Izrada CM sustava za web stranice s turističkim sadržajem

Klešić, Ivan

Undergraduate thesis / Završni rad

2017

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:200:223733

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni preddiplomski studij računarstva

IZRADA CM SUSTAVA ZA WEB STRANICE S TURISTIČKIM SADRŽAJEM

Završni rad

Ivan Klešić

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek,12.09.2017.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Ivan Klešić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 3652, 23.07.2014.
OIB studenta:	56360121376
Mentor:	Doc.dr.sc. Zdravko Krpić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Izrada CM sustava za web stranice s turističkim sadržajem
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	12.09.2017.
Datum potvrde ocjene Odbora:	

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 27.09.2017.

Ime i prezime studenta:	Ivan Klešić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 3652, 23.07.2014.
Ephorus podudaranje [%]:	8

Ovom izjavom izjavljujem da je rad pod nazivom: **Izrada CM sustava za web stranice s turističkim sadržajem**

izrađen pod vodstvom mentora Doc.dr.sc. Zdravko Krpić

i sumentora

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. CMS-OVI ZA PONUDU TURISTIČKOG SADRŽAJA 2
2.1. Zahtjevi na programsko rješenje 2
3. TEHNOLOGIJE ZA RAZVOJ WEB APLIKACIJA 4
3.1. HTML 4
3.2. CSS 5
3.2.1. Bootstrap 5
3.3. JavaScript 5
3.3.1. jQuery 6
3.4. PHP 6
3.5. Laravel 6
3.5.1. MVC arhitektura 7
3.5.2. Artisan i Tinker 8
3.5.3. Eloquent 9
4. REALIZACIJA ZADATKA U RAZVOJNOJ CJELINI LARAVEL 10
4.1. Korisnički profili 10
4.1.1. Ažuriranje profila korisnika i dodavanje događaja u raspored 12
4.2. Objavljivanje sadržaja 15
4.3. Postavljanje slika i videozapisa 16
5. OPIS I VREDNOVANJE PROGRAMSKOG RJEŠENJA 19
6. ZAKLJUČAK 20
LITERATURA 21
SAŽETAK 22
ABSTRACT 23
ŽIVOTOPIS 24

1. UVOD

Tema ovog završnog rada je izrada CM sustava tj. CMS-a za web stranice s turističkim sadržajem. **CMS** (eng. *Content management system*) je web aplikacija za upravljanje sadržajem web stranica. Radi se o aplikaciji koja je napisana u nekom od programskih jezika te koja ima podršku odgovarajuće baze podataka. Takve aplikacije omogućuju razdvajanje dizajna i sadržaja web stranica. Ovaj pristup je danas vrlo zastupljen jer se od modernih web stranica očekuje ažuriranje postojećeg sadržaja te dodavanje novog. Za izvršavanje promjena na web stranici koristeći CMS nije potrebno znanje programiranja niti web dizajna, već se cijeli postupak svodi na ispunjavanje odgovarajućih modula potrebnim podacima. Korištenjem CMS-a se također eliminira trošak ažuriranja web stranica, jer u tom slučaju ažuriranja vrši sam korisnik.

Prvo je ukratko objašnjena potreba za ovakvim rješenjem te je napravljena usporedba sa postojećim rješenjima, a zatim su dani zahtjevi na programsko rješenje. Obrađena je teorija vezana uz razvoj web aplikacija, nakon čega dolazi opis razvoja CMS-a i mogućnosti istoga. Zatim je programsko rješenje opisano i vrednovano te je na kraju dan zaključak.

1.1. Zadatak završnog rada

Potrebno je izraditi vlastiti CMS koji će biti namijenjen upravljanju web stranicama s ponudom turističkih usluga. CMS treba omogućiti dodavanje multimedijskih sadržaja (slike, video zapisi), objavljivanje turističkih ponuda, dinamičko ažuriranje rasporeda za turističke vodiče te upravljanje profilom pojedinog turističkog vodiča. Za izradu CMS-a potrebno je koristiti HTML, CSS, PHP i MySQL. Mogućnosti CMS-a primijeniti na postojećoj web stranici na kojoj su zaposleni „freelance“ turistički vodiči. Podaci o njima bit će spremljeni u bazu podataka (MySQL). Kompanije koje organiziraju putovanja moći će na stranici dobiti informacije o vodičima te imati uvid u njihov raspored. Svaki vodič će imati mogućnost ažuriranja svog rasporeda, koji će biti prikazan u obliku kalendara na kojem je naznačeno kada pojedini vodič već ima dogovoren posao, odnosno kada je slobodan za nove poslove.

2. CMS-OVI ZA PONUDU TURISTIČKOG SADRŽAJA

Potreba za ovakvim programskim rješenjem dolazi od činjenice da je turizam u današnje vrijeme vrlo zastupljen, te je poželjno imati više ponuđenih rješenja za upravljanje sadržajem takvih web stranica.

Specifičnosti CMS-ova za stranice s turističkim sadržajem su mogućnosti objavljivanja turističkih ponuda, davanje informacija o turističkim vodičima, postavljanje multimedijskih sadržaja, podrška za više jezika, mogućnosti rezerviranja i slično. Neka od postojećih rješenja koja su inspirirala izradu ovoga su „eCMS“, „Magnolia CMS“ i „TourCMS“. Prednosti ovih rješenja su optimizacija za tražilice, elementi koji su specificirani za rezervaciju, standardizirana podrška za više web stranica (ne nužno jednakog izgleda) te orijentiranost prema većim turističkim tvrtkama.

CMS razvijen u okviru ovoga rada će biti orijentiran prema manjim turističkim obrtima koji povezuju velike tvrtke sa lokalnim destinacijama. Specifične mogućnosti koje ranije navedeni CMS-ovi ne sadržavaju su mogućnost sadržavanja galerije te održavanje profila turističkih vodiča, zajedno sa rasporedom.

2.1. Zahtjevi na programsko rješenje

U zadatku završnoga rada (poglavlje 1.1) su kratko objašnjeni elementi koje će CMS razvijen u okviru ovoga rada sadržavati. Ovdje će biti objašnjeni detaljnije.

Ideja je omogućiti korisnicima (u ovom slučaju turističkim vodičima) korištenje, održavanje i prikazivanje vlastitih profila. Vlastiti profili su objekti koji sadržavaju podatke o korisnicima te se koriste u svrhu prikazivanja istih. Podaci koje vlastiti profili sadržavaju se nalaze u bazi podataka. Svaki vodič će imati standardne atribute koji se koriste za prijavu (e-mail adresa te lozinka), te ime, prezime i opis koji će biti javno dostupni. U vlastiti opis će korisnici moći uvrstiti bilo kakve informacije koje smatraju bitnima npr. vještine, prijašnja iskustva itd. Također će biti realizirana i mogućnost objavljivanja sadržaja te postavljanja multimedijskih sadržaja na poslužitelj. Za prikaz multimedijskih sadržaja bit će napravljena galerija u kojoj će se moći odabrati željeni sadržaj strukturiran po kategorijama i potkategorijama. Za prikaz samih modula od kojih će se sastojati CMS (front end) će se koristiti HTML, koji nudi mnoštvo elemenata kao npr. polja za tekst, tipke, elemente za prikazivanje slika itd. Za definiranje estetskog prikaza HTML elemenata će biti korišten

CSS, odnosno razvojna cjelina Bootstrap koja nudi standardizirane načine prikaza pojedinih HTML i CSS elemenata. Javascript će biti korišten za skriptiranje na strani korisnika kao npr. prikazivanje određenih modula kada dođe do određenog događaja, ograničavanje opcija u izbornicima ovisno o drugim opcijama i slično. Za back end će biti korišten PHP odnosno razvojna cjelina Laravel, koja nudi potrebne koncepte upravo za razvoj ovakve vrste aplikacija, kao što su mnoštvo predefiniranih funkcija i objekata, podrška za rad s bazama podataka i pojednostavljen način vršenja upita i ulančavanja istih radi dobivanja željenih rezultata.

U nastavku rada bit će detaljnije objašnjene one tehnologije koje su korištene u ovome završnome radu, a iste su izabrane zbog svoje pristupačnosti, jednostavnosti te dobre dokumentacije dostupne na Internetu.

3. TEHNOLOGIJE ZA RAZVOJ WEB APLIKACIJA

Razvoj mrežnih aplikacija je proces dizajniranja, programiranja i testiranja aplikacija koje rade na mreži (web aplikacije). Neke od tehnologija koje se danas koriste za tu svrhu su:

- Ajax
- ASP.net
- CSS
- HTML
- Java
- Javascript
- Node.js
- PHP
- Python
- Ruby

Razvoj aplikacija je u današnje vrijeme napredovao do te razine da više nije potrebno učiti programske jezike od samog početka. Postoje razne razvojne cjeline (eng. *application frameworks*) koje objedinjuju svojstva programskih jezika, uvode nove koncepte te samim time olakšavaju razvoj aplikacija.

3.1. HTML

Prema [1], **HTML** (eng. *HyperTextMarkupLanguage*) je **opisni jezik** za izradu web stranica. Pomoću HTML-a stvaraju se hipertekstualni dokumenti, oblikuje se njihov sadržaj te se stvaraju hiperveze. Jednostavan je za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i činjenici da je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik.

3.2. CSS

Kao što je opisano u [1], **CSS** (eng. *Cascading Style Sheets*) je **stilski opisni jezik**, koji se rabi za opis prezentacije dokumenta napisanog pomoću markup (npr. HTML) jezika. Kako se web razvijao, prvotno su u HTML ubacivani elementi za definiciju prezentacije, ali je dovoljno brzo uočena potreba za stilskim jezikom koji će HTML osloboditi potrebe prikazivanja sadržaja i njegovog oblikovanja (čemu danas služi CSS). Drugim riječima, stil definira kako prikazati HTML elemente. CSS-om se uređuje sam izgled i raspored stranice. Stilski dokument ili stilski list (eng. *style sheet*) u CSS-u sastoji se od nekoliko pravila. Svako pravilo sastoji se od selektora i deklaracijskog bloka.

3.2.1. Bootstrap

Prema [1], **Bootstrap** je razvojna cjelina otvorenog koda koja služi za dizajn web stranica i web aplikacija. Sastoji se od predložaka za dizajn koji su bazirani na HTML-u i CSS-u te se odnose na tipografiju, forme, tipke, navigaciju te druge komponente sučelja, kao i JavaScript ekstenzije. Za razliku od većine drugih razvojnih cjelina za web, Bootstrap se odnosi samo na front-end razvoj. Svaka Bootstrap komponenta se sastoji od HTML strukture, CSS deklaracije, te u nekim slučajevima također i pripadajućeg JavaScript koda. Bootstrap nudi stilske listove koji omogućavaju osnovne stilske definicije za sve ključne HTML elemente.

3.3. JavaScript

Kao što je navedeno u [1], **JavaScript** je **skriptni programski jezik** koji se izvršava u web pregledniku na strani korisnika. Izvorno ga je razvila tvrtka Netscape kao primjenu ECMAScript standarda. JavaScript s **AJAX** (Asynchronous JavaScript and XML) tehnologijom omogućuje web stranicama komunikaciju sa serverskim programom, što čini web aplikacije interaktivnijim i lakšim za korištenje. JavaScript sadrži aplikacijska programska sučelja (API) za rad sa tekstom, poljima, datumima, regularnim izrazima, te manipulaciju **DOM**-om (eng. *Document Object Model*), što zapravo predstavlja sve elemente web stranice kao elemente hijerarhije koja ima strukturu stabla).

3.3.1. jQuery

Prema [1], **JQuery** je višeplatformska JavaScript biblioteka napravljena sa svrhom pojednostavljenja skriptiranja na strani klijenta. Istraživanja pokazuju da je jQuery najzastupljenija JavaScript biblioteka. Sintaksa je dizajnirana tako da olakšava pretraživanje dokumenta, odabir određenih elemenata u DOM-u, kreiranje animacija, rukovanje događajima (eng. *events*), te korištenje AJAX-a. U suštini, jQuery je biblioteka za manipuliranje DOM-om. Na primjer, može se koristiti za lociranje točno određenog elementa unutar HTML dokumenta, promjenu atributa, ili za kreiranje rukovatelja događajima koji omogućavaju elementima da reagiraju na određene događaje kao klik miša i slično.

3.4. PHP

Kao što je opisano u [1], **PHP** (eng. *PHP: Hypertext Preprocessor*) je **programski jezik** koji je baziran na C i Perl sintaksi, namijenjen prvenstveno programiranju dinamičnih web stranica. PHP je kao slobodni softver distribuiran pod PHP licencnim uvjetima. Ističe se širokom podrškom raznih baza podataka i internet protokola kao i raspoloživosti brojnih programskih biblioteka. PHP kod se može ugraditi u HTML strukturu, a može se i koristiti u kombinaciji sa raznim modulima za obrasce (eng. *templating engines*), sustavima za upravljanje sadržajem (CMS, eng. *Content Management System*), te razvojnim cjelinama. PHP kod procesira PHP prevoditelj (eng. *interpreter*) koji je implementiran kao modul na web poslužitelju ili preko standardnog klijent-poslužitelj sučelja (eng. *Common Gateway Interface*).

3.5. Laravel

Prema službenoj web stranici [2], **Laravel** je razvojna cjelina otvorenog koda (eng. *open source*) za web aplikacije bazirana na PHP-u odnosno Symfony razvojnoj cjelini. Razvio ga je Taylor Otwell s ciljem razvoja web aplikacija prateći „model - pogled - upravitelj“ (eng. *model - view - controller*) arhitekturu. Smatra se jednim od najpopularnijih PHP razvojnih cjelina te se cijeli izvorni kod Laravel-a nalazi na GitHub-u pod MIT licencom. U kratkom vremenu je postao popularan među PHP programerima zbog svoje jednostavnosti i „čistog koda”. Koristeći jasnu, izražajnu sintaksu, ova razvojna cjelina olakšava razvoj web aplikacija.

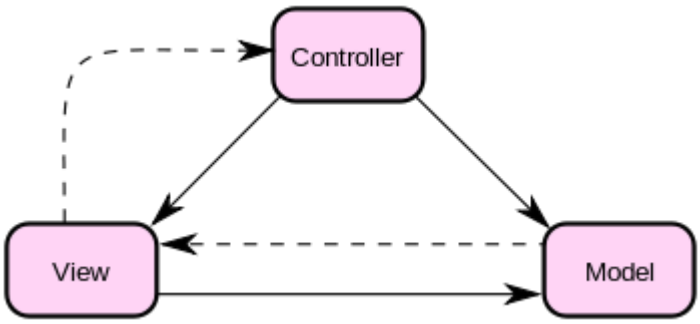
Može se reći da Laravel na vrlo efikasan i intuitivan način objedinjuje različite komponente i koncepte potrebne za razvoj i dizajn web aplikacija. Postoji velika količina **biblioteka** za Laravel koje proširuju funkcionalnost aplikacija, npr. „Intervention Image“ biblioteka koja nudi podršku za rad sa fotografijama.

Fasade (eng. *facades*) u Laravel-u nude sučelje prema klasama koje su dostupne u aplikacijama kao svojevrsne usluge. Pri instalaciji Laravel-a, također se instalira i velik broj fasada koje nude pristup različitim mogućnostima, te se tako proširuje funkcionalnost aplikacija. Fasadama se pristupa tako da se napiše ime fasade, dvostruka dvotočka, te ime funkcije kojoj se pristupa.

Rute (eng. *routes*) u Laravel-u služe za usmjeravanje HTTP zahtjeva prema određenim upraviteljima. Nalaze se u datoteci routes.php te je za njihovo definiranje potrebno specificirati tip zahtjeva (GET, POST, DELETE, PUT), web adresu i funkciju koju ruta poziva (ova funkcija se može definirati odmah pri definiciji same rute ili se ruta može usmjeriti na određenu funkciju unutar određenog upravitelja, korištenjem ključne riječi „uses“).

3.5.1. MVC arhitektura

Kao što je objašnjeno u [3], Model–View–Controller (MVC) je koncept softverske arhitekture. Koristi se u izradi aplikacija, te služi za odvajanje pojedinih dijelova aplikacije u komponente ovisno o njihovoj namjeni. **Model** se sastoji od podataka te poslovnih pravila i logike. U modelima se mogu stvoriti veze koje služe za definiranje logike između modela. **Pogled** je bilo kakav prikaz prethodno modeliranih podataka kao što je obrazac, tablica ili dijagram. Moguć je prikaz podataka kroz više različitih pogleda. Pogledi u Laravel-u su datoteke sa ekstenzijom .blade.php, te će se u ovom radu pri imenovanju pogleda podrazumijevati da iza imena slijedi ekstenzija „.blade.php“ (ekstenzija nije samo „.php“, nego „.blade.php“). U pogledima je vrlo praktično koristiti Laravel-ov modul za obrasce „**Blade**“, detaljnije opisan u [5], koji omogućava izvršavanje upita if, else itd., pristupanje fasadama te varijablama koje su poslane pogledu. **Upravitelj** prihvaća ulazne naputke (eng. *input*, najčešće zahtjevi korisnika) i pretvara ih u naredbe modelu ili pogledu. Ovakva arhitektura olakšava neovisan razvoj, testiranje i održavanje određene aplikacije.



Slika 3.1. Prikaz MVC arhitekture. Puna crta prikazuje direktnu povezanost dok isprekidana prikazuje indirektnu povezanost kroz korisnika

Osim podjele aplikacije u tri vrste komponenti, MVC dizajn utvrđuje interakcije između njih, kao što je vidljivo na slici 3.1.:

- Upravitelj može slati naredbe modelu kojima ažurira njegovo stanje (npr. uređivanje dokumenta). Također može slati naredbe u poglede kojima mijenja prikaz modela (npr. kretanje kroz dokument).
- Model dojavljuje sebi pridruženim pogledima i upraviteljima kada je došlo do promjene u njegovom stanju. Ove dojave omogućuju pogledu da prikaže obnovljeno stanje modela, a upravitelju promjenu dostupnog skupa naredbi.
- Pogled zahtjeva od modela informacije potrebne za stvaranje prikaza modela korisniku.

3.5.2. Artisan i Tinker

Prema [2], **Artisan** je naredbeno sučelje sadržano u Laravel-u. Nudi mogućnost korištenja velikog broja naredbi koje pomažu u stvaranju aplikacija. Moguće je vidjeti popis svih naredbi Artisan-a pomoću naredbe „php artisan list“. Svaka naredba ima i „help“ opciju koja prikazuje popis mogućih argumenata i opcija.

Sve aplikacije u Laravel-u sadrže i **Tinker**, interaktivno REPL (eng. *Read-eval-print-loop*) sučelje koje omogućava interakciju sa aplikacijom putem naredbi. Za pokretanje Tinker-a, potrebno je u razvojnom okruženju ili IDE-u (eng. *Integrated Develpoment Environment*) izvršiti naredbu „php artisan tinker“.

3.5.3. Eloquent

Kao što je opisano u [4], **Eloquent** je ORM (eng. *Object-relational mapping*) alat koji nudi vrlo jednostavnu implementaciju za rad s bazama podataka. Svaka tablica u bazi podataka ima pripadajući model koji se koristi za interakciju s istom. Modeli omogućavaju upite nad podacima bez korištenja klasičnih SQL upita, kao i ubacivanje podataka u tablice. Naredbom „php artisan make:model“ koja se pokreće u razvojnom okruženju se kreira model, a opcijom „–m“ se također kreira i migracija za taj model. **Migracije** su svojevrsni recepti za implementaciju tablica u bazi podataka. Pri kreiranju migracije, automatski se dodaju 2 parametra: „increments“ i „timestamps“. Parametar increments predstavlja primarni ključ, a timestamps predstavlja vremenske oznake koje se same ažuriraju pri kreiranju određenog reda u tablici.

4. REALIZACIJA ZADATKA U RAZVOJNOJ CJELINI LARAVEL

Budući da je aplikacija razvijena u okviru ovog rada napravljena za turističke vodiče, te radi dostupnosti široj publici, turističkim kompanijama itd., front-end dio je napisan na engleskom jeziku, ali će se u ovom radu prevesti odnosno objasniti na hrvatskom jeziku. Za testiranje je korišten virtualni poslužitelj TurnKey.

4.1. Korisnički profili

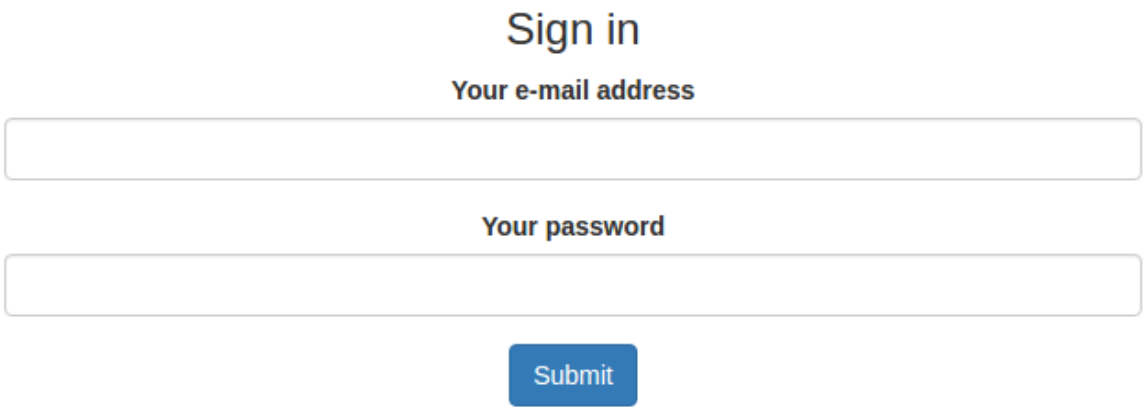
Da bi se uopće mogli definirati atributi modela korisnika, potrebno je kreirati sam model te migraciju, što je napravljeno izvršavanjem naredbe „php artisan make:model User -m“. Rezultat izvršavanja te naredbe su kreirana datoteka „User.php“, koja se odnosi na sam model, te migracija za model korisnika u koju su dodani atributi. Svaki će korisnik imati e-mail adresu, šifru, ime, prezime, opis te varijablu koja označava da li je korisnik administrator ili ne. Nakon pokretanja migracije naredbom „php artisan migrate“ stvara se tablica u MySQL bazi podataka koja sadržava prethodno navedene parametre. Radi demonstriranja funkcionalnosti ove aplikacije, pomoću Tinker-a je dodan novi korisnik u tablicu Users, naredbama prikazanim na slici 4.1..

```
php artisan tinker
$user = new App\User
$user->email = primjer@gmail.com
$user->password = bcrypt('pass')
$user->is_admin = true
$user->save()
```

Slika 4.1.: Naredbe Tinker-a za stvaranje novog korisnika

Sljedeći korak je omogućiti korisnicima prijavu i odjavu iz aplikacije, te kreirati korisničko sučelje. Za prijavu korisnika je kreiran novi pogled „login“ koji nasljeđuje pogled „master“ naredbom „@extends('layouts.master')“, te uključuje pogled „msg-block“ naredbom „@include('includes.msg-block')“. Pogled „master“ je kreiran kao predložak koji sadrži elemente koji će biti sadržani u više pogleda (npr. navigacijska traka, poveznice za uključivanje Bootstrap

stilskih listova itd.). U pogledu „msg-block“ se provjerava da li postoje pogreške (eng. *errors*) ili poruke poslane pogledu, te ako postoje, ispisuje ih se. Pogodno je uključiti pogled „msg-block“ u svakom pogledu koji zahtjeva mogućnost ispisivanja poruka korisniku (poput poruke da je došlo do pogreške, uspješno objavljene zadaće i slično).

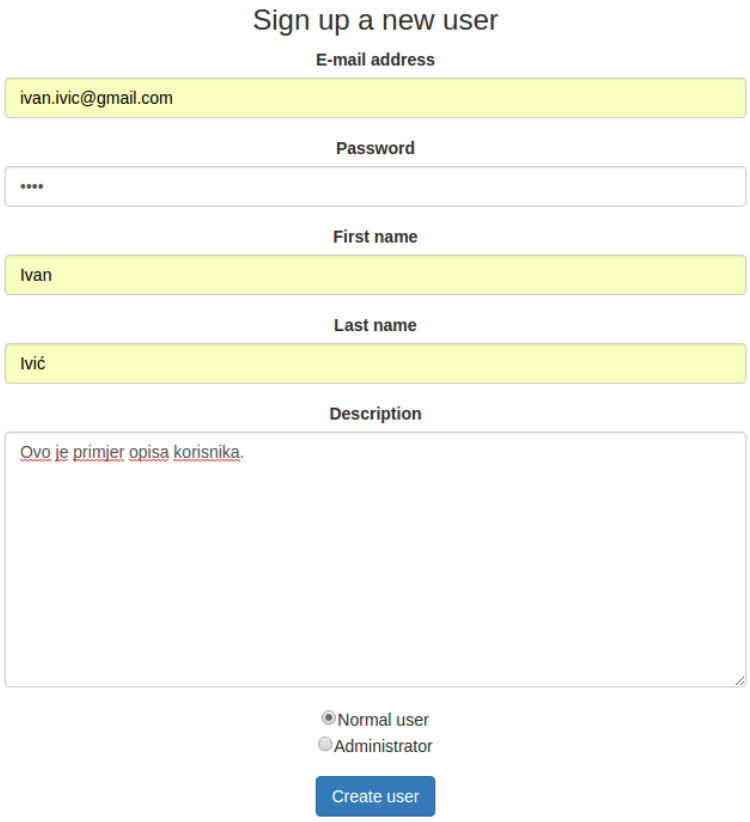


Slika 4.2.: Pogled za prijavu korisnika

Na slici 4.2. se može vidjeti forma za prijavu korisnika. Korisnik treba upisati e-mail adresu te šifru, te stisnuti tipku za prijavu „Sign in“. Ista poziva rutu „signin“ te dolazi do izvršavanja pripadajuće funkcije „postSignIn“ u upravitelju „UserController“. Parametri korisničkog računa se prvo provjeravaju metodom „validate“ a zatim se poziva funkcija „attempt“ iz fasade „Auth“, koja kao parametre prima e-mail adresu i šifru koji su poslani u HTTP zahtjevu. Ako je prijava uspješna, korisnika se metodom „redirect“ usmjerava na pogled sa objavama „posts“, a ako nije, usmjerava ga se na pogled za prijavu uz odgovarajuću poruku. Nakon prijave, korisnik može pristupati pogledima za kreiranje novog korisnika (samo ako je administrator), kreiranje objave, postavljanje slika ili videozapisa te ažuriranje vlastitog profila.

Klikom na tipku „Create user“ učitava se pogled za kreiranje novog korisnika koji se može vidjeti na slici 4.3.. Sastoji se od tekstualnih polja u koja se upisuju parametri novog korisnika. Klikom na tipku „Create User“ poziva se odgovarajuća ruta i funkcija za kreiranje korisnika. Metodom „validate“ se ograničavaju dopušteni unosi, kreira se nova instanca objekta User, pridružuju se parametri poslani HTTP zahtjevom, te se objekt sprema kao novi red u tablici

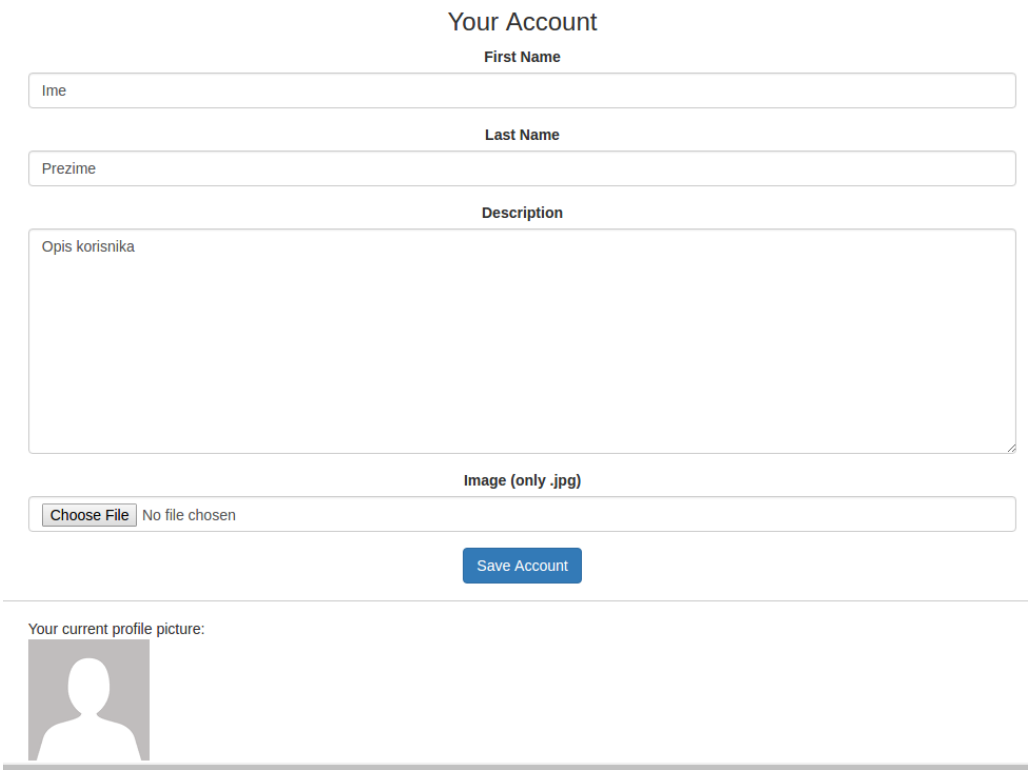
funkcijom „save“. Bitno je naglasiti da se šifra ne sprema u obliku u kojem je unesena, nego se prvo enkriptira funkcijom „bcrypt()“.



Slika 4.3.: Pogled za kreiranje korisnika

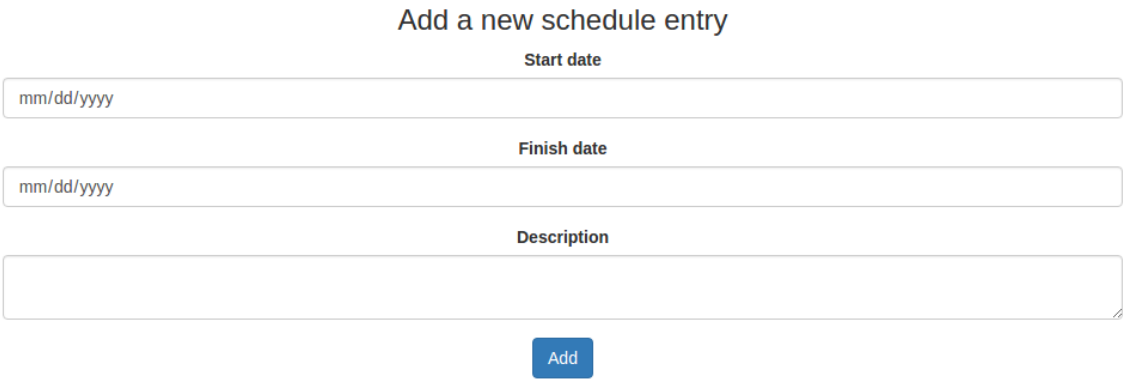
4.1.1. Ažuriranje profila korisnika i dodavanje događaja u raspored

Klikom na tipku „account“ učitava se istoimeni pogled u kojem su tekstualna polja za ime, prezime i opis vodiča. Pri učitavanju samog pogleda, polja se popunjavaju trenutnim vrijednostima atributa, tako da korisnik može napraviti izmjene nad podacima bez ponovnog upisivanja. Također je moguće postaviti profilnu sliku korisnika. Ako postoji profilna slika, ona se prikazuje u pogledu, a isti je vidljiv na slici 4.4..



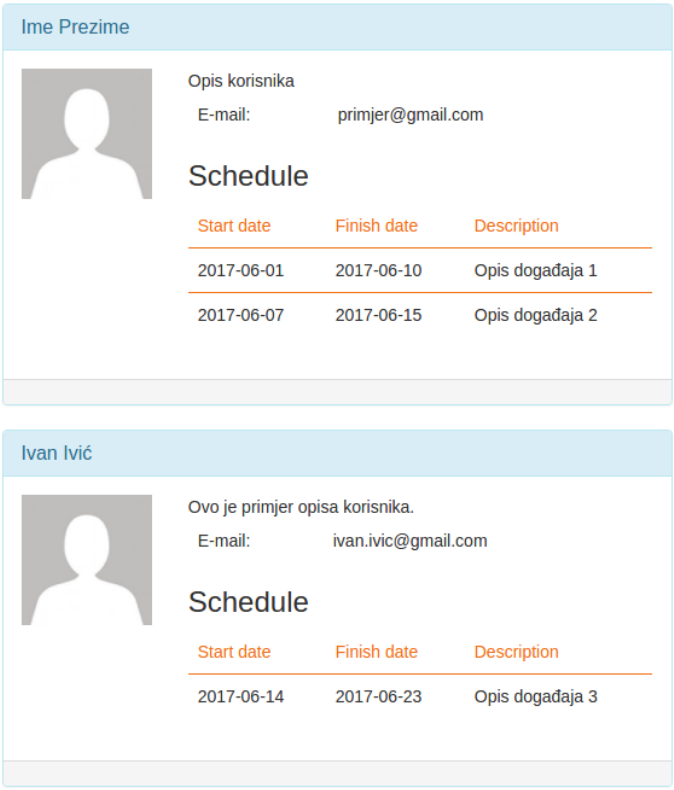
Slika 4.4.: Pogled za ažuriranje korisničkog profila

Da bi korisnici mogli kreirati te ažurirati raspored, potrebno je stvoriti novi model te definirati vezu između modela korisnika i modela događaja. To je napravljeno pomoću naredbe „php artisan make:model Date -m“. U migraciji su definirani atributi: dvije varijable tipa „date“ koje označavaju početak i kraj događaja, te varijabla tipa „string“ koja označava opis događaja) Zatim su u modelima „User.php“ i „Date.php“ definirane veze, korištenjem funkcije „hasMany()“ u modelu korisnika (jedan korisnik može imati više događaja u rasporedu), odnosno „belongsTo()“ u modelu događaja (jedan događaj pripada samo jednom korisniku). Na taj način je korisnicima omogućeno dodavanje događaja, prikazano na slici 4.5..



Slika 4.5.: Pogled za dodavanje događaja

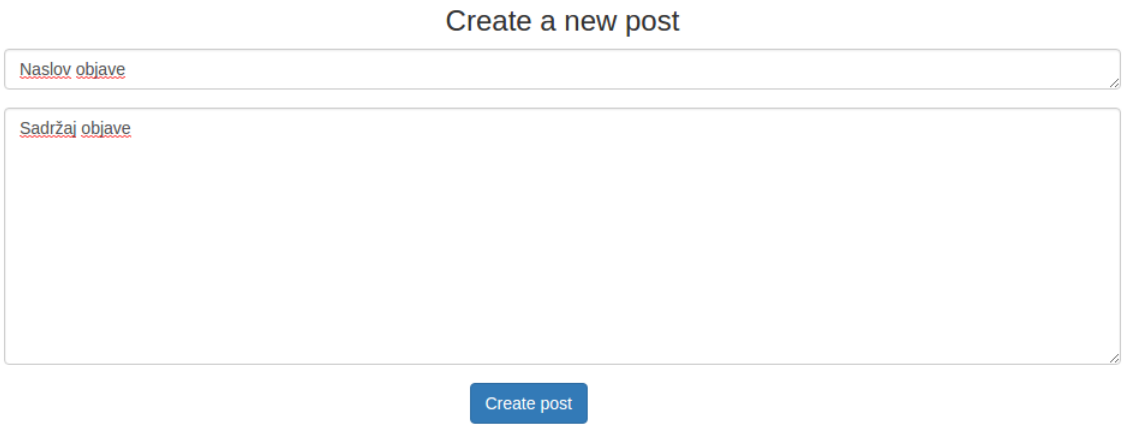
Klikom na tipku „Guides“ može se pristupiti pogledu prikazanom na slici 4.6., u kojem se prikazuje profil svakog postojećeg korisnika. Ovdje se mogu dobiti informacije o korisnicima (ime, prezime, opis, profilna slika te raspored).



Slika 4.6.: Pogled za prikazivanje profila korisnika

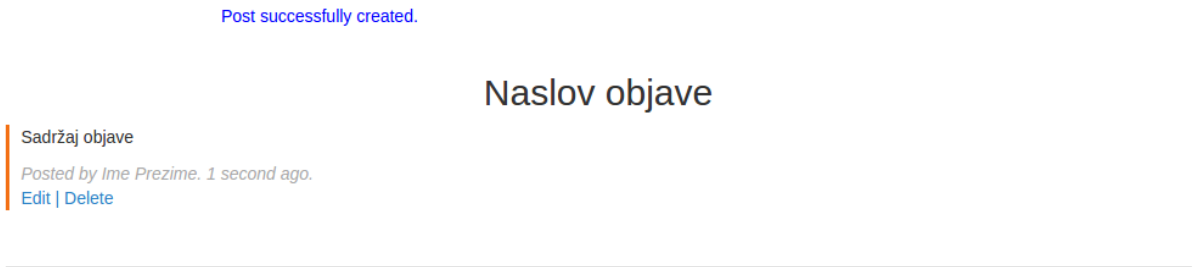
4.2. Objavljivanje sadržaja

Prateći istu logiku kao i kod modela događaja, kreiran je i model objave, te su u migraciji dodani atributi naslov i sadržaj. Klikom na tipku „Create Post“ učitava se pogled za kreiranje objave vidljiv na slici 4.7.. Tu korisnik može upisati naziv objave i sadržaj objave.



Slika 4.7.: Pogled za kreiranje objave

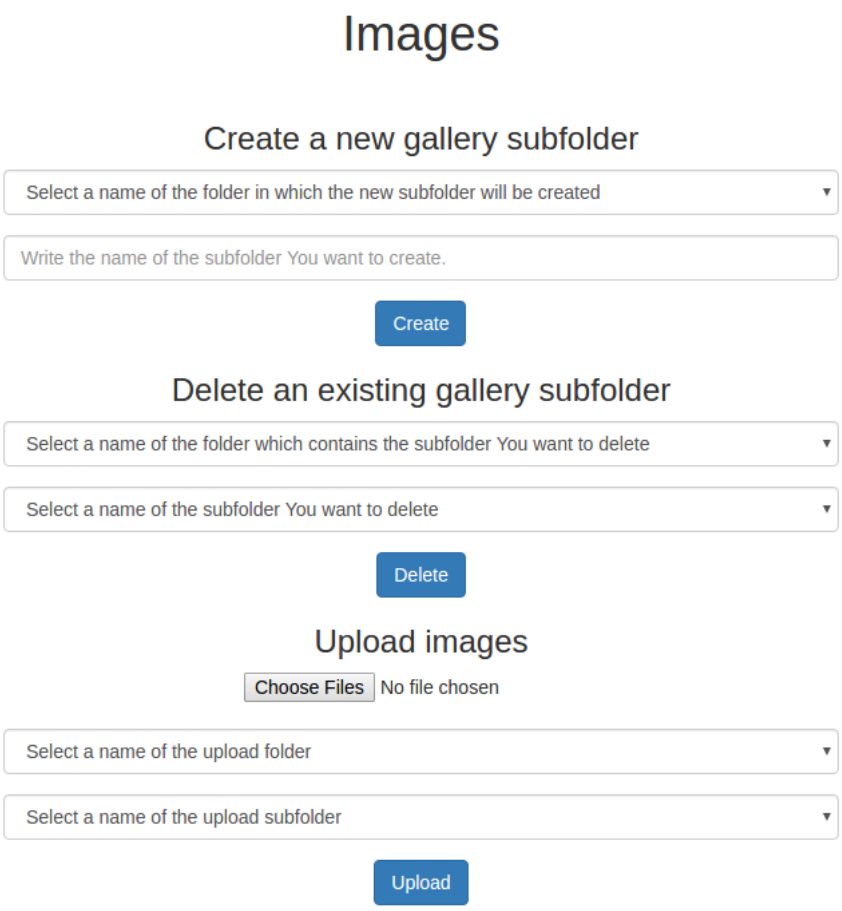
Tipka „Create Post“ poziva odgovarajuću rutu i funkciju u upravitelju za objave „PostController“, te se kreira nova objava i sprema u tablicu. U pogledu „Posts“ se dohvaćaju sve postojeće objave te ih se prikazuje kao na slici 4.8.. Uz svaku objavu se ispisuje i vrijeme kada je objavljena. Također su implementirane i opcije „Edit“ i „Delete“ kojima se objava može izmijeniti odnosno obrisati iz baze, a vidljive su samo ako je korisnik vlasnik objave ili administrator. Izmjene su realizirane pomoću Javascript elementa „modal“ koji se pojavljuje kada korisnik stisne tipku „Edit“.



Slika 4.8.: Pogled sa objavama

4.3. Postavljanje slika i videozapisa

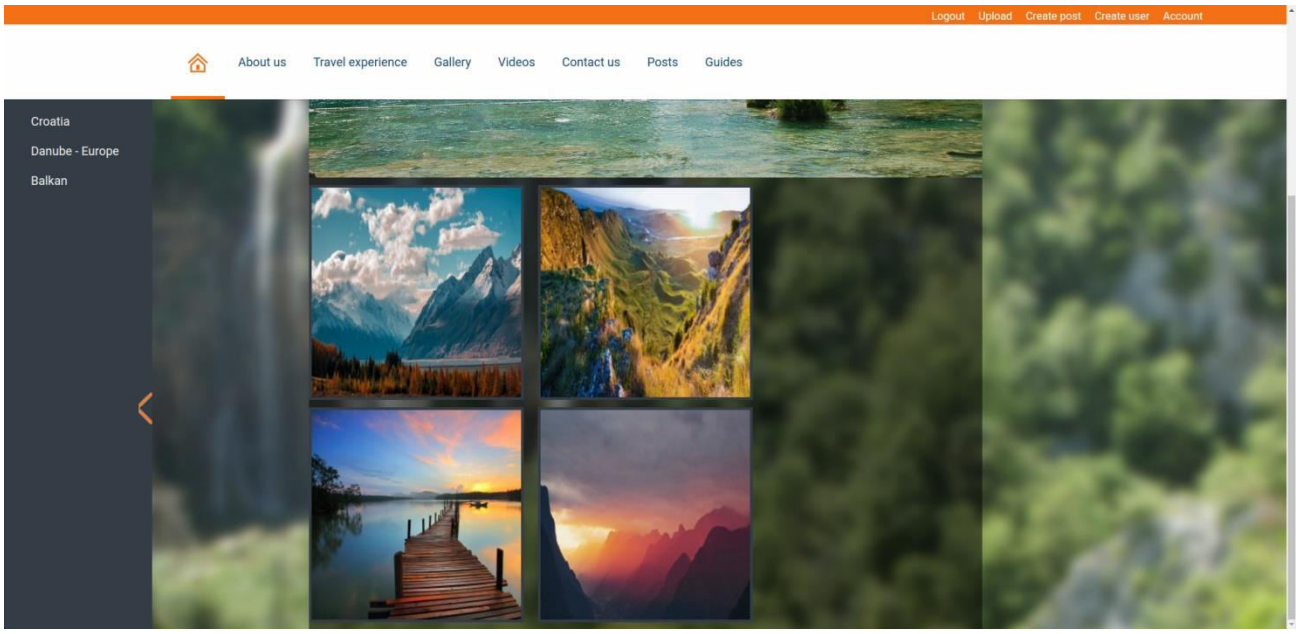
Ovaj dio CMS-a je realiziran na jednak način za slike i videozapise, pa će se ovdje detaljno objasniti postupak samo za slike. Naime, kreirani su pogledi za postavljanje (upload) slika i videozapisa na poslužitelj, te pogled u kojem će se prikazivati slike odnosno videozapisi.



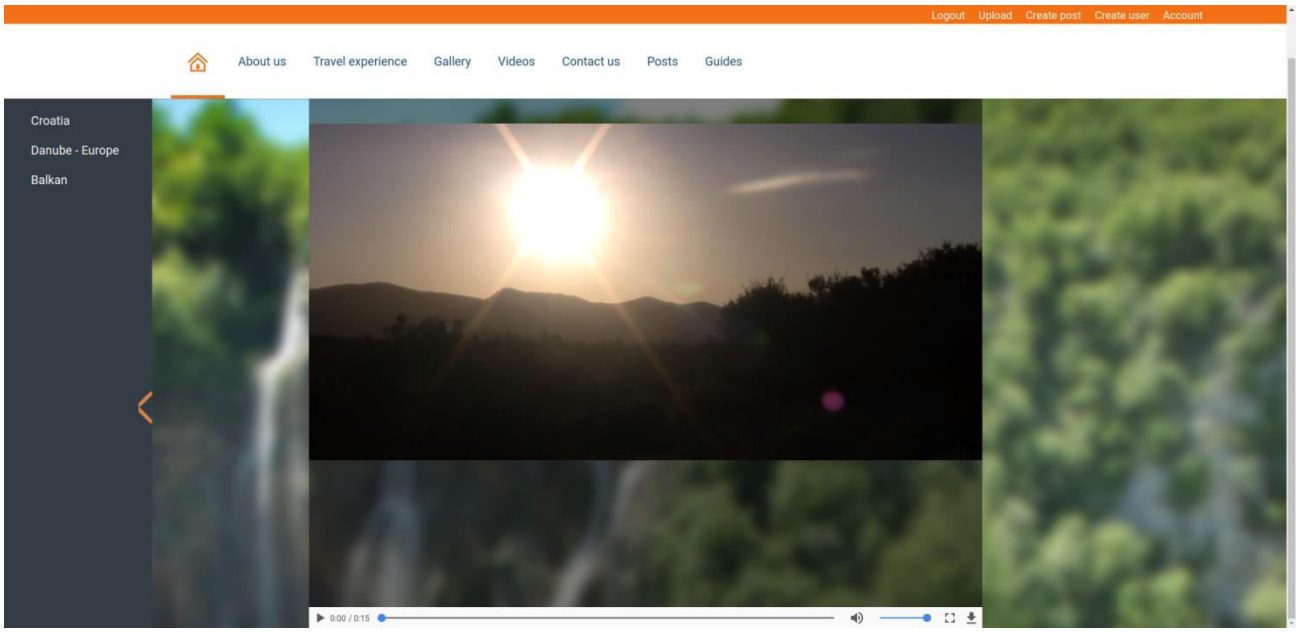
Slika 4.9.: Pogled za dodavanje slika i videozapisa

Preko sučelja prikazanog na slici 4.9., korisnik može kreirati ili obrisati potkategoriju u galeriji, a u postojeće potkategorije može dodavati slike. HTML element za dodavanje slika je realiziran tako da se može odjednom dodati više slika, a to je omogućeno deklaracijom „enctype="multipart/form-data“ u HTML formi. Kada korisnik klikne na tipku „Upload“, poziva se funkcija u upravitelju „UploadControler“ koja dohvaća sve datoteke, prolazi kroz njih u petlji, te svaku sprema u određeni direktorij koji je specificirao korisnik. Isto tako, od svake slike se pravi

umanjena slika koja služi za prikazivanje u galeriji (eng. *thumbnail*). Umanjivanje slika je postignuto funkcijom „resize“, koja je dio biblioteke „Intervention Image“.



Slika 4.10.: Pogled za prikazivanje slika



Slika 4.11.: Pogled za prikazivanje videozapisa

Kao što je vidljivo na slikama 4.10. i 4.11., sa lijeve strane galerije se nalazi izbornik sa kategorijama. Tu se može odabrati određena kategorija, što rezultira pojavljivanjem novog izbornika sa potkategorijama (npr. u kategoriji „Hrvatska“ mogu postojati potkategorije „Dubrovnik“ ili „Istra“ i sl.). Odabiranjem određene potkategorije se prikazuju slike ili videozapisi iz te potkategorije. Kategorije su predefinirane, a u svaku korisnik može dodavati potkategorije po svom izboru u koje može postavljati sadržaj. Ovaj sustav je realiziran strukturom direktorija koja se nalazi u javnom direktoriju (public) na poslužitelju.

5. OPIS I VREDNOVANJE PROGRAMSKOG RJEŠENJA

Potrebne funkcionalnosti su riješene na sljedeći način: za navigaciju se koriste tipke u gornjem desnom dijelu sučelja, a pomoću njih korisnik može pristupati određenim dijelovima aplikacije. Za primanje podataka od korisnika su korištene HTML forme sa pripadajućim poljima za tekst, koje su estetski uređene pomoću CSS-a. Sami korisnici su podijeljeni u dvije skupine: administratori i obični korisnici, a ovisno o skupini neke mogućnosti će biti ograničene: obični korisnici će moći izvršavati izmjene te brisati samo vlastite objave, dok će administratori moći izvršavati izmjene te brisati sve objave, neovisno o tome koje mkorisniku objava pripada. Također, administratori će, za razliku od običnih korisnika, imati mogućnost kreiranja novih korisnika te brisanja postojećih. Za objavljivanje turističkih ponuda se koristi model objave, a isti se može koristiti i u druge svrhe (obavijesti, informacije o vodičima itd.). Korisnici imaju svoje profile, koje mogu ažurirati zajedno sa svojim rasporedom, koji se sastoji od redova u tablici gdje svaki red predstavlja jedno razdoblje definirano datumima, s pripadajućim opisom. Također, implementiran je i sustav za objavljivanje slika i video zapisa s mogućnosti dodavanja (i brisanja) kategorija odnosno direktorija u koje se postavlja sadržaj. Za realizaciju ovog zadatka bilo je potrebno je koristiti i bazu podataka u koju su spremljeni podaci o korisnicima, objavama i rasporedima.

Tijekom testiranja CMS-a razvijenog u okviru ovog rada, nisu uočene greške koje bi mogle bitno utjecati na rad te je svaka mogućnost radila upravo onako kako je i opisano u ovom radu.

6. ZAKLJUČAK

CMS je web aplikacija koja omogućava upravljanje sadržajem web stranice. To se najčešće odnosi na kreiranje novog sadržaja (objave, ponude, proizvodi, usluge), ažuriranje starog sadržaja, te postavljanje multimedijskih sadržaja na web stranicu.

Zadatak ovog završnog rada je bio razviti CMS za primjenu u turistički orijentiranim web stranicama te omogućiti korisnicima (turističkim vodičima) kreiranje i ažuriranje vlastitog profila, kreiranje i ažuriranje rasporeda, objavljivanje tekstualnog sadržaja, te objavljivanje slika i video zapisa. CMS je uspješno razvijen i implementiran na računalu pomoću razvojnog okruženja Laravel. Korisnici koriste e-mail adresu i lozinku za prijavu, te tada mogu pristupati mogućnostima CMS-a, koristeći navigacijsku traku. Korisnici unose podatke preko HTML formi u koje se upisuje tekst. Podaci koji definiraju pojedine korisničke profile i objave su spremljeni u MySQL bazi podataka.

CMS bi se mogao poboljšati dodavanjem novih funkcionalnosti (npr. mogućnost rezervacija), složenijim tipovima objava specijaliziranim za određene sadržaje (npr. mogućnost dodavanja slike ili video zapisa u turističku ponudu), te optimizacijom za tražilice.

LITERATURA

[1] J.Wales, L.Sanger, Wikipedia, Wikimedia Foundation, 2001., dostupno na: https://en.wikipedia.org [25.6.2017.]

[2] T.Otwell, Laravel, 2016., dostupno na: https://laravel.com [27.6.2017.]

[3] Laracasts, Laravel fundamentals, 2015., dostupno na: https://laracasts.com/series/laravel-5-fundamentals [27.6.2017.]

[4] M. Schwarzmüller, Laravel PHP Create a social network, 2016., dostupno na: https://www.youtube.com/watch?v=_dd4-HEPejU&list=PL55RiY5tL51oloSGk5XdO2MGjPqc0BxGV [27.6.2017.]

[5] Tutsplus, PHP to Laravel essentials, 2014., dostupno na: https://tutsplus.com/bundles/php-to-laravel-essentials [27.6.2017.]

SAŽETAK

U današnje vrijeme Internet je preplavljen web sjedištima koje nude turistički sadržaj. Problem postojećih stranica je nedostatak jednostavnog upravljanja sadržajem istih, kao i izrade rasporeda za turističke vodiče. U ovom radu predstavljena je izrada jednog CMS-a za primjenu u turističkim web stranicama, korištenjem sljedećih tehnologija: HTML, CSS, PHP i MySQL. Na osnovu spomenutih tehnologija, uspješno je razvijen CMS koji omogućuje jednostavno upravljanje tekstualnim i multimedijskim sadržajem. Tekstualni sadržaj je prikazan u obliku objava, a multimedijski sadržaj u obliku galerije koja može sadržavati slike i video zapise. U radu je također prikazano kako navedeno web rješenje funkcionira, te su objašnjeni najbitniji koncepti korišteni u izradi CMS-a. Opisani su pojedini funkcijski dijelovi navedenog sustava, te je isti na kraju vrednovan kao cjelina.

Ključne riječi: mreža, CMS, Laravel, korisnik, PHP

ABSTRACT

DEVELOPMENT OF A CM SYSTEM FOR WEBSITED WITH TOURIST CONTENT

In today's time, the Internet is flooded with web sites that offer tourist content. The problem with existing web sites is the absence of simple content management, as well as creating schedules for tour guides. This paper presents the development of a CMS, designed for tourist sites application, using the following technologies: HTML, CSS, PHP and MySQL. Based on above mentioned technologies, a CMS, which enables simple management of textual and multimedia content, was successfully developed. Textual content is shown using the form of posts, and multimedia content is shown in galleries which can contain images and videos. The paper also shows how the mentioned web solution functions, and the most important concepts used in the development of this CMS are also explained. Each functional part of this system has been described separately and in the end of the paper the system was evaluated as a whole.

Key words: web, CMS, Laravel, user, PHP

ŽIVOTOPIS

Ivan Klešić rođen je u Osijeku, Republika Hrvatska, 24. studenog 1995. Godine. Pohađao je osnovnu školu „Vladimir Becić“ u Osijeku. U osmom razredu je sudjelovao na državnoj smotri radova iz robotike.

Nakon osnovne škole, 2010. godine upisuje Elektrotehničku i prometnu školu Osijek te završava sva četiri razreda sa odličnim uspjehom. U drugom razredu srednje škole je sudjelovao na državnom natjecanju iz elektrotehnike. Maturirao je 2014. godine.

2014. godine upisuje Elektrotehnički fakultet u Osijeku, smjer računarstvo. Nakon završetka preddiplomskog studija, planira upisati diplomski studij na istom fakultetu.

Vlastoručni potpis

_____

Ivan Klešić