Srednjoj školi Valpovo upisujući smjer Opća gimnazija. Nakon završetka srednjoškolskog obrazovanja upisuje Preddiplomski stručni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Trenutno zaposlen kao backend developer u Osijeku.

_____

Potpis autora