# Android aplikacija za pregled ponuda zabavnih sadržaja

**Pejić, Petar**

**Undergraduate thesis / Završni rad**

**2019**

*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:200:703332

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-20**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

# IZJAVA O ORIGINALNOSTI RADA

**Osijek, 12.09.2019.**

| | |
|---|---|
| **Ime i prezime studenta:** | Petar Pejić |
| **Studij**: | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. studenta, godina upisa:** | R3969, 19.09.2018. |
| **Ephorus podudaranje [%]:** | 13 |

Ovom izjavom izjavljujem da je rad pod nazivom**: Android aplikacija za pregled ponuda zabavnih sadržaja**

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

i sumentora Izv. 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:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

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

**Osijek, 04.09.2019.**

**Odboru za završne i diplomske ispite**

# Prijedlog ocjene završnog rada

| | |
|---|---|
| **Ime i prezime studenta:** | Petar Pejić |
| **Studij, smjer:** | Preddiplomski sveučilišni studij Računarstvo |
| **Mat. br. studenta, godina upisa:** | R3969, 19.09.2018. |
| **OIB studenta:** | 39213705040 |
| **Mentor:** | Prof. dr. sc. Dominika Crnjac Milić |
| **Sumentor:** | Izv. prof. dr. sc. Krešimir Nenadić |
| **Sumentor iz tvrtke:** | Ivan Sudar |
| **Naslov završnog rada:** | Android aplikacija za pregled ponuda zabavnih sadržaja |
| **Znanstvena grana rada:** | **Informacijski sustavi (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: 3 bod/boda<br>Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda<br>Jasnoća pismenog izražavanja: 3 bod/boda<br>Razina samostalnosti: 3 razina |
| **Datum prijedloga ocjene mentora:** | 04.09.2019. |
| **Datum potvrde ocjene Odbora:** | 11.09.2019. |

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

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

**Sveučilišni preddiplomski studij računarstva**

# ANDROID APLIKACIJA ZA PREGLED PONUDA ZABAVNIH SADRŽAJA

**Završni rad**

**Petar Pejić**

**Osijek, 2019**

# SADRŽAJ

**1. UVOD** ..... 1

**1.1. Zadatak završnog rada** ..... 2

**2. VAŽNOST TURIZMA ZA REPUBLIKU HRVATSKU** ..... 2

**3. POVIJESNI RAZVOJ PRIMJENE MOBILNIH APLIKACIJA** ..... 5

**3.1. ANALIZA TRENUTNE PONUDE ANDROID APLIKACIJA VEZANIH ZA PONUDU ZABAVNOG SADRŽAJA** ..... 6

**4. FRONT-END I BACK-END** ..... 6

**4.1. Frot-end** ..... 6

**4.2. Back-end** ..... 8

**5. TEHNOLOGIJE KORIŠTENE ZA IZRADU APLIKACIJE** ..... 10

**5.1. Ionic – razvojno okruženje mobilnih aplikacija** ..... 10

**5.1.1. Instalacija Ionica** ..... 11

**5.2. Angular – razvojno okruženje web stranica** ..... 12

**5.3. Baza podataka** ..... 12

**5.3.1. NoSQL** ..... 13

**5.3.2. MongoDB** ..... 13

**6. IZRADA APLIKACIJE** ..... 15

**6.1. Ideja i osnovni koncept** ..... 15

**6.2. Izrada mobilne aplikacije** ..... 17

**6.3. Izrada web sučelja** ..... 23

**7. ZAKLJUČAK** ..... 28

**LITERATURA** ..... 29

**SAŽETAK** ..... 31

**ABSTRACT** ..... 32

**ŽIVOTOPIS** ..... 33

# 1. UVOD

Današnji svijet gotovo je nemoguće zamisliti bez mobilnih aplikacija. Skoro svaka osoba na svijetu posjeduje barem jedan mobilni uređaj na kojemu ima instalirane aplikacije te na njih troši svoje slobodno vrijeme. Koriste se za uređivanje slika, igranje igrica, organiziranje planova puta, zabavu, edukacijske svrhe i mnoge druge stvari. Lako je zaključiti da su vrlo važan dio ljudskih života.

Aplikacije se mogu definirati kao softverski alati koji pružaju razne pogodnosti i olakšavaju život u različitim aspektima. Da bi se utvrdila raširenost njihovog korištenja dovoljno je otići na internetski portal koji se bavi tehnologijom i pogledati naslove članaka. "*Smartphone aplikacija pouzdano dijagnosticira bolesti dišnog sustava djece*", "*Španjolska nogometna liga putem aplikacije prisluškivala gledatelje*", "*Aplikacije za bolje spavanje donose-nesanicu*" [1] su samo neki naslovi članaka koji se mogu pročitati te se vrlo brzo shvati koliko su aplikacije ukorijenjene u naš svakodnevni život.

Turizam je jedna od najvažnijih, ali i uspješnijih gospodarskih grana Republike Hrvatske. O tome dovoljno svjedoči činjenica da je početkom 2019. godine Republika Hrvatska osvojila drugo mjesto za najbolju europsku turističku destinaciju prema *Travvy Awards* [2]., dok je britanski magazin *Business Destinations* Republici Hrvatskoj dodijelio nagradu za najbolju destinaciju za poslovni turizam u Europi u 2018. godini [3].

Aplikacija predstavljena u radu ima cilj poboljšati turističku ponudu zabavnog sadržaja u svim dijelovima Republike Hrvatske. Njezino korištenje nudi mogućnost unaprijeđivanja poslovanja gospodarskih subjekata koji se bave pružanjem takve vrste usluga.

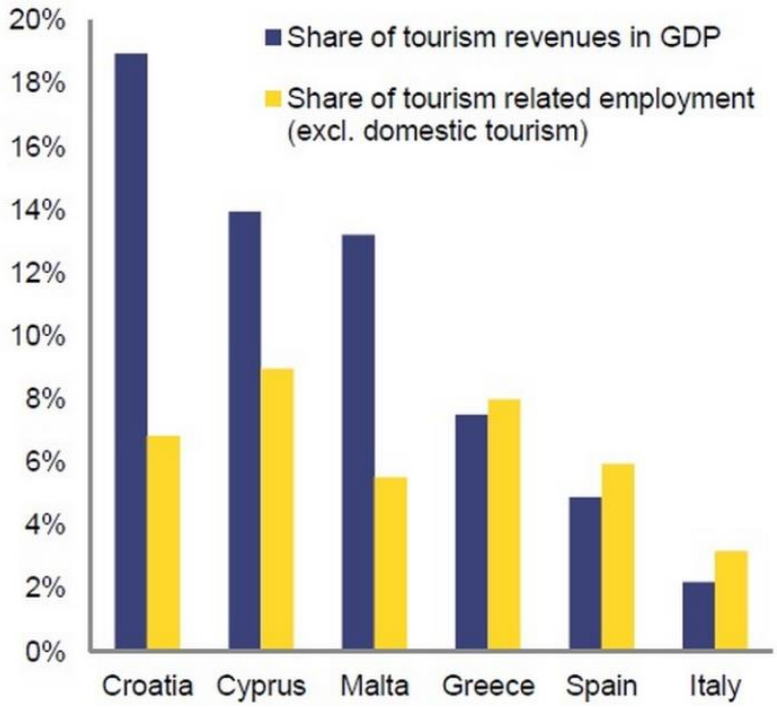
U drugom poglavlju rada opisana je važnost turizma za Republiku Hrvatsku. Cijelo poglavlje potkrijepljeno je statističkim brojkama dobivenih s različitih izvora. Nakon toga, u trećem poglavlju slijedi dio vezan za mobilne aplikacije i trenutnu ponudu sličnih aplikacija. Četvrto poglavlje odnosi se na opis glavnih značajki backend i frontend programiranja. Prije samog opisa izrade, u petom poglavlju, opisane su tehnologije kojima se aplikacija razvila, Ionic i Angular, te korištena baza podataka MongoDB. Za kraj, u šestom poglavlju opisan je postupak izrade aplikacije.

## 1.1. Zadatak završnog rada

Ovim radom je potrebno napraviti istraživanje trenutne ponude android aplikacija pomoću kojih korisnik može pregledati ponudu zabavnih sadržaja koja nude pojedina mjesta u Republici Hrvatskoj. Na osnovi rezultata pretrage postojećih aplikacija treba razviti android aplikaciju pomoću koje bi svi korisnici mogli saznati pojedine detalje o ponuđenim zabavnim sadržajima na osnovi lokacije i vremena. Organizatorima događaja i vlasnicima objekata koji nude zabavne sadržaje omogućiti postavljanje obavijesti o njihovim ponudama zabave ako su se registrirali za korištenje aplikacije. Predložiti model i komercijalizaciju razvijene aplikacije.

# 2. VAŽNOST TURIZMA ZA REPUBLIKU HRVATSKU

Kako bi se provjerilo je li Republika Hrvatska zadržala ugled kao top destinacija za turiste, najbolje je pogledati statističke podatke koji se odnose na prethodne godine te ih usporediti s drugim mediteranskim destinacijama budući da se najveći prihodi od turizma ostvaraju ljeti. Usporedba mediteranskih država vidljiva je na slici 2.1.

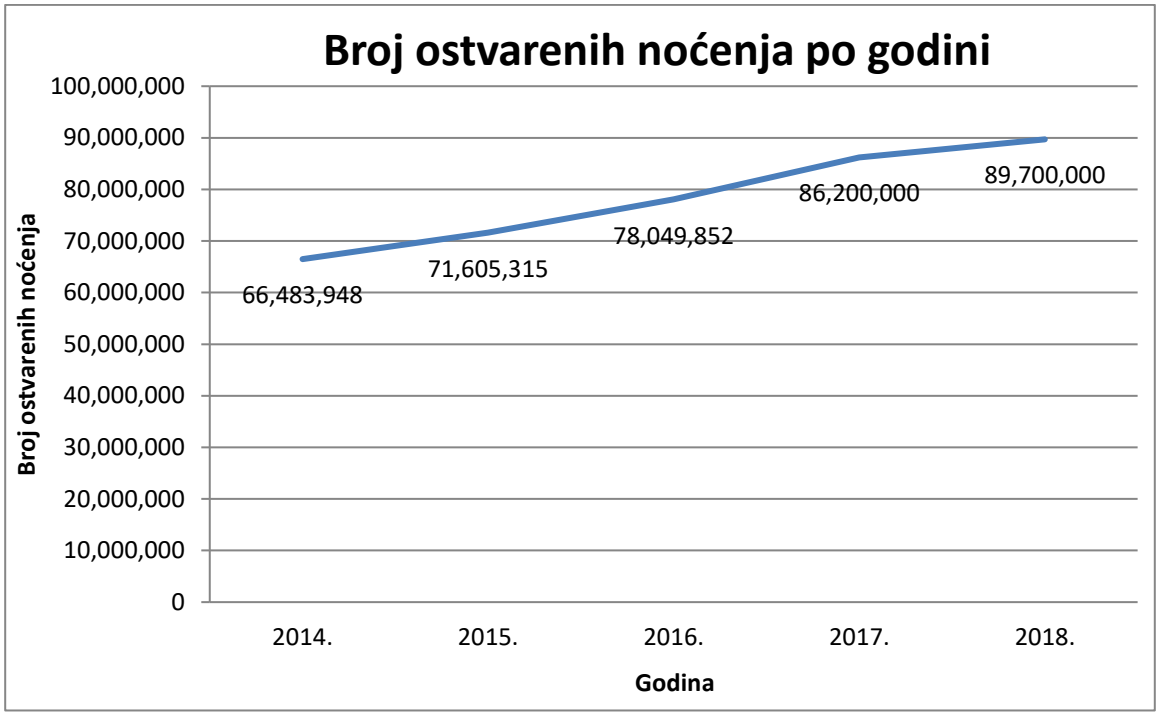



**Sl. 2.1**. Prihodi od turizma i zaposlenost,2016. godina [4]

Prema podacima iz platne bilance, prihodi od stranih turista činili su oko 19% BDP-a u 2016., odnosno 35% ukupnog prihoda od izvoza roba i usluga. Turizam je, isključujući domaću komponentu, odgovoran za 7% ukupnog broja zaposlenih u RH [5].

Noćenja je, u 2018. godini, više od 105 milijuna i četiri posto više nego 2017. godine. Stranaca je, uobičajeno, bilo neusporedivo više, točnije 17, 3 milijuna i ostvarili su 92, 6 milijuna noćenja, što je porast od šest, odnosno četiri posto [6].

Porast je zabilježen i u razdoblju od siječnja do kraja rujna 2018. kada su prihodi od stranih gostiju iznosili 9,24 milijarde eura, što je 6,4 **%** više u odnosu na isto razdoblje u 2017. godini[7]. Broj ostvarenih noćenja proteklih godina vidljiv je na slici 2.2.




**Sl. 2.2.** Broj noćenja u proteklih pet godina u Republici Hrvatskoj [8]

Kao što je vidljivo iz brojki svake godine broj turista se povećava što znači da postoji veći broj ljudi koji su ciljana skupina za ovu vrstu aplikacije. Kako bi turistima olakšali boravak u nepoznatoj sredini, potrebno im je pružiti dovoljno informacija o svemu što se događa ondje. Budući da trenutno ne postoji niti jedna poznata i široko prihvaćena platforma za pružanje pregleda zabavnog sadržaja, prostor za popularizaciju aplikacije koja bi to omogućavala je ogroman te bi zasigurno podignula hrvatski turizam na jedan viši nivo. Osobito bi značila promociji ponuđenih sadržaja za

osobe mlađe i srednje životne dobi koje se tehnologijom služe vezano za sve segmente svoga života. Stoga, takva vrsta aplikacije bila bi idealna za pravovremeno informiranje jer bi korisnici, na jednom mjestu i u svega par klikova, mogli provjeriti koji se događaji u narednom periodu održavaju u određenom gradu u kojem borave.

Trenutno je sadržaj uz pomoć informacijsko komunikacijske tehnologije (ICT-ja) moguće reklamirati putem Google oglasa, plaćenih oglasa na Facebooku ili na drugim društvenim mrežama, no pravo je pitanje koliko će ljudi iz ciljane skupine vidjeti taj oglas.
U skladu sa svim navedenim javlja se potreba za ICT potporom u vidu aplikacije čiji će razvoj biti opisan u daljnjem tekstu.

# 3. POVIJESNI RAZVOJ PRIMJENE MOBILNIHAPLIKACIJA

U srpnju 2008. godine Apple je objavio 552 mobilne aplikacije koje su se mogle preuzeti na iPhone uređajima. U samo tjedan dana, korisnici iPhonea preuzeli su oko deset milijuna aplikacija. U jesen iste godine, Apple je dobio konkurenciju -Android Market. Google je postao velika konkurencija Apple-u jer je uz Android Market, gdje se moglo besplatno skidati aplikacije, pružao usluge kupovine glazbe, e-knjiga i filmova. Godine 2012. nastaje Google Play koji povezuje Android Market, Google Music i Google eBookStore [9].

U 2014. godini je preuzeto i instalirano gotovo 139 milijardi mobilnih aplikacija, što predstavlja veliki rast u odnosu na samo pet godina starije podatke, kada ih se u 2009. godini preuzelo 2.5 milijardi. Broj preuzetih aplikacija za 2017. i 2018. godinu iznosi 178 odnosno 205 milijardi [10]. Iz ovih podataka vidljivo je kako se broj preuzimanja aplikacija i dalje povećava, no razlika je iz godine u godinu sve manja.

Prema podacima koje je obavila tvrtka IDC – *International Data Corporation*, tržište je zastupljeno operacijskim sustavima prema prikazu na slici 3.1., gdje se nalaze predviđanja i za naredne godine.

| Year | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 |
|---|---|---|---|---|---|---|---|
| Android | 85.1% | 85.1% | 86.7% | 86.6% | 86.9% | 87.0% | 87.1% |
| iOS | 14.7% | 14.9% | 13.3% | 13.4% | 13.1% | 13.0% | 12.9% |
| Others | 0.2% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **TOTAL** | **100.0%** | **100.0%** | **100.0%** | **100.0%** | **100.0%** | **100.0%** | **100.0%** |

**Sl. 3.1.** Tržišni udjeli mobilnih aplikacija prema operacijskim sustavima [11]

## 3.1. Analiza trenutne ponude android aplikacija vezanih za ponudu zabavnih sadržaja

Prije pristupa rješavanju ovog problema provedeno je istraživanje prisutnosti sličnih proizvoda na tržištu. Kao dobar primjer sličnog proizvoda nađen je proizvod *Eventos* čiji su autori studenti s Fakulteta elektrotehnike i računarstva Zagreb (FER-a).

Eventos je android aplikacija koja nudi širok spektar događaja poput festivala, izložbi, tečaja yoge, škole plese ili pak stand-up večeri, a uz onu mobilnu inačicu, postoji i verzija za web za sve one koji su više privrženi iOS, Windows Phone ili nekoj drugoj platformi [12]. Nažalost, navedena aplikacija više nije dostupna korisnicima na Google Play trgovini.

Također postoji aplikacija *Meetup: Find events near you*. Njezino korištenje nije usko vezano za Republiku Hrvatsku, već za cijeli svijet, a time ne može osigurati dovoljno brz protok informacija nužnih kod ovakvog pružanja usluga.

# 4. FRONT-END I BACK-END

U gotovo svim programerskim tvrtkama postoje programeri koji su opredijeljeni ili za frontend ili za backend dio razvoja aplikacije. Tijekom same izrade aplikacije razvojni inženjeri moraju surađivati jer često i zbog najmanje sitnice odnosno nesporazuma može doći do zastoja u razvoju, a s time lako dolazi do probijanja vremenskog roka za izradu aplikacije što nije dobro za poslovanje tvrtke.

## 4.1. Frot-end

Frontend programiranje odnosi se na izradu vizualnog sučelja aplikacije. Pri tome ne treba miješati frontend programiranje s dizajnom. Nakon što dobije dizajn razvojni inženjer programira svaki element na stranici. Naravno, pri tome mora poštivati dogovorene standarde. Također, zadatak frontend razvojnog inženjera je da osigura prilagodbu sadržaja na različite preglednike kao i na različite veličine ekrana. Na web i mobilnoj aplikaciji za frontend korišten je HTML (*HyperText MarkupLanguage*) i CSS (*Cascading Style Sheets*) te TypeScript koji je vrlo sličan JavaScriptu. Treba napomenuti kako HTML i CSS zapravo i nisu programski jezici, već opisni .

HTML zajedno s TypeScriptom dodaje funkcionalnosti te upravlja sučeljem koje je u kontaktu s posjetiteljem odnosno korisnikom web stranice.

```
<input type=”submit” value=”Login” (click)=”dohvatiKorisnika(korisnik)”>
```

**Sl.4.1.** Kod za dodavanje gumba

Jednostavno objašnjenje slike 4.1. je da tom linijom koda dodajemo gumb na našu stranicu. No, tu možemo vidjeti da postoje parametri koji se zadaju kao što su *type* i *value*. Pomoću input elementa u

dokument se mogu dodati različite vrste kontrola, a nad tim kontrolama se mogu obavljati različite akcije. Parametar *type* definira tip unosa podatka, a neke od mogućnosti su :

- checkBox - odabire se jedna ili više ograničenih opcija iz ponuđenog izbora
- date – unos datuma u odabranom formatu
- text – unos teksta
- radio – odabire se jedna opcija od ponuđenog izbora. Primjerice odabir spola
- password – unosi se tekst koji se automatski prikriva
- submit – definira gumb kojim prihvaćamo ranije unesene podatke

Parametar *value* označava tekst koji će se pojaviti na gumbu. Na klik gumba dolazi do izvršavanja funkcije *dohvatikorisnika(korisnik)* kojom se, u ovom slučaju, provjerava je li korisnik koji se želi prijaviti registriran u aplikaciji, a kao argumenti se predaju unesena adresa e-pošte i zaporka.

Dodavanjem linije koda navedene na slici 4.1. dobivamo gumb izgleda prikazanog na slici 4.2. :

**Sl.4.2.** Izgled gumba nakon dodavanja linije iz slike 4.1.

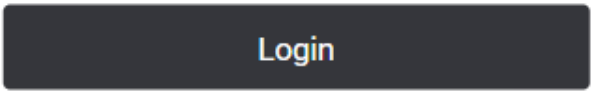
Kako bi se polje za unos teksta prilagodilo izgledu stranice dodaju mu se stilske značajke u CSS-u.

```
input[type=submit]{
background-color: #35363B;
color:white;
border-radius: 3px;
width: -webkit-fill-available;
}
```

**Sl.4.3.** CSS kod za za stiliziranje gumba

Ovim kodom zadana je siva pozadinska boja u hex[1] formatu. Određena je boja teksta koja je bijela te su mu zaobljeni rubovi u iznosu od 3 pixela. Za kraj svojstvom *width* gumb je proširen do njegovih granica. CSS kod za dodavanje stila gumbu nalazi se na slici 4.3. dok je izgled gumba prikazan na slici 4.4.

[1]Šesteroznakovni kod gdje svaki par znakova predstavlja intenzitet crvene,zelene ili plave boje




**Sl.4.4.** Izgled gumba nakon stiliziranja

Ovakvu manipulaciju izgleda potrebno je primijeniti na sve elemente na stranici koji to zahtijevaju.

## 4.2. Back-end

Backend u svijetu programiranja predstavlja drugo naličje gdje samo administratori imaju pristup i služi za upravljanje web poslužiteljem, korisnicima i sadržajem tog sjedišta[13]. Često se ne razmišlja o kodu i načinima na koji su omogućene sve funkcionalnosti koje se koriste. U njemu se pišu linije koda kojim se ostvaruju veze s poslužiteljima, odnosno bazama podataka. Popularno zvani "backendaš" mora izvrsno poznavati arhitekturu sustava (WordPress, Magento, Joomla, Typo3) te vladati širokim spektrom programskih jezika uključujući HTML, PHP (engl. *Hypertext Preprocessor*), MySQL (engl. *Structured Query Language*), JavaScript. Također, potrebno je poznavanje određenih programskih okruženja kao što su Zend, Laravel itd.

Kao nastavak na već spomenuti gumb za prijavu dodat je kod, koji se nalazi u backendu, koji provjerava nalazi li se adresa e-pošte s odgovarajućom zaporkom u bazi podataka.

Najprije je potrebno napraviti API poziv prikazan na slici 4.5. koji se sastoji od HTTP (engl. *Hypertext Transfer Protocol*) metode, dodatka na rutu ('/login) te imena funkcije koja se poziva (slika 4.6.), u ovome slučaju to je funkcija *loginUser*.

```
router.post('/login',
master(),
loginUser)
```

**Sl.4.5.** Mongo API poziv

Nakon napravljenog API poziva potrebno je dodati kod za provjeru korisnika u funkciju *loginUser*.

```
Export const loginUser = (req, res, next) =>
User2.find({ email:req.body.email, password:req.body.pass})
.then(notFound(res))
.then((user) => {
if(user[0]){
res.json({success: true, rez: user, message: "Ulogiran!"})
    }
else{
res.json({success: false,message: "Wrong email or password"})}
    })
.then(success(res))
.catch(next)
```

**Sl.4.6.** Funkcija koja se poziva API-jem

Pomoću funkcije *find* vrši se provjera postoji li u bazi podataka adresa e-pošte i zaporka koju je korisnik unio. Ako korisnik postoji, kao odgovor dobiva se niz objekata koji odgovaraju zahtjevu. Budući da među registriranim korisnicima ne može postojati više njih s istom adresom e-pošte, u nizu postoji samo jedan objekt i on se nalazi na indexu 0. Također u odgovoru se dobiva podatak o uspješnosti te odgovarajuću poruku koja može biti koristina na frontendu.

# 5. TEHNOLOGIJE KORIŠTENE ZA IZRADU APLIKACIJE

U ovom poglavlju dana je teorijska podloga za tehnologije koje su korištene u izradi web i mobilne aplikacije kao i opis baze podataka koja se upotrebljava za spremanje podataka.

## 5.1. Ionic – razvojno okruženje mobilnih aplikacija

Za programiranje mobilne aplikacije korišten je Ionic razvojno oruženje. Ionic je SDK (engl. *Software Development Kit*) otvorenog tipa za hibridne mobilne aplikacije koji su razvili Max Lynch, Ben Sperry i Adam Bradley 2013. godine. Omogućava razvoj mobilnih aplikacija koristeći HTML, CSS i JavaScript odnosno TypeScript. Budući da je Ionic izgrađen na temeljima Angulara, znanje iz Angulara može se vrlo lako primijeniti na njega. Za razumjevanje pojma hibridnih aplikacija najprije je potrebno razumjeti što su to izvorne (engl. *native*) i web aplikacije. Izvorne aplikacije razvijene su po mjeri za podržanu platformu te su pisane programskim jezicima kao što su C++ i Java. Web aplikacije su jednostavno responzivne web stranice. Hibridne mobilne aplikacije su kombinacija tih dvaju pristupa jer pružaju mogućnosti nativnih aplikacija, ali su pisane pomoću web tehnologija (HTML5, CSS, JavaScript) i omogućuju razvoj na više platformi [15]. Iako činjenica da se pisanjem samo jednog koda automatski dobivaju verzije za sve platforme zvuči sjajno, postoje određene restrikcije. Hibridne aplikacije, naspram izvornih aplikacija i mobilnih web stranica imaju nekoliko prednosti, a to su: mogućnost razvoja jedne aplikacije za jednu platformu te je tada uz minimalna ulaganja moguće razviti istu aplikaciju za sve druge podržane platforme, omogućavanje razvoja mobilnih aplikacija sa znanjem tehnologija korištenih u razvoju web stranica i web aplikacija, pristup svim resursima i mogućnostima kao i kod izvornih aplikacija, jednostavnost i brzina razvoja. Osim velikog broja prednosti hibridnih aplikacija postoje i neki nedostaci pri korištenju i razvoju kao što: ograničenja web pogleda – aplikacija može biti pokrenuta samo onoliko dobro koliko i sam web pogled, što znači da je izvođenje aplikacije vezano i ograničeno na kvalitetu izvođenja preglednika na pojedinoj platformi, pristupanje izvornim mogućnostima pomoću programskih dodataka – potreban pristup izvornim API-jima možda trenutno nije dostupan pa je u takvom slučaju potrebno prvo razviti programski dodatak koji će podržavati takav pristup, nedostatak izvornog korisničkog sučelja – bez alata[15]. Neke od najpoznatiji aplikacija izrađenih u Ionicu su *Sworkit*, *JustWatch*, *Pacifica*, *Untappd* koje imaju po nekoliko milijuna preuzimanja.

### 5.1.1. Instalacija Ionica

Instalacija Ionic aplikacije odvija se pomoću Ionic CLI-a (engl. *Command Line Interface*) koje prevedeno na naš jezik odgovara pojmu Ionic komandna linija. Ionic CLI nudi osnovne naredbe kojima se pokreće aplikacija, povezuje ju s drugim uređajima te mnoge druge funkcionalnosti.

Neke od najčešće korištenih naredbi su:

- Start – kreiranje novog projekta
- Docs – otvara Ionic dokumentaciju
- Generate – dodavanje komponenti, stranica, tabova i sl. u projekt
- Serve – pokreće aplikaciju na lokalnom poslužitelju
- Link – povezivanje lokalne aplikacije sa Ionicom
- Build – kompiliranje aplikacije za određenu platformu

Prije početka instalacije potrebno je provjeriti koriste li se najnovije verzije Node.js–a i npm-a (engl. *node package manager*).

Node.js je poslužiteljska platforma otvorenog tipa te se može pokretati na različitim operacijskim sustavima. Služi za dodavanje, brisanje i upravljanje podacima u bazi podataka.

Npm je najveća svjetska baza različitih paketa za programiranje. Korisnici mogu preuzimati već postojeće pakete i koristiti ih u svojim projektima. Instalacija Ionic-a te kreiranje aplikacije prikazani su na slici 5.1. odnosno 5.2.

```
$ npm install -g ionic
```

**Sl.5.1** Naredba za instalaciju ionic CLI-a

```
$ ionic start myApp tabs
```

**Sl.5.2.** Naredba za kreiranje aplikacije

Osim *tabs* predloška, predlošci koji se mogu koristiti su *sidemenu, my-first-app, conference*, a moguće je cijeli projekt započeti od praznog ekrana.
Za pokretanje već napravljene aplikacije potrebno je otići u mapu projekta pomoću naredbe "cd" nakon čega slijedi ime mape u koju se želi ući. Kada se dođe do projekta potrebno je upisati naredbu *ionic serve*, kao što je vidljivo na slici 5.3., te se aplikacija otvara u web pretraživaču:

```
$ cd myApp
$ ionic serve
```

**Sl.5.3.** Pokretanje aplikacije

## 5.2. Angular – razvojno okruženje web stranica

Angular je razvojno okruženje otvorenog tipa za dinamičke web aplikacije baziran na TypeScriptu. Njegov je cilj povećati aplikacije temeljene na preglednicima pomoću mogućnosti Model-View-Controller-a (MVC) i smanjiti količinu JavaScripta potrebnu za funkcionalnost web-aplikacija.

Razvila ga je kompanija Google prije 8 godina. Do sada postoji 8 verzija Angulara (Angular 1-Angular 8), a zadnja puštena je u uporabu 28.5.2019. godine.

Angular radi tako da prvo pročita HTML stranicu, koja ima dodatne ugrađene HTML atribute. Nakon toga interpretira te atribute kao smjernice za povezivanje ulaznih ili izlaznih dijelova stranice s modelom koji je predstavljen standardnim TypeScript varijablama. Vrijednosti tih TypeScript varijabli mogu se ručno postaviti unutar koda ili preuzeti iz statičkih ili dinamičkih JSON resursa.

## 5.3. Baza podataka

Baza podatka je vrlo važan dio svake aplikacije, ali i sastavni dio svakodnevnog života. Većina nas, nalazi se svakodnevno u interakciji s bazama bez da smo svjesni toga.

Najjednostavnije rečeno baza podataka je organizirani skup podataka. Predstavlja skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala koji su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Sustav za upravljanje bazom podataka (*Data Base Management System*- DBMS) je poslužitelj (server) baze podataka. On oblikuje fizički prikaz baze u

skladu s traženom logičkom strukturom i obavlja sve operacije s podacima. Podaci u bazi su logički organizirani u skladu s nekim modelom podataka [16]. Glavna podjela baza je na: NoSQL i SQL.

### 5.3.1. NoSQL

SQL je strukturirani jezik za pretraživanje koji omogućuje različite zadaće nad podacima u bazi. Kod SQL baza podataka podaci su organizirani u tablice gdje svaki redak predstavlja entitet, a stupci predstavljaju atribute entiteta.

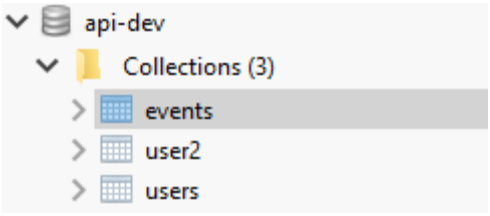
NoSQL baze su ne-relacijske baze podataka novije generacije, većinom su otvorenog koda, horizontalno skalabilne, imaju jednostavan API, rade s velikom količinom podataka itd. Neke od njih su CouchDB, MongoDB, SimpleDB, Cassandra, Amazon Dynamo, GraphDB, DEX …

Prednosti NoSQL baza su da nemaju shemu potrebnih podataka, a podaci se mogu dodavati bez da se prethodno definira cjelokupna shema. Nadalje, postoji automatska elastičnost zbog čega NoSQL automatski širi podatke na više poslužitelja bez potrebe za primjenom programa. Važno za njihov rad je integrirano predmemoriranje - za povećanje podataka i povećanje performansi NoSQL tehnike unaprijed pohranjuju podatke u memoriju sustava [17]. Općenito govoreći NoSQL baze podatake obrađuju podatke brže od SQL, imaju jednostavnije modele podataka te su dovoljno fleksibilne da im programeri mogu prilagođavati aplikacije.

### 5.3.2. MongoDB

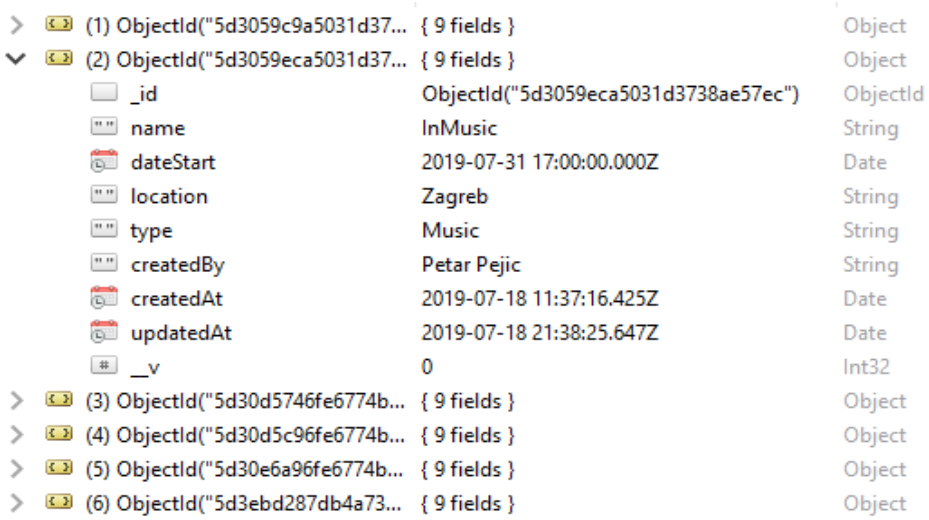
MongoDB razvili su Dwight Merriman i Eliot Horwitz prilikom rješavanja problema implementiranja tradicionalne relacijske baze pri programiranju njihove aplikacije. Zanimljivo, ime Mongo dolazi od riječi *humongous* koja znači ogroman te su vođeni tim razvili sustav za manipulaciju velikim količinama podataka.

MongoDB je ne-relacijska baza koja podatke prezentira kao kolekcije (slika 6.1.) JSON (engl. *JavaScript Object Notation*) dokumenata u binarnom zapisu koji se nazivaju BSON (engl. *Binary JSON*). Polja odnosno atributi se mogu razlikovati od dokumenta do dokumenta te nema potrebe za unaprijed definiranim strukturama.




**Sl.6.1.** Kolekcije u aplikaciji

Dokumenti su modeli podataka koji su predstavljeni kao mape što se vidi na slici 6.2. Oni se u aplikaciji predstavljaju kao objekti čime programerima znatno olakšavaju učenje i korištenje. Svaki dokument sadrži atribute koji ga opisuju, a ukoliko se želi dodati novo polje dodaje se bez ikakvog utjecaja na ostale dokumente u kolekciji.



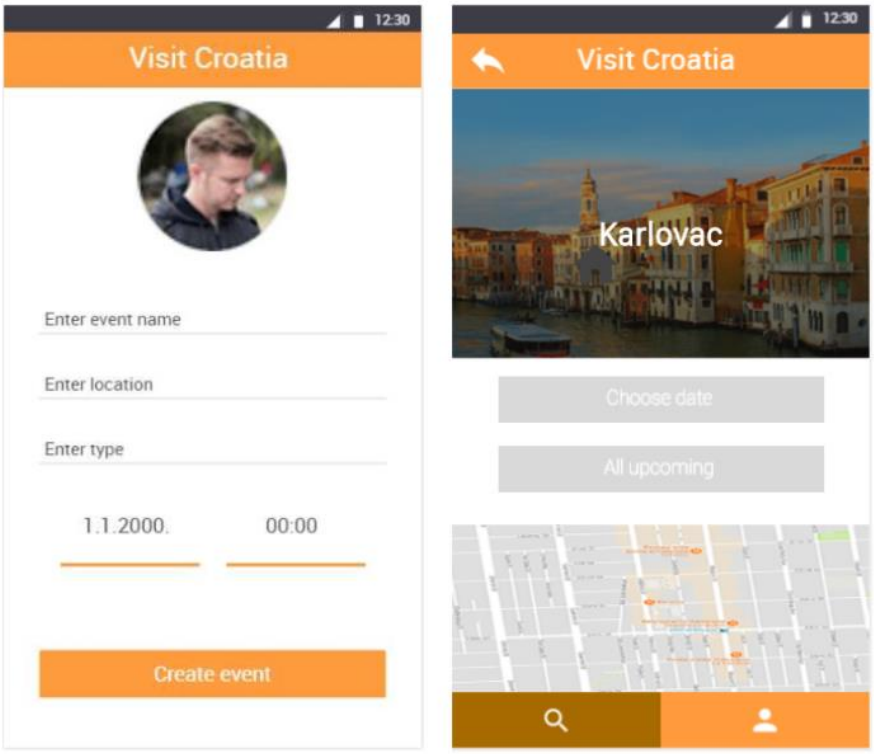

**Sl.6.2.** Dokumenti u kolekciji *eventi*

# 6. IZRADA APLIKACIJE

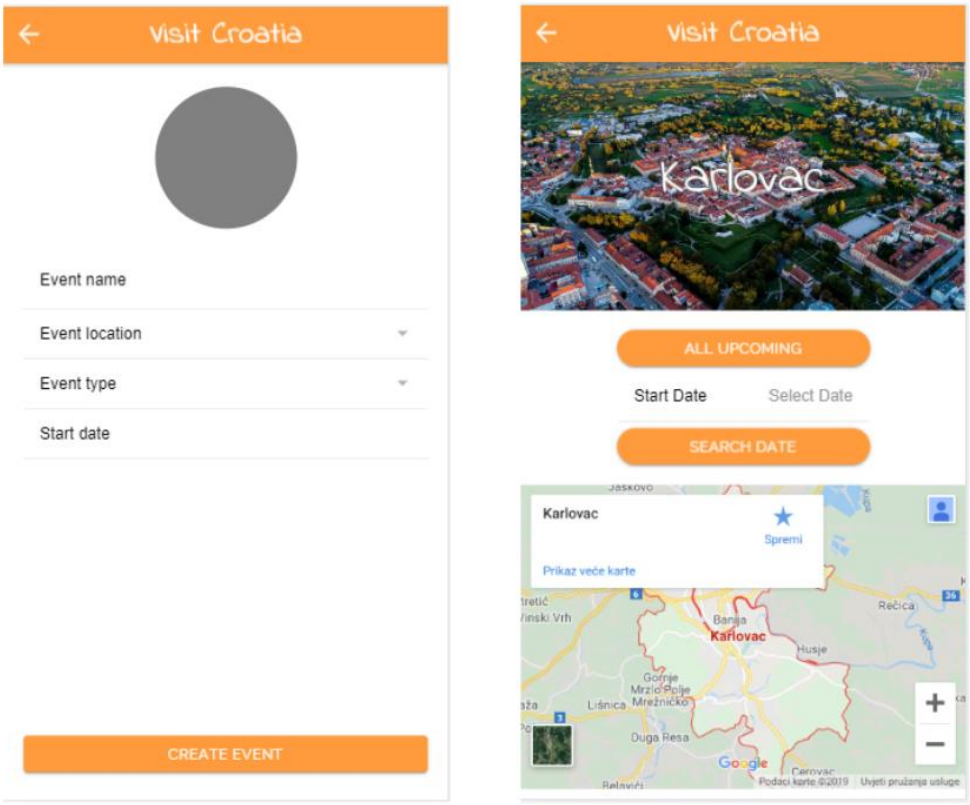
U nastavku je prikazana i opisana izrada aplikacije. Prije same izrade potrebno je napraviti dizajn. Za izradu web sučelja korišten je Angular dok je za izradu mobilne aplikacije korišten Ionic. Dizajn je izrađen u FluidUI-u.

## 6.1. Ideja i osnovni koncept izrade aplikacije

Za izradu aplikacije korišten je FluidUI za dizajn, Angular razvojna okolina za web dio te Ionic razvojna okolina za mobilni dio aplikacije. Budući da se korisnicima daje mogućnost kreiranja događaja odnosno stvaranja vlastitog sadržaja nužno je napraviti administratorsko sučelje. To je važno kako ne bi došlo do postavljanja neprimjerenog sadržaja, ali i mogućih tehničkih grešaka. Uspješnim se dizajnom smatra svaki dizajn koji sam za sebe govori i opisuje pojedine funkcije proizvoda, ne zbunjuje korisnika i olakšava upotrebu proizvoda. Izrada uspješnog dizajna započinje skiciranjem. Skiciranje je osnova i skraćuje proizvodni tijek i do 50% [18]. Važno je istaknuti kako aplikacija trenutno pokriva dvadeset najvećih gradova Republike Hrvatske što znači da je moguće samo u njima kreirati i pretraživati događaje.




**Sl.6.1.**Cilj izgleda zaslona *create-event* i *city* izrađen u FluidUI-u




**Sl.6.2.** Krajnji izgled zaslona *create-event* i *city*

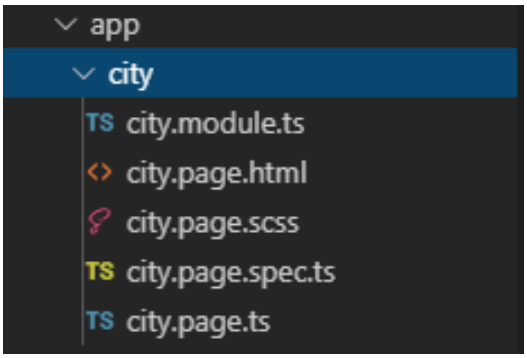
Kako bi se vidjela razlika između početnog dizajna te krajnjeg izgleda zaslona izrađene aplikacije dodane su slike 6.1. i 6.2. na kojima su prikazani zasloni za kreiranje događaja i grada Karlovca.

Prilikom pokretanja aplikacije pojavljuje se ekran na kojemu je prikazan logo. Na njemu se zadržava tri sekunde nakon čega se automatski prelazi na sljedeći zaslon za prijavu. Ukoliko se korisnik ne prijavi onemogućeno mu je postavljanje događaja. Ako korisnik nije registriran moguće je odabrati opciju za registriranje.

Na sljedećem zaslonu nalazi se popis od dvadeset najvećih hrvatskih gradova te se odabirom na pojedini grad otvara novi zaslon na kojem imamo sliku i naslov odabranog grada, prikaz na Google karti i gumbove za pregled događaja. Moguće je pregledati sve nadolazeće događaje ili događaje na odabran datum.

## 6.2. Izrada mobilne aplikacije

Unutar Ionic projekta nalaze se datoteke značajne za održavanje strukture koda i realizaciju aplikacije. Za uspješno obavljanje zadatka važno je aplikaciju podijeliti na određen broj cjelina koje predstavljaju zaslone pri čemu svaka ima svoj izgled i funkcionalnosti koje obavlja. Svaka cjelina sastoji se od osnovnih komponenata; *imezaslona.module.ts, imezaslona.page.html, imezaslona.page.scss, imezaslona.page.spec.ts, imezaslona.page.ts.*

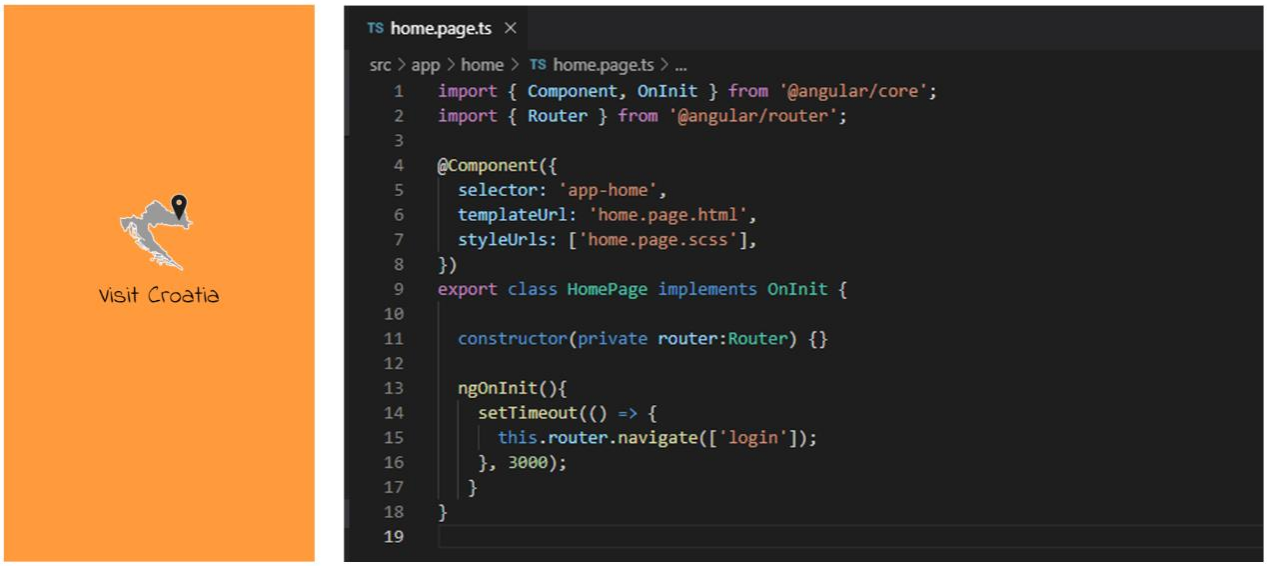



**Sl.6.3.** Struktura zaslona *city*

Unutar datoteke "app" na slici 6.3., nalazi se popis svih korištenih zaslona: *home*, *login*, *registration*, *citylist*, *create-event*, *city*, *eventlist*.

Prilikom izrade sučelja aplikacije najprije je potrebno definirati izgled za svaki zaslon. To se realizira pisanjem koda u HTML-u, te njegovim stiliziranjem u CSS-u. Nakon toga u komponenti s nastavkom *module.ts* dodajemo funkcionalnosti.

Pri pokretanju aplikacije na ekranu se pojavljuje logo ispod kojeg se nalazi naziv aplikacije čiji su izgled i klasa prikazani na slici 6.4.

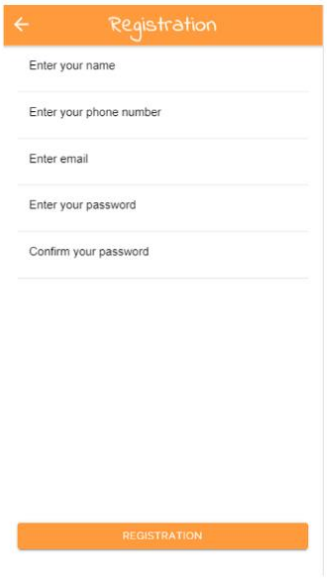



**Sl.6.4.**home.html i home.page.ts

Izgled zaslona je vrlo jednostavan. Sastoji se od jedne klase *img* koja prikazuje logo aplikacije. Prilikom pisanja koda elementi se automatski stavljaju jedan ispod drugog ovisno o tome koji je prije postavljen. Pomoću jedinice vh (engl. *viewport height*) pozicioniraju relativno u odnosu na ekran. Primjerice, "*margin-top:33vh;"* znači da se početak elementa nalazi na jednoj trećini ekrana.

Za kraj je dovoljno pozicionirati elemente u horizontalno središte ekrana. Pri ulasku na ekran dolazi do izvršavanja funkcije *OnInit*. Funkcijom *setTimeout()* određuje se akcija koja će se izvršiti, ali nakon zadanog vremena u milisekundama. U ovom slučaju to iznosi 3000 ms odnosno 3 sekunde.

Registracija korisnika značajan je dio aplikacije jer je to prvi korak ka povezivanju sučelja s bazom podataka.




```
registration()
{
  if(this.regKorisnik.password !== this.regKorisnik.confPassword){
    this.passwordAlert();
  }
  else if(this.regKorisnik.name == ''|| this.regKorisnik.phone == '' ||
  this.regKorisnik.email == '' || this.regKorisnik.password == ''||
   this.regKorisnik.confPassword == ''){
  this.emptyAlert();
  }
  else if(this.regKorisnik.password.length < 6){
    this.lenghtAlert();
  }
  else{
    this.parametriSlanje={
      'name':this.regKorisnik.name,
      'phoneNumber':this.regKorisnik.phone,
      'email':this.regKorisnik.email,
      'password':this.regKorisnik.password
    }
    this.registriraj();
  }
}
```

**Sl.6.5.** home.html i home.page.ts

Prilikom registracije korisnika najprije se provjeravaju uvjeti koji su nužni za uspješnu registraciju, što se nalazi na slici 6.5. Niti jedno polje ne smije biti prazno, unešene zaporke moraju biti jednake te zaporka ne može imati manje od šest znakova. Ako neki od uvjeta nije ispunjen prikazuje se upozorenje.

Nakon provjere svih uvjeta provjerava se postoji li već korisnik s unešenom adresom e-pošte ili brojem mobitela u bazi podataka. Ukoliko postoji, upozorenje će se prikazati na ekranu te će posjetitelj morati ponovno unijeti svoje podatke.

Ako nakon svih provjera ne postoji nikakav problem poziva se funkcija *registrirajKorisnika(this.parametriSlanje)* (slika 6.6.) kojoj se, kao argument, predaje objekt korisnika koji se želi registrirati.

```
registrirajKorisnika(a:any){
returnthis._http.post('http://localhost:9000/' + 'user2s', a).pipe();
  }
```

**Sl.6.6.**Spremanje dokumenta u kolekciju user2

Komponente ne bi smjele dohvaćati ili spremati podatke. Njih se koristi za predstavljanje podataka te određivanje pristupa podacima usluge. Zato je vrlo važno da pri dohvaćanju i spremanju podataka imamo *imezaslona.service.ts*. Usluge su izvrstan način za razmjenu informacija između klasa koje se

međusobno ne poznaju. To su uglavnom obične klase, ali s dobro definiranom svrhom. Ionic razlikuje komponente od usluga zbog povećanja modularnosti i ponovne upotrebe.

Funkcija *registrirajKorisnika(a:any)* nalazi se u *registration.service.ts.* Nakon toga slijedi pozivanje API-ja (engl. *Application programing interface*) koji na temelju predanog parametra, tj. objekta koji je predan, poziva funkciju *create* koja kreira dokument u bazi podataka s dodjeljenim atributima što je vidljivo na slikama 6.7. i 6.8.

```
router.post('/',
//master(),
body({name, email, password, phoneNumber }),
create)
```

**Sl.6.7.** Api poziv za kreiranje korisnika

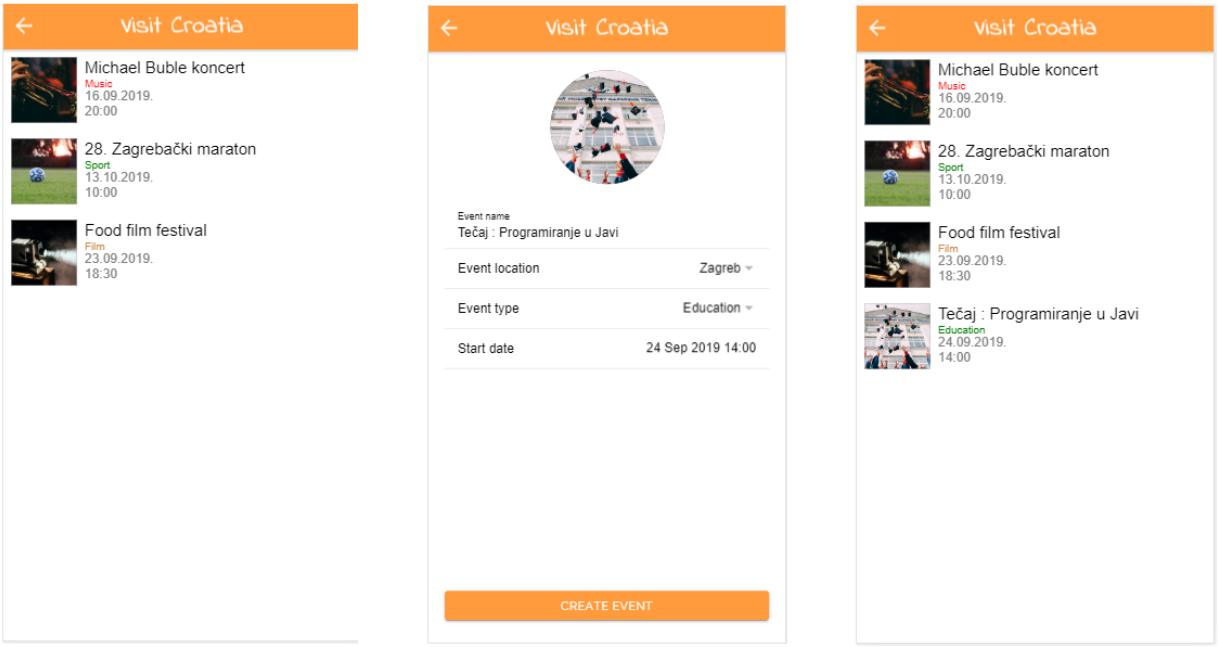
```
exportconstcreate = ({ bodymen: { body } }, res, next) =>
User2.create(body)
    .then((user2) =>user2.view(true))
    .then(success(res, 201))
    .catch(next)
```

**Sl.6.8.** Kreiranje korisnika u bazi podataka

Dodavanje događaja radi se u klasi *create-event*. Svaki događaj određen je svojim imenom, mjestom u kojem se održava, vremenom u obliku datuma i sata te tipom. Tip događaja odnosi se na predstavljanje vrste događaja. Primjerice, za nogometnu utakmicu odgovarajući tip bio bi *sport*, dok bi za umjetničku izložbu bio *art*. U ovisnosti o tipu, fotografija se automatski generira. Za svaki je određena jedna slika koja se povlači s web stranice *https://unsplash.com.* Dio koda koji provjerava i postavlja sliku vidljiv je na slici 6.9.

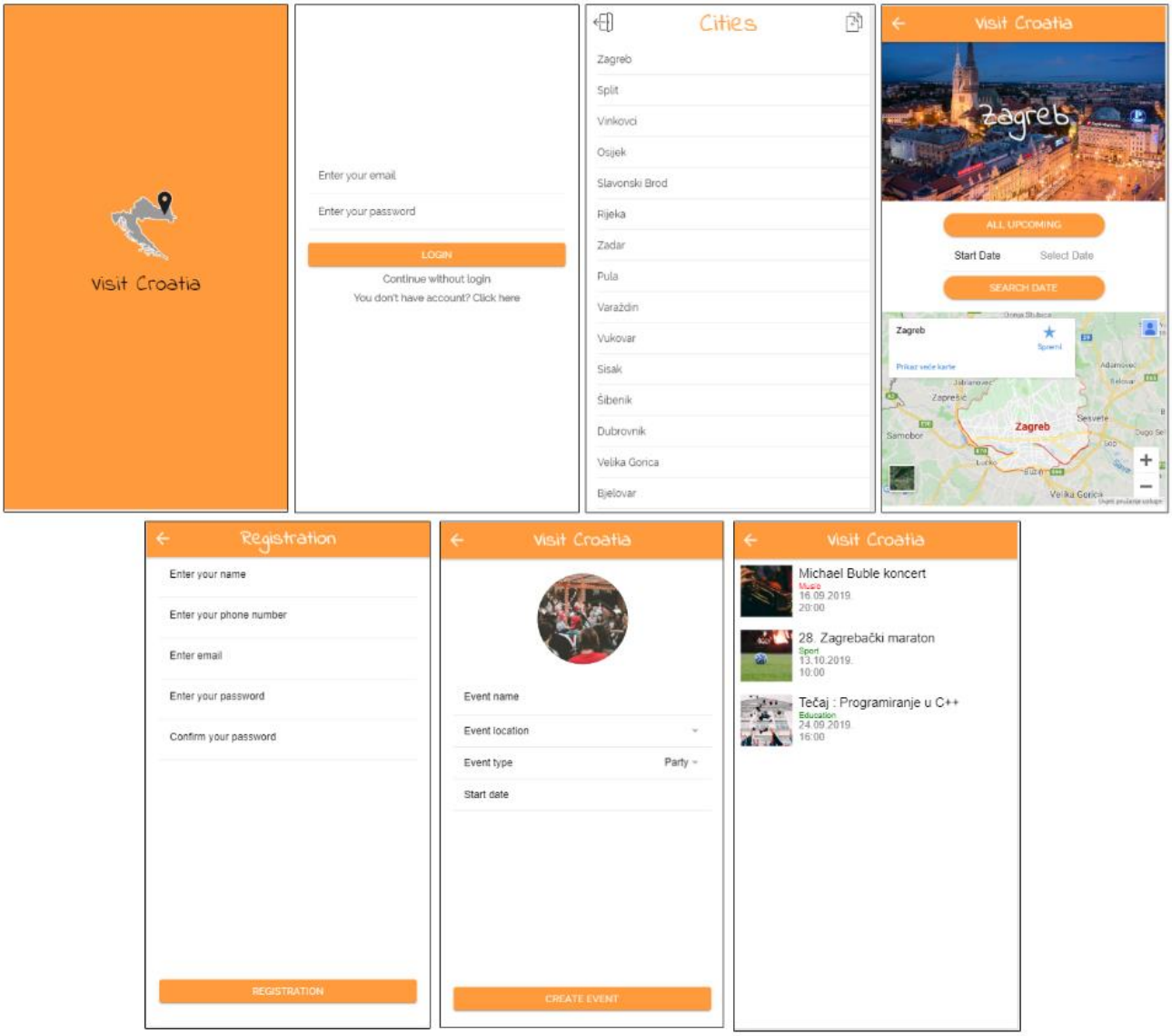
```
if(this.event.type === "Education")
document.getElementById('typeImage').innerHTML = '<img
src="https://source.unsplash.com/collection/6782142/" style="height: 100%;object-fit:
cover; width:100%"; >';
```

**Sl.6.9.** Postavljanje zadane slike s obzirom na tip događaja




**Sl.6.10.** Kreiranje događaja

Na slici 6.10 predstavljen je primjer dodavanja događaja *Tečaj:Programiranje u Javi*. Također, može se vidjeti kako je učitan iz baze podataka u listu događaja. Za kraj dodani su izgledi svih zaslona mobilne aplikacije, vidljivi su na slici 6.11.




**Sl.6.11.** Prikaz svih zaslona aplikacije

## 6.3. Izrada web sučelja

Sučelje za upravljanje događajima izrađeno je u Angularu. Korisnicima koji su registrirani omogućeno je postavljanje događaja u aplikaciju i putem web sučelja. Web aplikacija sastoji se od sljedećih stranica: *cities*, *event*, *eventlist*, *login*, *update-event*. Prva stranica koja se otvara na webu je *login* (slika 6.12). Ovisno o tome je li korisnik koji se prijavljuje administrator otvara se sljedeća stranica.

**Sl.6.12.** Zaslon za prijavu

Ako je prijava uspješna dobiva se odgovor od poslužitelja kao "*success*" nakon čega se lokalno spremaju podaci iz baze za potrebe daljnjeg korištenja (slika 6.13). U slučaju da prijava nije uspješna dolazi do prikaza poruke upozorenja. Ako je korisnik koji se prijavljuje administrator, otvorit će mu se drugačija stranica od one koja se otvara "običnim" korisnicima stranice.

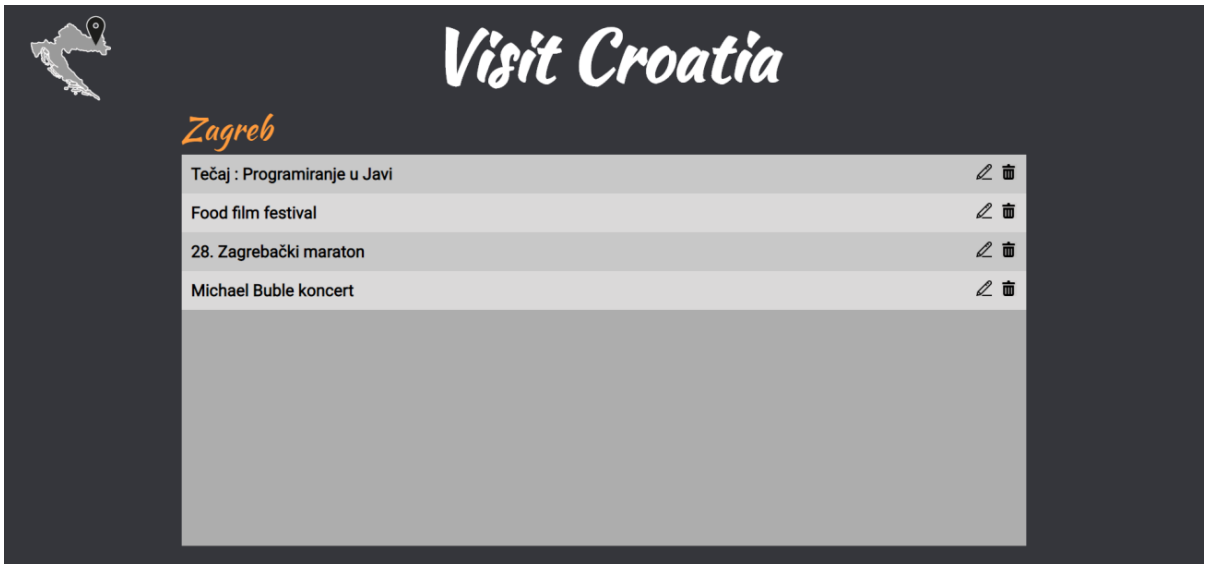
```
this._loginService.dohvatiKorisnika(this.parametriSlanje).subscribe(res=> {
console.log(res['success']);
if (res['success'] === true) {
this.error = false;
localStorage.setItem("ime", res['rez'][0].name);
localStorage.setItem("telefon", res['rez'][0].phoneNumber);
localStorage.setItem("email", res['rez'][0].email);
if(localStorage.getItem('email') === 'petar.pejic@outlook.com'){
this.router.navigate(['/cities']);
          }
else{
this.router.navigate(['/event']);
          }
       } else {
this.error = true;
this.errorMessage = res['message'];
       }
    });
  }
```

**Sl.6.13.** Prijava korisnika na web aplikaciju

Adresa e-pošte koja je određena kao administratorska glasi *petar.pejic@outlook.com* i njezinom se prijavom otvara stranica administratorskog sučelja koje je prikazano na slici 6.14.

**Sl.6.14**. Administratorsko sučelje

Odabirom na određeni grad otvara se stranica za upravljanje događajima. Događaji su prikazani u listi te je svaki moguće obrisati ili po želji promijeniti informacije. Listu događaja u gradu Zagrebu moguće je vidjeti na slici 6.15.




**Sl.6.15.** Prikaz događaja u Zagrebu

```
dohvatiEvent() {
this._updateEventService.dohvatiEvent(this.eventId).subscribe(res=> {
this.event={
'name' :res['name'],
'dateStart':new Date(res['dateStart']),
'time':moment(res['dateStart']).format("HH:mm"),
'location':res['location'],
'type':res['type']
      }
this.selectedLocation = this.event.location;
this.selectedType=this.event.type;
    });
  }
```

**Sl.6.16.** Dohvaćanje događaja prilikom ažuriranja

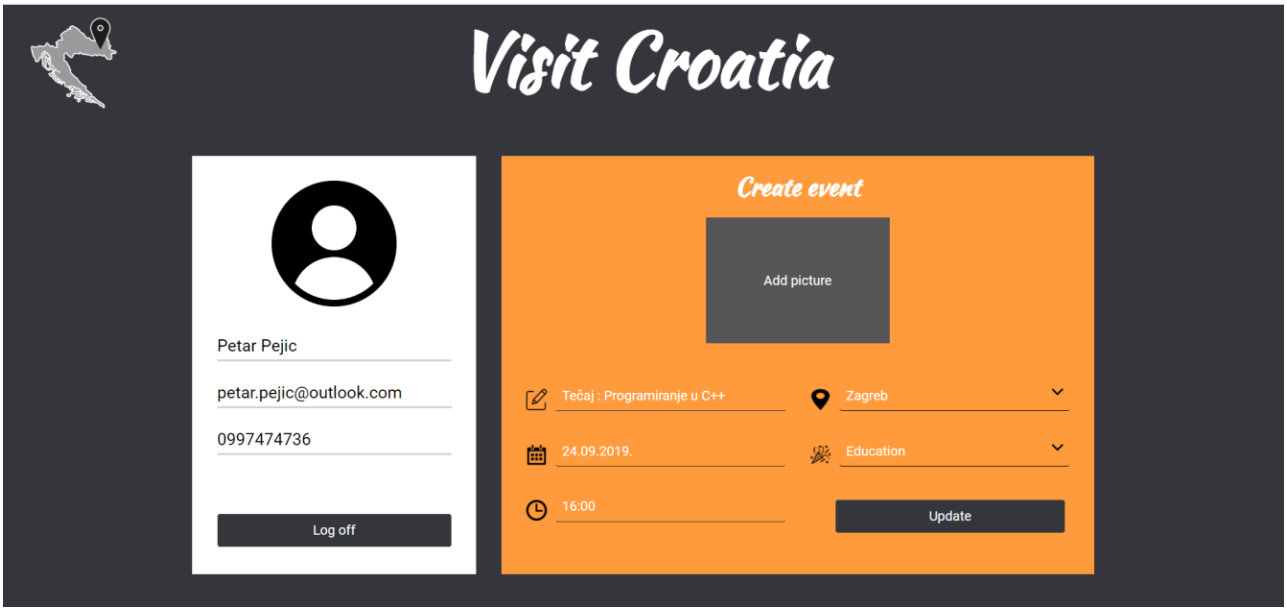
Svaki događaj posjeduje svoj *id* preko kojega mu se pristupa te se njime dohvaćaju podaci o događaju iz baze podataka. Budući da je odgovor koji se dobiva objekt, podacima se pristupa preko atributa koje svaki objekt posjeduje. Dohvaćanje događaja putem njegovog id-a prikazano je na slici 6.16.

Ono što je važno napomenuti je format vremena u kojem se spremaju događaji. Kada se događaj dohvaća iz baze podataka, sati i minute se dohvaćaju kao stringovi u posebnom atributu objekta-*time*. Najprije ih je potrebno pretvoriti u brojeve te uz metode *setHours* i *setMinutes* spremiti u zajednički atribut s datumom što je vidljivo na slici 6.17. Sve je to omogućeno bibliotekom *Moment.js* koja služi za manipulaciju datuma i vremena. Potrebno je odabrati format u kojemu se datum prikazuje te vremensku zonu.

```
Let hours = Number(moment(this.event.time).format("HH:mm").substr(0,2))
Let minutes = Number(moment(this.event.time).format("HH:mm").substr(3,5))
this.parametriSlanje.dateStart.setHours(hours)
this.parametriSlanje.dateStart.setMinutes(minutes)
this.parametriSlanje.dateStart=moment(this.parametriSlanje.dateStart).format("YYYY-MM-
DDTHH:mm:00.000");
```

**Sl.6.17.** Postavljanje datuma

Za potrebe rada događaj *Food film festival* bit će obrisan dok će *Tečaj: Programiranje u javi* biti preimenovan, a vrijeme početka će biti izmjenjeno kao što je vidljivo u nastavku na slici 6.18.




**Sl.6.18.** Ažuriranje događaja *Tečaj : programiranje u Javi*

Sve manipulacije koje se s podacima naprave u aplikaciji bit će spremljene u bazu te će se automatski sinkronizirati na mobilnom i web dijelu aplikacije što je prikazano na slici 6.19.

**Sl.6.19.** Prikaz događaja nakon izvšenih akcija

# 7. ZAKLJUČAK

Tehnologija napreduje ogromnim koracima. Svakim se danom pomiču granice, a ono što je do prije par godina zvučalo nemoguće, danas je sasvim normalno. Razvojem tehnologije raste i upotreba mobilnih uređaja bez čijih funkcionalnosti, već sada, ne možemo zamisliti svoju svakodnevicu. U ovom radu se isprepliću, pa i spajaju, jedne od vodećih industrija svijeta. Turistička i mobilna industrija.

Turizam je jedna od najvažnijih gospodarskih grana Republike Hrvatske. Detaljnom analizom turističke ponude utvrđeno je kako ova vrsta aplikacija ima potencijala na ovome području. Osim ljetne sezone koja se svake godine produžuje sve je popularniji ruralni turizam uz već, od prije, afirmirani zimski turizam. Napravljena aplikacija omogućuje svakoj osobi da napravi vlastiti događaj te da na taj način doprinese ukupnom poboljšanju ponude. Također, putnicima nudi brojna rješenja u vidu posjeta alternativnim događanjima.

Izuzetno je značajno potaknuti lokalno stanovništvo da ulaže u svoju ponudu, ali i osvijestiti ih da i oni mogu sudjelovati u stvaranju sadržaja. Svim korisnicima koji žele kreirati događaj omogućeno je jednostavno sučelje, a sam događaj kreira se u nekoliko klikova bez ikakvih troškova što znači da je korištenje aplikacije besplatno.

Svrha aplikacije je malim stvarima obogatiti raznolikost ponude. Povećanjem posjetitelja koji traže zabavu i događaje koje mogu posjetiti povećava se i broj onih koji nude i organiziraju događaje. Tijekom izrade aplikacije fokus je zadržan na jednostavnosti kako bi posjetitelji u što manje koraka došli do traženih informacija.

Postoje brojna rješenja za komercijalizaciju aplikacije. Neka od njih su omogućavanje korisnicima kreiranje vlastitog plana posjećivanja događaja, slanje informacija o promjenama u događaju ukoliko je korisnik označio da će ga posjetiti, mogućnost ostavljanja povratne informacije korisnika itd. Kao dopuna ovoj aplikaciji može biti pokrivenost za sva postojeća naseljena mjesta u Republici Hrvatskoj. Također, postoji mogućnost proširenja aplikacije u vidu prikazivanja događaja koji se nalaze u okolici odabranog, većeg mjesta.

## LITERATURA


[1] https://www.bug.hr/ (stranica posjećena 25.lipnja 2019)

[2] https://punkufer.dnevnik.hr/clanak/kakvo-priznanje-hrvatska-na-drugom-mjestu-najboljih-destinacija-u-europi---546758.html (stranica posjećena 2.kolovoza 2019)

[3] https://www.tportal.hr/biznis/clanak/hrvatska-najbolja-destinacija-poslovnog-turizma-u-europi-u-2018-20190320 (stranica posjećena 2.kolovoza)

[4] https://novac.jutarnji.hr/rasprave-i-rjesenja/velika-analiza-turizam-u-hrvatskoj-i-usporedba-s-drugim-mediteranskim-destinacijama/7272516/ (stranica posjećena 25.lipnja 2019)

[5] https://arhivanalitika.hr/blog/turizam-u-hrvatskoj-usporedba-s-drugim-mediteranskim-destinacijama/ (stranica posjećena 24.lipnja 2019)

[6] https://www.vecernji.hr/vijesti/hrvatsku-u-2018-izabralo-gotovo-20-milijuna-turista-1290275 (stranica posjećena 24.lipnja)

[7] https://www.cimerfraj.hr/aktualno/statistike-turisticke-2018-godine (stranica posjećena 2.srpnja)

[8] Državni zavod za statistiku, https://www.dzs.hr/Hrv/DBHomepages/Turizam/Turizam.htm, (stranica posjećena 2.kolovoza 2019)

[9] https://www.fortuno.hr/kratka-povijest-mobilnih-aplikacija/ (stranica posjećena 2. Srpnja 2019)

[10] https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/ (stranica posjećena 24.lipnja 2019)

[11] https://www.idc.com/promo/smartphone-market-share/os (stranica posjećena 25.lipnja 2019)

[12] https://zimo.dnevnik.hr/clanak/fer-ovci-razvili-mobilnu-aplikaciju-za-pronalazak-dogadjaja-u-zagrebu-i-okolici---400334.html (stranica posjećena 2.kolovoza 2019)

[13] https://znatko.com/1233/(stranica posjećena 2.kolovoza 2019)

[14] https://safrandesign.hr/razlika-izmedu-web-dizajnera-i-web-developera/ (stranica posjećena 4.kolovoza 2019)

[15] https://repozitorij.unin.hr/islandora/object/unin%3A2015/datastream/PDF/view (stranica posjećena 4.kolovoza 2019)

[16] http://www.znanje.org/abc/tutorials/accessMMX/01/Baze_podataka.htm (stranica posjećena 5.kolovoza 2019)

[17] Difference between SQL and NoSQL, https://www.geeksforgeeks.org/difference-between-sql-and-nosql/ (stranica posjećena 10.srpnja 2019)

[18] K. Nenadić, S. Bošnjak, T. Pekanov, J. Balen, A. Baumgartner, B. Zorić, M. Hanzer, Razvoj mobilnih aplikacija-priručnik za edukaciju Osijek: Elektrotehnički fakultet Osijek, 2013

[19] Ionic developer community, https://ionicframework.com/community (stranica posjećena 25.lipnja 2019)

[20] What is MongoDB? Introduction, Architecture, Features & Example (stranica posjećena 4.kolovoza 2019)

[21] https://www.guru99.com/what-is-mongodb.html (stranica posjećena 12.srpnja 2019)

[22] Ivana Šimić: Uvod u TypeScript, http://www.mono.hr/2016/12/15/uvod-u-typescript-code-camp/ (stranica posjećena 25.lipnja 2019)

[23] Eurostat Statistic Explained, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Tourism_statistics_for_Croatia&oldid=202498 (stranica posjećena 2.kolovoza 2019)

[24] Angular documentation, https://angular.io/docs (stranica posjećena 25.lipnja 2019)

[25] Moritz Christian: Mobile Application Development in the Tourism Industry and its Impact on On-Site Travel Behavior, https://www.modul.ac.at/index.php?eID=dumpFile&t=f&f=5212&token=574c2c521a177ac124fa57d70a466744635b16ac (stranica posjećena 23.lipnja 2019)

## SAŽETAK

Tema rada bila je izrada aplikacija čija je svrha pružanje pregleda ponude zabavnih sadržaja u Republici Hrvatskoj na osnovi vremena i mjesta. Cilj je bio cijeli postupak zadržati na jednostavnosti te pristupačnosi. Na temelju pretrage već postojećih aplikacija određen je broj zaslona, boje koje predvladavaju, funkcionalnosti i druge stvari važne za rad. Aplikacija je izrađena u razvojnom programskom okruženju Ionic, dok je web sučelje namjenjeno za upravljanje događajima izrađeno u Angularu. Za izgled zaslona pobrinuli su se opisni jezikci HTML CSS, a funkcionalnosti su rađene u TypeScriptu.

Baza podataka koja je korištena za spremanje objekata iz aplikacije je MongoDB. Kako bi baza podataka bila vizualno prikazana potreban je Robo 3T koji pomaže pri upravljanju podacima.

Krajnji cilj realizacije aplikacije je poboljšanje ponude i potražnje turističke ponude u Republici Hrvatskoj.



**Ključne riječi**: Android, Ionic, TypeScript, turizam, MongoDB

## ABSTRACT

The task of this work was to build an application whose purpose is providing an offer of entertainment content in the Republic of Croatia based on time and place. The goal was to keep the whole process of creating and reviewing content simple and accessible. Number of activities are specified by reviewing already existing applications, same as colors that prevail, functionalities and other things important for work.

Application was built in Ionic framework, while the web interface which providing event management was build in Angular. The layout is coded in HTML markup language and functionalities were made by TypeScript. The database used for storing objects from the application is MongoDB. For a visual representation of data, it was necessary to install Robo 3T which serves as a data management tool in the database.

The ultimate goal of application implementation is to improve the supply and demand of tourist offers in the Republic of Croatia.



**Key words**: Android, Ionic, TypeScript, tourism, MongoDB

# ŽIVOTOPIS

Petar Pejić rođen je 13. svibnja 1997. godine u Vinkovcima, Hrvatska. Svoje školovanje započinje 2004. godine u Osnovnoj školi Antuna Gustava Matoša,Vinkovci. Nakon osnovne škole svoje srednjoškolsko obrazovanje nastavlja u Tehničkoj školi Ruđera Boškovića u Vinkovcima gdje upisuje smjer elektrotehnika. Trenutno pohađa preddiplomski studij, smjer računarstva na „Fakultetu elektrotehnike, računarstva i informacijskih tehnologija“ u Osijeku koji je u sklopu Sveučilišta Josipa Jurja Strossmayera.