

TEHNIČNA DOKUMENTACIJA

za

študentsko socialno omrežje

studentPal

Matevž Mak, Gregor Sotlar

Koper, 9.7.2020

Kazalo vsebine

Definicija naloge	3
Opis zahtev	3
Uporabniške zahteve	3
Tok podatkov	4
Podatkovni model	4
Vmesniki	5
Vmesnik registracije in prijave	5
Vmesnik nadzorne plošče	6
Podatkovna baza	7
Uporabljene knjižnice	8
Express	8
Connect-flash	8
Method-override	8
Passport	8
Mongoose	9

1. Definicija naloge

StudentPal je socialno omrežje namenjeno študentom/dijakom/fakultetam da izboljšajo študentske/dijaške medsebojne odnose in povečajo kvaliteto izmenjanega znanja. Fakulteta pa lahko u uporabo platforme izboljšajo svoj tutorski sistem.

Cilji projekta je torej izdelati socialno omrežje, kjer bi si študentje izmenjavali znanje in zapiske.

2. Opis zahtev

Za izdelavo platforme smo uporabili MEAN stack, kjer lahko zelo enostavno:

- Definiramo komponente naše platforme
- Predstavimo dinamične podatke
- Definiramo storitve
- Uporabljamo NPM knjižnice
- Manipuliramo izgled platforme
- Povežemo na naš API

MEAN stack je sestavljen iz podatkovne baze MongoDB, express-a, Angularja in Nodejs. Celoten projekt je razdeljen na dva dela. Prvi del je frontend, kjer uporabljamo framework Angular, ki je odličen za manipulacijo videza spletne strani in definiranje storitev, ki kličejo podatke na naš API. Ostale komponente MEAN stacka spadajo pod backend. S pomočjo Node.js smo definirali API, ki pobira podatke iz podatkovne baze MongoDB.

Celoten projekt je gostovan na oblačni platformi Heroku za gostovanje podatkovne baze mongoDB pa smo uporabili oblačno platformo MongoAtlas. Oba serverja sta gostovana v Evropi.

2.1 Uporabniške zahteve

- Celoten sistem je namenjen študentom. Potrebujemo vmesnik, kjer lahko uporabnik ustvari račun in se prijavi v sistem. Vsak uporabnik lahko unikatno generira svoje socialno okolje na spletni aplikaciji. Potrebujemo vmesnik, ki omogoča kreiranje spletne učilnice za posamičen predmet ali celo skupino predmetov.
- Ob kreiranju spletne učilnice za posamičen predmet mora vmesnik omogočati dodajanje novih članov, dodajanje administrativnih pravic, spremembo imena skupine ali predmeta.
- Vmesnik spletne učilnice mora biti zgrajen iz več modulov, kot so: omogočanje objavljjanja zapiskov, kreiranje anket, postavljanje vprašanj, odgovarjanje na vprašanja, objavljjanje pomembnih rokov in klepetalnica.
- Vsak modul mora prikazati podatke, ki so povezani na specifičen predmet. Z vsemi podatki, ki krožijo po specifičnem razredu ali skupini, opravljajo ti moduli, posledično morajo biti podatki prikazani dinamično. - Sistem mora omogočati sistem glasovanja. Sistem glasovanja se upošteva pri modulih, kot so: nalaganje zapiskov, postavljanje vprašanj, kreiranje anket in odgovarjanje na vprašanja. Uporabnik lahko glasuje, ali so mu

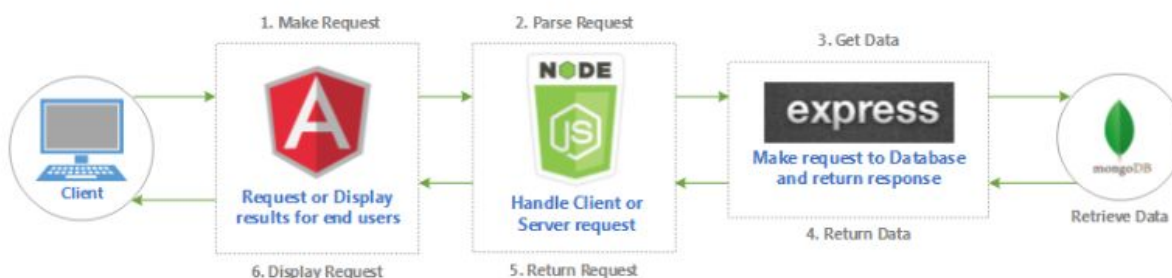
zapiski, vprašanja ali odgovori koristili. Tukaj moramo upoštevati, da lahko uporabnik za specifično vsebino glasuje le enkrat.

- Vsi moduli morajo omogočati shranjevanje vsebine spletne učilnice v podatkovno bazo. Pri klepetalnici se morajo uporabljati socketi, ker gre za pogovor med uporabniki v realnem času.

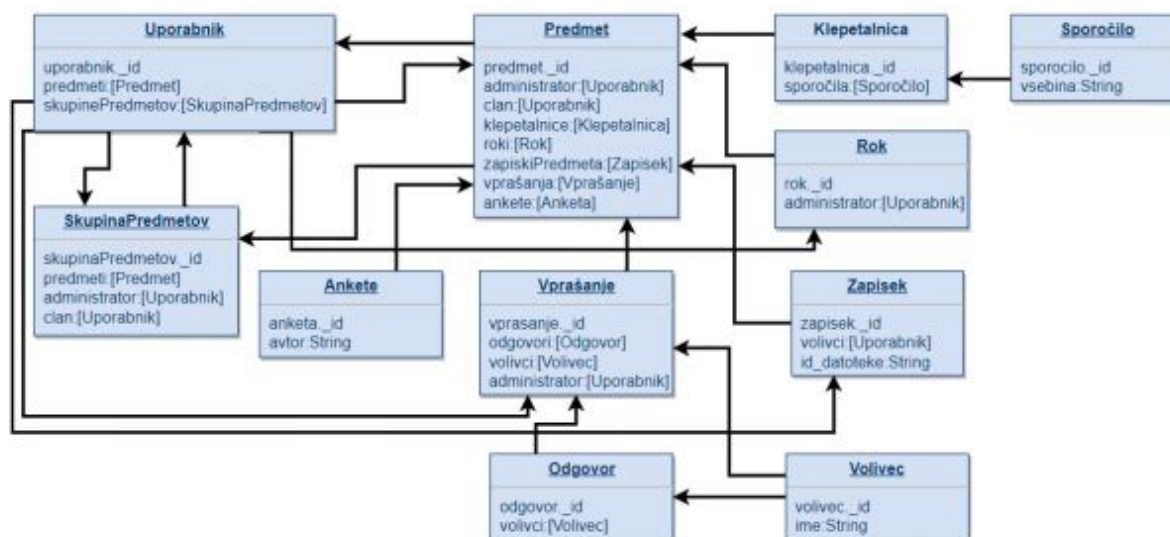
- Ko uporabnik objavi vprašanje ali zapiske, morajo pri tem biti obveščeni vsi uporabniki.

2.2 Tok podatkov

Celotna platforma je sestavljena iz statičnih in dinamičnih komponent. Statične komponente so komponente, ki ostanejo vidne ne glede na operacijo uporabnika. Ker vsi procesi delujejo na enak način bom opisal samo en proces. Uporabnik lahko kreira skupino. Pri kreaciji izpolni podatke, ki se pošljejo specifični storitvi, ki je definirana v frameworku Angular. Ta storitev pošlje prošnjo na API v obliki GET, POST, PUT, DELETE. Naš API nato procesira podatke in poišče iskane podatke iz podatkovne baze in jih nato vrne nazaj uporabniku. Ti podatki se nato dinamično prikažejo s pomočjo Angular frameworka.



3. Podatkovni model



Zbirka lahko vsebuje tudi sklice na ostale objekte oz. zbirke. S pomočjo metode populate() lahko dobimo vse gnezdene podatke posamezne zbirke. V primeru SQL moramo dve tabeli združiti in nato filtrirati podatke, v primeru MongoDB podatkovne baze pa se uporabi metoda populate() za vse gnezdene zbirke posamezne zbirke.

V zbirki vprašanja se nahajajo vsa vprašanja specifičnega predmeta ali predmeta v skupini. Prav tako vsako vprašanje vsebuje reference dokumentov ostalih zbirk, kot so volivci in odgovori. Torej za vsako vprašanje lahko ugotovimo kateri odgovori so povezani na specifično vprašanje in kdo je že glasoval specifično vprašanje.

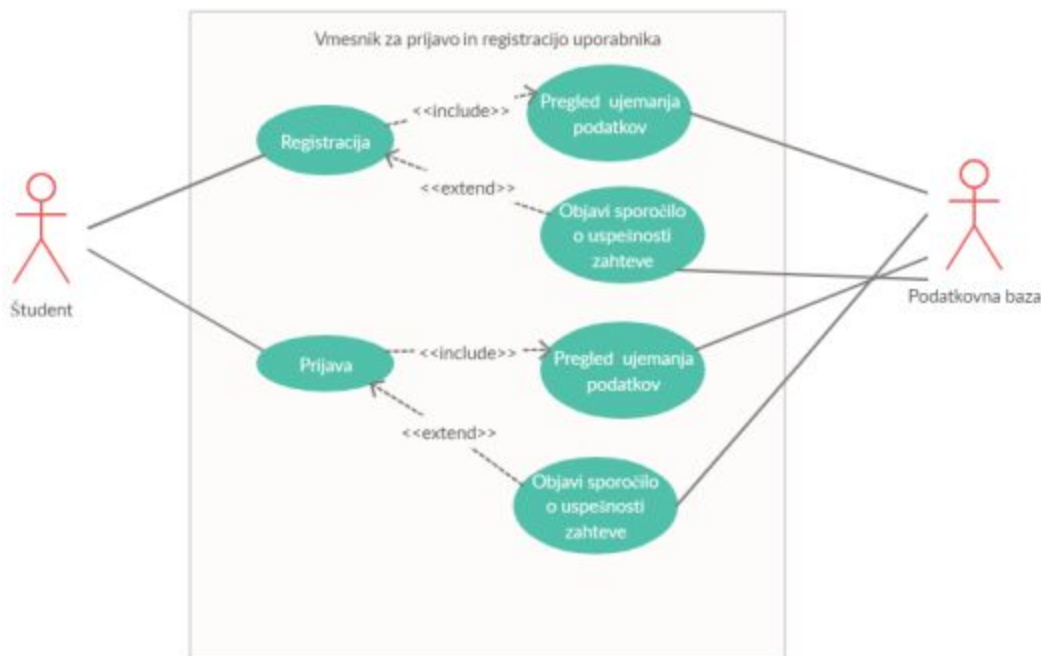
Predmet ima največjo zbirko. V njej lahko najdemo reference na več ostalih zbirk, to so npr. administratorji, udeleženci predmeta, klepetalnica, pomembni roki, zapiski, vprašanja in ankete. Tako lahko za specifičen predmet dobimo vse podatke, ki so povezani na specifičen predmet s specifičnim indeksom.

Zbirka zapiski vsebuje attribute, kot so avtor, ime datoteke, tip datoteke in referenca na zbirko volivci. Uporabnik, ki bo za določeno datoteko volil, ali mu je koristila ali ne, se bo povezal na to datoteko, tako bomo lahko v vsakem trenutku vedeli, kateri uporabnik je že volil in kateri ni.

Zbirka uporabnik vsebuje osnovne attribute, kot so e-naslov, ime, geslo in referenca na zbirko predmet in skupine predmetov. Tako točno vemo, v katerih predmetih ali skupinah predmetov uporabnik sodeluje.

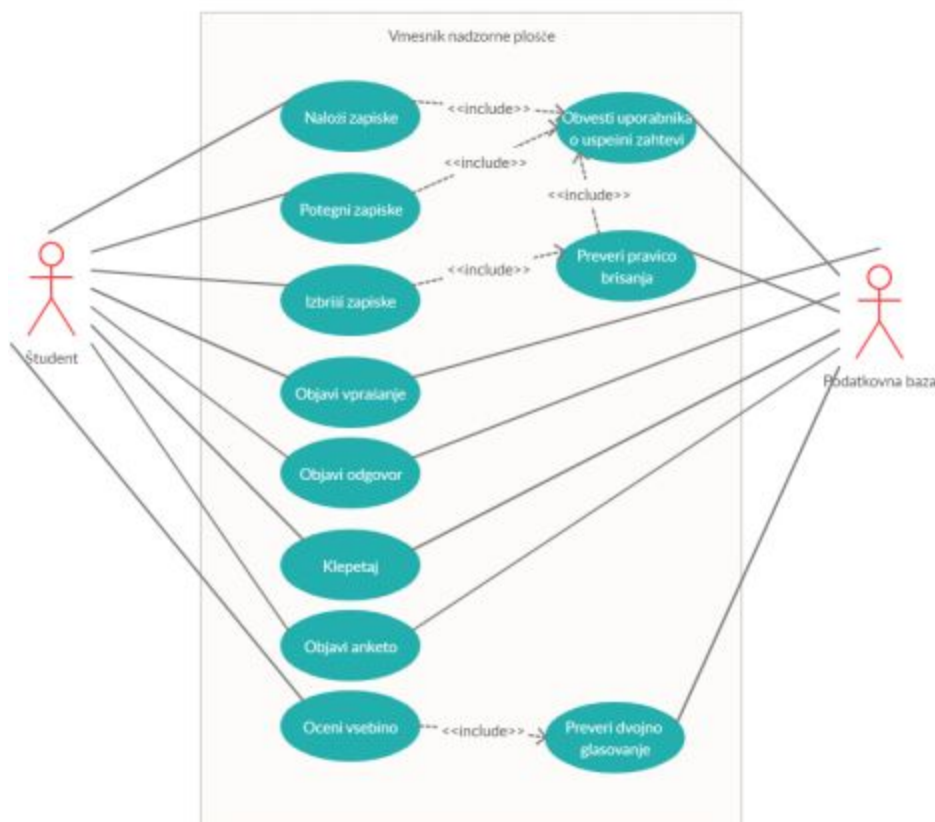
4. Vmesniki

4.1 Vmesnik registracije in prijave



Grafični vmesnik domače strani omogoča prijavo in registracijo uporabnika. Uporabnik mora pri prijavi ali registraciji izpolniti vsa polja. Poslani podatki s forme na strežnik morajo biti skriti s POST metodo. Na strežnikovi strani moramo preveriti vse robne pogoje, ali je elektronski naslov že uporabljen, ali se vneseni gesli ujemata, ali se geslo sklada z geslom v podatkovni bazi itd. V primeru težav ali uspešne prijave ali registracije moramo uporabnika obvestiti, ali je bila njegova zahteva uspešna.

4.2 Vmesnik nadzorne plošče

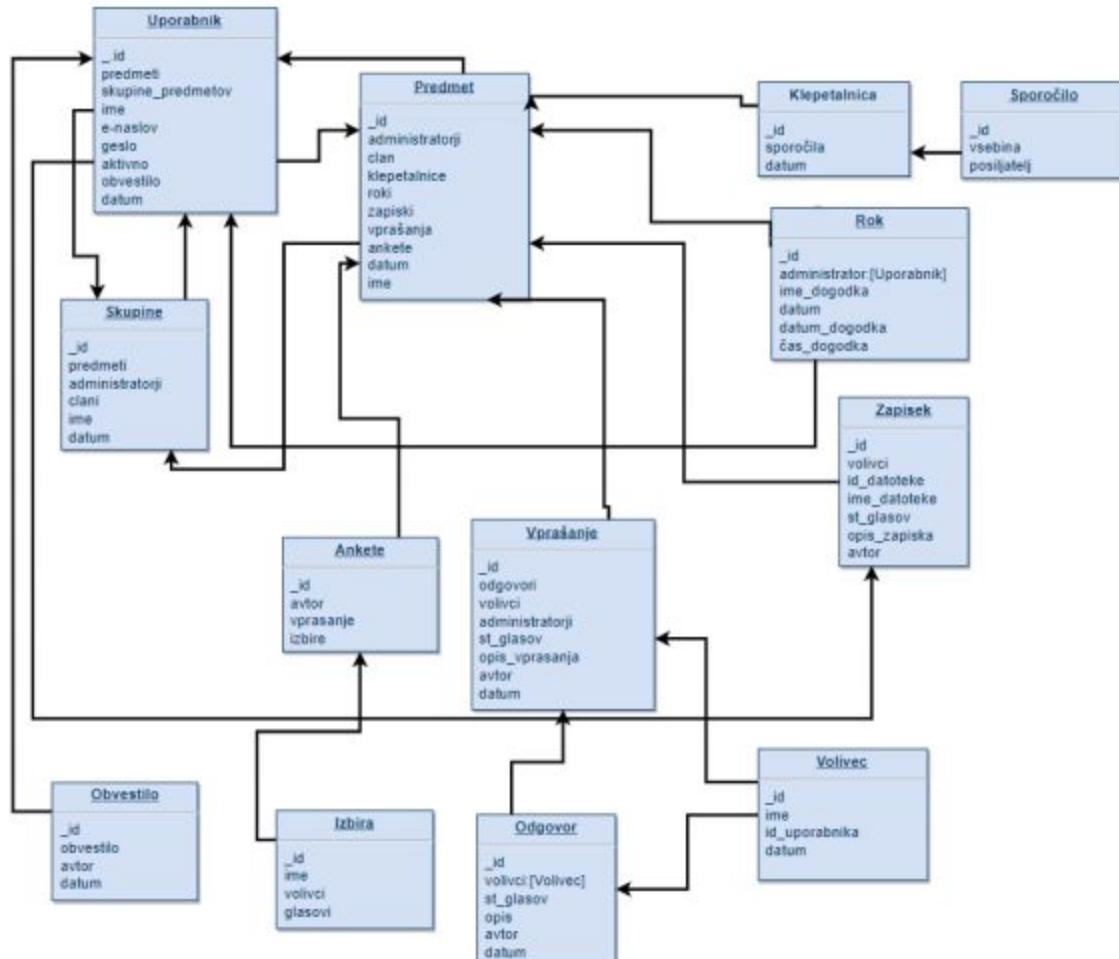


Vmesnik nadzorne plošče nam omogoča pregled vseh operacij uporabnika v sistemu. Vsebino spreminjamo na podlagi izbranega predmeta ali skupine predmetov. Na predmet so vezani številni podatki, kot so klepetalnica, zapiski, vprašanja, odgovori na vprašanja. V sistemu je na voljo tudi glasovanje o vsebini, kot so: vprašanja, odgovori na vprašanj in zapiski. Vse glavne funkcionalnosti so porazdeljene v zavihke. Podatki, ki jih pošilja uporabnik na strežnik, se ustavijo v posameznem modulu, ta jih procesira in pošlje v podatkovno bazo. O tem, ali je zahteva uporabnika uspešna ali ne, ga obvestimo. Vsebino v posameznem predmetu lahko izbrišejo le administrator predmeta oz. skupine ali pa sam avtor vsebine... Vmesnik nadzorne plošče nam omogoča pregled vseh operacij uporabnika v sistemu. Vsebino spreminjamo na podlagi izbranega predmeta ali skupine predmetov. Na predmet so vezani številni podatki, kot so klepetalnica, zapiski, vprašanja, odgovori na vprašanja in ankete. V sistemu je na voljo tudi glasovanje o vsebini, kot so: vprašanja, odgovori na vprašanja, zapiski in ankete.

Vse glavne funkcionalnosti so porazdeljene v zavihke. Podatki, ki jih pošilja uporabnik na strežnik, se ustavijo v posameznem modulu, ta jih procesira in pošlje v podatkovno bazo. O tem, ali je zahteva uporabnika uspešna ali ne, ga obvestimo. Vsebino v posameznem predmetu lahko izbrišejo le administrator predmeta oz. skupine ali pa sam avtor vsebine.

5. Podatkovna baza

Pri izdelavi podatkovne baze moramo paziti na poimenovanje zbirk in njenih atributov. Moramo si zastaviti neka slovnična pravila, ki se bodo skladala skozi celoten projekt zaradi preglednosti. Pri implementaciji zbirk in atributov se poskušamo izogibati šumnikov zaradi potencialnih napak. Struktura dokumentov v zbirki je v obliki JSON.



Gostovanje podatkovne baze omogoča oblčna platforma MongoDB Atlas, ki ponuja brezplačno uporabo do 500 MB podatkov. Ustvarili smo svoj cluster v srednji Evropi, platforma generira unikaten link z vgrajenim geslom uporabnika, ki ga nato uporabimo pri

povezavi na strežnikovi strani, kjer poteka pretok informacij od uporabniškega vmesnika v podatkovno bazo. Cluster je mreža računalnikov, ki sodelujejo skupaj in predstavljajo enoten sistem za izvajanje specifičnih nalog.

6. Uporabljene knjižnice

Za razvoj sistema smo uporabili 19 knjižnic, kar nam je omogočalo hitrejši in enostaven razvoj. V tem poglavju bomo opisali nekaj glavnih knjižnic. Vse knjižnice se shranijo v datoteko, imenovano `package.json`, tako lahko sistem prenesemo na katerokoli napravo in izvedemo ukaz `npm install`, ki naloži vse knjižnice na napravo.

6.1 Express

Express je ena od najbolj uporabljenih knjižnic pri razvoju Node.js aplikacij. V le nekaj vrsticah lahko generiramo lokalni server aplikacije. Je zelo prilagodljiva in minimalistična knjižnica. Uporaba nam omogoča implementacijo robustnih poti oz. URL naslovov, na katere »posluša aplikacija«. Prav tako nam omogoča posredovanje podatkov preko http klicev. Podatke lahko dobimo preko url naslova ali preko forme v HTML dokumentu. Ko podatki pridejo na definirano pot, lahko v tej poti izvedemo številne funkcije in operacije. Nato lahko posodobljene podatke dinamično prikažemo s funkcijo `res.render()`, v katero lahko vstavimo objekt, ki vsebuje posredovane spremenljivke, ki nosijo poljubne vrednosti.

6.2 Connect-flash

Pri razvoju projekta je pomembno, da uporabniku pošiljamo povratne informacije, če se je proces, ki ga je izvajal končal, ali je prišlo do napake. To obveščanje omogoča knjižnica `connect-flash`, ki deluje skupaj s knjižnico `express`. Preden uporabimo funkcijo `redirect()`, zapišemo sporočilo v poseben del seje. Ko se grafični vmesnik uspešno naloži se sporočilo prebere in izbriše iz seje. Tako uporabniku povemo, ali je bila operacija uspešna ali ne.

6.3 Method-override

Node.js omogoča številne tipe metod pri definiranih poteh Node.js aplikacije. Te metode so POST, GET, PUT, DELETE. Knjižnica `method-override` mi omogoča, da lahko pošiljamo podatke preko HTML forme na pot, ki je definirana z metodo DELETE. Knjižnica služi, da je projekt bolj pregleden in da točno vemo, kaj katera pot počne, ali poskušamo pridobiti informacije iz podatkovne baze, ali poskušamo izbrisati podatke iz podatkovne bazi, ali poskušamo posodobiti podatke v podatkovni bazi in ali želimo naložiti podatke v podatkovno bazo.

6.4 Passport

Passport knjižnica je ena najbolj uporabljenih knjižnic za avtentikacijo uporabnika v sistem. Je fleksibilna knjižnica, kjer lahko definiramo mnogo strategij, preko katerih lahko opravljamo proces registracije in prijave. Za registracijo in prijavo lahko uporabljamo socialne platforme, kot so Google plus, Facebook, Twitter, ki jih definiramo v svojih strategijah. Ko se uporabnik prijavi v sistem, shranimo njegov unikatni ID v sejo. Ko uporabnik izvaja operacije v sistemu, pride do številnih obiskov poti na strežnikovi strani, kjer uporabimo njegov ID tako, da v vsakem trenutku vemo, kateri uporabnik izvaja specifične operacije.

6.5 Mongoose

Uporaba knjižnice Mongoose nam omogoča, da se povežemo na podatkovno bazo. Gre za orodje, ki nam pomaga pri modeliranju MongoDB objektov, ki so v JSON obliki. Ukazi se izvajajo asinhrono. Pri projektu smo definirali številne modele oz. sheme, in sicer, kakšne attribute mora vsebovati specifičen objekt. Pri tem je pomagala Mongoose knjižnica. Mongoose omogoča eleganten način poizvedovanja podatkov iz podatkovne baze. Vsak klic poizvedovanja vsebuje povratno funkcijo, kjer preverimo, ali je prišlo do napake.