

GISSPOM LOKALIZIRANA BAZA ZNAMENITOSTI TEMELJENA NA URBANOM OKRUŽENJU

GISSPOM LOCALIZED BASE OF SIGHTS BASED ON URBAN ENVIRONMENT

Alen Šimec¹, Filip Penezić²

¹Tehničko veleučilište u Zagrebu, Vrbik 8, 10 000 Zagreb, Hrvatska

²Tehničko veleučilište u Zagrebu, Vrbik 8, 10 000 Zagreb, Hrvatska, Student

SAŽETAK

U ovom radu opisan je razvoj GisSpom aplikacije i integracije s Telegram platformom za izmjenu poruka u obliku chatbota. Ideja iza GisSpom Web aplikacije je napraviti bazu spomenika i ostalih znamenitosti u Hrvatskoj. Web aplikacija mora biti prilagodljivog (responzivnog) dizajna kako bi olakšala korištenje na mobilnim uređajima. Za pretraživanje znamenitosti korisnik ima dvije opcije, filtriranjem rezultata popisa svih znamenitosti ili putem interaktivne karte. Dodavanje novih znamenitosti omogućeno je svim registriranim korisnicima. Web aplikacija je izrađena korištenjem različitih tehnologija otvorenog koda, a to su PHP i MySQL za pozadinu, te Bootstrap i JavaScript za sučelje. Aplikacija se oslanja na vanjske servise poput Google Maps APIa, Google OAuth i Telegram aplikacije za razmjenu poruka.

Ključne riječi: *GisSpom, baza podataka, spomenici, kultura, Internet aplikacija, realno okruženje*

ABSTRACT

This paper describes the development of the GisSpom application and integration with the Telegram platform for exchanging messages in the form of a chatbot. The idea behind the GisSpom Web application is to create a database of monuments and other sights in Croatia. The web application must be of a flexible (responsive) design to facilitate use on mobile devices. To search for sights, the user has two options, by filtering the results of the list of all sights or via an interactive map. Adding new sights is enabled

for all registered users. The web application was created using various open source technologies, namely PHP and MySQL for the background, and Bootstrap and JavaScript for the interface. The application relies on external services such as Google Maps API, Google OAuth and Telegram messaging application.

Keywords: *GisSpom, database, monuments, culture, Internet application, real environment*

1. UVOD

1. INTRODUCTION

GisSpom aplikacija nastala je u želji da se napravi interaktivna baza znamenitosti gdje korisnici mogu pronaći znamenitosti putem interaktivne karte i na taj način dobiti informacije o mjestu u kojemu se nalaze. Ponekad prolazimo pokraj neke građevine, niti ne primijetimo neki reljef ili skulpturu na njezinom pročelju, GisSpom aplikacija služi kako bi se upravo takve interesantne, zaboravljene znamenitosti približile njezinim korisnicima. GisSpom osim pretraživanja nudi opciju dodavanja novih znamenitosti svim registriranim korisnicima. Kako bi se maksimalno olakšao unos nove znamenitosti, aplikacija može dohvatiti korisnikovu trenutnu lokaciju putem HTML5 geolokacijskog APIa. Na taj način korisnik koji je naišao na neku znamenitost koja još nije u sustavu može na brzinu dodati znamenitost u sustav i za lokaciju postaviti korisnikovu trenutnu lokaciju, bez potrebe za ručnim postavljanjem na karti. Prilikom dodavanja fotografija korisniku koji se služi mobilnim uređajem nudi se opcija otvaranja kamere za brzo postavljanje fotografije

znamenitosti. Za lakšu prijavu na aplikaciju koristi se Google OAuth sustav, time se izbjegla potreba za klasičnim unosom korisničkog imena, lozinke, mail adrese, bilo tko s Google računom može kreirati račun na Web stranici i prijaviti se kroz svega par sekundi. Kako u zadnje vrijeme sve više korisnika provodi većinu svog vremena tipkajući u raznim aplikacijama za razmjenu poruka, napravljena je integracija GisSpom aplikacije s Telegram aplikacijom za razmjenu poruka u obliku chatbota. Korisniku Telegram aplikacije dijeljenjem svoje (ili željene) lokacije direktno GSBotu ili u grupu s GSBotom bot će vratiti popis 10 najbližih znamenitosti uz mogućnost pregleda njezinih detalja.

2. KORIŠTENE TEHNOLOGIJE ZA RAZVOJ APLIKACIJE

2. TECHNOLOGIES USED FOR APPLICATION DEVELOPMENT

Razvojne tehnologije podijelili smo na dva dijela, poslužiteljski sloj – aplikacijski dio koji se izvršava na poslužiteljskoj strani i klijentski sloj – dio koji se izvršava na klijentskoj strani.

Tehnologije korištene na poslužiteljskom sloju su PHP – skriptni jezik otvorenoga koda koji se koristi pri programiranju dinamičkih Web stranica i MySQL – sustav za upravljanje bazom podataka otvorenog koda.

PHP je jedan od najpopularnijih programskih jezika za izradu dinamičnih Web stranica, prema statistikama w3techs.com¹ Web stranice, PHP se koristi na 79% svih Web stranica s poznatim poslužiteljskim programskim jezicima. PHP u GisSpomu je korišten za generiranje dinamičkih Web stranica i strukturirane podatke koji se koriste za prikaz oznaka na karti, generiranje dinamičkih tablica i kao API sučelje za Telegram.² Isto tako PHP se koristi za dohvatanje i pospremanje podataka u MySQL bazu podataka. MySQL je strukturirana baza podataka relacijskog tipa, gdje se podaci pohranjuju unutar tablica koje

1 W3Techs – Server-side Languages – PHP – URL: <https://w3techs.com/technologies/details/pl-php> – 11.11.2020

2 Nixon, Robin; *Learning PHP, MySQL, JavaScript, CSS & HTML5, 3rd Edition*; O'Reilly Media; 2014.; str. 34.

sadrže kolone i redove. GisSpom koristi nekoliko različitih tablica kao što su tablica za pohranu podataka o korisniku, tablica znamenitosti, tablica autora i ostale.

Klijentski sloj aplikacije koristi Internet preglednik korisnika za generiranje i prikaz podataka. Kombinacija HTML, CSS i JS koda generira podatke na ekranu korisnika i prilagođava prikaz dimenzijama zaslona uređaja korisnika.³ Samo korisničko sučelje izrađeno je koristeći Bootstrap programski okvir koji sadržava sve standardne elemente Web stranice kao što su navigacijska traka, gumbi, polja za unos podataka i ostale, a s obzirom na to da je Bootstrap kreiran s „Mobile First“ (prvo mobilno) pristupom, njegov rešetkasti dizajn je prilagodljiv različitim dimenzijama zaslona i na taj način olakšava i ubrzava razvoj Web aplikacija.⁴

3. REALIZACIJA PLATFORME ZA MOBILNE UREĐAJE I DESKTOP RAČUNALA

3. REALIZATION OF A PLATFORM FOR MOBILE DEVICES AND DESKTOP COMPUTERS

Osnova GisSpom aplikacije je baza podataka, u koju će se pospremati podaci o korisnicima aplikacije, znamenitosti, kategorije znamenitosti, autori i komentari korisnika. Podaci su raspoređeni u 5 tablica. Slika 1 - Arhitektura baze prikazuje arhitekturu baze i relacije između podataka.

Tablica „users“ sadržava podatke o korisniku koji se dohvaćaju putem Google OAuth servisa i koji se automatski osvježavaju prilikom prijave korisnika. Polje „type“ označava razinu prava korisnika, a ta prava mu dodjeljuje administrator aplikacije.

Informacije o znamenitosti poput naziva, slike, lokacije, opisa pospremaju se unutar tablice „spomenici“. Ova tablica koristi strane ključeve iz tablica „autori“ i „kategorije“.

3 Šimec, Alen; *Programiranje i optimizacija Internet stranica u HTML5 okruženju*; Tehničko veleučilište u Zagrebu; 2015; str. 65.

4 Šimec, Alen; *Uvod u HTML, XHTML i CSS*; Tehničko veleučilište u Zagrebu; 2011.; str. 44.

Unutar tablice „kategorije“ definirana je lista mogućih kategorija znamenitosti, tipa bista, fontana, arhitektura i ostale, a tablica „autori“ sadržava imena autora.

Dodatno, tu je još i tablica „status“ unutar koje se zapisuju komentari korisnika, ova tablica koristi strane ključeve iz tablica „spomenici“ i „users“.

Uz gore navedenih pet tablica unutar baze nalaze se i „pogledi“ koji objedinjuju podatke iz tablica i olakšavaju kreiranje upita prilikom dohvaćanja podataka.⁵

Većina PHP koda podijeljena je u nekoliko PHP datoteka, dio koda zadužen za generiranje WEB stranica nalazi se unutar „php“ direktorija i podijeljen je prema logičkim cjelinama WEB stranice, tipa header, footer, komentari itd. U početnom direktoriju nalaze se datoteke unutar kojih su definirane razne funkcije, konfiguracijska datoteka koja sadrži različite varijable poput pristupnih podataka za bazu, Google API ključeva i ajax.php, zadužen za generiranje strukturiranih podataka ovisno o vrsti upita.

Podatak dohvaćen iz baze podataka PHP ubacuje unutar dinamički generiranog HTML koda. Kako bi se olakšao razvoj aplikacije, PHP kod je razdvojen u logičke cjeline, glava stranice, tijelo i podnožje. Ovisno o Web stranici koja se pregledava tijelo stranice učitava potrebnu PHP datoteku koja sadržava dio HTML i programskog koda zadužen za generiranje traženog sadržaja. Primjer programskog koda index.php dokumenta.

```
include 'php/header.php';

if (isset($_GET['map'])){
    include 'php/map.php';
} else if (isset($_GET['popis'])){
    include 'php/ispis.php';
} else if (isset($_GET['login'])){
    include 'php/login.php';
} else if (isset($_GET['profil'])){
    include 'php/profil.php';
} else if (isset($_GET['spomenik'])){
    include 'php/spomenik.php';
} else if (isset($_GET['novi'])){
```

```
include 'php/novi.php';
} else if (isset($_GET['korisnici'])){
    include 'php/korisnici.php';
} else {
    include 'php/home.php';
}

include 'php/footer.php';
```

*Kod 1 – struktura datoteka na serveru
file structure on the server*

Iz priloženog koda vidljivo je da se header.php i footer.php učitavaju bez obzira na stranicu koju korisnik pregledava, a ovisno o traženoj stranici učitava se tijelo Web stranice. Tako na primjer ako je otvorena stranica gisspom.site/map, aplikacija će generirati tijelo s interaktivnom kartom, čiji se HTML i programski kod nalaze unutar php/map.php datoteke. S obzirom na to da je dio stranice zaštićen, odnosno nije pristupačan svim korisnicima, na tim stranicama uključene su provjere je li korisnik ulogiran i ima li sa svojom razinom pristupa pravo otvaranja tražene stranice. Provjera je li korisnik ulogiran vrši se funkcijom loginProtection() koja se dodaje na vrhu zaštićene stranice. U slučaju da korisnik nije ulogiran biti će vraćen na početnu stranicu. Na sličnom principu funkcioniraju i funkcije koje provjeravaju razinu pristupa korisnika (moderator, administrator). Kako je običaj JS kod dodavati u podnožje stranice zbog bržeg učitavanja sadržaja i s obzirom na to da neke skripte nije potrebno pozivati na svim lokacijama (npr. skripta zadužena za generiranje interaktivne karte) postavljena su pravila unutar footer.php datoteke koja će učitati odgovarajuću PHP datoteku samo ako se korisnik nalazi na odgovarajućoj stranici, a to se može vidjeti u priloženom kodu:

```
<?php if (isset($_GET['map'])) {
include 'php/mapfooter.php';} ?>
```

Kod 2 -Skripta zadužena za generiranje interaktivne karte

Script for generating interactive map

HTML kod sastavni je dio PHP dokumenata zaduženih za generiranje sadržaja i na stranicama na kojima je potrebno prikazati vrijednost iz baze koristi se kombinacija HTML i PHP koda. Svaka PHP datoteka počinje s otvaranjem PHP taga <?php unutar kojeg se nalaze funkcije potrebne za dohvaćanje sadržaja iz baze, tada se zatvara PHP tag s ?> i nastavlja s HTML kodom.

⁵ Seyed M.M. "Saied" Tahaghoghi; Hugh E. Williams; Learning MySQL; O'Reilly Media; str. 70.

Unutar HTML koda na željenom mjestu koristi se „short echo“ (kratki echo) PHP kod koji ispisuje željenu vrijednost. Primjer „short echo“ taga možemo vidjeti kod komentara znamenitosti:

```
<p><?=$value['comment'] ?></p>
```

*Kod 3 - PHP kod koji ispisuje željenu vrijednost
PHP code prints value*

Unutar functions.php datoteke definirane su funkcije koje se pozivaju prilikom učitavanja stranice, poput funkcija za provjeru razine pristupa korisnika, ispisa podataka uz baze i ostalih. Kako bi se mogle izvršiti tražene funkcije potrebno je uključiti functions.php datoteku prilikom učitavanja stranice, a to je napravljeno na početku index.php datoteke naredbom require 'functions.php'; i na taj način smo osigurali da su funkcije dostupne na svim stranicama aplikacije.⁶ Primjer funkcije koja provjerava je li korisnik ulogiran:

```
function loginProtection()
{
    if (!isset($_SESSION['user']['type'])) {
        header('Location: /');
    }
}
```

Kod 4 - Funkcije dostupne na svim stranicama aplikacije

Functions available on all pages of the application

Osnovna funkcija datoteke ajax.php je generiranje strukturiranih setova podataka koji se koriste za razne JavaScript skripte kao što su generiranje markera interaktivne karte, podataka koji se koriste za podatkovne tablice. Osim toga koristi se i prilikom dodavanja nove znamenitosti i kao API lokacija za integraciju Telegram chatbota. Ovisno o vrsti upita („POST“ ili „GET“) i unaprijed definiranoj varijabli „mode“, izvršit će se željena aktivnost. Varijablu „mode“ definiramo prilikom AJAX (Asynchronous JavaScript and XML) POST upita ili, ovisno o lokaciji GET upita, unutar ajax.php datoteke. Primjer JS koda zaduženog za uređivanje naziva znamenitosti i PHP koda unutar ajax.php datoteke:

```
$('#naziv').editable({
    type: 'text',
    placement: 'bottom',
    title: 'Naziv:',
```

⁶ Šimec Alen, Davor Lozić; *Extending PHP with modules; Polytechnic & Design; Vol. 3, No. 1, 2015.; str. 3.*

```
url: 'ajax.php',
validate: function(value) {
    if($.trim(value) == '') {
return 'Naziv ne može biti prazan!';}
    },
params: function(params){
    params.mode = 2;
    return params;
}
});
```

*Kod 5 – Javascript datoteka
Javascript files*

```
else if ($mode == 2) {
    $q = $pdo-
        >prepare("UPDATE spomenici
SET naziv = ? WHERE id = ?");
    $q->bindParam(1, $_POST['value']);
    $q->bindParam(2, $_POST['pk']);
    $q->execute();
}
```

*Kod 6 – PHP datoteka
PHP files*

4. REGISTRACIJA I PRIJAVA KORISNIKA POMOĆU GOOGLE OAUTH SERVISA

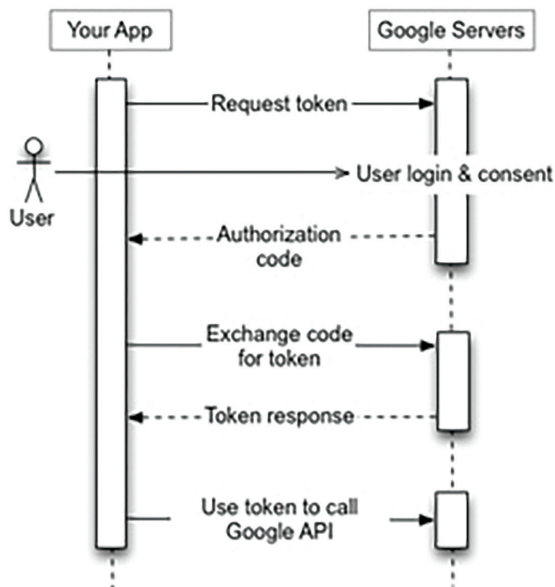
4. USER REGISTRATION AND LOGIN USING GOOGLE OAUTH

Registracija i prijava korisnika implementirana je korištenjem Google OAuth servisa koji se bazira na OAuth 2.0 protokolu. Korištenje OAuth sustava eliminira potrebu za pospremanjem vjerodajnica korisnika (najčešće kombinacija korisničkog imena i lozinke) na serveru, odnosno aplikaciji u koju se korisnik prijavljuje.⁷ Slika 1 prikazuje proceduru prijave u aplikaciju korištenjem Google OAuth sustava. Prilikom prijave aplikacija preusmjerava korisnika na Google poslužitelj i stranicu suglasnosti i prijave korištenjem Google računa. OAuth sustav vraća autorizacijski kod aplikaciji, koja ako je sve u redu ponovno radi upit prema Google serverima i dobiva token za daljnju API komunikaciju.⁸

⁷ *The OAuth 2.0 Authorization Framework – URL: <https://tools.ietf.org/html/rfc6749> – 11.11.2020*

⁸ Šimec, Alen; Lozić, Davor; *Basic PHP Implementations, Opcodes and Internal Work; str. 3*

Iako izgleda komplicirano, krajnjem korisniku zapravo znatno olakšava prijavu u sustav jer nema potrebe za pamćenjem dodatnih lozinki, nema potrebe za unosom korisničkih podataka, verifikacija email adrese i cijeli proces prijave i registracije obavljen je korištenjem Google korisničkog računa. S druge strane, sama aplikacija može dohvatiti osnovne korisnikove podatke postavljene na Google računu kao što su ime, prezime, mail adresa ili korisnička slika.



Slika 1 Google OAuth - <https://developers.google.com/identity/protocols/oauth2>

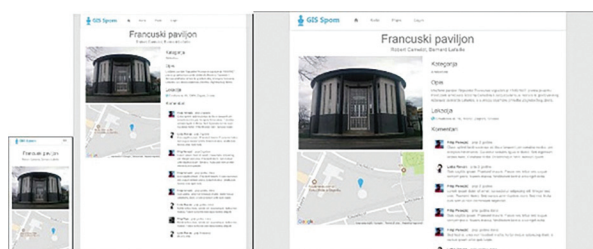
5. GOOGLE MAPS I REVERSE GEOCODING API

5. GOOGLE MAPS AND REVERSE GEOCODING API

Osim Google OAuth aplikacija koristi i druge Google API servise kao što su Google Maps i Google reverse Geocoding API servise. Google Maps koristi se za generiranje interaktivne karte na kojoj su vidljive sve znamenitosti u sustavu, prikaze lokacije znamenitosti prilikom pregledavanja detalja znamenitosti, odnosno odabira lokacije prilikom unosa nove znamenitosti. Prilikom dodavanja znamenitosti korisnik pribadačom odabire željenu lokaciju na karti, a reverse geocoding API pretvara koordinate pribadače u ljudski čitljivu adresu. Tako će na primjer za lokaciju 45°48'20.3" sjeverno i 15°58'43.4" istočno dohvatiti adresu Trg kralja Tomislava 10, 10000, Zagreb, Croatia.

GisSpom u svoju bazu posprema i točne koordinate i adresu znamenitosti i na taj način omogućava pretraživanje rezultata prema nazivu grada ili ulice.

Prilikom izrade korisničkog sučelja koristili smo Bootstrap programski okvir (framework). Iako u današnje vrijeme ima sve više programskih okvira za izradu Web stranica s „mobile first“ pristupom, Bootstrap je i dalje jedan od najzastupljenijih i s najboljom podrškom Internet zajednice. Bootstrap je besplatni programski okvir za izradu Web stranica prema HTML5 i CSS3 standardima i sadržava sve potrebne elemente za izradu Web stranice. Prilagodljiv dizajn postignut je korištenjem takozvanog „rešetkastog“ sistema u kojem je cijela stranica podijeljena u 12 stupaca i širina svakog elementa određuje se brojem stupaca koji on zauzima. Ukoliko se sadržaj pregledava na mobilnim uređajima sadržaj stranica prilagođava svoj dizajn na način da stupce pretvara u redove i slaže sadržaj jedan ispod drugog. Korištenjem različitih klasa možemo definirati kako će se stranica prilagođavati promjenama dimenzija zaslona. Za izradu GisSpom aplikacije korištene su klase za izradu gumba, navigacijska traka, prilagodljive slike i tablice, prilagodljivo umetanje (embed), polja za unos teksta i ostali. Slika 2 prikazuje kako se dizajn stranice prilagođava različitim dimenzijama ekrana, krajnje lijevo je izgled na mobilnom uređaju, u sredini su uređaji srednje velikih ekrana (npr. tableti) i desno je izgled stranice na uređajima velikog ekrana (stolna / prijenosna računala).



Slika 2 Primjer responzivnog ili prilagodljivog dizajna

Na stranici s popisom znamenitosti koristi se „Datatables“ JavaScript dodatak koji generira dinamičke tablice i mogu koristiti različite izvore podataka. GisSpom `ajax.php` datoteka zadužena je, između ostalog, i za generiranje strukturiranog skupa podataka u JSON formatu iz kojeg Datatables generiraju popis svih spomenika u bazi.

Osim za popis znamenitosti Databases se koristi i za generiranje popisa registriranih korisnika. U prilogu je primjer JSON generiranog koda Kod 7 i Slika 3 prikazuje tablicu generiranu iz JSONa.

```
{
  "data": [
    {
      "id": "4",
      "naziv": "Vatroslav Lisinski",
      "adresa": "Trg Stjepana Radi\u0107a 4, 10000, Zagreb, Croatia",
      "opis": "Vatroslav Lisinski [Zagreb, 1819. - Zagreb, 1854.], hrvatski skladatelj. Prvi je hrvatski profesionalni glazbenik koji je ujedno postavio temelje nacionalnom smjeru u Hrvatskoj",
      "komentar": "Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit. ",
      "autor": "Mile Grgas",
      "kategorija": "Bista"
    },
    {
      "id": "61",
      "naziv": "Njegovateljica ru\u0107eva",
      "adresa": "Ul. kneza Vukobratovi\u0107a 16, 10000, Zagreb, Croatia",
      "opis": "Skulptura izra\u0107ena 1953. postavljena 1955. godine.",
      "komentar": "Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet. ",
      "autor": "Frano Kr\u0107ini\u0107",
      "kategorija": "Skulptura"
    }
  ]
}
```

Kod 7 – Primjer JSON generiranog koda
Example of JSON generated code

Spomenici u sustavu

Ispis svih spomenika u bazi

Dodaj spomenik

Naziv	Autor	Kategorija	Adresa
Andrija Medak	Ivan Meštrović	Skulptura	Trg knaza Tomislava 22, 10000, Zagreb, Croatia
Ante Starčević	Josip Poljan	Skulptura	Trg Ante Starčevića 7, 10000, Zagreb, Croatia
August Šenoa	Ivan Rendić	Bista	Trg Josipa Jurja Strossmayera 3, 10000, Zagreb, Croatia
Blai Josip Jelačić	Alfonso Domínguez Fernández	Spomenik	Trg J. Jelačića, 10000, Zagreb, Croatia
Bazilika Sv. Sreće	Janko Velić	Arhitektura	Platošćeva ul. 33, 10000, Zagreb, Croatia
Benedikt Kružević	Ante Despot	Spomenik	Ul. kneza Branimira 2, 10000, Zagreb, Croatia
Biznani 1	Hermann Balle	Fontana	Trg kneza Šubića Zrinskog 15, 10000, Zagreb, Croatia
Biznani 2	Hermann Balle	Fontana	Trg Nikole Šubića Zrinskog 7-6, 10000, Zagreb, Croatia
Cathedral Defense Tower	Autor nepoznat	Arhitektura	Kaptol ul. 25A, 10000, Zagreb, Croatia
Crkva sv. Marije	Autor nepoznat	Arhitektura	Vrška ul. 36, 10000, Zagreb, Croatia

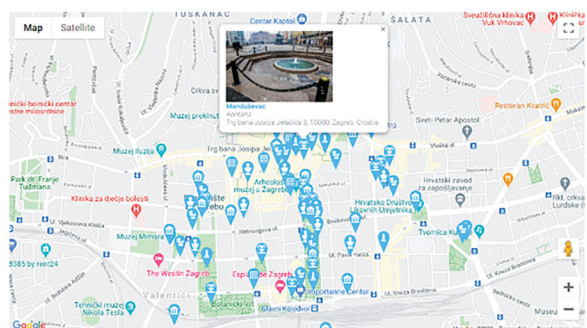
Slika 3 Ispis spomenika u bazi – Databases
Print monuments in the database - Databases

Pomoću JavaScripta generiramo i pribadače (oznake) na karti. Ovdje se isto koristi ajax.php datoteka, ali umjesto JSON formata informacije o oznakama generiramo u XML formatu i JavaScriptom generiramo kartu. U prilogu je primjer XML koda za generiranje oznaka Kod 8, a Slika 4 prikazuje interaktivnu kartu i oznake.

```
<marker id="3" naziv="John Lennon"
adresa="Ul. Pavla Hatza 16, 10000, Zagreb, Croatia" kategorija="Bista" img="slike/3.jpg" lat="45.807598" lng="15.981676" />
<marker id="4" naziv="Vatroslav Lisinski" adresa="Trg Stjepana Radi\u0107a 4, 10000, Zagreb, Croatia" kategorija="Bista" img="slike/4.jpg" lat="45.800697" lng="15.979743" />
</markers>
```

Kod 8 - Primjer XML koda za generiranje oznaka
Example XML code for generating tags

Interaktivna karta

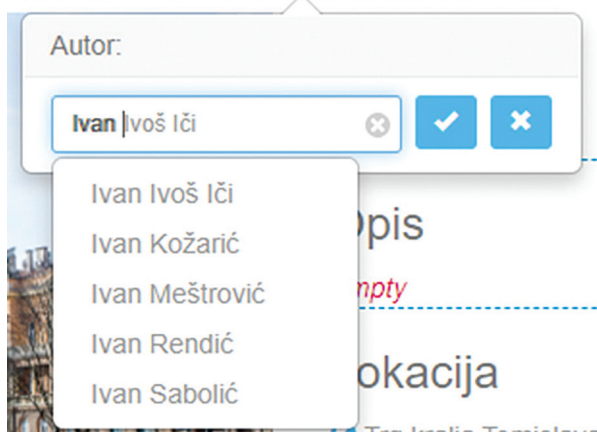


Slika 4 Interaktivna karta – Google Maps
Interactive map - Google Maps

6. TYPEAHEAD I TIMEAGO JAVASCRIPT DODACI

6. TYPEAHEAD AND TIMEAGO JAVASCRIPT PLUGINS

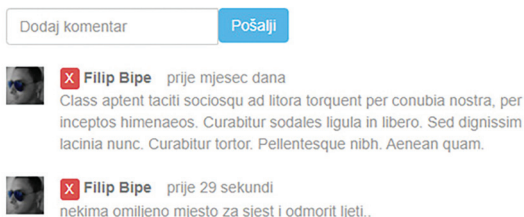
Prilikom dodavanja ili uređivanja spomenika polje autora koristi Typeahead JavaScript dodatak koji također dohvaća podatke iz baze koristeći ajax.php datoteku. Typeahead je dodatak koji automatski pretražuje bazu dok korisnik unosi naziv autora spomenika i na taj način olakšava dodavanje novog spomenika ako je autor već u bazi i smanjuje potencijalne greške prilikom unosa. Slika 5 prikazuje primjer Typeahead dodatka kod uređivanja autora spomenika.



Slika 5 Primjer Typeahead JavaScript dodatka
Example of a Typeahead JavaScript plugin

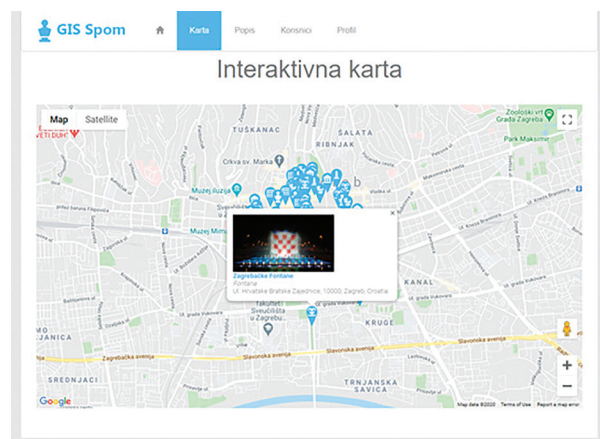
Timeago JavaScript dodatak pretvara točan datum i vrijeme u relativno vrijeme (bilo u prošlosti ili budućnosti) i podatak prikazuje u pisanom obliku poput prije 5 sekundi, 5 minuta, dana, mjeseci ili godina. Typeahead dodatak koristimo na komentarima korisnika koji se prikazuju od najstarijeg do najnovijeg. Slika 6 prikazuje primjer Timeago JavaScript dodatka.

Komentari



Slika 6 Timeago primjer u komentarima
Timeago example in the comments

Na stranici s interaktivnom kartom znamenitosti nalazi se prilagodljivo umetnuta Google Maps karta s pribadačama svih znamenitosti u sustavu. Različite ikonice unutar pribadača predstavljaju o kojoj se vrsti znamenitosti radi: bisti, spomeniku, fontani, reljefu, skulpturi ili arhitekturi. Klikom na pribadaču prikazuju se osnovne informacije o znamenitosti, a to su: slika, naziv, kategorija i lokacija u ljudski čitljivom obliku (puna adresa lokacije). Slika 7 prikazuje stranicu interaktivne karte.



Slika 7 Interaktivna karta znamenitosti
Interactive landmark map

7. INTEGRACIJA S TELEGRAM APLIKACIJOM

7. INTEGRATION WITH THE TELEGRAM APPLICATION

Kao proširenje već postojeće aplikacije odlučili smo se integrirati GisSpom s Telegram aplikacijom za brzu razmjenu poruka. Telegram nudi mogućnost kreiranja chatbotova i integracije s drugim sustavima. Chatbot je računalni program (bot) koji automatizira određene zadatke, obično razgovorom s korisnikom putem konverzijskog sučelja.

Kroz Telegram aplikaciju kreirali smo novog Bot korisnika s imenom GSBot, a integracija s GisSpom aplikacijom napravljena je korištenjem Webhook metode. Kako bi mogli aktivirati webhook potrebno je napraviti POST upit prema Telegram API destinaciji koji u svojem URLU sadržava ID i token bota koji smo dobili prilikom kreiranja bota. POST upit mora sadržavati tijelo JSON sadržaja unutar kojeg definiramo „url“ lokaciju na koju će webhook raditi upit.

```
{
  "url": "https://gisspom.site/ajax.
  php?TelegramBotToken"
}
```

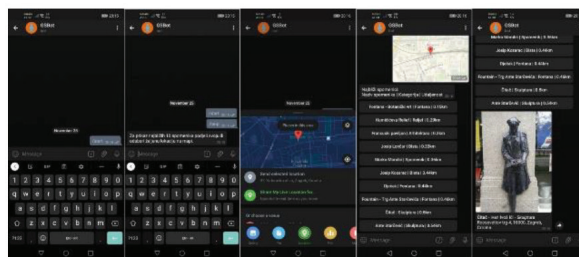
Kod 9 - Telegram API

Sada chatbot preusmjerava sve poruke prema definiranoj lokaciji potrebno je programirati na što će odgovarati. U config.php datoteci definirali smo varijablu \$botToken koja sadržava tajni token koji se koristi u komunikaciji s botom. Telegram API prihvaća slanje parametara putem GET i POST metode s podrškom zahtjeva u obliku URL niza upita (URL query string), application/x-www-form-urlencoded, application/json ili multipart/form-data. Za izradu GSBota odabrali smo POST metodu s URL niz upita.⁹

Unutar functions.php datoteke definirali smo nekoliko novih funkcija koje služe za slanje poruka botu, prvo smo definirali funkciju „slanjePoruke“ s parametrima url i content. Parametar url sadržavati će URL na koji bot treba poslati odgovor, a content će biti sadržaj poruke. Funkcija će odgovor poslati korištenjem PHP curl biblioteke na zadani URL. Zatim smo kreirali tgHelp funkciju s parametrom chatId koja će na korisnikovu poruku /help poslati kratke upute o korištenju bota.

Kroz par slika možemo vidjeti primjer korištenja novokreiranog chatbota. Slika 16 prikazuje korištenje GSBot chatbot proširenja, s lijeva na desno:

- Početak razgovora s botom.
- Slanje /help poruke.
- Slanje željene lokacije.
- Bot odgovara s 10 najbližih znamenitosti, njihovim nazivima, kategorijama i udaljenosti od podijeljene lokacije.
- Detaljne informacije o željenoj znamenitosti

Slika 8 Primjer Telegram chatbot integracije
Example Telegram chatbot integration

8. ZAKLJUČAK

8. CONCLUSION

Ovim radom opisana je aplikacija GisSpom – online baza znamenitosti, tehnologije korištene prilikom izrade i integracija s Telegram aplikacijom. Opisali smo sve važne komponente aplikacije i kako su realizirane korištenjem tehnologija otvorenog koda poput PHPa, MySQLa i Bootstrapa.

Upoznali smo se sa strukturom podataka unutar MySQL baze i strukturu PHP datoteka. Obradili smo korištenje Google OAuth usluge za registraciju i prijavu korisnika u aplikaciju, integraciju Google Maps karte i Google reverse geocoding API.

Napravili smo kratki pregled Bootstrap programskog okvira, njegovih mogućnosti i kako smo ga primijenili u izradi aplikacije. Iako većinu podataka na stranici generira PHP, dio podataka prikazuje se korištenjem JavaScript AJAX upita, na primjer interaktivna karta ili tablice kod ispisa znamenitosti.

Napravili smo pregled svih stranica aplikacije i opisali kako smo realizirali određene funkcionalnosti. Na kraju smo napravili integraciju aplikacije s Telegram aplikacijom u obliku chatbota. U današnje vrijeme chatbotovi postaju dio svakodnevice i kroz primjer proširivanja funkcionalnosti postojeće aplikacije pokazali smo samo jednu od mogućih opcija za proširivanje, ali dovoljno kako bi se shvatio koncept i mogućnosti koje nam nude platforme poput Telegrama.

⁹ Telegram – Bots – URL: <https://core.telegram.org/bots/api#making-requests> – 21.11.2020

9. REFERENCE

9. REFERENCES

- [1.] Šimec, Alen; Programiranje i optimizacija Internet stranica u HTML5 okruženju; Tehničko veleučilište u Zagrebu; 2015; ISBN 978-953-7048-55-6
- [2.] Šimec, Alen; Osnove HTML, XHTML i CSS; Tehničko veleučilište u Zagrebu; 2011; ISBN 978-953-7048-19-8
- [3.] Nixon, Robin; Learning PHP, MySQL, JavaScript, CSS & HTML5, 3rd Edition; O'Reilly Media; 2014.; ISBN-13: 978-1491949467
- [4.] Seyed M.M. "Saied" Tahaghoghi; Hugh E. Williams; Learning MySQL; O'Reilly Media; 2007; ISBN-13: 978-0596008642
- [5.] Filipova Olga; Learning Vue.js 2; Packt Publishing, 2016.; ISBN: 9781786461131
- [6.] Šimec Alen, Davor Lozić; Extending PHP with modules; Polytechnic & Design; Vol. 3, No. 1, 2015.; ISSN: 1849 – 1995
- [7.] Šimec, Alen; Lozić, Davor; Basic PHP Implementations, Opcodes and Internal Work; MIPRO 2015 - 38. međunarodni skup; ISSN: 1847-3946.

AUTORI · AUTHORS

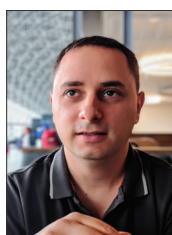
● **Alen Šimec**

Rođen je 19.01.1979. U Zagrebu završava III. Gimnaziju, stručni dodiplomski studij na Tehničkom veleučilištu u Zagrebu (2002.), te stječe stručno zvanje inženjer informatike. Diplomirao je na

stručnom dodiplomskom studiju na Visokoj školi za sigurnost, te je stekao zvanje diplomiranog inženjera sigurnosti (2006.). Diplomirao je na Specijalističkom diplomskom stručnom studiju politehnike, smjer informatika (2008). Doktorirao je na Filozofskom fakultetu Sveučilišta u Zagrebu, smjer Informacijske i komunikacijske znanosti, te stječe titulu Doktora znanosti, područje društvene znanosti, polje informacijske i komunikacijske znanosti. Dr. sc. Alen Šimec sudjelovao je na predavanjima i razgovorima o unaprjeđenju suradnje tvrtke Microsoft s akademskom zajednicom, pohađao je i sudjelovao na stručnim i znanstvenim konferencijama sa radovima koji su objavljeni u tiskanom i digitalnom izdanju. Napisao je tri knjige objavljene u Nacionalnoj sveučilišnoj biblioteci, te radi na Tehničkom veleučilištu kao viši predavač, te je član neprofitne organizacije Rotary klub Zagreb.

Korespondencija · Correspondence

alen.simec@tvz.hr

● **Filip Penezić**

Rođen 1984. u Zagrebu, 2017. završio preddiplomski studij na Veleučilištu Velika Gorica smjer Održavanje računalnih sustava. 2019. upisuje specijalistički studij na Tehničkom Veleučilištu

Zagreb, smjer E-Poslovanje. Od 2014. do 2019. vodi hrvatsku podružnicu informatičke tvrtke BlueBoxIT sa sjedištem u Australiji i obavlja poslove sistemskog inženjera. Trenutno zaposlen kao sistem inženjer u Bulb d.o.o. u Zagrebu.