

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ivan Lovrić

**Interaktivna spletna aplikacija za
slovenska narečna besedila**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Alenka Kavčič

SOMENTOR: red. prof. dr. Vera Smole

Ljubljana, 2018

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuira, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Interaktivna spletna aplikacija za slovenska narečna besedila

Tematika naloge:

V okviru diplomske naloge zasnujete in izdelajte interaktivno spletno aplikacijo za prikaz narečnih besedil na karti slovenskih narečij. Aplikacija naj za podane kraje omogoča predvajanje zvočnih posnetkov narečnih besedil ter prikaz njihove transkripcije, poknjžitve in analize besedil. Aplikacija naj vključuje tudi administrativni del, v katerem lahko uporabnik dodaja nove kraje slovenskega govornega področja s pripadajočimi zvočnimi posnetki in pisnimi materiali. Uporabniški vmesnik aplikacije naj bo intuitiven in enostaven za uporabo.

Zahvaljujem se viš. pred. dr. Alenki Kavčič za mentorstvo in pomoč pri realizaciji diplomske naloge, somentorici prof. dr. Veri Smole z Oddelka za slovenistiko na Filozofski fakulteti Univerze v Ljubljani za predloge, popravke ter besedilne in zvočne vsebine, Poloni Štefanič za vektorsko različico narečnega zemljevida in dekletu Francesci za pomoč pri vnašanju vsebin v sistem, motivacijo in moralno podporo.

Posebna zahvala gre mojim staršem za podporo, potrpežljivost in vse, kar so mi v življenju omogočili.

Hvala tudi vsem kolegom in kolegicam s fakultete, zaradi katerih je bil študij računalništva in informatike pravi užitek.

Mojim staršem in starim staršem, ki so vsa
ta leta (bolj ali manj) potrpežljivo čakali
na to diplomsko delo.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Predstavitev problema	3
2.1	Pregled obstoječih narečnih zemljevidov	3
2.1.1	Zemljevid narečij bolgarskega jezika	4
2.1.2	Zvočna karta naglasov in narečij Velike Britanije	5
2.2	Uporabniška aplikacija	5
2.3	Skrbniška aplikacija	7
3	Uporabljene tehnologije, orodja in knjižnice	9
3.1	Strežnik Apache	9
3.2	Podatkovna baza MySQL	10
3.3	HTML5	10
3.4	CSS	10
3.5	Bootstrap	11
3.6	JavaScript	11
3.7	AJAX	11
3.8	AngularJS	12
3.9	PHP	12
3.10	Format GeoJSON	13

3.11 Spletni zemljevidi Google Maps	13
3.12 Leaflet	13
3.13 Knjižnica WavesurferJS	14
3.14 Sistem ZRCola	14
3.15 Git	14
4 Izdelava aplikacije	15
4.1 Priprava razvojnega okolja	15
4.2 Izdelava zemljevida narečij	15
4.3 Podatkovni model	18
4.4 REST API	20
4.5 Skrbniška aplikacija	21
4.5.1 Pregled vnosov	21
4.5.2 Urejanje vnosov	22
4.5.3 Komunikacija z bazo	23
4.5.4 Avtentikacija in avtorizacija	25
4.6 Preoblikovanje kode za delovanje v ogrodju AngularJS	27
4.7 Preoblikovanje kode za uporabo knjižnice Leaflet	27
5 Sklepne ugotovitve	29
Literatura	31

Seznam uporabljenih kratic

kratica	angleško	slovensko
AJAX	Asynchronous JavaScript and XML	asinhroni JavaScript in XML
API	application programming interface	aplikacijski programski vmesnik
CSS	Cascading Style Sheets	kaskadne stilske podloge
DOM	Document Object Model	objektni model dokumenta
HTML	Hyper Text Markup Language	jezik za označevanje nadbese-dila
JSON	JavaScript Object Notation	objektna notacija JavaScript
MP3	MPEG-1 Audio Layer III	format zapisa glasbe z izgu-bami
MVC	Model—view—controller	model-pogled-krmilnik
PHP	PHP: Hypertext Preprocessor	skriptni programski jezik
REST	Representational state transfer	arhitektura za izmenjavo po-datkov med spletnimi stori-tvami
SVG	Scalable Vector Graphics	umerljiva vektorska grafika
URL	Uniform Resource Locator	enolični krajevnik vira
XML	Extensible Markup Language	razširljivi označevalni jezik

Povzetek

Naslov: Interaktivna spletna aplikacija za slovenska narečna besedila

Avtor: Ivan Lovrić

Slovenščina pozna šestinpetdeset narečij, združenih v sedem narečnih skupin. Veliko literature o slovenskih narečjih v fizični obliki že obstaja, cilj te diplomske naloge pa je bil razviti zanimivo, interaktivno, spletno aplikacijo, ki bi na zemljevidu omogočala pregled vseh narečij, njihovih značilnosti in poslušanje zvočnih zapisov narečnih govorov. Poleg tega je bilo potrebno izdelati skrbniški del za dodajanje novih narečnih vnosov. Začetni del aplikacije smo razvili z uporabo tehnologij, kot so HTML5, CSS, JavaScript, Google Zemljevidi in ogrodje AngularJS, zaledni del pa z jezikom PHP in podatkovno bazo MySQL. Z diplomsko nalogo smo izdelali prvo takšno aplikacijo za slovenska narečja.

Ključne besede: narečje, narečna skupina, spletna aplikacija, zemljevid, Slovenija, GeoJSON.

Abstract

Title: Interactive web-based application for Slovenian dialectal texts

Author: Ivan Lovrić

The Slovene language has 56 dialects, grouped into 7 dialect groups. A lot of literature on Slovene dialects already exists in physical form, but the goal of this thesis was to develop an interesting, interactive web application which would let its users see all the dialects on a map, read their characteristics and listen to their sound recordings. In addition, an administrative part was created to allow the addition of more examples. The front end part was developed using technologies such as HTML5, CSS, JavaScript, Google Maps and the AngularJS framework. The back end was created using the PHP language and the MySQL database. Consequently we created the first application of this kind for Slovenian dialects.

Keywords: dialect, dialect group, web application, map, Slovenia, GeoJSON.

Poglavje 1

Uvod

Med slovanskimi jeziki je slovenščina narečno najbolj členjena. Pozna namreč šestinpetdeset narečij, ki jih združujemo v sedem narečnih skupin: gorenjsko, dolensko, štajersko, panonsko, koroško, primorsko in rovtarsko.

Najstarejši slovenski zapisi, Brižinski spomeniki, ki so nastali v 10. ali 11. stoletju, še ne kažejo narečnih značilnosti, Celovski (14. stoletje) in Stiški rokopis (15. stoletje) pa že kažeta znake narečne členjenosti [8].

Na nastanek narečij je vplivalo več dejavnikov. En izmed teh so geografske ovire, npr. visoka gorovja, ki jih ljudje niso pogosto prehajali. Poleg tega se ljudje niso dnevno selili v mesta in so se tako jezikovne skupnosti oblikovale znotraj posameznih far. Prometne poti, ki so vodile v večja mesta, so ponavadi potekale po sredini narečnih skupin. Na narečja ob državnih mejah pa je vplivala tudi bližina drugih jezikov. Na Primorskem se tako pojavljajo italijanizmi, na Dolenjskem hrvatizmi, germanizmi so pogosti v štajerskih in panonskih narečjih, v Prekmurju zaznamo vplive madžarščine.

O slovenskih narečjih obstaja že veliko literature, veliko je dostopne tudi na internetu. Med prve pomembnejše slovenske narečjeslovce štejemo Frana Ramovša (1890-1952) in Tineta Logarja (1916-2002). Slednji je prehodil celotno slovensko jezikovno območje in zapisal več kot 200 govorov za Slovenski lingvistični atlas [11]. Sodeloval je tudi pri mnogih drugih pomembnih delih na področju slovenskih narečij in izkušnje kot profesor prenašal na nove

rodove slovenistov [14].

Od študijskega leta 2009/10 se na Oddelku za slovenistiko na Filozofski fakulteti Univerze v Ljubljani (odslej: Oddelek za slovenistiko) izvaja izbirni predmet Slovenska narečja, pri katerem študentje pri seminarski nalogi posnamejo zvočni zapis nekega govora, zapišejo transkripcijo posnetka s fonetičnim zapisom in poknjžitev (zapis istega besedila v knjižni slovenščini) ter izdelajo analizo jezikovnih značilnosti govora [4]. Del teh zapisov je zbran v publikaciji Stara kmečka hiša, Narečna besedila z analizo I [2], ki sta jo pripravili prof. dr. Vera Smole in dr. Mojca Horvat.

Glavni namen te diplomske naloge je bil izdelati interaktivni spletni zemljevid slovenskih narečij, ki bi omogočal enostaven in uporabniku prijazen pregled vseh elementov zbranih govorov: zvočnih zapisov, transkripcij, poknjžitev in analiz ter lingvistom omogočiti vnašanje novih.

Aplikacija je tako sestavljena iz dveh delov: uporabniškega in skrbniškega. Uporabniški del vsebuje spletni zemljevid, na katerem so barvno označena vsa slovenska narečja, podnarečja in narečne skupine, ter legendo. Na zemljevid so nato dodane oznake na lokacijah, kjer so bili zvočni zapisi posneti. Ob kliku na oznako se odpre pojavno okno, v katerem lahko uporabnik predvaja zvočni zapis in si prebere transkripcijo, poknjžitev in analizo govora.

Skrbniški del vsebuje avtentikacijski obrazec za dostop do urejevalnika, ob uspešni prijavi se prikaže seznam vseh vnesenih govorov. Prijavljeni uporabnik ima možnost urejati obstoječe ali dodajati nove vnose.

Poglavje 2

Predstavitev problema

Cilj naloge je bil izdelati atraktiven in uporabnikom prijazen interaktivni spletni zemljevid slovenskih narečij z možnostjo poslušanja posameznih zvočnih zapisov, pregleda transkripcij, poknjžitev in analiz posnetkov ter s skrbniško aplikacijo omogočiti urejanje in dodajanje novih narečnih vnosov.

Zahtevani končni rezultat je spletna aplikacija, zato je bilo najprej potrebno izbrati tehnologije in orodja za njeno izdelavo, kot so spletni strežnik, podatkovna baza, programski jeziki za čelni (angl. *front end*) in zaledni del (angl. *back end*) aplikacije ipd. Na izbiro tehnologij so vplivali dejavniki, kot so avtorjevo predhodno poznavanje le-teh, brezplača uporaba in preprosta integracija na produkcijski strežnik, ki ga bo upravljal Oddelek za slovenistiko.

2.1 Pregled obstoječih narečnih zemljevidov

Na spletu lahko najdemo veliko spletnih strani s podatki o narečjih raznih jezikov. Prav tako lahko najdemo tudi nekaj interaktivnih aplikacij, med katerimi so nekatere bolj, druge manj prilagojene spletnemu mediju in se zato med njimi razlikuje tudi uporabniška izkušnja. Poblížje si oglejmo dve izmed njih.

2.1.1 Zemljevid narečij bolgarskega jezika

Zemljevid narečij bolgarskega jezika [17] je bil zgled za to diplomsko nalogo. Izdelal ga je Oddelek za dialektologijo in jezikovno geografijo Inštituta za bolgarski jezik in je prvi elektronski interaktivni zemljevid bolgarskih narečij. Prikazuje glavne narečne skupine celotnega jezikovnega ozemlja in predstavi vsa glavna narečja in njihove meje. Po zemljevidu so razporejene ikone, ki ob kliku prikažejo krajšo analizo ali predvajalnik zvočnega zapisa narečja. Poleg tega omogoča dopolnjevanje karte z novimi zvočnimi zapisi [15]. Zemljevid je prikazan na sliki 2.1.



Slika 2.1: Zemljevid narečij bolgarskega jezika

Uporabniška izkušnja pri uporabi bolgarskega zemljevida narečij je dobra, vseeno pa bi se jo dalo še izboljšati. Npr. legenda je po vzoru fizičnih kart fiksna v zgornjem desnem kotu zemljevida. Če uporabnik zemljevid približa in se po njem z miško pomika, legenda ni več vidna. Za izboljšanje izkušnje

bi lahko legendo naredili lebdečo, da je vidna vedno, ne glede na to kateri del zemljevida si uporabnik ogleduje.

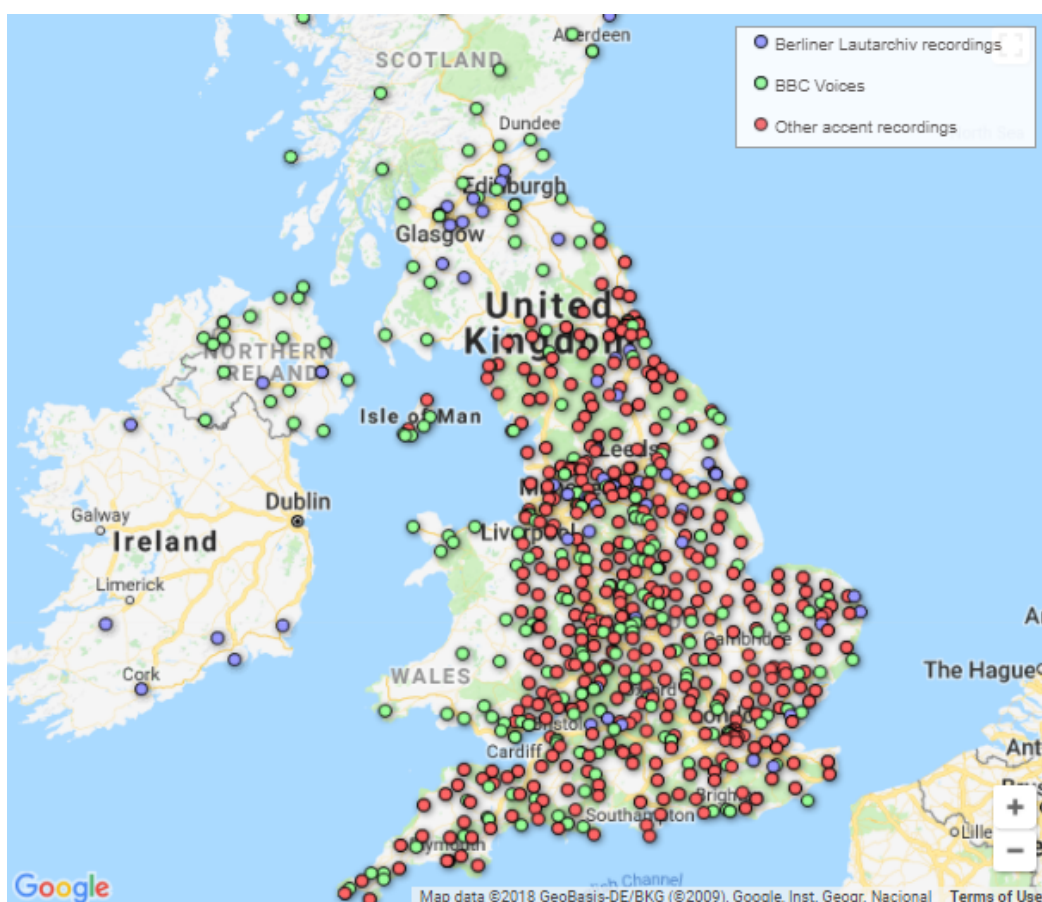
2.1.2 Zvočna karta naglasov in narečij Velike Britanije

Zvočna karta naglasov in narečij Velike Britanije [1] je dostopna na spletnih straneh The British Library in je primer karte, ki za prikaz podatkov uporablja zemljevide Google Maps. Ne vsebuje tekstovnih besedil, temveč le zvočne zapise. Uporabniška izkušnja ni najboljša, saj se predvajalnik zvoka ne odpre v modalnem oknu, ampak uporabnika ob kliku na zvočni zapis z novo zahtevo (brez tehnologije AJAX) preusmeri na drugo stran. Karta je prikazana na sliki 2.2.

2.2 Uporabniška aplikacija

Za uporabniško aplikacijo je bilo potrebno izbrati sistem oz. knjižnico za upodobitev geografskih podatkov – zemljevida z vrisanimi narečnimi skupinami, narečji, podnarečji in oznakami na točno določenih zemljepisnih dolžinah in širinah. Narečnih vnosov bo na zemljevidu vedno več, zato je bilo potrebno izbrati takšen sistem, ki omogoča poljubno približevanje in premikanje po zemljevidu.

Izdelati je bilo treba tudi vektorski izris zemljevida slovenskih narečij. Za osnovo smo vzeli Karto slovenskih narečij [3], ki sta jo pripravila Tine Logar in Jakob Ringler na osnovi Ramovševe Dialektološke karte slovenskega jezika. Vektorsko grafiko smo nato morali geokodirati – pretvoriti v format, ki vsebuje geografske podatke za vsako točko in ki ga lahko uporabimo oz. uvozimo v obstoječo knjižnico za izdelavo zemljevidov. To smo želeli doseči zato, da lahko v aplikaciji omogočamo več interaktivnosti – dinamično spreminjanje oblikovnih stilov narečnih področij in mej med njimi, odzivanje na uporabniške vnose in geste z miško, kot je npr. lebdenje z miškinim kazalcem nad posameznim narečjem ipd. Enostavnejša, vendar neinteraktivna in zato uporabnikom manj prijazna rešitev bi bila uvoz bitne slike zemljevida narečij



Slika 2.2: Zvočna karta naglasov in narečij Velike Britanije

kot prekrivek (angl. *overlay*) obstoječemu zemljevidu.

Na zemljevidu je bilo potrebno prikazati tudi vse oznake lokacij, kjer so bili posneti zvočni zapisi. Za izvedbo tega dela diplomske naloge smo morali najprej pripraviti skrbniško aplikacijo za vnašanje podatkov, ki smo jih dobili od Oddelka za slovenistiko.

Zaradi dobre uporabniške izkušnje nismo želeli omejiti predvajanja zvočnega posnetka na funkciji predvajaj in ustavi, temveč zvok tudi upodobiti z valovnimi oblikami (angl. *waveforms*) in omogočiti premikanje naprej in nazaj po posnetku, za kar smo morali uporabiti ustrezno knjižnico za vizualizacijo zvoka.

Ključnega pomena za izdelavo aplikacije je bila priprava pisave (angl. *font*) ZRCola za uporabo na svetovnem spletu. Vse transkripcije in analize govorov so namreč bile pripravljene z vnašalnim sistemom ZRCola in istoimensko pisavo [18], ki ju je na Znanstvenoraziskovalnem centru SAZU v Ljubljani razvil Peter Weiss. Pisavo je bilo potrebno pretvoriti v ustrezen format za uporabo v brskalnikih in jo vključiti v stilske podloge CSS, ki določajo obliko spletne strani.

2.3 Skrbniška aplikacija

Skrbniška aplikacija je na prvem mestu zahtevala avtentikacijo, saj je namenjena narečjeslovcem in nismo želeli dovoliti vsem uporabnikom dostopa do zaledja, kjer bi lahko poljubno spreminjali ali celo brisali vnose. Uporabnikom, ki bi se uspešno prijavi v sistem, smo želeli omogočiti čim bolj preprost, minimalističen pregled, urejanje in dodajanje novih narečnih vnosov. Na temelju tega in same narave podatkov je bilo potrebno pripraviti strukturo podatkovne baze. Kljub temu da ne pričakujemo zelo velikega števila hkratnih uporabnikov aplikacije, smo podatkovno bazo tudi malenkostno optimizirali za hitrejše delovanje, kar bomo predstavili v poglavju o izdelavi aplikacije 4.3.

Poglavje 3

Uporabljene tehnologije, orodja in knjižnice

Sveta brez spleta si v današnjem času sploh ne moremo več predstavljati, saj je prisoten skoraj povsod. Področje spletnih tehnologij se zato izjemno hitro razvija. Če želimo pritegniti čim večje število uporabnikov, moramo pri delu s spletnimi aplikacijami ves čas spremljati nove tehnologije, saj nam lahko zelo olajšajo delo in uporabnikom ponudijo boljšo izkušnjo. Pri izdelavi narečnega zemljevida smo zato poleg nekaterih dolgo uveljavljenih tehnologij uporabili tudi nekaj novejših, kot so HTML5, ogrodje AngularJS (v času izdelave te diplomske naloge je prišla v uporabo že novejša različica ogrodja pod imenom Angular) in nekatere knjižnice, kot je npr. Wavesurfer.js za prikaz valovnih oblik zvoka v brskalniku.

3.1 Strežnik Apache

Strežnik Apache je odprtokodni strežnik za spletne strani in je že od leta 1996 najbolj razširjen strežnik HTTP na svetovnem spletu [13]. Razvijalci strežnika Apache stremijo k temu, da bi bil karseda varen, učinkovit in razširljiv in da bi zagotavljal najnovejše storitve sočasno z razvojem novih standardov protokola HTTP.

3.2 Podatkovna baza MySQL

MySQL je odprtokodni sistem za upravljanje z relacijskimi podatkovnimi bazami. Tako kot pri vseh relacijskih podatkovnih bazah so tudi v sistemu MySQL podatki združeni v tabele, ki so razdeljene na vrstice, kjer vsaka vrstica predstavlja en vnos v tabelo, in stolpce, ki predstavljajo attribute vnosov. Po podatkih poizvedujemo in jih upravljamo s strukturiranim poizvedovalnim jezikom SQL.

Za pridobivanje vseh nazivov „name“ iz tabele narečij „dialects“, ki spadajo pod primorsko narečno skupino, bi uporabili npr. poizvedbo na sliki 3.1.

```
SELECT name
FROM dialects
WHERE dialect_group = 'primorska_narecna_skupina';
```

Slika 3.1: Primer poizvedbe v podatkovni bazi MySQL

3.3 HTML5

HTML5 je standard, ki obsega dva koncepta: zadnjo različico jezika za označevanje nadbesedila HTML (angl. *Hyper Text Markup Language*), ki vsebuje nove elemente, attribute in delovanja, ter zbirko tehnologij, ki omočajo bolj raznoliko gradnjo spletnih strani in aplikacij, kot so tehnologije za zvok, video in druge [7]. HTML je osnova za izdelavo vsake spletne strani.

3.4 CSS

Kaskadne stilske podloge CSS (angl. *Cascading Style Sheets*) so slogovni jezik za oblikovanje dokumentov, zapisanih v označevalnih jezikih, kot je HTML. Uprabljam ga za oblikovanje spletnih strani in z njim določamo lastnosti, kot so barve, pisave, velikosti, odmiki in podobno. Zadnja različica

jezika, CSS3, med ostalim omogoča tudi izdelavo pravil za izdelavo animacij na spletnih straneh.

3.5 Bootstrap

Bootstrap je brezplačno odprtokodno ogrodje za oblikovanje spletnih strani, ki vsebuje predpripravljene predloge HTML in CSS (opcijsko tudi JavaScript za interaktivne komponente) za tipografijo, obrazce, gumbe, navigacijo in podobno. Privzeto ima vgrajeno tudi odzivno oblikovanje (angl. *responsive design*) za prilagojen prikaz na mobilnih napravah. V skrbniški aplikaciji smo Bootstrap uporabili za oblikovanje tabel, obrazcev in gumbov, v uporabniški aplikaciji pa za pojavna okna (angl. *modal windows*), ki jih prikažemo ob miškinem kliku na oznake na zemljevidu in ki vsebujejo podatke, besedila in predvajalnik zvočnega zapisa izbranega narečja.

3.6 JavaScript

JavaScript je visokonivojski šibko tipiziran skriptni jezik, ki je bil razvit za dodajanje interaktivnosti spletnim stranem in se ga lahko vstavi neposredno v kodo HTML (slika 3.2). Je zelo vsestranski, saj omogoča imperativno, objektno orientirano (temelječ na prototipih), funkcionalno in dogodkovno gnano (angl. *event-driven*) programiranje. Po sintaksi je podoben jeziku C. Najpogosteje se uporablja kot jezik na strani odjemalca (angl. *client-side*), v uporabnikovem brskalniku. Z razvojem platforme Node.js se JavaScript vse bolj pogosto uporablja tudi kot jezik na strani strežnika (angl. *server-side*). V diplomski nalogi smo ga uporabili na strani odjemalca.

3.7 AJAX

AJAX je naveza več tehnologij, ki omogočajo komunikacijo odjemalca (brskalnika) s strežnikom v ozadju brez potrebe po ponovnem nalaganju celotne

```
<script type="text/javascript">
    alert("Pozdravljen svet!");
</script>
```

Slika 3.2: Primer vključevanja kode JavaScript v HTML

spletne strani. Večina modernih ogrodij za izdelavo aplikacij na strani odjemalca danes uporablja AJAX za nemoteno pridobivanje podatkov s strežnika in s tem prijaznejšo uporabniško izkušnjo.

3.8 AngularJS

AngularJS je odprtokodno ogrodje za razvoj spletnih aplikacij na strani odjemalca v JavaScriptu z arhitekturo MVC. Deluje tako, da AngularJS najprej prebere stran v označevalnem jeziku HTML. HTML mora na elementih vsebovati posebne attribute, ki jih AngularJS interpretira kot direktive za povezovanje vhodnih in/ali izhodnih delov spletne strani z modelom, ki ima obliko navadne spremenljivke v JavaScriptu. Takšna povezava omogoča, da se vsaka sprememba modela (recimo, če krmilnik sproži asinhrono zahtevo AJAX za pridobivanje novih podatkov s strežnika) takoj odzove in manifestira tudi na pogledu. Prikazovanje podatkov je tako preprostejše, saj ni več potrebe po ročnem manipuliranju kode HTML in modela DOM ob vsaki uspešni zahtevi za pridobivanje podatkov.

3.9 PHP

PHP je zelo razširjen odprtokodni programski jezik. Tako kot JavaScript je tudi PHP po sintaksi podoben jeziku C. Za razliko od JavaScripta pa PHP deluje na strežniški strani in se upravlja predvsem za generiranje spletnih strani na strežniku in za komunikacijo s podatkovno bazo. V diplomski nalogi smo PHP uporabili za izdelavo enostavnega aplikacijskega programskega vmesnika REST za pridobivanje in upravljanje podatkov o slovenskih

narečjih in uporabnikov (za avtentikacijo) ter za komuniciranje s podatkovno bazo z uporabo PHP-jeve razširitve MySQLi.

3.10 Format GeoJSON

GeoJSON je format za kodiranje raznih zemljepisnih podatkovnih struktur: točk, linij, poligonov in podobno v notaciji JSON. Format je natančno definiran v spletnem dokumentu RFC7946 [10]. Zemljevid slovenskih narečij smo zakodirali v format GeoJSON in ga uvozili v zemljevide Google Maps in Leaflet.

3.11 Spletni zemljevidi Google Maps

Google Maps je spletna storitev za kartiranje, ki jo je razvilo podjetje Google. Preko programske knjižnice Google Maps API omogočajo zelo veliko možnosti za prilagajanje in prikazovanje najrazličnejših geografskih podatkov. Storitve smo uporabili za prikaz slovenske narečne karte, kodirane v formatu GeoJSON, in oznak na zemljepisnih lokacijah, kjer so bili zvočni zapisi narečij posneti.

3.12 Leaflet

Leaflet [5] je odprtokodna knjižnica v jeziku JavaScript za izdelavo spletnih kartografskih aplikacij. Poleg Google Maps je ena izmed najbolj priljubljenih spletnih knjižnic za kartiranje. Prednost knjižnice Leaflet pred knjižnico Google Maps je odprtokodnost in možnost uporabe zelo velikega števila razširitev, ki so jih razvili uporabniki.

3.13 Knjižnica WavesurferJS

Knjižnica Wavesurfer.js je prilagodljiva vizualizacija zvočnih valovnih oblik, zgrajena na osnovi spletnih tehnologij HTML5 in aplikacijskega programskega vmesnika Web Audio API, ki se uporablja za predvajanje in upravljanje zvočnih posnetkov na spletu. Uporabili smo jo za grafično upodobitev zvočnih zapisov slovenskih narečij in za lažje premikanje naprej in nazaj po posnetku.

3.14 Sistem ZRCola

Sistem ZRCola sestavljata vnašalni sistem in pisava. Vnašalni sistem uporabniku omogoča, da pri pisanju črke dopolnjuje s poljubnimi diakritičnimi znaki in tako v besedilo enostavno vstavlja nove, kompleksne znake. Pisava temelji na standardu Unicode in vsebuje velik nabor znakov vseh slovanskih pisav. V uporabniški in skrbniški aplikaciji smo za pravilen prikaz transkripcij zvočnih posnetkov slovenskih narečij uporabili pisavo ZRCola, ki je dostopna na spletnih straneh Znanstvenoraziskovalnega centra SAZU [18] .

3.15 Git

Git je sistem za upravljanje z izvorno kodo, ki vse spremembe shranjuje in nudi vpogled v zgodovino ter možnost hitre povrnitve na katero koli izmed prejšnjih stanj izvorne kode. Omogoča tudi hkratno urejanje izvorne kode več uporabnikov, saj ponuja reševanje konfliktov, ko več programerjev poskuša urediti isti del kode. Sistem smo uporabili zaradi sprotnega shranjevanja in vpogleda v zgodovino ter lažjega razhroščevanja kode.

Poglavje 4

Izdelava aplikacije

4.1 Priprava razvojnega okolja

Izdelave diplomske naloge smo se lotili s pripravo razvojnega okolja na operacijskem sistemu Microsoft Windows. Namestili smo razvojno okolje Wamp-Server, ki že vsebuje vse potrebno za pričetek dela: spletni strežnik Apache2, podatkovno bazo MySQL, programski jezik PHP in knjižnico OpenSSL za podporo varni povezavi med strežnikom in odjemalcem. Kot urejevalnik izvirne kode smo največ uporabljali Microsoftov urejevalnik Visual Studio Code.

4.2 Izdelava zemljevida narečij

Bitno sliko Karte slovenskih narečij Tineta Logarja in Jakoba Ringlerja, ki je dostopna tudi na spletnih straneh Znanstvenoraziskovalnega centra Slovenske akademije znanosti in umetnosti [3], smo uvozili v program za risanje vektorskih grafik, Adobe Illustrator, in z miško prerasali vse meje narečnih območij. Novonastalo karto je bilo potrebno nekako predelati v obliko GeoJSON, ker pa Adobe Illustrator tega ne omogoča, smo na spletu poiskali orodje, s katerim bi si lahko pomagali. Našli smo JavaScriptov paket `svg-to-geojson` [12], ki zna pretvoriti v format GeoJSON vektorske gra-

fike SVG. Funkcija `svgToGeoJson` (na sliki 4.1) sprejme tri parametre: dve zemljepisni koordinati, ki predstavljata skrajno severovzhodno in skrajno jugozahodno točko vektorske grafike, element SVG in parameter `CurveComplexity`, ki predstavlja število ravnih linij, s katerimi bo upodobljena vsaka krivulja. Format GeoJSON namreč ne podpira idealnih krogov, zato jih pretvori v poligone z N ravnimi linijami.

```
var geoJson = svgtogeojson.svgToGeoJson(  
    [[46.98, 16.67], [45.385, 13.12]],  
    document.getElementById('mysvg'),  
    3  
);
```

Slika 4.1: Primer klica funkcije za zagon pretvorbe elementa SVG v format GeoJSON

Dobljeni rezultat je v formatu GeoJSON, ki smo ga z metodo *addGeoJson* programskega vmesnika Google Maps API dodali na naš zemljevid. Z metodo *setStyle* smo nato nastavili barvo in pozicijo z-index vsakemu izmed območij GeoJSON (angl. *GeoJSON features*), kjer vsako območje predstavlja narečno skupino, narečje ali podnarečje. Da bi ob lebdenju z miško nad posameznim območjem lahko to območje vizualno poudarili, smo narečnim skupinam nastavili najnižjo vrednost z-index, podnarečjem pa najvišjo, zato da večja območja ne prekrijejo manjših.

Nato smo na območjih inercializirali poslušanje dogodkov *mouseover*, ob katerem smo sprožili metodo *overrideStyle*, s katero smo nastavili novo začasno debelino mejnih linij izbranega območja in glede na atribut *id* v spodnjem delu zaslona izpisali njegovo ime.

Poleg tega smo v desnem delu aplikacije zgradili barvno legendo narečnih skupin, narečij in podnarečij, ki smo ji prav tako nastavili poslušanje miškinih dogodkov (angl. *mouse events*). Ko uporabnik postavi miško nad vnos v legendi, se dotično narečno območje na zemljevidu poudari.

```

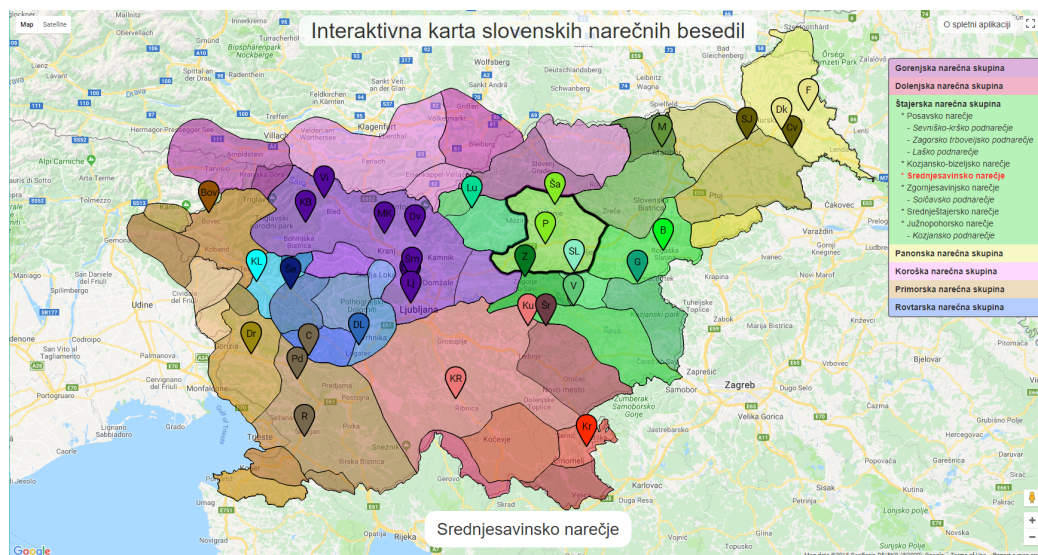
var marker = new google.maps.Marker({
  position: {lat: -25.363, lng: 131.044 },
  map: map,
  title: 'LJ'
});

```

Slika 4.2: Dodajanje oznak na zemljevid Google Maps

Na zemljevidu smo morali označiti tudi vse lokacije, kjer so bili posneti zvočni zapisi narečij. Vmesnik Google Maps API omogoča dodajanje oznak na način, ki ga prikazuje slika 4.2.

V kasnejši fazi, ko smo že zgradili skrbniško aplikacijo in vnesli podatke o narečnih zapisih, smo oznake na zemljevid (slika 4.3) dodali z zanko po vseh vpisanih krajih.



Slika 4.3: Interaktivna karta slovenskih narečnih besedil

Za prikaz podatkov o posameznem narečju smo uporabili Bootstrapovo modalno okno. Na oznake na zemljevidu smo dodali poslušalce dogodkov (angl. *event listeners*) za zaznavanje miškinih klikov in ob kliku sprožili prikaz modalnega okna (slika 4.4). V njem smo prikazali podatke, kot so ime

kraja, kjer je bil posnetek narejen, pod katero narečje in narečno skupino govor spada, njegovo transkripcijo, poknjižitev in analizo. Za predvajanje posnetka smo uporabili knjižnico Wavesurfer.js, ki zvočni zapis tudi vizualno upodobi z valovno obliko.

Dolnji Logatec (DL): horjulsko narečje - rovtarska narečna skupina

Transkripcija

Uččasix so bilē: xišš prečej dargāčne. Niso ble ta'ku velike in visōke kot z'dej. Mēle so mējixna 'akna in strēxe is cigla. 'Jest se ene take še spōmnam. Blā: je zidana is kāmna in ko si ustōpu, je bla najprij vēža, pōl pa kūxna, kjēr je biy špōrgert pa krodēna. Iz vēže se je slō: na ēno strān u xišo, naprēj pa še u štibelē. U xiš je bla u ēnem kōtu miza sa klopmi: in 'kašān štōkarj; vo drūgam pa pēč, ki je imēla dvā: za'pačka in klōp. U 'tam prostōru je blā: še pōstla, kōdar sta spāla oče in māt, zrāven je biy kōš za najmāšja otrōka. Cūn niso mēl dōst. Sprāglene so blē: u kōsmu al pa u skrīn (!). Eni so mēl še omārce u zīdu, u njix pa žēgnano vōdo pa šnopec za arēnije. Tlā: so bla lāsēna, u kūxni pa betōnska. U xiš (!) je biy še špājs, uččasix tūd kēpdar. Strānišče je blo zūnej, rēklo smo mu sakrēt. Ene xiše so mēle prōstor za spāt še zgōri, 'temu se je rēklo ispa. Gōr se je paršlō: po štēngax. Pri 'košnix xišax so pozī'm (!) sušil mesū, vīsu je u kāmbrī na potstrēx. Prōstōru pot strēxo smo rēkal potstrēxax. Zrāvan xiše je bla še štā'la za žvī'no (!), svināč, s'kadān, 'kazuc, kāšča, kūrank, ēbslnāk, lūpa in še'kej. * (!) opozarja na izgubo akuriranega naglasa, vēčinoma je vzrok postopno krajšanje i-ja kot visokega samoglasnika.

Poknjižitev

Včasih so bile hiše precej drugačne. Niso bile tako velike in visoke kot zdaj. Imele so majhna okna in strehe iz cegla 'opeke'. Jaz se ene take še spomnim. Bila je zidana iz kamna in ko si vstopil, je bila najprej veža, poli 'potem, nato' pa kuhinja, kjer je bil šporgert 'štedilnik' pa kredenca 'kuhinjska omara'. Iz veže se je šlo na eno stran v hišo 'dnevni prostor', naprej pa še v stibelje 'manjšo sobo'. V hiši je bila v enem kotu miza s klopmi in kakšen štokerelej 'stol brez naslonjala'; v drugem pa peč, ki je imela dva zapečka in klop. V tem prostoru je bila še postelja, koder sta spala oče in mati, zraven je bil koš za najmanjšega otroka. Cunj niso imeli dosti 'veliko'. Spravljene so bile v kosmu 'omari' ali pa v skrini. Eni 'nekateri' so imeli še omarice v zidu, v njih pa žegnano 'blagoslovljeno' vodo pa šnopec 'žganje' za arēnije 'zdravila'. Tla so bila lesena, v kuhinji pa betonska. V hiši je bil še špajz 'shramba', včasih tudi kelder 'klet'. Stranišče je bilo zunaj, rekli smo mu sekret. Ene 'nekateri' hiše so imele prostor za spat(i) še zgoraj, temu se je reklo ispa 'podstrešna soba'. Gor se je prišlo po štengah 'stopnicah'. Pri kakšnih hišah so pozimi sušili meso, viselo je v kambri na podstrehi 'podstrešju'. Prostoru pod streho smo rekli podstreha. Zraven hiše je bila še štala 'hlev' za živino, svinjak, skedenj, kozolec, kašča, kurmik 'kokošnjak', čebelnjak, lopa in še kaj.

Analiza

1	Naglas	
1.1	Tonemski; količnostna nasprotja so ohranjena, kratki samoglasniki so naglašeni dinamično:	kūxna : ta'ku
1.2	Izveden je naglasni umik na prednaglasna e in o, a in pogosto z vsake končne kračine:	visōke, vēža, otrōka, prōstor, s'kadān, 'kašān, za'pačka, 'kazuc
2	Dolgi samoglasniki	
2.1	ē > e;/ē:	mēle so, strēxe, na potstrēx; so mēl
2.2	cirkumflektirani o > ō;/u:	se je slōt, se je paršlōt, mesū, lūpa

Slika 4.4: Prikaz podatkov o zvočnem zapisu govora

Ob kliku na gumb za zapiranje ali izven modalnega okna smo morali poskrbeti tudi za to, da se posnetek, ki se predvaja, ob zaprtju okna ustavi, kar smo dosegli s klicem metode *pause* oz. *destroy* na wavesurferjevem API-ju.

4.3 Podatkovni model

Ob uspešno pripravljenem prototipu uporabniške aplikacije smo imeli podatke o enem narečju začasno vstavljene kar v izvorno kodo. Za potrebe

dodajanja, urejanja in prikazovanja vedno več novih zapisov smo morali izdelati skrbniški del, ki podatke shranjuje v podatkovno bazo. Zato smo se lotili priprave podatkovnega modela.

Podatkovni model je abstraktna, a razumljiva predstavitev podatkov, s katerimi neka aplikacija upravlja oz. jih bere, dodaja, posodablja ali briše. Za izdelavo konceptualnega modela na sliki 4.5 smo si pomagali z orodjem MySQL Workbench.



Slika 4.5: Konceptualni model podatkov narečnega zemljevida

Glavna tabela *dialect_entries* vsebuje osnovne podatke o narečnem vnosu. Poleg teh vsebuje tudi metapodatke, kot so kdo je vnos ustvaril (in kdaj) ter kdo je nazadnje vnos spremenil (in kdaj). Metapodatka *created_by* in *modified_by* sta tako tuja ključa (angl. *foreign keys*) tabele *users*, ki vsebuje podatke o uporabnikih, ki imajo skrbniški dostop do aplikacije. Vsak narečni vnos ima lahko enega ali več analitičnih vnosov, vsak analitični vnos pa ima lahko enega ali več podvnosov oz. primerov.

Po analizi konceptualnega modela smo ugotovili, da lahko model poenostavimo tako, da naslove analitičnih vnosov premaknemo v tabelo *dialect_entries*, saj jih je vedno natanko sedem in smo tako eno tabelo povsem izpustili, kar v praksi pomeni hitrejše izvajanje poizvedb po podatkovni bazi. Rezultat poenostavitve podatkovnega modela je prikazan na sliki 4.6.

#	Name	Type
1	id	int(11)
2	dialect_key	varchar(50)
3	location_name	varchar(50)
4	location_label	varchar(10)
5	audio	varchar(255)
6	location_latitude	decimal(10,6)
7	location_longitude	decimal(10,6)
8	transcription	text
9	standard_slovene	text
10	analysis1	text
11	analysis2	text
12	analysis3	text
13	analysis4	text
14	analysis5	text
15	analysis6	text
16	analysis7	text
17	created	timestamp
18	modified	timestamp
19	created_by	int(11)
20	modified_by	int(11)

#	Name	Type
1	id	int(11)
2	dialect_entry_id	int(11)
3	section	int(11)
4	maintext	text
5	examples	text

Slika 4.6: Fizična baza podatkov narečnega zemljevida: tabela narečnih vnosov (levo) in tabela analitičnih vnosov (desno)

4.4 REST API

AngularJS je ogrodje za JavaScript, ki deluje na strani odjemalca, zato ne omogoča neposrednega dostopa do strežnika in podatkovne baze. V ta namen smo morali zgraditi vmesnik, s katerim kličemo strežniške skripte v jeziku PHP. V praksi je to najpogosteje REST API, ki sprejema zahteve HTTP (angl. *HTTP request*) z metodami GET, POST, PUT, DELETE ipd. in vrača odzive HTTP (angl. *HTTP response*) o uspehu oz. neuspehu operacij v formatu JSON. Metoda GET se ponavadi uporablja za pridobivanje po-

datkov s strežnika, POST za ustvarjanje novih vnosov, PUT za modificiranje obstoječih in DELETE za brisanje, je pa implementacija povsem prepuščena programerju. Poleg tega REST določa tudi relacije [9] med obliko naslova URL in zahtevo HTTP, kar je predstavljeno na sliki 4.7.

URL	GET	PUT	POST	DELETE
api.com/dialects/	pridobi seznam narečij	zamenja celo zbirko narečij z novimi	ustvari novo narečje v zbirki	zbriše celo zbirko narečij
api.com/dialects/3	pridobi narečje z ID-jem 3	zamenja narečje z ID-jem 3 z novim	se ne uporablja	zbriše narečje z ID-jem 3

Slika 4.7: Relacije med obliko naslova URL in zahtevo HTTP v vmesnikih REST

Pri razvoju aplikacije na strežniški strani nismo uporabili nobenega ogrodja PHP, temveč smo vse funkcionalnosti razvili sami, pri tem pa smo se držali prej omenjenih uveljavljenih praks.

4.5 Skrbniška aplikacija

Želeli smo izdelati minimalističen in intuitiven skrbniški del aplikacije. To smo dosegli z uporabo ogrodij AngularJS in Bootstrap, za delo s podatkovno bazo MySQL pa smo uporabili jezik PHP z razširitvijo MySQLi.

4.5.1 Pregled vnosov

Najprej smo razvili pogled za pregled že vnesenih vnosov. To je enostavna tabela HTML, prikazana na sliki 4.8. Narejena je bila z ogrodjem Bootstrap, ki seznam vnosov pridobi z zahtevo GET na naslov */api/dialects* našega

aplikacijskega programskega vmesnika. Za vsak vnos je prikazan kraj, kjer je posnetek nastal, kratica, ki jo uporabimo za prikaz oznake na zemljevidu, ime narečja oz. podnarečja, v katerega govor spada, ter gumba *Uredi* in *Izbriši*. Prvi gumb je povezava ogrodja AngularJS do pogleda za urejanje s parametrom ID vnosa, drugi gumb pa s tehnologijo AJAX pošlje zahtevo tipa DELETE za izbrani vnos.

Poleg tabele smo dodali še gumb *Dodaj nov vnos*, ki z zahtevo POST ustvari nov vnos in odpre pogled za njegovo urejanje.

Urejevalnik narečnih besedil Dodaj nov vnos

#	Kratica	Kraj	Narečje	Urejanje
1	B	Brestovec	srednjestajersko_narecje	Uredi Izbriši
2	Bov	Bovec	obsosko_narecje	Uredi Izbriši
3	C	Col	notransko_narecje	Uredi Izbriši
4	Cv	Cven	prlesko_narecje	Uredi Izbriši
5	Dk	Dokležovje	prekmursko_narecje	Uredi Izbriši
6	DL	Dolinj Logatec	horjulsko_narecje	Uredi Izbriši
7	Dr	Domberk	krasko_narecje	Uredi Izbriši
8	Dv	Dvorje pri Cerkljah na Gorenjskem	gorenjsko_narecje	Uredi Izbriši
9	F	Filovci	prekmursko_narecje	Uredi Izbriši
10	G	Gubno pri Lesičnem	kozjansko-bizeljsko_narecje	Uredi Izbriši
11	KB	Koprivnik v Bohinju	gorenjsko_narecje	Uredi Izbriši
12	KL	Kanalski Lom	tolminsko_narecje	Uredi Izbriši
13	KR	Kot pri Ribnici	dolenjsko_narecje	Uredi Izbriši
14	Kr	Krasinec	severnobelokranjsko_narecje	Uredi Izbriši
15	Ku	Kumpolje pri Gabrovki	dolenjsko_narecje	Uredi Izbriši

Slika 4.8: Skrbniška aplikacija: pogled seznama vnosov

4.5.2 Urejanje vnosov

Ob kliku na gumb *Dodaj nov vnos* ali povezavo *Uredi* v pogledu za pregled vnosov se prikaže pogled za urejanje vnosa na sliki 4.9. Zgradili smo ga iz treh panelnih plošč ogrodja Bootstrap, ki smo jih poimenovali *Osnovni podatki*, *Besedilo* in *Analiza*. V prvega vpisujemo podatke o lokaciji, kjer je bil posnet zvočni zapis narečnega govora: ime kraja, njegove zemljepisne koordinate, oznako. Iz spustnega menija izberemo narečje oz. podnarečje, kamor govor umeščamo, ter naložimo zvočni zapis govora v formatu MP3.

V drugem panelu sta dve polji za vnos besedil: transkripcije in poknjžitve. Transkripcija je vedno pripravljena v sistemu ZRCola, zato je na tem mestu za prikaz bilo nujno uporabiti tudi pisavo ZRCola. Zaradi grafične doslednosti smo isto pisavo uporabili tudi v polju za poknjžitev, čeprav ta ne uporablja fonetičnih znakov.

Tretji, največji panel služi vnašanju analize narečnega besedila. Vsaka analiza je sestavljena iz sedmih sekcij za analizo: naglasov, dolgih samoglasnikov, naglašanih samoglasnikov, nenaglašanih samoglasnikov, soglasnikov, oblikoslovnih pojavov in leksike. Vsaka sekcija lahko vsebuje poljubno število podvnosov s primeri iz besedila, ki jih lahko sproti dodajamo s klikom na gumb *dodaj vnos*.

4.5.3 Komunikacija z bazo

Ne glede na to, kakšno aplikacijo razvijamo, moramo vedno biti zelo previdni, če aplikacija zapisuje v podatkovno bazo. Pred vsakim vpisovanjem uporabniških vnosov v bazo moramo najprej poskrbeti, da uporabnik ne more v našo bazo vnašati zlonamerne kode s t. i. vrivanjem SQL (angl. *SQL injection*). Za ilustracijo nevarnosti si oglejmo stavek SQL na sliki 4.10.

Predpostavimo, da glede na parameter *id*, ki ga v spletni aplikaciji vnese uporabnik, naredimo poizvedbo po bazi in uporabniku prikažemo podatke o izbranem narečju. Če uporabnik vnese npr. številko 7, je poizvedba neškodljiva in se glasi:

```
SELECT * FROM dialects WHERE id = '7';
```

Če pa vnese zlonamerni vrivek SQL, npr.

```
' ; DELETE FROM users WHERE 1 OR '
```

dobimo poizvedbo, ki pobriše vse iz tabele uporabnikov:

```
SELECT * FROM dialects WHERE id = '';
```

```
DELETE FROM users WHERE 1 OR '';
```

Uporabnik z vnosom narekovaja podatkovno bazo prepriča, da je vneseni niz zaključen, zato od narekovaja naprej baza interpretira besedilo kot ukaze

Urejevalnik narečnih besedil

Shrani spremembe
Prekliči

Osnovni podatki

Oznaka

Lokacija

Narečje oz. podnarečje

Zemljepisna širina in dolžina

Zvočni zapis (priporočeno: mp3)
 No file chosen

Besedilo

Transkripcija

Učá:šix so bilé: xí:šá pre'čej dargá:čne. Niso ble ta'ku veli:ke in visó:ke kot z'dej. Mě:le so mějxna 'uakna in stré:xe is ci:glá. 'Jest se ene tá:ke še spómnam. Blá: je zídana is kámna in ko si ustó:pu, je bla ná:jprej vé:ža, pól pa kú:čna, kjér je biyú špórgert pa kradé:čna. Iz vé:že se je sló: na éno strán u xí:šo, napréj pa še u ští:bole. U xí:š je bla yó énem kó:tu mí:za so klopmi: in 'kašan

Poknjížitev

Včasih so bile hiše precej drugačne. Niso bile tako velike in visoke kot zdaj. Imele so majhna okna in strehe iz cegla 'opeke'. Jaz se ene take še spomim. Bila je zidana iz kamna in ko si vstopil, je bila najprej veža, poli 'potem, nato' pa kuhinja, kjer je bil špogert 'štedilnik' pa kredenca 'kuhinjska omara'. Iz veže se je slo na eno stran v hišo 'dnevni prostor', naprej pa še v štibclje

Analiza

Sekcija	Opis	Primeri
1	Naglas	
1.1	Tonemski; količnostna nasprotja so ohranjena, kratki samoglasniki so naglašeni dinamično:	kú:čna : ta'ku
1.2	Izveden je naglasni umik na prednaglasna e in o, a in pogosto z vsake končne kračine:	visó:ke, vé:ža, otró:ka, pró:stor, s'kadan, 'kašan, za'pačka, 'kazue
+ dodaj vnos		
2	Dolgi samoglasniki	
2.1	ě > e/é:	mě:le so, stré:xe, na potstré:x; so měj:l

Slika 4.9: Skrbniška aplikacija: pogled urejanja vnosov

SQL. Da se izognemo vrivkom SQL, moramo torej pri vsaki poizvedbi SQL ustrezno spremeniti narekovaje v uporabniških vnosih. PHP-jeva razširitev MySQLi to omogoča s funkcijo `mysqli_real_escape_string`, ki sprejme dva argumenta. Prvi argument je povezava z bazo MySQL, iz katere funkcija pridobi podatke o kodiranju znakov, drugi argument je niz, ki ga želimo spremeniti.

Še boljša in tudi bolj varna rešitev je uporaba pripravljenih stavkov SQL (angl. *prepared SQL statements*). S pripravljenim stavkom SQL podatkovni bazi že vnaprej sporočimo natančno obliko poizvedbe in tipe parametrov, ki jih bomo s poizvedbo posredovali. Poleg tega, da se zna zdaj baza izogniti vrivkom SQL, tudi poizvedb ne interpretira vsakič znova sproti, kar pomeni,

```
$id = $_POST['id'];
$query = "SELECT *
        FROM dialects
        WHERE id = '" . $id . "'";
```

Slika 4.10: Primer poizvedbe po podatkovni bazi MySQL, ki ni zaščiten pred vrivanjem SQL

da se izvajajo hitreje.

V aplikaciji smo za varnost pred vrivanjem SQL poskrbeli tako s funkcijo *mysqli_real_escape_string* kot tudi z uporabo pripravljenih stavkov SQL.

4.5.4 Avtentikacija in avtorizacija

Avtentikacija je proces potrjevanja identitete uporabnika. Najbolj pogosta oblika identifikacije je uporabniško ime, najpogostejša oblika avtentikacije pa geslo.

Avtorizacija je dodeljevanje pravic za dostop do določenih vsebin. V računalniških sistemih je pri avtorizaciji pogosta uporaba več uporabniških vlog (angl. *user roles*), kot so skrbnik (angl. *administrator*), urednik (angl. *editor*), naročnik (angl. *subscriber*), ki imajo vsi različne pravice za dostop in urejanje.

AngularJS je začetno ogrodje, kar pomeni, da se celotna izvorna koda prenese k uporabniku in izvaja v brskalniku. Uporabnik lahko s poseganjem v kodo pridobi dostop do vseh pogledov in elementov vmesnika, zato je avtentikacija in avtorizacija v ogrodju AngularJS ne obstajata. Za avtentikacijo, avtorizacijo in s tem integriteto podatkov smo zato morali poskrbeti v zalednem delu aplikacije.

V podatkovni bazi smo naredili tabelo *users*, kjer so vneseni podatki o uporabnikih, ki jim je dovoljen dostop do skrbniškega dela aplikacije. V tabeli so med ostalim tudi uporabnikov elektronski naslov, ki ga uporabimo

za identifikacijo uporabnika, in geslo, ki služi za avtentikacijo.

Gesel iz varnostnih razlogov v podatkovno bazo nikoli ne zapisujemo v nekriptirani obliki, zato smo jih šifrirali z varno razpršitveno funkcijo (angl. *hashing function*) `password_hash` jezika PHP.

Pri prijavi uporabnika v skrbniški del sistema opravimo avtentikacijo s funkcijo `password_verify`, ki vneseno geslo ponovno razprši z uporabo soli in ostalih parametrov, ki jih preberemo iz podatkovne baze, ter rezultat primerja z nizom v podatkovni bazi. Če se ujemata, uporabnika spustimo v skrbniški del aplikacije, sicer pa izpišemo obvestilo o neuspešni prijavi in ga preusmerimo nazaj na obrazec za avtentikacijo.

HTTP je protokol brez stanj (angl. *stateless protocol*), kar pomeni, da nima mehanizma, s katerim bi si zapomnil, da je bil uporabnik uspešno avtentificiran. Zato moramo podatek o uspešni prijavi uporabnika hraniti npr. v PHP-jevi seji. V ta namen ob uspešni avtentikaciji uporabnika začnemo novo sejo s klicem PHP-jeve funkcije `session_start`, nato pa v globalni objekt `$_SESSION` pod ključ „user“ dodamo id prijavljenega uporabnika.

Zdaj lahko ob vsakem dostopu do API-ja preverimo, če je uporabnik prijavljen, tako da v vse skripte PHP na začetku dodamo klic za vključitev skripte `check_login.php`. Skripta v datoteki `check_login.php` preveri, če je uporabnik avtentificiran, kot je vidno na sliki 4.11.

```
if (isset($_SESSION["user"])) {  
    // uporabnik je že avtentificiran,  
    // dovolimo mu dostop do API-ja  
} else {  
    // uporabnik ni avtentificiran,  
    // preusmerimo ga na prijavnno stran  
    header("Location: admin#!/login");  
}
```

Slika 4.11: Preverjanje, če je uporabnik prijavljen v sistem

4.6 Preoblikovanje kode za delovanje v ogrodju AngularJS

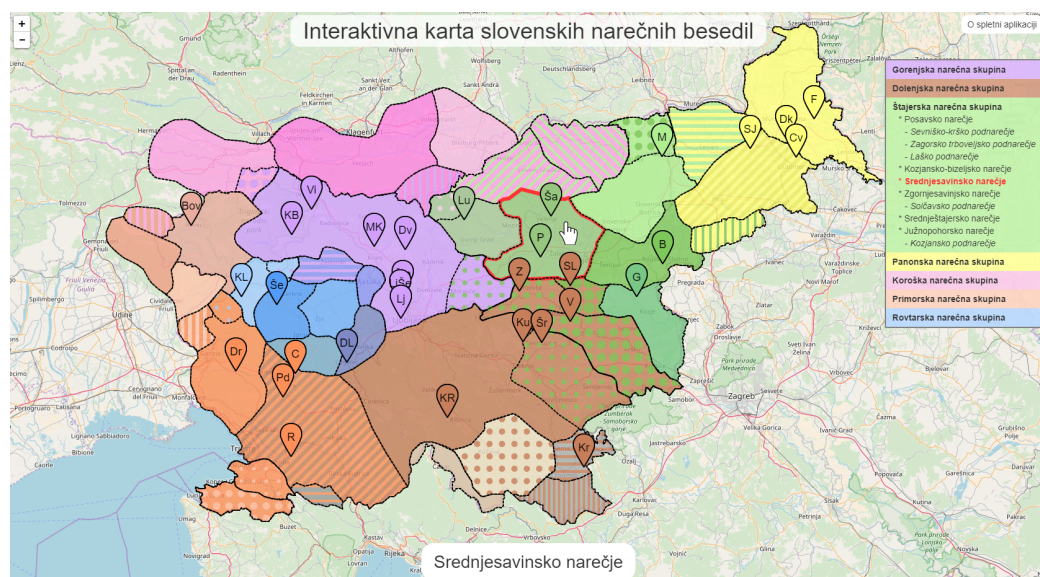
Ko smo uspešno zgradili skrbniški del aplikacije, smo vanj vnesli vse podatke o narečjih in zvočne posnetke, ki smo jih dobili od Oddelka za slovenistiko.

Nato je bilo potrebno narečja dinamično prikazati v uporabniški aplikaciji, črpajoč podatke iz podatkovne baze. Tu smo se odločili, da za pridobivanje in prikaz podatkov namesto knjižnice jQuery, ki smo jo uporabili za izdelavo prototipa aplikacije, raje uporabimo ogrodje AngularJS in, tako kot v skrbniškem delu aplikacije, podatke prikažemo z uporabo predlog tega ogrodja in s povezovanjem spremenljivk (angl. *data binding*). Izvorna koda je na tak način bolj berljiva in lepše strukturirana, pa tudi manj možnosti imamo za napake. Izognemo se namreč potrebi po sklicevanju na elemente HTML znotraj nadzornika (npr. glede na njihov atribut *id*). Če v kodi HTML kakšnemu elementu spremenimo npr. atribut *id*, ni potrebe po iskanju in prilagajanju kode v nadzorniku.

4.7 Preoblikovanje kode za uporabo knjižnice Leaflet

Eden od ciljev diplomskega dela je bil ustvariti karto, ki bo grafično čim bolj podobna Karti slovenskih narečij Tineta Logarja in Jakoba Riglerja. Ta namreč z izbranimi barvami, šrafurami in vzorci vsebuje tudi vizualne informacije o tem, kako se na nekaterih območjih prepletajo narečja in podnarečja.

Knjižnica Google Maps omogoča le enobarvne ploskve, zato smo se nazadnje odločili še enkrat prepisati aplikacijo – za rabo s knjižnico Leaflet. Z razširitvijo *Leaflet.pattern* [6] smo na ploskve dodali šrafure in vzorce s krogci tako, da se po gostoti in velikosti ujemajo s Karti slovenskih narečij, kot je vidno na sliki 4.12.



Slika 4.12: Interaktivna karta slovenskih narečnih besedil

Poglavje 5

Sklepne ugotovitve

Razvili smo zanimivo aplikacijo, interaktivno spletno karto slovenskih narečnih besedil, ki omogoča pregled vseh slovenskih narečnih skupin, narečij in podnarečij ter omogoča poslušanje zvočnih zapisov narečnih govorov iz vseh koncev slovenskega govornega območja, ogled njihovih fonetičnih zapisov, prevodov v knjižno slovenščino in analiz narečnih posebnosti. Poleg tega skrbnikom z Oddelka za slovenistiko omogoča dodajanje novih vnosov in tako zagotavlja vsebinsko vedno bolj bogato spletno aplikacijo.

Poleg samega dodajanja novih narečnih vnosov aplikacijo lahko nadgradimo in ji izboljšamo uporabniško izkušnjo na več načinov.

Ena izmed možnih izboljšav je uporaba programske opreme za združevanje in kompresijo datotek za rabo na spletnih straneh (angl. *bundler*), s čimer zmanjšamo velikost in število zahtev HTTP, ki potekajo med odjemalcem in strežnikom. Med najbolj priljubljenimi orodji, ki se danes uporabljajo v ta namen, je Webpack [16], ki je že privzeto dodan novejšim različicam ogrodja Angular.

Skrbniška aplikacija bi lahko skrbnikom omogočala ponastavljanje pozabljenega gesla. V primeru, da uporabnik pozabi geslo, ga zahteva tako, da vpiše svoj elektronski naslov, in če uporabnik s tem elektronskim naslovom obstaja, mu na ta naslov pošljemo povezavo, ki je veljavna le za enkratno uporabo, na kateri si lahko vpiše novo geslo.

Čeprav smo za varnost že poskrbeli, ni v aplikaciji nikoli odveč dodati še kakšen varnostni mehanizem. En tak primer je omejevanje števila poskusov prijave v skrbniški del sistema. Če uporabnik petkrat zapored vnese napačno geslo, mu za eno uro onemogočimo nadaljne poskuse.

Za prijetnejšo uporabniško izkušnjo bi bila vsekakor dobrodošla tudi profesionalna grafična podoba aplikacije, možnih izboljšav pa je še veliko več.

Aplikacijo je vsekakor potrebno povezati z orodjem za analitiko, saj bo najbrž zelo zanimiva tako za Slovence kot tudi za tujce, ki jih zanimajo tuji jeziki in narečja. Verjamemo, da lahko z minimalno promocijo, npr. z uporabo socialnih omrežij, aplikacija pridobi veliko število uporabnikov.

Literatura

- [1] The british library: Sound map - accents & dialects. Dosegljivo: <https://sounds.bl.uk/Sound-Maps/Accents-and-Dialects>. [Dostopano: 1. 3. 2018].
- [2] Vera Smole in Mojca Horvat. *Stara kmečka hiša, Narečna besedila z analizo I*. Znanstvena založba Filozofske fakultete Univerze v Ljubljani, 2016.
- [3] Karta slovenskih narečij. Dosegljivo: http://sla.zrc-sazu.si/eSLA/Karta_narecij_z_naselji.pdf. [Dostopano: 12. 4. 2017].
- [4] Jožica Škofic, Peter Weiss in Danila Zuljan Kumar. 3. Slovenski dialektološki posvet, Povzetki. Tehnično poročilo, Inštitut za slovenski jezik Frana Ramovša ZRC SAZU, 2016.
- [5] Leaflet. Dosegljivo: <http://leafletjs.com/>. [Dostopano: 10. 3. 2018].
- [6] Leaflet.pattern. Dosegljivo: <https://github.com/teastman/Leaflet.pattern>. [Dostopano: 10. 3. 2018].
- [7] MDN Web Docs. Dosegljivo: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>. [Dostopano: 6. 2. 2018].
- [8] Klementina Podvršnik, Marijana Klemenčič Glavica, Nataša Kralj in Mateja Črv Sužnik. *Slovenščina 1, i-učbenik za slovenščino v 1. letniku gimnazij*. Zavod RS za šolstvo, 2015.

-
- [9] Representational state transfer. Dosegljivo: https://en.wikipedia.org/wiki/Representational_state_transfer. [Dostopano: 13. 2. 2018].
- [10] RFC 7946: The GeoJSON format. Dosegljivo: <https://tools.ietf.org/html/rfc7946>. [Dostopano: 15. 1. 2018].
- [11] O projektu Slovenski lingvistični atlas. Dosegljivo: http://sla.zrc-sazu.si/eSLA/SLA_ia_informacija_o_projektu_JS.html. [Dostopano: 6. 8. 2017].
- [12] Svg to geojson. Dosegljivo: <https://github.com/davecranwell/svg-to-geojson>. [Dostopano: 1. 6. 2017].
- [13] The apache http server project. Dosegljivo: <https://httpd.apache.org/>. [Dostopano: 13. 2. 2018].
- [14] Tine Logar. Dosegljivo: https://sl.wikipedia.org/wiki/Tine_Logar. [Dostopano: 6. 2. 2018].
- [15] Vseki može da dopālni audiokartata na bālgarskite dialekti. Dosegljivo: https://www.dnevnik.bg/bulgaria/2014/12/10/2434953_vseki_moje_da_dopulni_audiokartata_na_bulgarskite/. [Dostopano: 1. 3. 2018].
- [16] Webpack. Dosegljivo: <https://webpack.js.org/>. [Dostopano: 20. 2. 2018].
- [17] Zemljevid narečij bolgarskega jezika. Dosegljivo: http://ibl.bas.bg/bulgarian_dialects/. [Dostopano: 1. 3. 2018].
- [18] Zrcola, sistem za vnašanje in pisava. Dosegljivo: <http://zrcola.zrc-sazu.si/>. [Dostopano: 1. 6. 2017].