



## **Šolski center Celje**

Srednja šola za kemijo, elektrotehniko in računalništvo

# **Maturanc**

**Raziskovalna naloga**

### **Avtorja**

Amadej Skornšek

Jaka Rizmal

### **Mentor**

mag. Boštjan Resinovič

MESTNA OBČINA CELJE

CELJE, 2017

## **Zahvala**

Za raziskovalno nalogo gre posebna zahvala mentorju, gospodu Boštjanu Resinoviču, ki nama je dal mnogo nasvetov, brez katerih izdelava naloge nebi bila uspešna.

Zahvaljujeva se tudi profesorici, gospe Tanji Jelenko, ki je najino delo jezikovno pregledala.

## Kazalo vsebine

1	Povzetek .....	8
2	Uvod.....	9
	2.1 Hipoteze in cilji .....	9
3	Maturanc.com .....	10
	3.1 Maturantski ples .....	10
	3.2 Maturantski izlet .....	10
	3.3 Prodaja.....	10
4	Analiza ankete.....	11
	4.1 Maturantski ples .....	12
	4.2 Maturantski izlet .....	16
	4.3 Prodaja.....	18
5	Izdelava spletne aplikacije.....	20
	5.1 Django.....	20
	5.2 Spletna aplikacija .....	21
	5.2.1 Datoteka z nastavitvami .....	21
	5.2.2 Datoteka s povezavami – urls.py.....	22
	5.2.3 Aplikacije.....	22
	5.3 Podatkovna baza .....	25
	5.3.1 Shramba podatkov .....	26
	5.3.2 Predloge.....	27
	5.4 Materialize.....	28
	5.4.1 Postavitve strani in oblikovanje .....	28
	5.4.2 Prilagajanje .....	30
	5.4.3 Dodatki.....	31
	5.5 Gostovanje.....	32
	5.6 Domena .....	32
	5.7 Optimizacija pozicije v iskalniku Google .....	33
6	Delovanje aplikacij.....	34
	6.1 Domov.....	34
	6.2 Profil.....	34
	6.3 Urejanje profila .....	35
	6.4 Klepetalnik .....	36

6.4.1	AJAX.....	36
6.4.2	JSON.....	38
6.5	Iskanje.....	39
7	Ovrednotenje hipotez in ciljev.....	40
7.1	Hipoteze.....	40
7.2	Cilji.....	40
8	Zaključek.....	41
9	Viri.....	42

## Kazalo slik

Slika 1 Struktura spletne aplikacije (Skornšek & Rizmal, 2017). .....	21
Slika 2 Povezave (Skornšek & Rizmal, 2017). .....	22
Slika 3 Zgradba aplikacije (Skornšek & Rizmal, 2017). .....	22
Slika 4 Zgradba pogleda (Skornšek & Rizmal, 2017). .....	23
Slika 5 Odziv strežnika (Skornšek & Rizmal, 2017). .....	23
Slika 6 Primer modela (Skornšek & Rizmal, 2017). .....	24
Slika 7 Podatkovna baza (Skornšek & Rizmal, 2017). .....	25
Slika 8 Primer ustvarjanje novega zapisa podatkov (Skornšek & Rizmal, 2017). .....	26
Slika 9 Primer pridobivanja podatkov iz baze (Skornšek & Rizmal, 2017). .....	26
Slika 10 Primer oznake v JavaScript dokumenta (Skornšek & Rizmal, 2017). .....	27
Slika 11 Primer oznak v html dokumentu (Skornšek & Rizmal, 2017). .....	27
Slika 12 Primer odseka (Skornšek & Rizmal, 2017). .....	27
Slika 13 Povezava z nadstranjo (Skornšek & Rizmal, 2017). .....	28
Slika 14 Odsek vsebine (Skornšek & Rizmal, 2017). .....	28
Slika 15 Dvanajst stolpcev (Materialize, 2017). .....	28
Slika 16 Html koda postavitve stolpcev (Materialize, 2017). .....	29
Slika 17 Sence (Materialize, 2017). .....	29
Slika 18 Html določevanja senc (Materialize, 2017). .....	29
Slika 19 Kartica z besedilom (Materialize, 2017). .....	30
Slika 20 Html kartice z besedilom (Materialize, 2017). .....	30
Slika 21 Primer menjave barve elementa (Skornšek & Rizmal, 2017). .....	31
Slika 22 Dodatek Dropify v aplikaciji maturanc (Skornšek & Rizmal, 2017). .....	31
Slika 23 Dodatek Datepicker v aplikaciji Maturanc (Skornšek & Rizmal, 2017). .....	31
Slika 24 Dodatek noUiSlider v aplikaciji Maturanc (Skornšek & Rizmal, 2017). .....	32
Slika 25 Dodatek Autocomplete v aplikaciji Maturanc (Skornšek & Rizmal, 2017). .....	32
Slika 26 Dodatek Validate v aplikaciji Maturanc (Skornšek & Rizmal, 2017). .....	32
Slika 27 Spletna stran v Googlovi iskalni konzoli (Google, 2017). .....	33
Slika 28 Domača stran (Skornšek & Rizmal, 2017). .....	34
Slika 29 Profil uporabnika (Skornšek & Rizmal, 2017). .....	34
Slika 30 Iskanje soplesalcev nastavitve profila (Skornšek & Rizmal, 2017). .....	35
Slika 31 Klepetalnik (Skornšek & Rizmal, 2017). .....	36

Slika 32 Primer kode AJAX klica (Skornšek & Rizmal, 2017).....	37
Slika 33 Sintaksa JSON kodiranja (Skornšek & Rizmal, 2017). ....	38
Slika 34 Filter za iskanje soplesalcev (Skornšek & Rizmal, 2017).....	39
Slika 35 Rezultat poizvedbe (Skornšek & Rizmal, 2017). ....	39
Slika 36 Osnovne informacije poizvedenega profila (Skornšek & Rizmal, 2017).....	39

## Kazalo grafov

Graf 1 Anketiranci po spolu. ....	11
Graf 2 Anketiranci, ki so plesali oziroma si želijo plesati na maturantskem plesu.....	12
Graf 3 Anketiranci, ki so imeli težave pri iskanju soplesalca. ....	12
Graf 4 Ženske, ki so imele težave z iskanjem soplesalca. ....	13
Graf 5 Moški, ki so imeli težave pri iskanju soplesalke.....	13
Graf 6 Anketiranci, ki bi uporabljali spletno aplikacijo.....	14
Graf 7 Ženske, ki bi uporabljale spletno aplikacijo. ....	14
Graf 8 Moški, ki bi uporabljali spletno aplikacijo.....	15
Graf 9 Zadovoljstvo anketirancev o obveščanju plesnih šol.....	15
Graf 10 Anketiranci, ki bi želeli prejemati obvestila plesnih šol. ....	16
Graf 11 Anketiranci, ki so bili na maturantskem izletu.....	16
Graf 12 Način organizacije maturantskega izleta.....	17
Graf 13 Anketiranci, ki so na spletu dobili dovolj informacij glede maturantskega izleta.....	17
Graf 14 Zadovoljstvo z informacijami na spletu glede maturantskega izleta. ....	18
Graf 15 Anketiranci, ki bi prodali šolske potrebščine.....	18
Graf 16 Odstotek anketirancev, ki bi za prodajo uporabili spletno aplikacijo. ....	19

## 1 Povzetek

Spletna aplikacija Maturanc je namenjena dijakom zaključnih letnikov. Vizija aplikacije je maturantom pomagati pri iskanju soplesalcev, pridobivanju informacij, prodaji šolskih potrebščin in boljši komunikaciji med ponudniki storitev. Glede na rezultate ankete bi srednješolci uporabljali takšno aplikacijo.

### **Ključne besede**

maturantski ples, maturantski izlet, Django, Materialize, JavaScript, HTML, CSS, SQL, spletna aplikacija, Google

### **Abstract**

Web application Maturanc is intended for students of the final year. The main goal is to help students find dancing partner for prom, acquiring important information, sale of school supplies and better communication between different service providers. According to the results of the survey, students would use an application like that.

### **Key words**

prom, prom trip, Django, Materialize, JavaScript, HTML, CSS, SQL, web application, Google



## 2 Uvod

Zaključna leta srednje šole so naporna, saj je v njih veliko pomembnih dogodkov, kot je na primer maturantski ples ali maturantski izlet. Na prvi pogled sta to dva zelo prijetna dogodka, a s seboj prineseta tudi nekaj problemov. Recimo: želiš plesati na maturantskem plesu in nimaš soplesalke. Ta problem je najbolj viden na šolah, na katerih prevladuje en spol, saj si je potrebno priskrbeti soplesalce iz drugih šol, na katerih ljudi ne poznaš. Da bi se dijaki brez soplesalcev lažje našli, sva izdelala spletno aplikacijo, ki omogoča iskanje soplesalcev med registriranimi uporabniki. Problem vidiva tudi v obveščanju dijakov. Meniva, da bi to področje lahko izboljšali, zato bova izdelala tudi komunikacijske poti, da bodo plesne in srednje šole lažje obveščale vse, ki bodo prijavljeni. S spletno aplikacijo želiva olajšati tudi postopek izbire in prijave na maturantski izlet za tiste, ki si maturantski izlet organizirajo sami. Aplikacija bo vključevala še prodajo učbenikov in ostalih šolskih potrebščin. Upava, da bova s to spletno aplikacijo pomagala prihodnjim generacijam, da bo njihova srednješolska pot lažja.

### 2.1 Hipoteze in cilji

Postavila sva naslednje hipoteze:

- Maturantje imajo težave pri iskanju soplesalcev.
- Maturantje bi za iskanje soplesalcev in pridobivanje informacij uporabljali spletno aplikacijo.
- Maturantje na spletu ne dobijo dovolj informacij glede maturantskega izleta.
- Stroški vzpostavitve sistema (brez lastnega dela) ne bodo presegli 100 €.
- Srednješolsko znanje zadostuje za izdelavo spletne aplikacije.
- Spletno aplikacijo je mogoče tržiti.

Zastavila sva si naslednje cilje:

- Spletna aplikacija bo omogoča iskanje soplesalcev.
- Spletna aplikacija bo omogoča obveščanje plesalcev s strani plesnih šol.
- Spletna aplikacija bo omogoča obveščanje dijakov s strani srednje šole.
- Spletna aplikacija bo omogoča prodajo rabljenih šolskih potrebščin.

Raziskovalne metode:

- Za raziskovanje sva uporabila spletno anketo in spletne vire.

### 3 Maturanc.com

Spletna aplikacija Maturanc je namenjena srednješolcem, srednjim šolam, plesnim šolam in turističnim agencijam. Z njo želiva pomagati predvsem maturantom, ki zaključujejo svojo srednješolsko pot. Spletna aplikacija je razdeljena na tri glavna področja, in sicer na maturantski ples, maturantski izlet in prodajo.

#### 3.1 Maturantski ples

H kategoriji maturantski ples sodijo funkcije iskanje soplesalcev, plesne šole in srednje šole. Iskanje soplesalcev bo omogočalo iskanje med registriranimi uporabniki, ki prav tako nimajo soplesalcev. V iskanje bo vgrajen filter, s katerim lahko iščemo po višini plesalca, regiji in srednji šoli. Komunikacijo bo omogočal klepet.

Plesne šole bodo ponujale možnost registracije skrbniških računov. Z njimi bodo lahko plesne šole neposredno obveščale njihove plesalce, a se bodo morali uporabniki na sporočila naročiti. S tem želiva izboljšati komunikacijo med plesnimi šolami in plesalci. Imele bodo svoje javne profile z vsemi kontaktnimi informacijami, da bodo v stik z njimi lahko prišli tudi plesalci.

Srednje šole bodo lahko obveščale svoje dijake na podoben način kot plesne šole. Prav tako bodo imele javni profil z vsemi kontaktnimi informacijami. Preko te spletne aplikacije bi lahko dijaki na šolo sporočili koliko vstopnic potrebujejo za maturantski ples in katere menije hrane bodo izbrali. Na ta način bi olajšali delo srednjim šolam in dijakom, ki lahko to storijo od doma.

#### 3.2 Maturantski izlet

Aplikacija bi izboljšala komunikacijo med maturanti in turističnimi agencijami. Agencije bi imele svoje skrbniške račune, s katerimi bi lahko urejale tekoče ponudbe in obveščale maturante. Maturantje bi se lahko na izlete prijavili preko spletne aplikacije, hkrati bi lahko dobili vse ponudbe na enem mestu.

#### 3.3 Prodaja

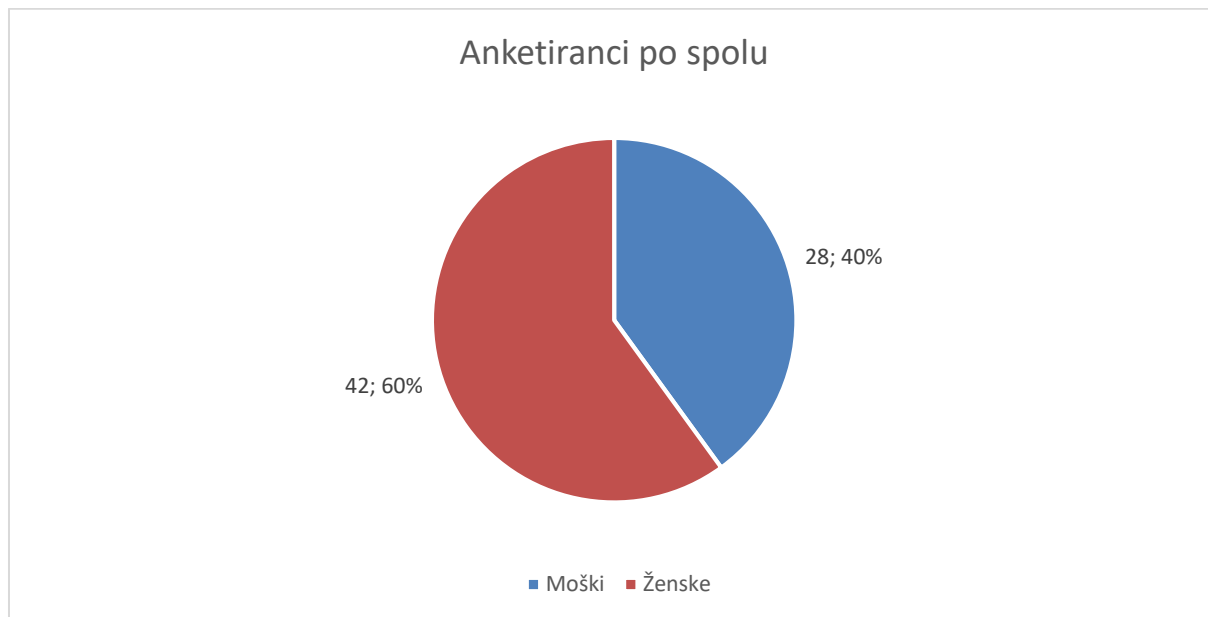
Pod kategorijo prodaja bi se nahajala spletna trgovina rabljenih izdelkov. Trgovino bi uporabljali predvsem dijaki. Po končanem šolanju ti ostanejo učbeniki in ostale šolske potrebščine, ki jih lahko prodaja naslednjim generacijam. Ko odplešeš maturantski ples, obleke morda ne potrebuješ več in bi jo lahko prodal v specializirani spletni trgovini za dijake.

Pod trgovino bi se lahko pojavile tudi razne promocije novih oblačil. Ponudbo bi bilo potrebno pripraviti v sodelovanju s spletnimi trgovinami, ki prodajajo takšna oblačila. Prav tako bi se v kategoriji lahko promovirale različne storitve, kot so na primer striženje las in ličenje.

## 4 Analiza ankete

Za raziskavo sva uporabila anketni vprašalnik, ki se deli na tri kategorije: maturantski ples, maturantski izlet in prodaja. Zanimalo naju je, če imajo maturantje probleme pri iskanju soplesalcev za maturantski ples in če so glede plesnih vaj dovolj obveščeni s strani plesnih šol. Prav tako naju je zanimalo, v kolikšni meri so bili maturantje zadovoljni z informacijami, ki so jih dobili na spletu za maturantske izlete. Glede prodaje naju je zanimalo, če želijo maturantje po končanem šolanju prodati šolske potrebščine.

### 1. Izberite spol

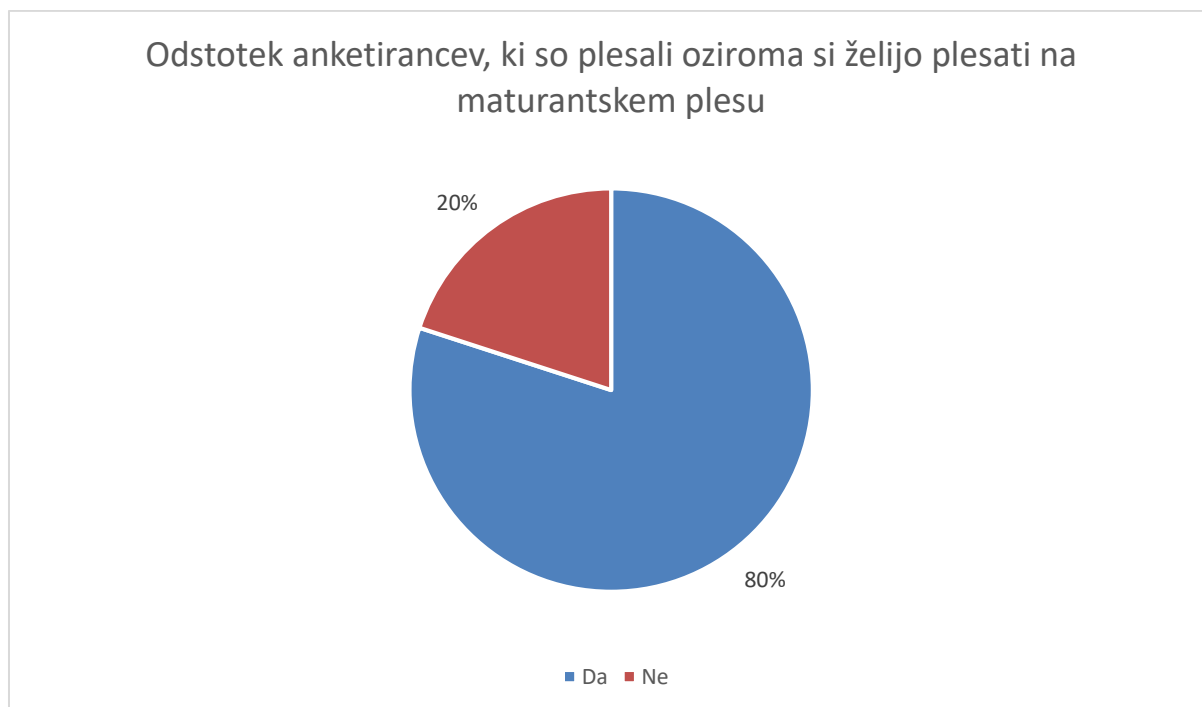


Graf 1 Anketiranci po spolu.

Anketo je rešilo 70 ljudi od tega je bilo 42 žensk in 28 moških. Anketiranci so bili srednješolci, ki obiskujejo tretji ali četrti letnik.

## 4.1 Maturantski ples

### 2. Ali ste plesali na maturantskem plesu oziroma želite plesati?



Graf 2 Anketiranci, ki so plesali oziroma si želijo plesati na maturantskem plesu.

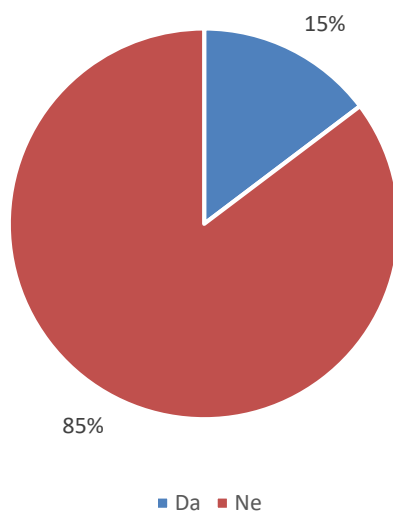
Na maturantskem plesu je plesalo oziroma si želi plesati 80% anketirancev.

### 3. Ste imeli ali imate težave z iskanjem soplesalca/soplesalke?



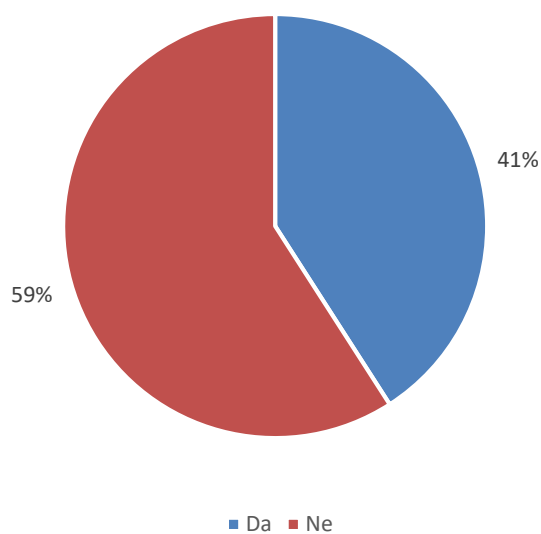
Graf 3 Anketiranci, ki so imeli težave pri iskanju soplesalca.

Odstotek ženski, ki so imele težave z iskanjem soplesalca



**Graf 4** Ženske, ki so imele težave z iskanjem soplesalca.

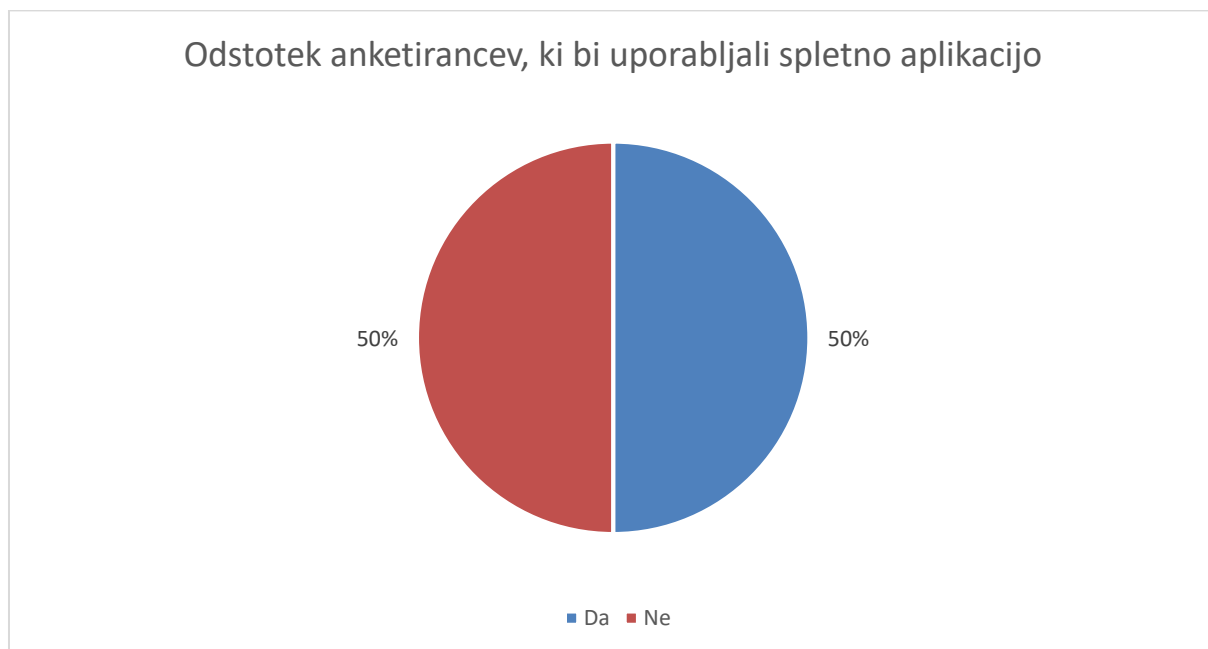
Odstotek moških, ki so imeli težave pri iskanju soplesalke



**Graf 5** Moški, ki so imeli težave pri iskanju soplesalke.

Med anketiranci, ki so plesali na maturantskem plesu oziroma si želijo plesati je pri iskanju soplesalcev imela četrtnina težave. Fantje so imeli več težav pri iskanju soplesalk. To je razumljivo, saj velja nepisano pravilo, da mora moški povabiti žensko na ples.

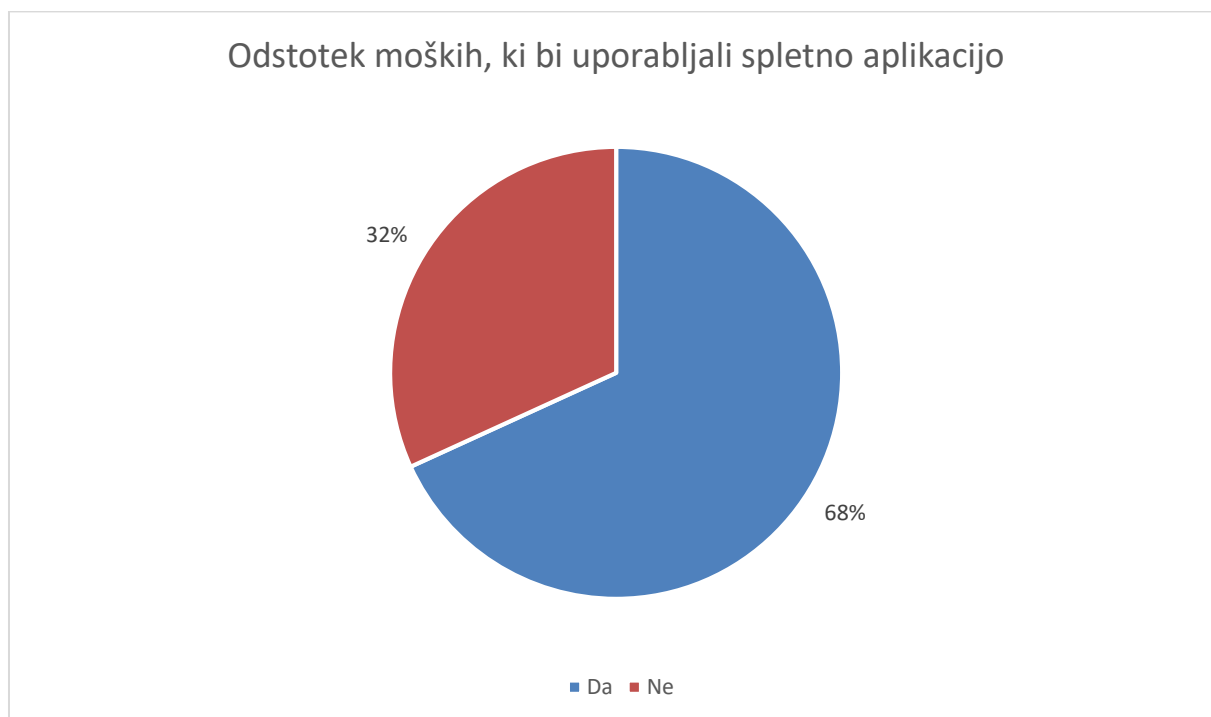
**4. Bi za iskanje soplesalca/soplesalke uporabili spletno aplikacijo, po kateri lahko soplesalca/soplesalko iščeš po regiji, šoli in višini?**



**Graf 6** Anketiranci, ki bi uporabljali spletno aplikacijo.



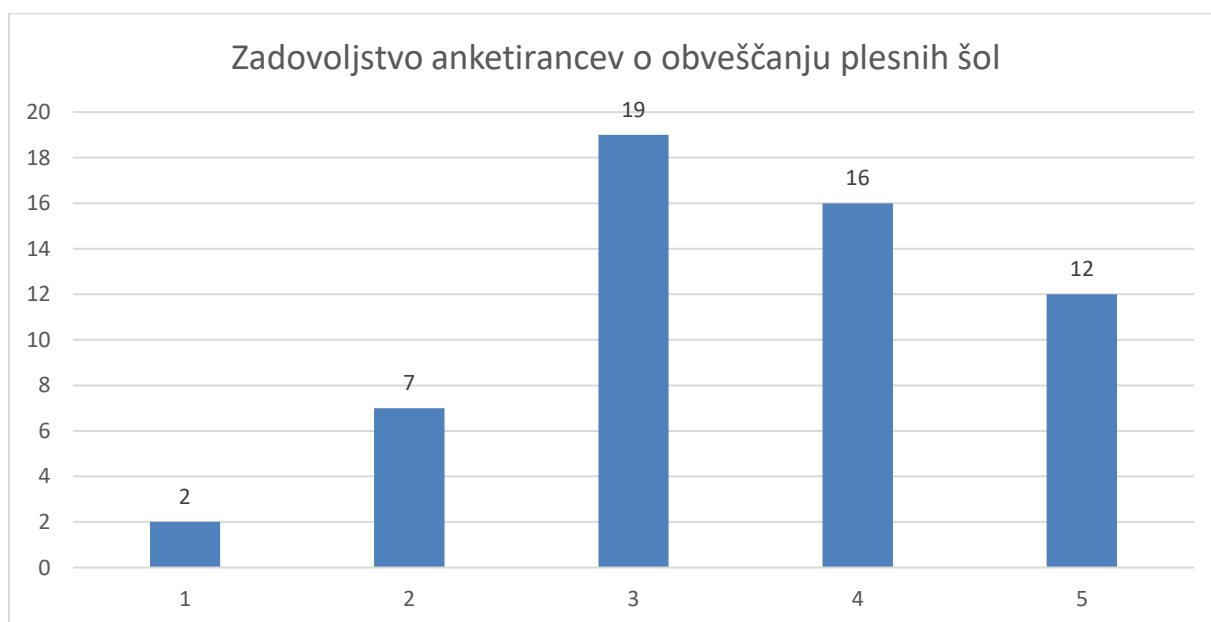
**Graf 7** Ženske, ki bi uporabljale spletno aplikacijo.



**Graf 8** Moški, ki bi uporabljali spletno aplikacijo.

Funkcijo za iskanje soplesalcev bi uporabljala polovica anketirancev. Razlika v uporabi aplikacije je zopet v spolu: moški bi jo raje uporabljali kot ženske.

### 5. Kako dobro ste bili obveščeni glede plesnih vaj s strani plesnih šol?



**Graf 9** Zadovoljstvo anketirancev o obveščanju plesnih šol.

Anketiranci so v večini zadovoljni z obveščanjem plesnih šol, saj je zadovoljstvo ocenjeno z oceno med 1 in 5, 5 pomeni zelo zadovoljen.

## 6. Bi želeli prejemati obvestila in urnike s strani plesnih šol?

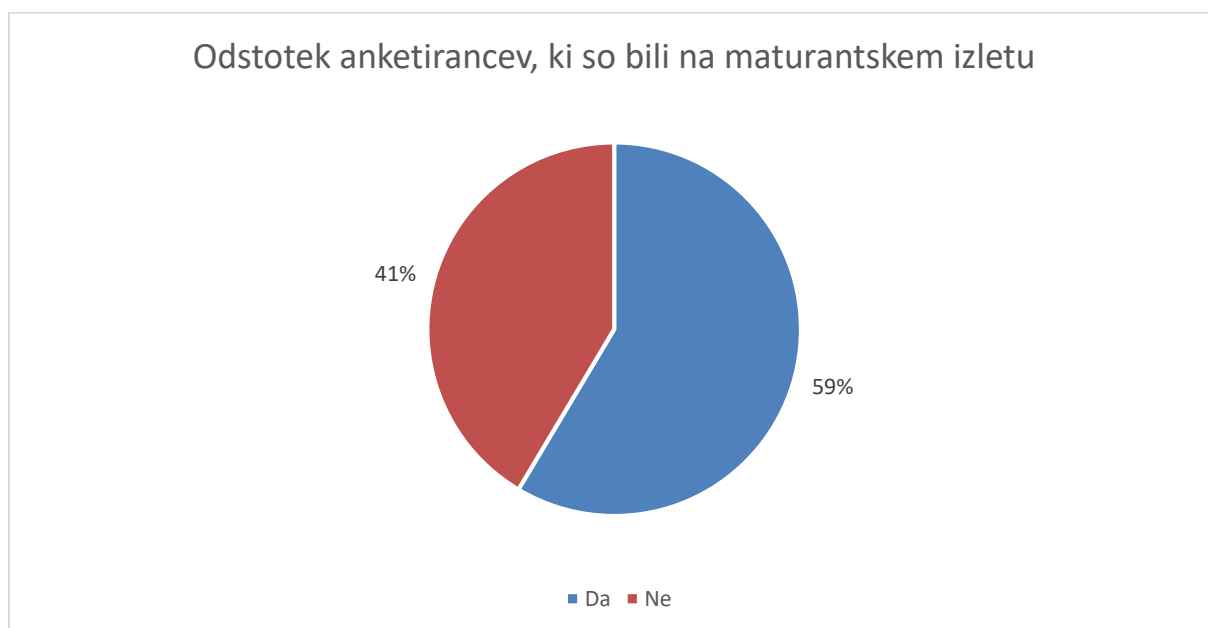


Graf 10 Anketiranci, ki bi želeli prejemati obvestila plesnih šol.

Ker želiva izboljšati obveščanje, sva anketirance vprašala, če želijo prejemati obvestila. Ugotovila sva, da si 79% anketirancev želi prejemati obvestila.

## 4.2 Maturantski izlet

### 7. Ali ste bili na maturantskem izletu?



Graf 11 Anketiranci, ki so bili na maturantskem izletu.



Na maturantskem izletu je bilo 59% anketirancev.

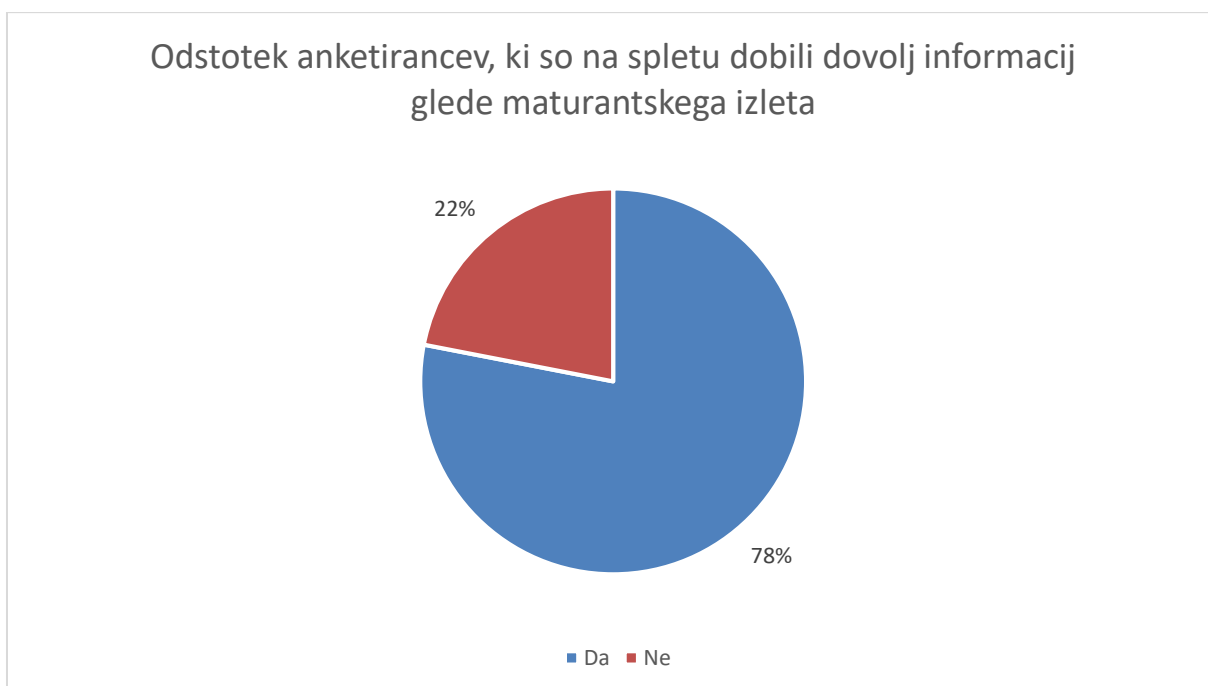
## 8. Kdo je organiziral vaš maturantski izlet?



Graf 12 Način organizacije maturantskega izleta.

Med anketiranci, ki so šli na maturantski izlet, jih je 71% potovalo pod vodstvom šole in 29% samostojno.

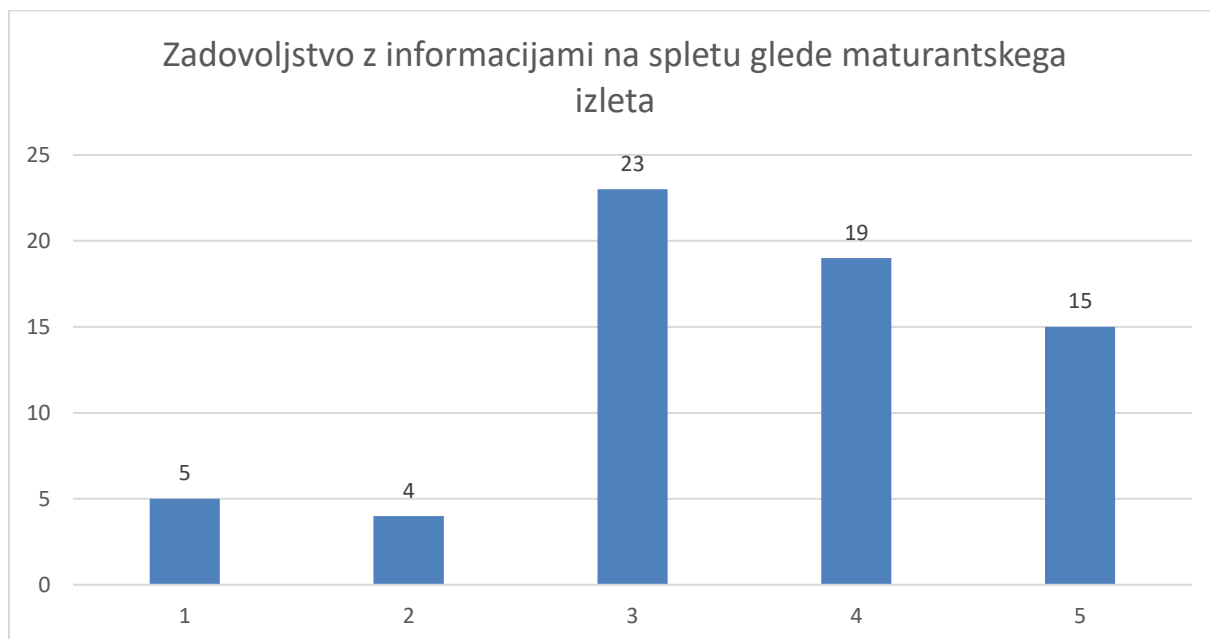
## 9. Ste lahko na spletu dobili dovolj informacij glede maturantskega izleta?



Graf 13 Anketiranci, ki so na spletu dobili dovolj informacij glede maturantskega izleta.

Na spletu je 78% anketirancev dobilo dovolj informacij, medtem ko jih 22% ni dobilo.

### 10. Kako ste bili zadovoljni z informacijami, ki ste jih dobili na spletu?

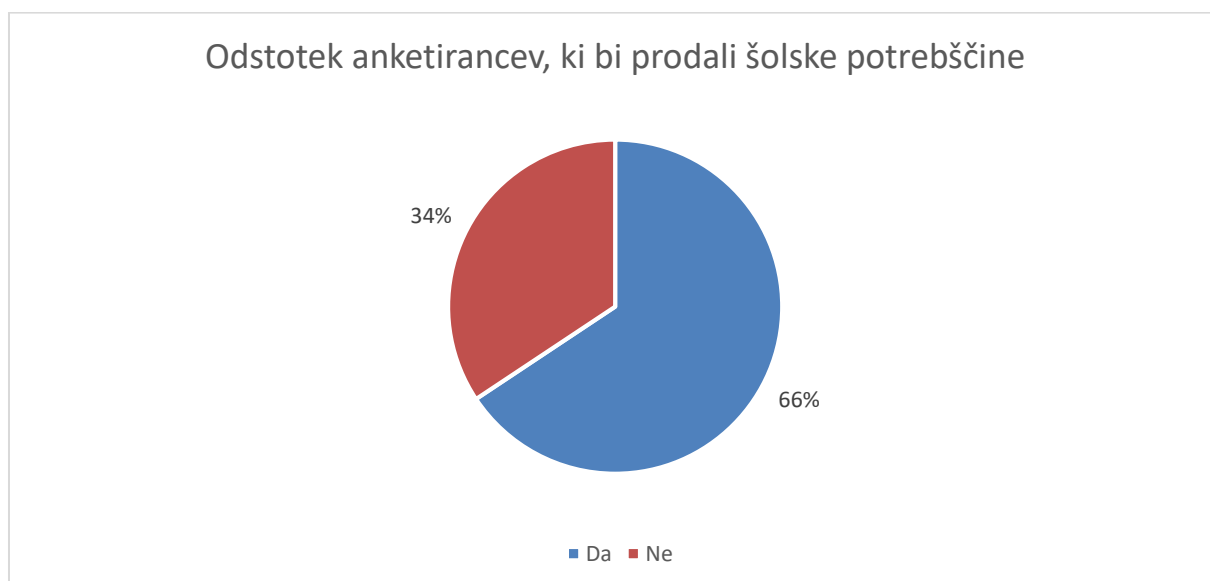


Graf 14 Zadovoljstvo z informacijami na spletu glede maturantskega izleta.

Anketiranci so bili v večini primerov zadovoljni z informacijami na spletu, vendar lahko to izboljšamo tako, da informacije združimo in jih predstavimo na enem mestu. Tako bodo maturantje lažje in hitreje našli prave informacije.

## 4.3 Prodaja

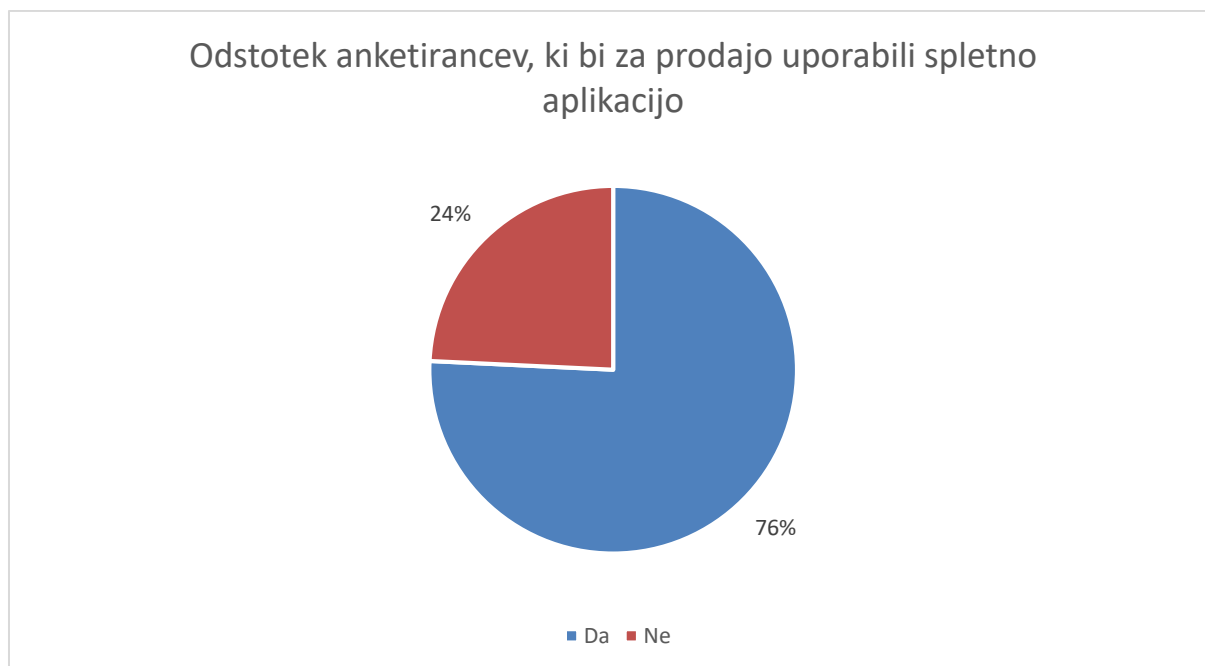
### 11. Želite po končanem šolanju prodati šolske potrebščine (učbenike...)?



Graf 15 Anketiranci, ki bi prodali šolske potrebščine.

Po končanem šolanju želi 66% anketirancev prodati šolske potrebščine.

### 12. Bi za prodajo uporabili specializirano spletno aplikacijo, katere uporabniki so predvsem srednješolci?



Graf 16 Odstotek anketirancev, ki bi za prodajo uporabili spletno aplikacijo.

Med anketiranci, ki bi prodali šolske potrebščine, bi jih 76% prodalo v spletni aplikaciji, katere uporabniki so predvsem srednješolci.

## 5 Izdelava spletne aplikacije

Za izdelavo spletne aplikacije sva se odločila, da uporabiva moderne tehnologije, ki so prijetne na pogled, hitre in hkrati varne. O teh platformah se ne učimo v šoli zato, sva se jih morala najprej naučiti. Dele spletne aplikacije, ki tečejo v ozadju poganja Django ogrodje, s pomočjo JavaScripta. Django skrbi za hitrost in varnost spletne aplikacije. Podatki se med serverjem in uporabniki prenašajo z JavaScriptom in jQueryjem. Dele aplikacije, ki so vidni uporabnikom, poganja ogrodje Materialize, ki je odgovorno za moderen videz v slogu material designa.

### 5.1 Django

Django je ogrodje za razvoj spletnih aplikacij, zgrajeno na osnovi programskega jezika Python. Odpravlja večino preglavic, ki se pojavljajo pri spletnem programiranju, da se lahko bolj posvetimo svoji aplikaciji. Django ni plačljiv in je odprtokoden.

Odlikujejo ga tri glavne lastnosti:

- Hitrost: Svojo idejo lahko s pomočjo tega ogrodja uresničimo zelo hitro;
- Varnost: Varnost je vzeta resno, zato so v samo ogrodje vgrajena orodja, ki razvijalcu pomagajo pri izgradnji varne aplikacije;
- Vsestranskost: Na osnovi tega ogrodja je bilo ustvarjenih mnogo strani, od socialnih omrežij do znanstvenih raziskovalnih platform.

Za to ogrodje sva se odločila, ker sva po kratki raziskavi ugotovila, da naj bi bil enostaven za uporabo in zanesljiv.

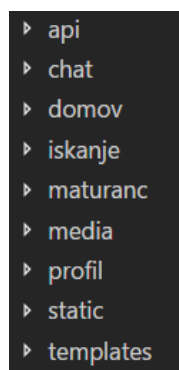
Glavne lastnosti, ki so bile všeč razvijalcu delov aplikacije, ki tečejo v ozadju:

- Zelo dobra uradna dokumentacija in veliko podpore po internetu;
- Preprosto delo s podatkovno bazo;
- Enostavno vključevanje podatkov v html dokument;
- Dobra podpora razvoju spletnega API-ja;
- Vgrajena stran za administratorja, prek katere se da spreminjati zapise v podatkovni bazi;
- Preprosto delo s serviranjem statičnih datotek (JavaScript, CSS...);
- Ogrodje je na splošno dobro organizirano in precej preprosto za razumevanje.

## 5.2 Spletna aplikacija

Spletna aplikacija je v osnovi sestavljena iz štirih modulov oz. »aplikacij«. Vsak ima svojo nalogo in skupaj omogočajo pravilno delovanje spletne strani.

1. »Klepetalnik« (chat) – skrbi za prikazovanje strani za klepet z drugimi uporabniki;
2. »Domov« – skrbi za prikazovanje domače strani;
3. »Iskanje« – omogoča uporabo iskalnika za iskanje spolesalca/ke;
4. »Api« – Sprejema zahteve in vrača rezultate v JSON formatu.



Slika 1 Struktura spletne aplikacije (Skornšek & Rizmal, 2017).

V aplikaciji so že po privzetem vgrajene komponente, brez katerih le ne bi deloval.

- Datoteka z nastavitvami – vsebuje veliko pomembnih informacij in nastavitvev za spletno aplikacijo;
- Datoteka s povezavami – v tej datoteki so napisani vzorci povezav. Vsak vzorec je nato povezan na svojo metodo, ki opravi določeno funkcijo;
- Predloge – html dokumenti, ki jih aplikacija zapolni s podatki in pošlje brskalniku;
- Statične datoteke – Datoteke JavaScript, CSS in slike.

Za lažje razumevanje bom najprej opisal že vgrajene komponente in nato še tiste, ki sva jih napisala sama.

### 5.2.1 Datoteka z nastavitvami

V datoteki z nastavitvami se nahajajo podatki, ki jih aplikacija nujno potrebuje za pravilno delovanje. Nekateri izmed teh podatkov so:

- Aplikacije, ki jih projekt uporablja – Ko ustvarimo nov projekt, so tukaj že navedene privzete aplikacije. Sem moramo vnesti tudi aplikacije, ki jih napišemo sami in različne razširitve, ki jih uporablja naša aplikacija;
- Lokacija statičnih datotek – pove aplikaciji, kje se nahajajo naše statične datoteke;
- Nastavitve podatkovne baze – navedena je vrsta podatkovne baze ter razne nastavitve za samo bazo, kot so:
  - Ime
  - Uporabnik
  - Geslo
  - Povezava
  - Prehod.
- Časovni pas

## 5.2.2 Datoteka s povezavami – urls.py

V datoteki s povezavami navedemo vse poti oz. url-je in jih povežemo s »pogledi«. »Pogledi« so metode, ki se nahajajo v aplikacijah. Podrobno so opisani v naslednjem poglavju.

```
url(r'^admin/', admin.site.urls),
url(r'^profil/$', profil_views.profil,name='profil'),
url(r'^iskanje/$', iskanje_views.iskanje,name='iskanje'),
url(r'^uporabnik/(?P<id_uporabnika>\d+)/$', profil_views.publicProfil,name='javni_profil'),
url(r'^chat/$', chat_views.chat,name='chat'),
url(r'^api/', include('api.urls'),name='api'),
url(r'^accounts/login/$', LoginView,name='socialLogin'),
url(r'^registracija/$', profil_views.registracija,name='registracija'),
url(r'^$', domov_views.domov,name='domov'),
```

Slika 2 Povezave (Skornšek & Rizmal, 2017).

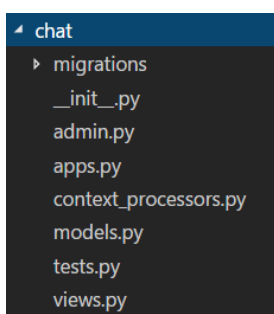
Vsaka povezava je sestavljena iz niza znakov, ki nakazuje samo strukturo url-ja, ki ga vpišemo v brskalnik. Na primer, če pogledamo prvo vrsto, je pot prikazana z '^admin/'. Ta povezava se bo uporabila, ko bo uporabnik vpisal v brskalnik »<https://maturanc.com/admin/>«. Za vejico je navedeno ime metode, ki se bo klicala ob uporabi te povezave. Na koncu je večina povezav še poimenovanih.

Nekatere povezave, kot na zgornji sliki '^api/', nimajo navedene metode, ampak drugo datoteko s povezavami. S tem dosežemo večjo modularnost in preglednost. Tak način uporabimo, ko imamo povezavo do aplikacije, iz katere je izpeljano še več povezav.

V povezavi '^uporabnik/(?P<id\_uporabnika>\d+)/\$' je vključen tudi parameter »id\_uporabnika«, ki je ob obisku na povezavo posredovan navedeni metodi. S pomočjo tega parametra metoda iz baze podatkov dobi ustrezne podatke in jih prikaže uporabniku.

## 5.2.3 Aplikacije

Kot sva že prej napisala, je aplikacija sestavljena iz štirih modulov, ki se v Django imenujejo »aplikacije«. Samo ime je lahko zavajajoče saj moduli sami brez ogrodja niso uporabni. V osnovi je vsaka aplikacija sestavljena iz sedemih datotek.



Slika 3 Zgradba aplikacije (Skornšek & Rizmal, 2017).

Osredotočil se bom na datoteki »models« in »views«, saj so te najbolj pomembne za delovanje aplikacije.

## 5.2.3.1 Views.py – pogledi

V datoteki views, slovensko »pogledi«, se opravlja večina dela v ozadju. V datoteki je zapisana vsa logika, ki je potrebna za delovanje vsakega modula.

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def domov(request):
5     return render(request, 'domov/domov.html')
```

Slika 4 Zgradba pogleda (Skornšek & Rizmal, 2017).

Zgornja slika predstavlja datoteko views.py iz modula »Domov«. Iz nje je lepo razvidna struktura. V datoteki views.py se nahaja več »pogledov«. Vsak je definiran s svojo metodo. Na zgornji sliki je ta metoda definirana v četrti vrstici. Vsak »pogled« pa mora vrniti nekakšen dokument, ki ga lahko naš brskalnik prikaže. To se zgodi v peti vrstici. Naloga te vrstice je, da vzame predlogo domov.html iz mape domov in jo pošlje brskalniku. Več o predlogah piše v nadaljevanju.

```
return render(request, 'profil/profil.html', {'profil':profil, 'social':socialnaomrezja, 'kontakti':kontakti})
```

Slika 5 Odziv strežnika (Skornšek & Rizmal, 2017).

V zgornji kodi je poleg lokacije predloge posredovan še tako imenovan kontekst. Kontekst je polje imen in vrednosti. To polje je posredovano predlogi, v kateri se ti podatki uporabijo.

»Pogled«, prikazan na sliki, je zelo preprost. V bolj zapletenih modulih so nekateri »pogledi« dolgi tudi do petdeset vrstic kode.

Da »pogled« postane uporaben, moramo imeti v datoteki s povezavami povezavo, ki kaže nanj.

### 5.2.3.2 Models.py – modeli

V datoteki models so definirani modeli za podatkovno bazo. Model si lahko predstavljamo kot tabelo v podatkovni bazi.

```
30 class SocialnaOmrezja(models.Model):
31     user = AutoOneToOneField(User, primary_key=True)
32
33     facebook=models.CharField(max_length=100, blank=True, default="")
34     snapchat=models.CharField(max_length=100, blank=True, default="")
35     instagram=models.CharField(max_length=100, blank=True, default="")
36     twitter=models.CharField(max_length=100, blank=True, default="")
37
38     def __str__(self):
39         return '%s' % (self.user)
40     def __unicode__(self):
41         return '%s' % (self.user)
```

Slika 6 Primer modela (Skornšek & Rizmal, 2017).

Vsak model je definiran z razredom, ki deduje iz razreda models.Model. V telesu razreda so definirani konkretni stolpci za podatkovno bazo. Vsakemu moramo določiti tip in v oklepajih še kakšne druge lastnosti.

Kot posebnost v zgoraj prikazanem modelu lahko omenim stolpec »user«. Stolpec je tipa »AutoOneToOneField«, kar v praksi predstavlja tuj ključ. Posebnost tega tipa pa je, da se vsakokrat, ko se ustvari nov primerek modela User, ustvari še en primerek tega modela (SocialnaOmrežja). V tem primeru pa se ta stolpec uporablja tudi kot primarni ključ.

Na sliki so vsi drugi stolpci tipa »CharField«, ki se uporablja za shranjevanje navadnega besedila. V oklepajih imajo določene še lastnosti:

- Max\_length – največja možna dolžina besedila ki ga je možno shraniti;
- Blank – če je nastavljeno na true, potem baza dovoljuje prazne zapise (ne null);
- Default – privzeta vrednost.

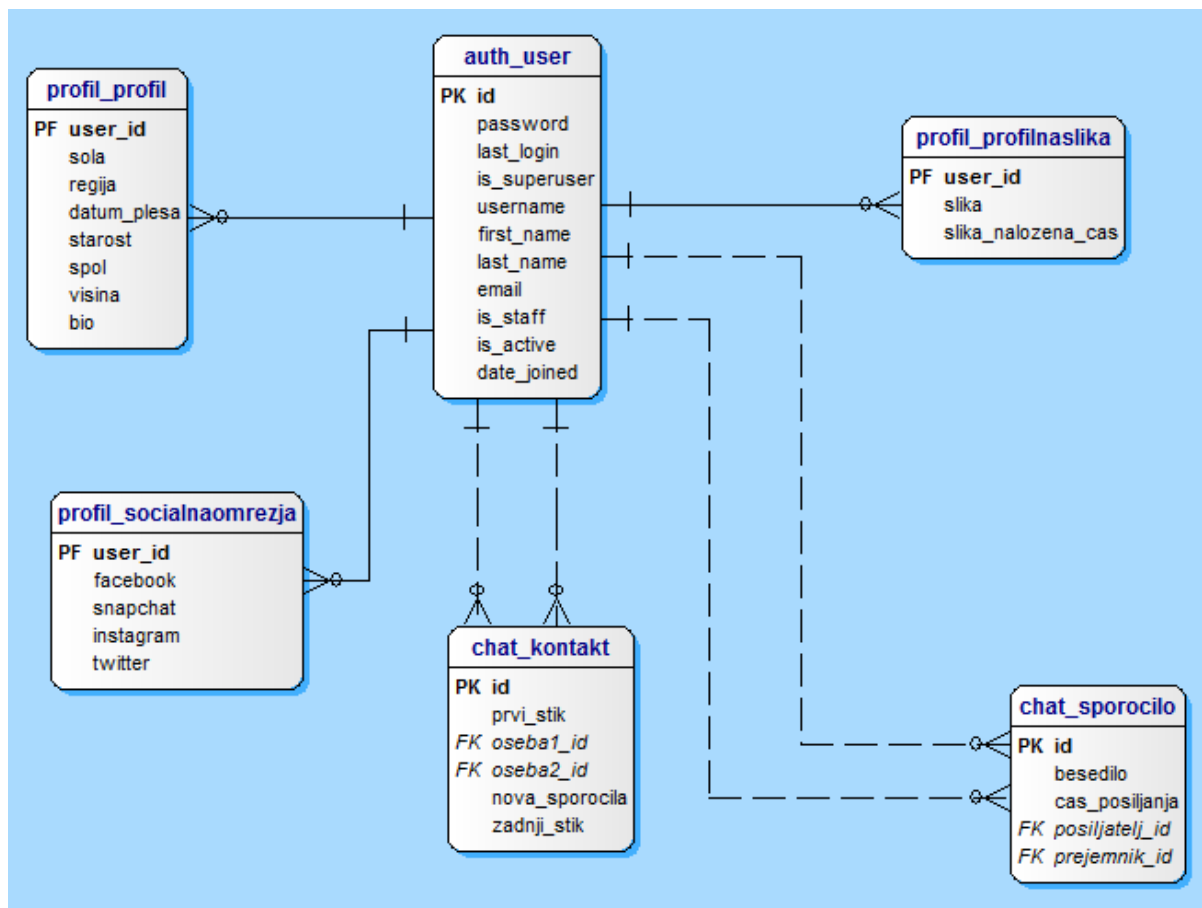
Predstavljeni bosta še dve metodi, ki se uporabljata za tekstovno predstavo modela. Tekstovna predstava se uporablja v primeru, ko prikazujemo nek primerek modela v tekstovni obliki. V tem primeru se bo prikazal uporabnik, s katerim je primerek modela povezan.



### 5.3 Podatkovna baza

Podatkovna baza temelji na sistemu PostgreSQL (SUPB). Skupaj vsebuje 22 tabel.

Podatkovna baza je zelo obsežna, saj Django sam ustvari veliko tabel, ki jih potrebuje za delovanje. Predstavila vam bova le del baze, specifičen za ta projekt.



Slika 7 Podatkovna baza (Skornšek & Rizmal, 2017).

Tabela `auth_user` je samodejno generirana in je ključnega pomena, saj se v njej identificirajo uporabniki.

Tabela `profil_profil` je razširitev tabele `auth_user`. V njej so podatki, pomembni za iskanje.

V tabeli `profil_socialnaomrezja` se nahajajo imena uporabnikov na socialnih omrežjih.

`Chat_kontakt` vsebuje kombinacije uporabnikov, ki se med sabo pogovarjajo.

`Profil_profilnaslika` vsebuje profilne slike uporabnikov in čas dodajanja slike.

`Chat_sporocilo` vsebuje besedilo in čas pošiljanja vseh sporočil.

### 5.3.1 Shramba podatkov

Kot je dandanes že standardno, se vsi podatki shranjujejo v podatkovno bazo. V Django je delo s podatki zelo enostavno in prijazno razvijalcu.

Kot sva omenila že prej, je baza definirana z modeli. Za vsako spremembo modela je treba bazo podatkov sinhronizirati s temi. To se naredi z ukazom **python manage.py migrate** v ukaznem pozivu. Pri tem je pomembno, da imamo ukazni poziv nastavljen na mapo, v kateri se nahaja naš projekt.

#### 5.3.1.1 Shranjevanje

Ko imamo podatkovno bazo sinhronizirano, lahko začnemo shranjevati podatke.

```
Sporocilo.objects.create(posiljatelj = user_posiljatelj, prejemnik = user_prejemnik, besedilo=msg_besedilo)
```

Slika 8 Primer ustvarjanje novega zapisa podatkov (Skornšek & Rizmal, 2017).

Vrstica na sliki prikazuje ustvarjanje novega zapisa v tabeli »Sporocilo«. V oklepajih so navedena imena stolpcev in vrednosti, ki se jim naj določijo.

#### 5.3.1.2 Pridobivanje

Podatke je treba iz podatkovne baze za obdelavo tudi pridobiti.

```
temp = Kontakt.objects.get(oseba1__id=id_user_1, oseba2__id=id_user_2)
```

Slika 9 Primer pridobivanja podatkov iz baze (Skornšek & Rizmal, 2017).

Vrstica na sliki pridobi iz baze zapis iz tabele Kontakt, ki ustreza pogojem v oklepajih in ga shrani v spremenljivko »temp«. Ukaz »get« uporabimo, ko pričakujemo le en rezultat. V primeru več rezultatov uporabimo ukaz »filter«.

## 5.3.2 Predloge

Predloge v Django predstavljajo html dokumenti, obogateni s posebnimi oznakami, ki jih Django podpira. Predloge se prikazuje s prej omenjenimi »pogledi«.

Poznamo dve vrsti oznak:

- `{{ }}` – v to oznako spada koda, ki vrne neko vrednost, naj bo to število ali niz znakov.
  - Primer: `{{user.first_name}}` – izpiše ime uporabnika
- `{% %}` – v to oznako spada vsa druga koda, kot na primer »if« stavki in »for« zanke.
  - `{% for k in kontakti %} {% endfor %}` – ponovi vse, kar je med začetkom in koncem oznake za vsak element iz polja »kontakt«

Te značke lahko vključujemo tako v html kot tudi JavaScript, vendar mora ta biti v predlogi.

Primer:

```
<script>
  var mojID = {{ request.user.id }};
```

Slika 10 Primer oznake v JavaScript dokumenta (Skornšek & Rizmal, 2017).

Ko bo uporabljena predloga, iz katere je zajeta zgornja slika, bo celotna značka zamenjana z id-jem prijavljenega uporabnika.

```
<div class="card-content chatpogovori">
  {% for k in kontakti %}
  <a href="#">
    <div class="chip hoverable {% if k.nova_sporocila %}red white-text{%endif%}">
      {% thumbnail k.oseba2.profilnaslika.slika "75x75" crop="center" as im %}
      
      {% endthumbnail %}
      {{k.oseba2.first_name}} {{k.oseba2.last_name}}
    </div>
  </a>
  {% endfor %}
</div>
```

Slika 11 Primer oznak v html dokumentu (Skornšek & Rizmal, 2017).

»For« zanka v zgornji kodi za vsak element iz polja kontaktov izriše html kodo, ki se nahaja do oznake »endfor«. Polje kontaktov se je dobilo v predlogo preko »konteksta«.

### 5.3.2.1 Base.html

Base.html oz. nadstran je posebna predloga, v kateri je po navadi napisana koda za navigacijsko vrstico, nogo itd., torej vse, kar se čez spletno stran ponavlja.

```
{% block content %}

{% endblock %}
```

Slika 12 Primer odseka (Skornšek & Rizmal, 2017).

Kjer bi po navadi prišla vsebina strani, postavimo oznako »block«. »Block« oz. odsek lahko poimenujemo kakorkoli. Odsekov je lahko v predlogi več.

## 5.3.2.2 Strani z vsebino

Namen strani z vsebino je, da prikažejo konkretne podatke. Strani z vsebino so ugnezdene v odsekih nadstrani. Da stran z vsebino ve, kam se vključiti, moramo na začetku navesti ime nadstrani:

```
{% extends 'base.html' %}
```

Slika 13 Povezava z nadstranjo (Skornšek & Rizmal, 2017).

Ko imamo stran z vsebino povezano z nadstranjo, dodajamo odseke.

```
{% block content %}  
  
{% endblock %}
```

Slika 14 Odsek vsebine (Skornšek & Rizmal, 2017).

Vsa html koda, ki jo bomo zapisali v odsek »content«, se bo vključila v isto imenovan odsek v nadstrani. Na eni strani lahko imamo spet več odsekov.

Tudi v straneh z vsebino lahko uporabljamo oznake za vključevanje podatkov.

Končni rezultat uporabe katerekoli predloge je html dokument, ki ga zna prikazati brskalnik, kjer so vse omenjene oznake zamenjane s konkretnimi podatki.

## 5.4 Materialize

Materialize je odprtokodno ogrodje, ki združuje veliko funkcij in dodatkov v eno povezano celoto. Večina ogrodja je sestavljena iz JavaScripta in CSS-ja vsebuje tudi nekaj HTML-ja. Služi oblikovanju in postavljanju strani. V dokumentaciji ogrodja so vse funkcije predstavljene s primeri in se jih lahko zlahka prilagodi našim potrebam. Oblikovanje je enostavno, saj se je potrebno naučiti le CSS razrede, ki so predhodno napisani in oblikovani.

### 5.4.1 Postavitve strani in oblikovanje

Postavitev strani deluje po sistemu dvanajstih stolpcev, ki so odvisni od velikosti ekrana, na katerem se vsebina prikazuje. Velikosti so pripravljene za mobilne naprave, tablice in običajne računalniške zaslone. Velikost s je namenjena majhnim napravam, m je namenjen srednje velikim naprav, l je namenjen velikim napravam. To pomeni, da se vsebina prilagaja velikosti zaslona, kar nam omogoča, da je uporabniška izkušnja enako dobra ne glede na napravo, ki jo uporabljamo.

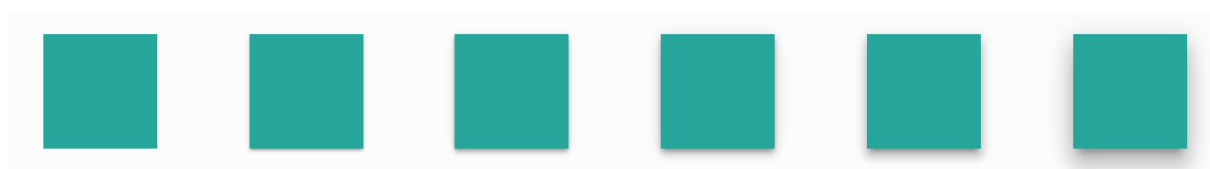


Slika 15 Dvanajst stolpcev (Materialize, 2017).

```
<div class="row">
  <div class="col s1">1</div>
  <div class="col s1">2</div>
  <div class="col s1">3</div>
  <div class="col s1">4</div>
  <div class="col s1">5</div>
  <div class="col s1">6</div>
  <div class="col s1">7</div>
  <div class="col s1">8</div>
  <div class="col s1">9</div>
  <div class="col s1">10</div>
  <div class="col s1">11</div>
  <div class="col s1">12</div>
</div>
```

**Slika 16** Html koda postavitve stolpcev (Materialize, 2017).

Ogrodje uporablja material design, ki je postal kot nekakšen standard za aplikacije na operacijskem sistemu Android. Posebnost takšnega oblikovanja so sence, ki elementom dodajo globino in s tem ustvarijo 3D učinek. Sence imajo pet stopenj.

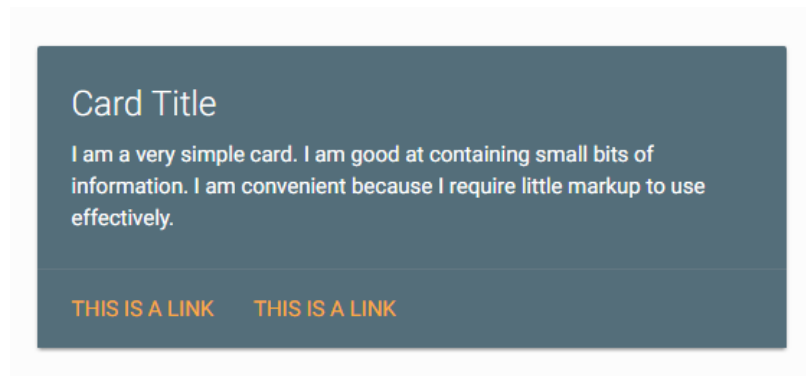


**Slika 17** Sence (Materialize, 2017).

```
<div class="col s12 m2">
  <p class="z-depth-1">z-depth-1</p>
</div>
<div class="col s12 m2">
  <p class="z-depth-2">z-depth-2</p>
</div>
<div class="col s12 m2">
  <p class="z-depth-3">z-depth-3</p>
</div>
<div class="col s12 m2">
  <p class="z-depth-4">z-depth-4</p>
</div>
<div class="col s12 m2">
  <p class="z-depth-5">z-depth-5</p>
</div>
```

**Slika 18** Html določevanja senc (Materialize, 2017).

Najbolj uporabljen element na spletni aplikaciji je kartica. S karticam lahko oblikujemo kot s kosi papirja, ki jih razporejamo po površini. Viaz aplikacije je zaradi tega bolj naraven.



Slika 19 Kartica z besedilom (Materialize, 2017).

```
<div class="row">
  <div class="col s12 m6">
    <div class="card blue-grey darken-1">
      <div class="card-content white-text">
        <span class="card-title">Card Title</span>
        <p>I am a very simple card. I am good at containing small bits of information.
        I am convenient because I require little markup to use effectively.</p>
      </div>
      <div class="card-action">
        <a href="#">This is a link</a>
        <a href="#">This is a link</a>
      </div>
    </div>
  </div>
</div>
```

Slika 20 Html kartice z besedilom (Materialize, 2017).

V kartice potem vstavljamo druge elemente, kot so vnosna polja, gumbi, slike, čipi, zbirke podatkov, tabele in še mnoge druge funkcije, ki jih ponuja Materialize. Uporaba le teh je prav tako enostavna kot vse stvari do sedaj. V večini primerov ustvarimo značko »<div></div>«, ji dodelimo CSS razred in funkcija je ustvarjena.

### 5.4.2 Prilagajanje

Materialize je zelo prilagodljivo ogrodje. Ustvarimo CSS dokument in ga povežemo s html datoteko. V CSS dokumentu razrede, ki jih želimo spremeniti, le prepíšemo. Pri tem moramo paziti, da je Materialize CSS dokument pred našim.

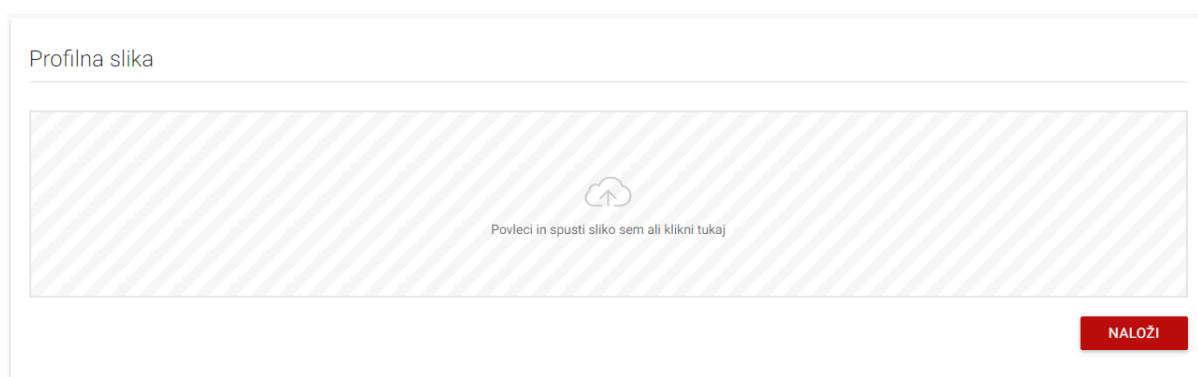
```
.btn {  
  background: #BA0D0B !important;  
}
```

Slika 21 Primer menjave barve elementa (Skornšek & Rizmal, 2017).

## 5.4.3 Dodatki

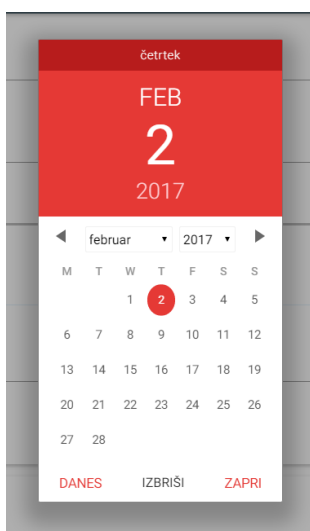
Samo ogrodje uporablja veliko tujih dodatkov, ki so vizualno prilagojeni, da se ujemajo videzu teme. Dodatke lahko dodajamo tudi sami, ampak jih je potrebno v večini primerov prilagoditi. V spletni aplikaciji so uporabljeni dodatki Dropify, Datepicker, noUiSlider, Autocomplete in Validate.

Dropify služi lažjemu nalaganju fotografij, saj omogoča »Povleci in spusti« sistem.



Slika 22 Dodatek Dropify v aplikaciji maturanc (Skornšek & Rizmal, 2017).

Datepicker je namenjen lažjemu izbiranju datuma.



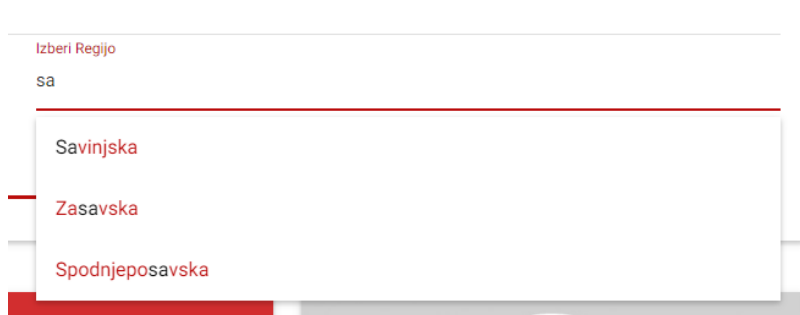
Slika 23 Dodatek Datepicker v aplikaciji Maturanc (Skornšek & Rizmal, 2017).

noUiSlider pomaga pri lažji nastavitvi višinskih parametrov.

Izberite višino: 150cm - 190cm

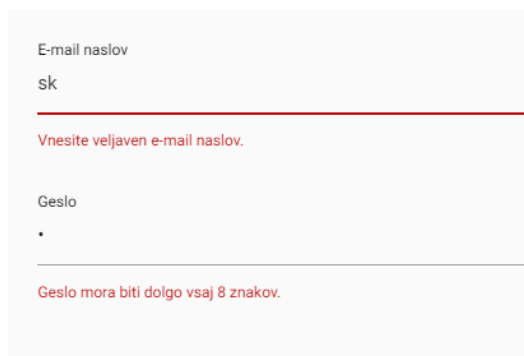
**Slika 24 Dodatek noUISlider v aplikaciji Maturanc (Skornšek & Rizmal, 2017).**

Autocomplete pomaga uporabniku končati parameter s predlogi.



**Slika 25 Dodatek Autocomplete v aplikaciji Maturanc (Skornšek & Rizmal, 2017).**

Validate preverja vnose uporabnika s pred pripravljenimi zahtevami.



**Slika 26 Dodatek Validate v aplikaciji Maturanc (Skornšek & Rizmal, 2017).**

## 5.5 Gostovanje

Celotno stran poganja Linux strežnik, najet pri Digitalocean.com. Do strežnika dostopava preko SSH in SFTP protokola. Prvi se uporablja za pošiljanje ukazov v Linux terminal, slednji pa za prenos novih datotek na strežnik.

Strežnik je v obliki virtualnega računalnika, ki teče poleg mnogih drugih virtualnih računalnikov na strežnikih storitve Digitalocean.com. Na strežniku so prednaloženi servisi, ključni za delovanje spletne strani, potrebna je le manjša konfiguracija.

Strežnik se nahaja v Nemčiji, zato je dostopni čas precej hiter.

## 5.6 Domena

Domeno maturanc.com sva kupila pri podjetju Godaddy.com.

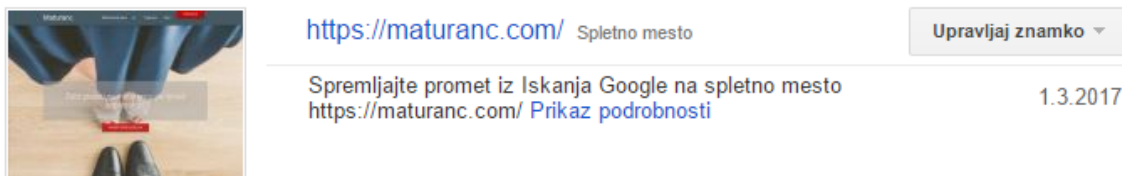
Za prikazovanje najine spletne strani je bila potrebna konfiguracija DNS nastavitvev. DNS je bilo potrebno nastaviti tako, da kaže na IP naslov najinega strežnika.



### 5.7 Optimizacija pozicije v iskalniku Google

Da bi spletno stran obiskalo čim več uporabnikov, je zelo pomembno, da ima dobro pozicijo na spletnih iskalnikih. Ko uporabnik vpiše v Google na primer »iskanje soplesalca«, mora biti najina spletna stran med prvimi zadetki, če ne, je možnost obiska precej majhna.

Najprej je potrebna registracija naslova na Googlovi konzoli za spletno iskanje.



Slika 27 Spletna stran v Googlovi iskalni konzoli (Google, 2017).

Ko imamo svojo stran vneseno v sistem, traja po navadi en dan, da Google stran pregleda in jo umesti med rezultate iskanja.

Na pozicijo med rezultati vpliva mnogo dejavnikov, ki se s časom tudi spreminjajo. Nekaj izmed najbolj pomembnih dejavnikov:

- Naslov strani – V naslovu moramo zajeti čim več ključnih besed kot so v tem primeru »maturantski ples« ali »iskanje soplesalke«;
- Opis strani – tudi v opisu morajo biti ključne besede;
- Naslovi na strani – Google pregleda tudi zgradbo same strani. Naslovi na straneh se morajo ujemati s ključnimi besedami;
- Povezave iz drugih strani – Google ima v bazi podatkov večino strani na spletu. S tem ima shranjene tudi vse povezave. Več povezav, ki kaže na spletno stran, višje je umeščena med rezultate iskanja;

Pozicija med rezultati je odvisna tudi od tega, koliko časa uporabniki povprečno ostanejo na strani;

Cilj optimizacije je, da se najina stran pojavi med prvimi rezultati pri čim več poizvedbah v zvezi z maturantskim plesom.

## 6 Delovanje aplikacij

### 6.1 Domov

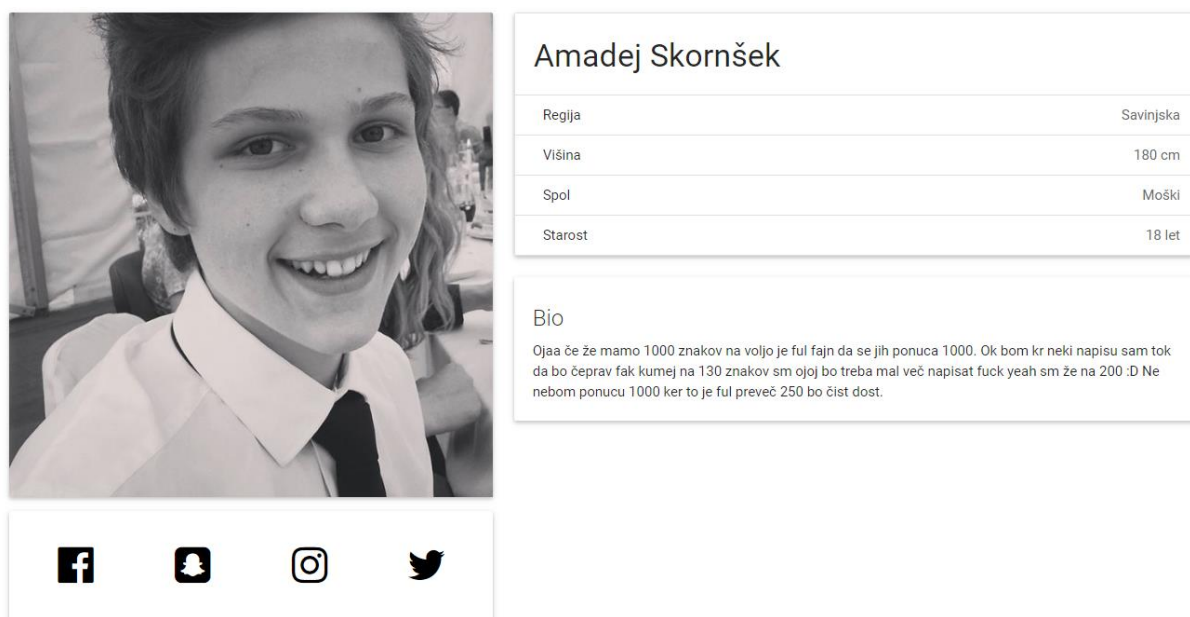
Aplikacija domov skrbi za prikazovanje domače strani. Je najbolj preprosta aplikacija v projektu, saj v njej ni definiranih modelov in ima samo en »pogled«.



Slika 28 Domača stran (Skornšek & Rizmal, 2017).

### 6.2 Profil

Aplikacija profil skrbi za prikazovanje našega osebnega profila in javnih profilov uporabnikov.



Slika 29 Profil uporabnika (Skornšek & Rizmal, 2017).

## 6.3 Urejanje profila

V predelu urejanje profila lahko uporabnik spreminja svoje osebne podatke. Vsak uporabnik si lahko nastavi:

- Profilno sliko
- Kratek opis
- Šolo ki jo obiskuje
- Regijo
- Spol
- Višino
- Datum maturantskega plesa
- Datum rojstva
- Socialna omrežja

Iskanje soplesalcev

Bio

Ime mi je Jaka. V prostem času programiram in poslušam glasbo :)

Izberi šolo

Solski center Celje, Srednja šola za kemijo, elektrotehniko in računalništvo

Izberi Regijo

Savinjska

Moški

Nastavite vašo višino: 182cm

Izberi datum plesa

petek, 03/03/2017

Izberi datum rojstva

torek, 17/11/1998

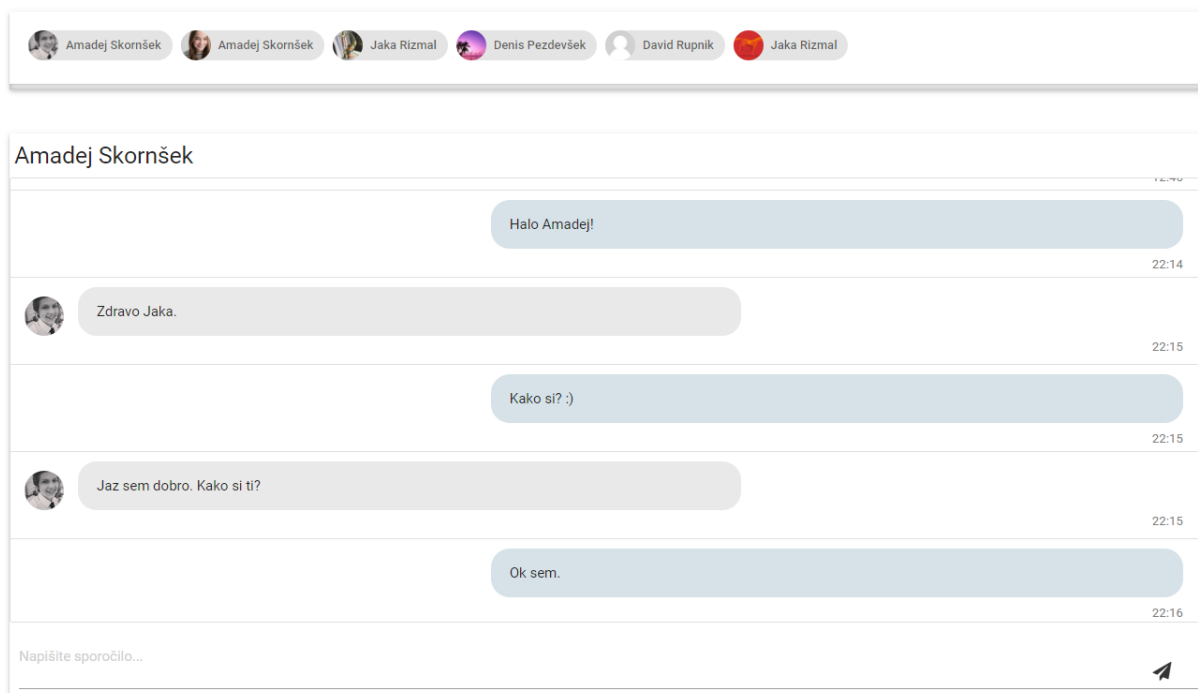
SHRANI SPREMEMBE

**Slika 30** Iskanje soplesalcev nastavitve profila (Skornšek & Rizmal, 2017).

Ob kliku na gumb shrani spremembe se podatki preko »POST« metode pošljejo na strežnik in shranijo v podatkovno bazo.

## 6.4 Klepetalnik

Aplikacija klepetalnik omogoča funkcijo klepeta med uporabniki. Preko klepeta se lahko uporabniki dogovorijo za maturantski ples itd.



Slika 31 Klepetalnik (Skornšek & Rizmal, 2017).

### 6.4.1 AJAX

AJAX (asynchronous JavaScript and XML) je tehnika, ki se uporablja v spletnem programiranju za asinhrono prenašanje podatkov med klientom in strežnikom. Asinhrono pomeni, da se prenosi odvijajo v ozadju in tako ne motijo uporabnika. V praksi to večinoma pomeni spreminjanje vsebine spletne strani brez osveževanja brskalnika.

AJAX je na modernih straneh nepogrešljiv, saj glede na navadne načine močno izboljša uporabniško izkušnjo na strani. Kot primer bom prikazal funkcijo, ki vsake 2 sekundi preverja, če je uporabnik dobil novo sporočilo in ga prikaže v klepetu.

```
function dobiNovaSporocila(){
  var zadnjeSporociloCas = $('#chat_list li').last().attr('cas');
  $.ajax({
    type: 'GET',
    url: 'https://maturanc.com/api/chat/vsasporocila/',
    data: {
      seen: 'da',
      starejsa: '',
      samo_nova: 'da',
      timestamp: zadnjeSporociloCas,
      oseba1: mojID,
      oseba2: sogovornikID
    },
    success: function(data){
      jsonResult = JSON.parse(data);
      $.each(jsonResult, function(i, item) {
        var cas_hms=jsonResult[i].cas_posiljanja.split("T")[1].split(":")[0].split(".");
        var cas_clean = cas_hms[0] + ':' + cas_hms[1];
        if(jsonResult[i].posiljatelj==mojID){
          $('#chat_list').append('<li class="collection-item" cas="'+jsonResult[i].cas_posiljanja+'><div class="chatmess
        }
        else{
          $('#chat_list').append('<li class="collection-item avatar" cas="'+jsonResult[i].cas_posiljanja+'><img src="'+s
        }
        var objDiv = document.getElementById("chat_list");
        objDiv.scrollTop = objDiv.scrollHeight;
      });
    }
  });
}
```

Slika 32 Primer kode AJAX klica (Skornšek & Rizmal, 2017).

V funkciji je definiran AJAX klic, ki za delovanje potrebuje naslednje podatke:

- Tip (type) – določi http metodo, ki se bo uporabila pri dostopu;
- Povezavo (url) – povezava, na katero bo klic poslal zahtevo;
- Podatki (data) – podatki, ki jih bo klic poslal v zahtevi;
- Uspeh (success) – sem vpišemo svojo metodo. Metoda se izvrši vsakič, ko je izvedeni klic uspešen.

### 6.4.1.1 Tip

Poznamo več tipov http zahtev. V tem primeru je uporabljena metoda »GET«, saj podatke iz strežnika pridobivamo.

### 6.4.1.2 Povezava

Povezava v tem klicu kaže na metodo v aplikaciji API, ki sprejme podatke in pošlje nazaj nova sporočila.

### 6.4.1.3 Podatki

Da strežnik ve, kakšna sporočila mora vrniti, mu moramo posredovati nekaj ključnih podatkov.

- Samo\_nova – nastavljeno na »da«. Tako strežnik ve, da zahtevamo samo nova sporočila;
- Timestamp – po slovensko časovni žig. V ta parameter vstavimo čas zadnjega prejetega sporočila. Tako strežnik ve od katerega časa naprej naj gleda sporočila.
- Oseba1 – sem vstavimo svojo uporabniško številko;
- Oseba2 – sem vstavimo številko sogovornika. V kombinaciji s podatkom oseba1 strežnik ve, čigava sporočila naj pošlje.

## 6.4.1.4 Uspeh

Tukaj se nahaja funkcija, ki prebere pridobljene podatke, in jih prikaže uporabniku. Funkciji je avtomatsko podan rezultat poizvedbe. Rezultat je razčlenjen, tako da ven dobimo ključne podatke, kot so čas pošiljanja in besedilo. Za vsako prejeto sporočilo funkcija generira html kodo in jo pripne na konec seznama vseh sporočil v klepetu.

Za ves prenos podatkov od strežnika k uporabniku se uporablja JSON kodiranje.

## 6.4.2 JSON

JSON – JavaScript object notation – je ena izmed najpogostejših oblik zapisa podatkov za pošiljanje med napravami. Zelo je razširjena zaradi tega, ker je podprt v večini sistemov, zato je možna komunikacija med zelo različnimi sistemi. Primer iz najinega projekta:

```
[
  {
    "posiljatelj": 1,
    "prejemnik": 22,
    "besedilo": "test api",
    "cas_posiljanja": "2017-03-03T22:10:59.448833"
  },
  {
    "posiljatelj": 1,
    "prejemnik": 22,
    "besedilo": "lep pozdrav",
    "cas_posiljanja": "2017-03-03T22:10:53.829516"
  },
  {
    "posiljatelj": 1,
    "prejemnik": 22,
    "besedilo": "primer sporočila",
    "cas_posiljanja": "2017-03-03T22:10:49.778642"
  },
  {↔},
  {↔},
  {↔},
  {↔},
  {↔},
  {↔},
  {↔}
]
```

**Slika 33** Sintaksa JSON kodiranja (Skornšek & Rizmal, 2017).

V primeru je vidna sintaksa JSON kodiranja. Oglati oklepaji na začetku naznanja, da se bodo znotraj njih ponavljali neki objekti. Vsak objekt je označen z zaviti oklepaji. Med sabo so ločeni z vejico. Znotraj vsakega objekta se nahajajo konkretni podatki, potrebni za delovanje aplikacije.

## 6.5 Iskanje

Aplikacija omogoča iskanje uporabnikov po kriterijih, ki jih vnesemo v filter.

Filter za iskanje soplesalcev

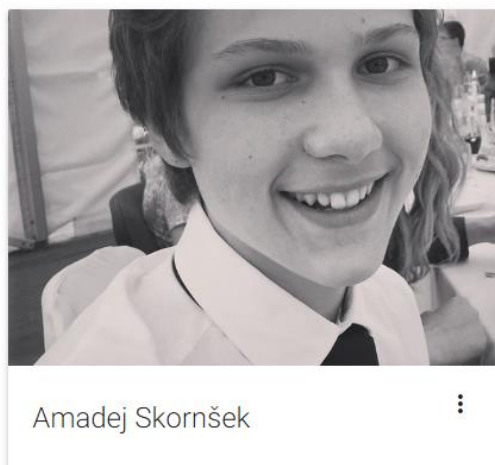
Izberi šolo

Izberi Regijo

Izberite višino: 150cm - 190cm

Slika 34 Filter za iskanje soplesalcev (Skornšek & Rizmal, 2017).

Uporabnike lahko iščemo po šoli, regiji in višini. Za vsako spremembo filtra se seznam uporabnikov osveži. Spremembe se kot pri klepetu naložijo prek metode AJAX.



Slika 35 Rezultat poizvedbe (Skornšek & Rizmal, 2017).



Slika 36 Osnovne informacije poizvedenega profila (Skornšek & Rizmal, 2017).

Uporabniki, ki ustrezajo filtru se prikažejo v obliki kartic. Na začetku je čez kartico prikazana profilna slika uporabnika. Ko kliknemo nanjo, se prikažejo podrobni podatki in povezava na profil uporabnika, preko katerega ji lahko pošljemo sporočilo.

## 7 Ovrednotenje hipotez in ciljev

### 7.1 Hipoteze

#### **Maturanti imajo težave pri iskanju soplesalcev.**

Z anketo sva ugotovila, da je imela le četrtina anketirancev probleme pri iskanju soplesalcev. Ampak ko sva se osredotočila na spol, sva ugotovila, da je imelo probleme 41% moških anketirancev. S tem potrjujeva hipotezo, saj morajo v večini primerov moški vprašati za ples.

#### **Maturanti na spletu ne dobijo dovolj informacij glede maturantskega izleta.**

Z anketo sva ugotovila, da so anketiranci dobili dovolj informacij, saj je informacije dobilo 78% anketirancev. Meniva da lahko informacije še izboljšava, če se nama uspe povezati s pravimi agencijami, zato te hipoteze ne moreva potrditi.

#### **Maturanti bi za iskanje soplesalcev in pridobivanje informacij uporabljali spletno aplikacijo.**

Anketa je pokazala, da bi polovica anketirancev uporabljala to spletno aplikacijo. Glede na spol bi jo uporabljal večji delež moških kot žensk; Moški imajo več težav z iskanjem soplesalk. To hipotezo potrjujeva.

#### **Stroški vzpostavitve sistema (brez lastnega dela) ne bodo presegli 100 €.**

Zaenkrat so stroški spletne aplikacije samo gostovanje in domena, ker so vsa spletna ogrodja in platforme odprtokodne in brezplačne. Cena trenutnega paketa gostovanja stane 5 € na mesec. Domena za eno leto stane 15 €. Za polno vzpostavitev bi bilo potrebno dokupiti še poštni strežnik, stroški tega bi bili 10 € na leto. Zato to hipotezo potrjujeva.

#### **Srednješolsko znanje zadostuje za izdelavo spletne aplikacije.**

Načeloma srednješolsko znanje zadostuje za izdelavo takšne spletne aplikacije. Vendar bi bila izdelava težja in dolgotrajna. Ker sva uporabila tehnologije, o katerih se jih v šolah ne učimo, to hipotezo delno potrjujeva.

#### **Spletno aplikacijo je mogoče tržiti.**

S povezovanjem turističnih agencij, plesnih in srednjih šol, bi aplikacijo lahko tržila. Za njihove skrbniške račune bi lahko zaračunavala mesečne naročnine. Ponudiva lahko tudi možnost oglaševanja. Ker aplikacije še ni gotova in je nisva še nikamor ponudila, te hipoteze ne moreva potrditi.

### 7.2 Cilji

Od štirih zastavljenih ciljev nama je uspelo doseči, da spletna aplikacija omogoča iskanje soplesalcev. Začetek ustvarjanja je bil težak, saj sva nove tehnologije uporabljala prvič in sva ustvarjala počasi. Pred izdelavo funkcij je bilo potrebno postaviti ogrodje spletne aplikacije. Zdaj ko je ogrodje postavljeno meniva, da bodo nove funkcije nastajale hitreje. Trenutno delujoči funkciji na strani sta iskanje soplesalcev in klepet.



### 8 Zaključek

Meniva, da srednješolci potrebujejo takšno aplikacijo. Iz ankete sva ugotovila, da imajo maturanti probleme pri iskanju soplesalcev in da bi jim takšna aplikacija lahko pomagala. Ker sva tudi sama maturanta, sva mnenja, da bi bilo lahko obveščanje o maturantskem plesu boljše, zato želiva to izboljšati s skrbniškimi računi ponudnikov storitev, ki bodo lahko uporabnike obveščali in promovirali sami sebe. To je lahko tudi eden izmed potencialnih virov zaslužka, saj bi bili skrbniški računi plačljivi. Več kot bo funkcij za ponudnike storitev, bolj bo aplikacija zanimiva za uporabnike, saj bodo imeli vse storitve na enem mestu. Po spletu sva iskala podobne produkte, a nisva našla nič podobnega, zato upava, da bo aplikacija v Sloveniji uspešna in bova lahko najin produkt širila tudi v druge države sveta.

### 9 Viri

Amsul. (marec 2017). *pickadate.js*. Pridobljeno iz <http://amsul.ca/pickadate.js/>

Django. (marec 2017). *Django documentation*. Pridobljeno iz <https://docs.djangoproject.com/en/1.10/>

*Django-allauth documentation*. (marec 2017). Pridobljeno iz <http://django-allauth.readthedocs.io/en/latest/signals.html>

Google. (marec 2017). *Search Console*. Pridobljeno iz <https://www.google.com/webmasters/tools/home?hl=sl>

JeremyFagis. (marec 2017). *Dropify - js*. Pridobljeno iz <https://github.com/JeremyFagis/dropify>

jQuery. (marec 2017). *jQuery Ui Autocomplete*. Pridobljeno iz <https://jqueryui.com/autocomplete/>

*jQuery Validation Plugin*. (marec 2017). Pridobljeno iz <https://jqueryvalidation.org/>

Materialize. (marec 2017). *Documentation - Materialize*. Pridobljeno iz Materialize: <http://materializecss.com/>

refreshless.com. (marec 2017). *noUiSlider*. Pridobljeno iz <https://refreshless.com/nouislider/>

Skornšek, A., & Rizmal, J. (marec 2017). <https://maturanc.com/>. Pridobljeno iz Maturanc.

## IZJAVA\*

Mentor (-ica) Boštjan Resinovič, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom

Maturanc,  
katere avtorji (-ice) so Amadej Skornjsek, Jaka Rizmal, \_\_\_\_\_ :

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo (-ičino) dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 10. 3. 2017



Podpis mentorja(-ice)

Podpis odgovorne osebe

\*

### POJASNILO

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.