

BOOKFACE

Strokovno poročilo

Mentor: Darjan Toth, prof. Avtor: Aljaž Rojina

Ljubljana, april 2024

# POVZETEK

Poročilo predstavlja postopek izdelave enostavne aplikacije za deljenje informacij, kateri ima centralni dostop, kjer uporabniki med seboj delijo ključne informacije. Izdelek, grajen po MVC načrt, uporabnikom omogoča registracijo, prijavo, ter deljenje objav, katere vsebujejo pomembne informacije za različne skupine ljudi. Zaradi potreb po varnosti podatkov, ter strokovni učinkovitosti je bil ta razvit s poudarkom na enostavni uporabi izmenjevanja informacij. Uporabljeni so jeziki PHP in HTML, podatkovna baza pa je implementirana v phpMyadmin. Bralcu ponuja vpogled v rešitev problema slabe komunikacije, izmenjave informacij med uporabniki, ter opisuje odločitve in postopke razvoja enostavnega omrežja za rešitev tega problema.

**Ključne besede:** socialno orežje, spletna stran, deljenje objav, PHP, HTML, MySQL

# KAZALO

[POVZETEK 2](#_Toc163922228)

[KAZALO 3](#_Toc163922229)

[KAZALO SLIK 4](#_Toc163922230)

[1. UVOD 5](#_Toc163922231)

[2. PROGRAMSKA ORODJA 6](#_Toc163922232)

[3. STRUKTURA PROJEKTA 7](#_Toc163922233)

[3.1. MVC: 7](#_Toc163922234)

[4. SESTAVA MVC ORODJA 8](#_Toc163922235)

[5. STRUKTURA PODATKOVNE BAZE 8](#_Toc163922236)

[6. UPORABNIKI 9](#_Toc163922237)

[7. OBJAVE 9](#_Toc163922238)

[7.1 Struktura datotečnega sistema 10](#_Toc163922239)

[7.2 LOČITEV MAP 10](#_Toc163922240)

[8. UPORABNIK 11](#_Toc163922241)

[9. REGISTRACIJA 12](#_Toc163922242)

[10. PRIJAVA 13](#_Toc163922243)

[11. PONASTAVITEV GESLA 13](#_Toc163922244)

[12. ADMINISTRATIVNI UPORABNIK 14](#_Toc163922245)

[13. OBJAVE 14](#_Toc163922246)

[14. DODAJANJE NOVE OBJAVE 15](#_Toc163922247)

[15. PRIKAZ OBJAVE 15](#_Toc163922248)

[16. UREJANJE OBJAV 16](#_Toc163922249)

[ZAKLJUČEK 17](#_Toc163922250)

[VIRI IN LITERATURA: 17](#_Toc163922251)

[IZJAVA O AVTORSTVU 18](#_Toc163922252)

# KAZALO SLIK

[Slika 1: MariaDB 6](#_Toc164184438)

[Slika 2: phpMyAdmin 6](#_Toc164184439)

[Slika 3: Viral Studio Code 6](#_Toc164184440)

[Slika 4: HTML 7](#_Toc164184441)

[Slika 5: Struktura 7](#_Toc164184442)

[Slika 6: Struktura podatkovne baze 9](#_Toc164184443)

[Slika 7: Uporabniki 10](#_Toc164184444)

[Slika 8: Objave 10](#_Toc164184445)

[Slika 9: Ločitev map 11](#_Toc164184446)

[Slika 10: Uporabnik 12](#_Toc164184447)

[Slika 11: Uporabnik 1.0 12](#_Toc164184448)

[Slika 12: Registracija 13](#_Toc164184449)

[Slika 13: Prijava 13](#_Toc164184450)

[Slika 14: Vpis 14](#_Toc164184451)

[Slika 15: Ponastavitev gesla 14](#_Toc164184452)

[Slika 16: Objave 15](#_Toc164184453)

[Slika 17: Dodajanje nove objave 16](#_Toc164184454)

[Slika 18: Prikaz objave 16](#_Toc164184455)

[Slika 19: Sprememba objave 17](#_Toc164184456)

[Slika 20: Poprava objave 17](#_Toc164184457)

# UVOD

Za svoj maturitetni izdelek sem si izbral izdelati spletno aplikacijo za socialno omrežje s centralizeranim dostopom. Za to temo sem se odločil, ker sem delal v podjetju, kjer sem opazil da je šibka točka širjenje informacij. Za posredovanje teh smo uporabljali Facebook, ampak se zaradi varnosti podatkov ni posredovalo vseh pomembnih informacij. Predlagal sem druge programe in spletne strani, ki imajo veliko funkcionalnosti, ki ustrezajo takim zahtevam. Vendar ni bilo privlačno zaradi visoke cene. Moj cilj je bil, da naredim bolj enostavno spletno stran za širjenje ključnih informacij in posledično nizkocenovno, ki bi lahko ustrezale tudi manjšim podjetjem z skromnim proračunom. Pri razvijanju aplikacije sem se najbolj osredotočil na uporabnost, varnost, ter pa nizke stroške, kateri bi ustregli tudi osebam z manjšim proračunom. Da bi dosegel zaželeni cilj sem izvedel podrobno analizo že obstoječih platform in tehnologij, ter poskušal prepoznati funkcionalnosti, ki so ključnega pomena za učinkovito izmenjavo informacij. Med razvojem sem uporabljal tehnologijo, ki je zagotavljala stabilnost, ter varnost omrežja. Prav tako sem se tudi osredotočil na intuitiven uporabniški vmesnik, ki bi uporabo olajšal tudi manj izkušenim uporabnikom. Izdelek se mi zdi inovativna rešitev za izboljšanje komunikacije, ter izmenjave informacij med uporabniki, ki stremijo k varnosti, učinkovitosti in manjšim stroškom. Rekel bi, da v današnjem hitro spreminjajočem se okolju je ključnega pomena učinkovito komuniciranje in izmenjava informacij. Vendar se številni soočajo s težavami pri širjenju ključnih informacij, kar lahko vodi v zamude, zmedo in izgubo produktivnosti. Klasična orodja, kot je Facebook, pogosto niso ustrezna rešitev, saj ne zagotavljajo zadostne ravni varnosti podatkov in prilagodljivosti. Namen moje raziskave je razviti socialno omrežje z enostavnim dostopom, ki bo poleg tega zagotavljal tudi varnost, ter učinkovito izmenjavo informacij. Moj cilj pa je je razviti platformo, ki bo omogočala hitro in enostavno deljenje informacij med zaposlenimi, zagotoviti visoko raven varnosti podatkov, ter zmanjšati stroške komunikacijskih orodij za podjetja.

# PROGRAMSKA ORODJA

Za svoj maturitetni izdelek sem uporabil več različnih orodji, ki so mi pomagali sestaviti celoto. Za sistem za upravljanjem s podatkovno bazo sem uporabil MariaDB, katero sem upravljal čez uporabniški vmesnik phpmyadmin. Za rokovanje z podatkovno bazo se je phpmyadmin izkazal za zelo dobro in enostavno za uporabo. Za vso kodo, ki sem jo spisal, sem uporabil Visal Studio Code. Microsoftov urejevalnik za pisanje in urejanje programske kode. Sem pa kasneje ugotovil tudi sam zakaj je tako popularen.Uporabil sem tudi paket strežniških programov XAMPP, ki na osebnem računalniku ustvari »mini« strežnik za testiranje kode. Organizacija mi je omogočila sledenje spremembam v kodi, ter varno shranjevanje in deljenje mojega projekta. Poleg tega sem za razvoj spletnega vmesnika uporabil HTML, ki je ključni jezik za oblikovanje in interaktivnost spletnih strani. Pri oblikovanju uporabniške izkušnje sem se opiral na koncepte uporabniškega vmesnika, pri čemer sem se lahko naslonil na različne spletne vire s tega področja.

Za reševanje morebitnih težav in iskanje odgovorov na vprašanja sem redno uporabljal spletna foruma, kot sta Stack Overflow in Reddit, kjer sem lahko dobil pomoč od izkušenih razvijalcev. Pri razvoju programske logike sem se opiral na znanje in koncepte objektno orientiranega programiranja, kar mi je omogočilo bolj strukturiran in vzdržljiv razvoj aplikacije.

Pri načrtovanju arhitekture in razvoju sem se lahko naslonil na spletne tečaje o razvoju programske opreme, ki so mi pomagali razumeti najboljše prakse in vzorce oblikovanja. Za zagotavljanje varnosti moje aplikacije sem uporabil različne tehnike, kot so šifriranje gesel, preverjanje uporabniških pravic.



Slika 1: MariaDB



Slika 2: phpMyAdmin



Slika 3: Viral Studio Code



Slika 4: HTML

# STRUKTURA PROJEKTA

Za strukturiranje sem se odločil za uporabo MVC ali »Model View Controller« načrta. To je arhitekturni načrt programske opreme. Baziran je na objektnem programiranju in ni vezan samo na PHP, uporablja se tudi v drugih programskih jezikih. Promovira organizacijo in loči funkcije aplikacije. Razdelimo jih na tri glavne.

## MVC:

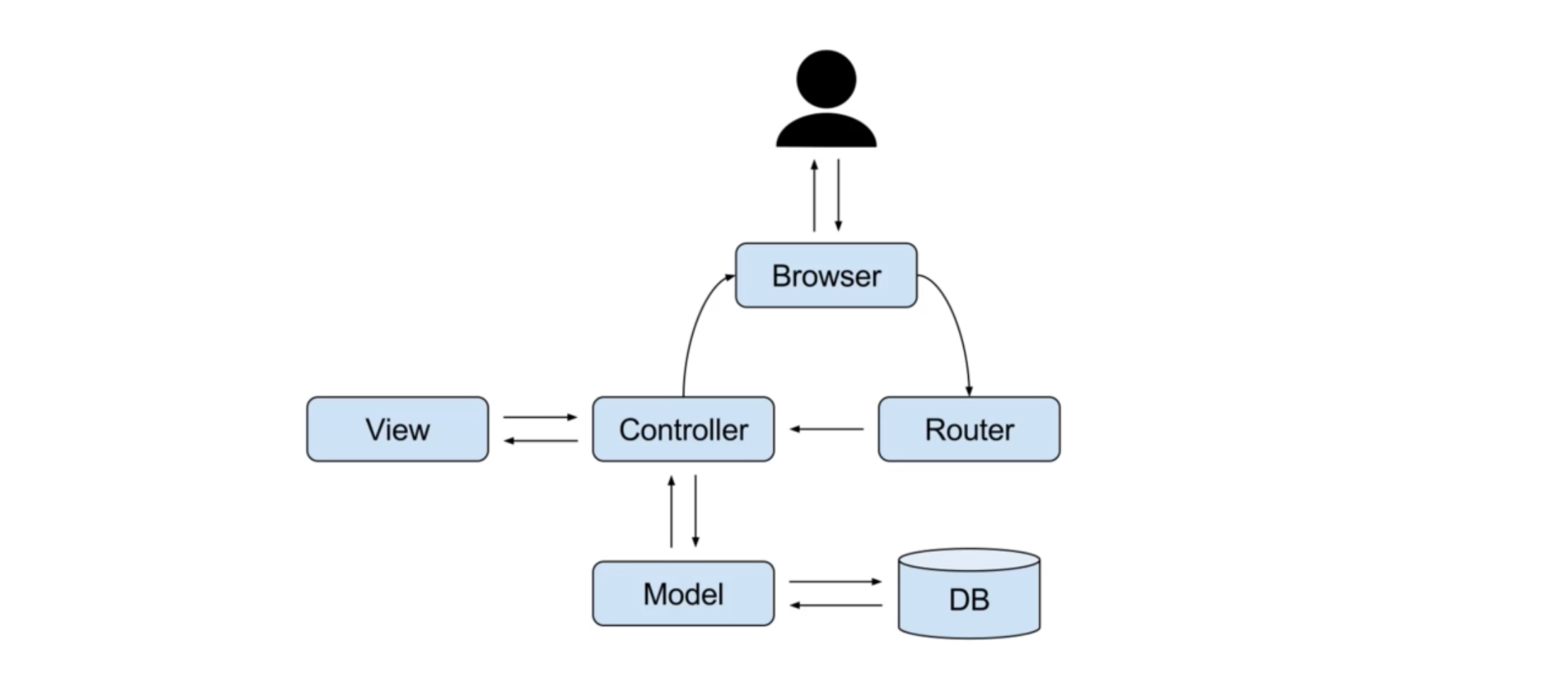
1. MODEL: sodeluje z podatkovno bazo in komunicira s krmilnikom

2.POGLED:

* Kar uporabnik vidi v brskalniku(Uporabniški vmesnik),
* Ponavadi iz HTML/CSS,
* komunicira z krmilnikom,
* so mu lahko podane dinamične vrednosti iz krmilnika.

3.KRMILNIK

* Prejme vnos podatkov iz URL-ja, pogleda ali obrazca,
* procesira prošnje(GET/POST),
* pošlje podatke pogledu.



Slika 5: Struktura

# SESTAVA MVC ORODJA

Za ogrodje sem si izbral MVC sestavo zaradi boljše organizacije. Sama sestava bo taka, da bo vse naslove v URL-ju usmerjalo čez glavno php datoteko index.php. To lahko naredimo z pomočjo .htaccess datoteke, ki je del Apache serverja in nam dovoli prepisovanje URL-ja. Strukturiral sem sestavo URL-ja, da ga razdelim prvo na kontroler, potem na funkcijo in vse za tem na parametre za funkcijo. Razdelitev URL-ja na kontroler, funkcijo in parametre za funkcijo so ključnega pomena za zagotovitev preglednosti in organiziranosti v MVC okvirju. S tem pristopom se da enostavno slediti potem zahtev URL-ja do ustrezne funkcije in posledično izvedbe določenih akcij v aplikaciji. Prav tako je pomembno, da kontrolerje in funkcije poimenujemo na način, ki jasno odraža njihovo funkcionalnost, da se da hitro ugotoviti, kje se nahaja določena logika in kateri deli aplikacije so odgovorni za posamezne naloge.

* Glavna\_mapa/javno/index.php/**kontroler/metoda/parametri**
* *Bookface/public/Index.php/****posts/show/45***

# STRUKTURA PODATKOVNE BAZE

Sistem za urejanje podatkovnih baz MariaDB sem si izbral, ker sem ga uporabljal na tečaju za PHP. Na tečaju smo ga uporabljali v povezavi z Apache Strežnikom v programu XAMPP. Program dovoljuje razvijalcem testiranja kode lokalno na osebnem računalniku.

Vsa struktura podatkovne baze je bil ustvarjena v uporabniškem vmesniku phpmyadmin, ki je bil ustvarjen za administracijo MySQL prek spletnega brskalinka. Privlačen mi je bil ker poenostavi upravljanje baze podatkov za spletno mesto. Nato sem uporabil njihov vmesnik, kateri generira sql kodo za vnos, urejanje iin limitacijo podatkov, ki jo kasneje tudi izvrši.

Podatkovna baza je sestavljena iz 2 tabel. Imamo tabele posts(»objave«) in users(»uporabniki«). Obe tabeli hranita čas ustvarjenja »created\_at«, ki se nastavi ob vnosu podatka s *CURRENT\_TIMESTAMP*. Vsi primarni ključi uporabljajo mehanizem AUTO\_INCREMENT, ki poskrbi za to, da se ključi ne podvajajo in jih ni potreno ročno/programsko vnašati.

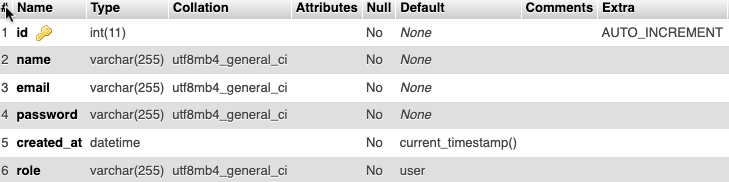


Slika 6: Struktura podatkovne baze

## 5.1 UPORABNIKI

Tabela uporabniki vsebuje vse potrebne podatke, ki jih od uporabnika potrebujemo za registracijo in vpis na spletno stran. Privzeta vloga je uporabnik(»user«), lahko pa jim dodelimo tudi opcijo admin, kateri lahko urejajo stran. Atributi tabele so:

* ID(Primary key)
* Ime(name)
* Email(email)
* Geslo(password)
* Ustvarjeno(created\_at)
* Vloga(role)



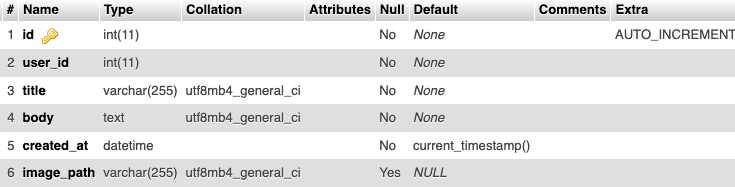
Slika 7: Uporabniki

## 5.2 OBJAVE

Tabela objave(»posts«) vsebuje vse potrebne podatke za nastanek ene objave. Vsebuje naslov in vsebino objave. Če se uporabnik odloči, da bi objavi dodal še sliko, lahko v tej tabeli hranimo tudi pot do slike(»image\_path«). Tako dostopamo do naloženih slik. Ta pa ni nujna, zato je njena privzeta vrednost NULL. Tabela objave je povezana z tabelo uporabnik, ker objavo lahko naredi le uporabnik.

Vsi atributi tabele so:

* ID,
* ID uporabnika,
* Naslov,
* Vsebina,
* Ustvarjeno,
* Pot do slike.



Slika 8: Objave

# Struktura datotečnega sistema

Struktura se je razvijala skozi izdelavo »ogrodja« spletne strani. Datoteke so med sabo ločene v mape za lažjo berljivost in organiziranost. Mapa je še naknadno ločeno na module pri kateri se uporablja. Poleg organizacije datok v mapah za večjo preglednost, lahko v vsaki mapi za boljš strukturo uporabimo še podmape. Poleg tega pa se lahko uporabljajo tudi posebne smernice za shranjevanje slik, stilov, skript in drugih vrst datotek, katere se uporabljajo za spletno stran. To pomaga pri ohranjanju urejenosti oziroma pregldnosti, ter olajša širjenje in vzdrževanje spletne strani.

## 6.1 LOČITEV MAP

Mapa »Public« bo sprednji del, skrbel bo za uporabniški vmesnik. V mapi CSS bo css datoteka, v mapi js bo javascript in v mapi img bodo slike, ki jih bom uporabil za nastanek spletne strani. Vsebuje še glavno php datoteko skozi katero se bo prevsmerjalo vse ostale php datoteke, index.php. To naredimo s pomočjo .htaccess datoteke.



Slika 9: Ločitev map

Mapa »App« bo poskrbela za delovanje v ozadju. Tudi tukaj uporabimo .htaccess datoteke, ampak za skrivanje same mape iz pogleda iz brskalnika. Mapa libraries bo delovala kot srce spletne aplikacije. Vsebovala bo naslednje mape.

Mapa Libraries(»Knjižnjice«) v kateri bodo datoteke core.php; središče, ki bo vzelo url in iz njega razbralo, kateri kontroler, metodo in parameter naloži. Database.php; klass, ki bo vseboval PDO, ogrodje za dostopanje do podatkovnih baz iz php-ja. Controller.php; za lažje nalaganje modelov in pogledov iz vseh ostalih kontrolerjev.

Mapa Modeli(»Models«) v kateri bodo datoteke Post.php in User.php. Te datoteke vsebujejo razdrede in funkcije za upravljanje z podatkovno bazo.

Mapa Pogledi(»Views«) v kateri bodo vse datoteke za uporabniški vmesnik. Razdeljene so na podmape »inc«, v kateri so spravljene dateteke glava, noga in navigacijska vrstica, ki jih vključimo v vsako datoteko pogleda. Ostali pogledi so razdeljeni na skripte za glavne strani, skripte za objave in skripte za uporabnika.

Mapa Kontrolerji(»controllers«) vsebujejo kontrolerje za Users, Posts in Pages.

V mapi Pomagači(»Helpers«) imamo skripte, ki vsebujejo različne funkcije. Ustvarimo jih v teh skriptah in jih kličemo v glavnem programu za boljšo preglednost.

V mapi Konfiguracija pa imamo skripto za konfiguracijo konstant programa.

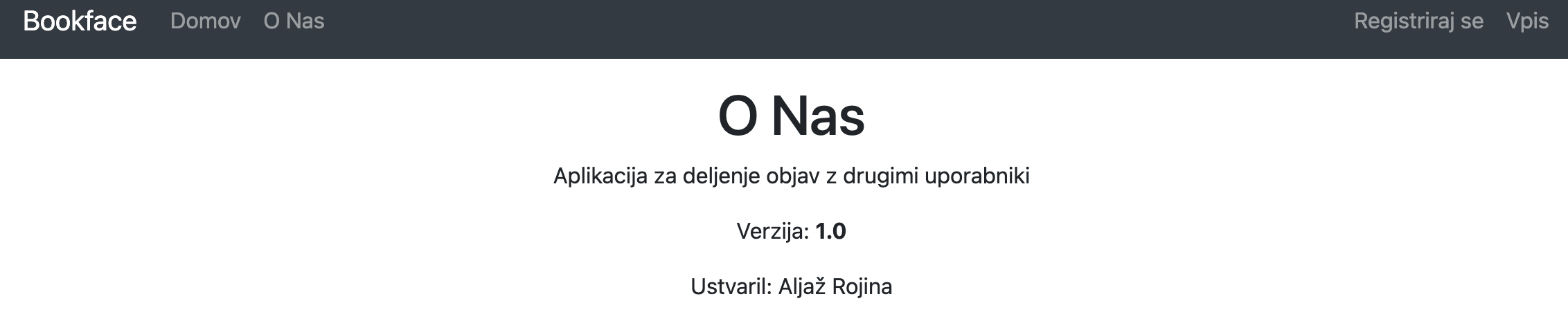
Vsebovala bo tudi bootstrap.php; vključila bo vse datoteke ki jih rabimo da naša spletna stran deluje.

# UPORABNIK

Za vstop na spletno stran, ustvarjanje objav in pregledovanje tujih je zahtevano da je uporabnik registriran in prijavljen. Brez prijave lahko vidimo samo Domačo stran in stran «O nas«.



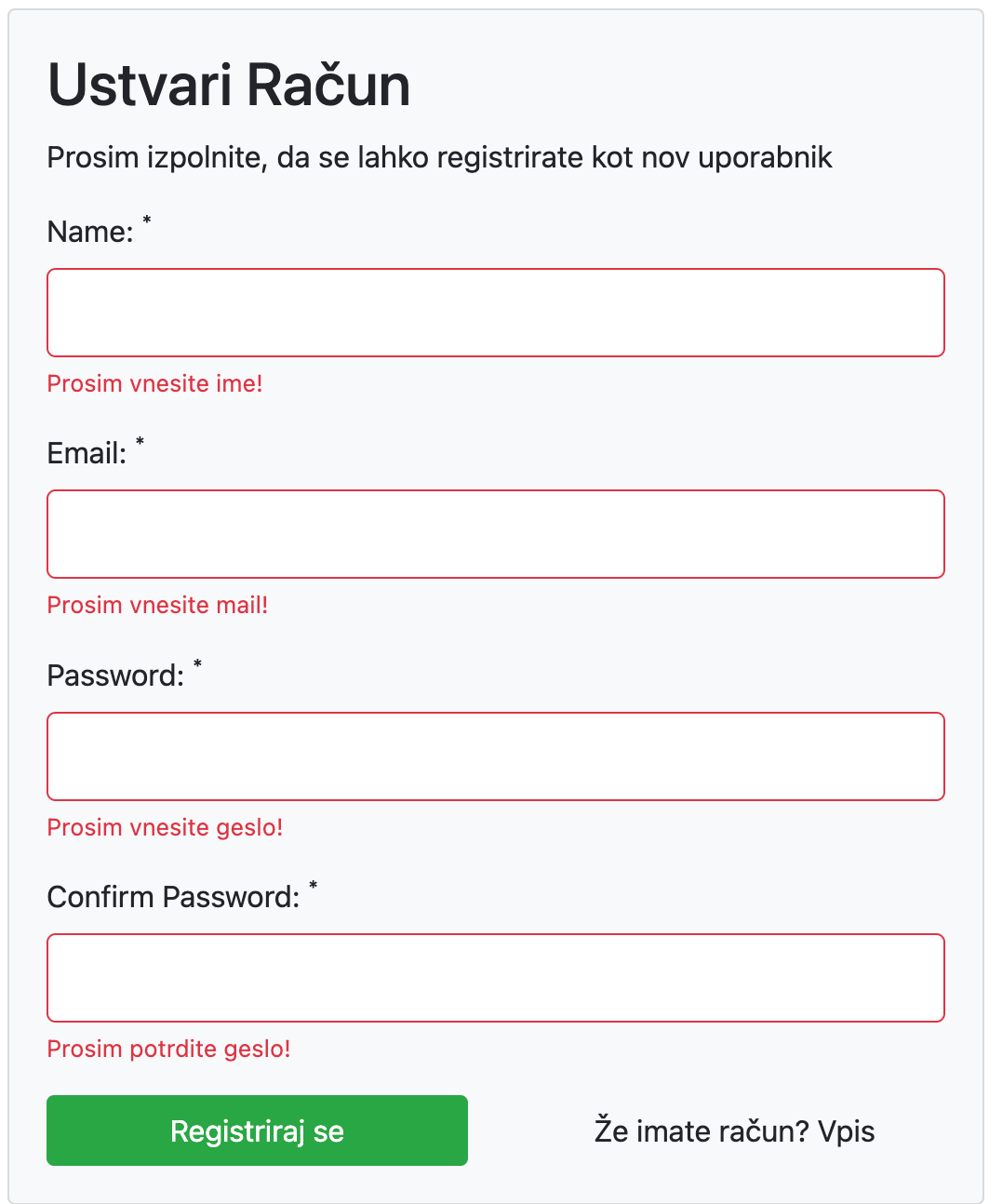
Slika 10: Uporabnik



Slika 11: Uporabnik 1.0

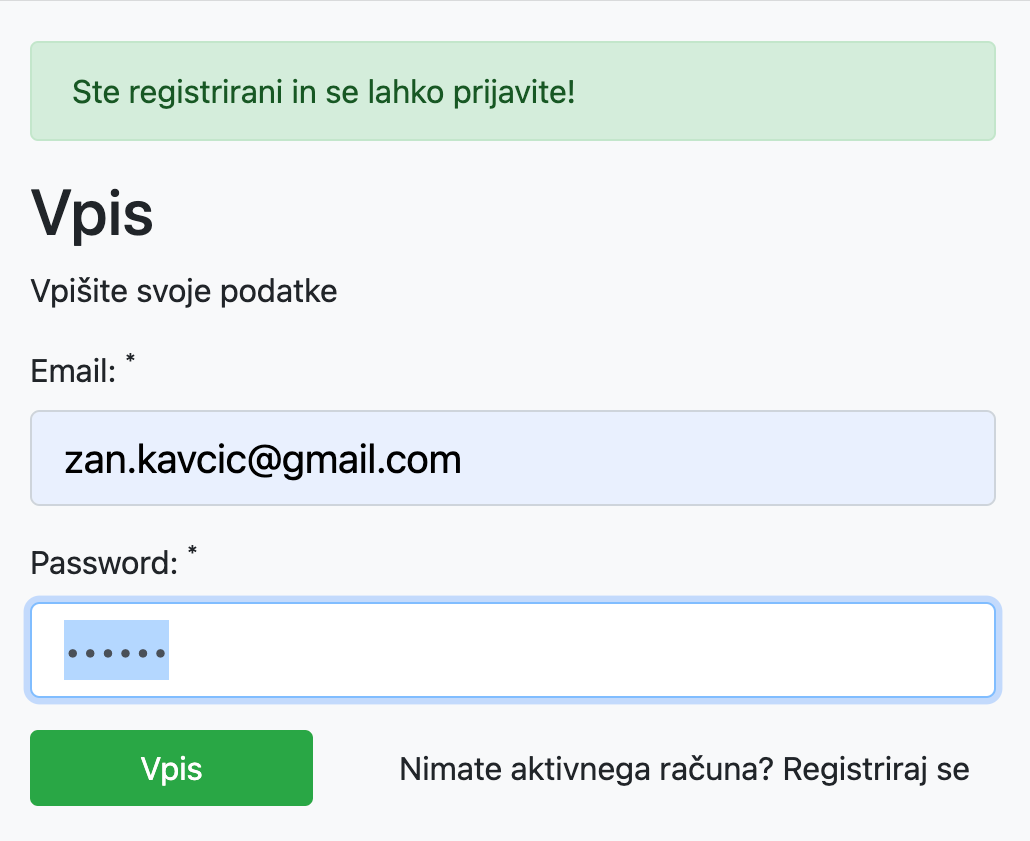
# REGISTRACIJA

Potek registracije je enostaven. Uporabnik mora vnesti svoje ime, email in pa geslo. Geslo mora z dodatnim vnosom potrditi, saj nam program preveri, če se gesli ujemata. Za dodatno varovalo mora biti geslo dolgo več kot 6 znakov. V primeru, če je geslo manjše od 6 znakov, moramo geslo vpisati ponovno. Email mora biti unikaten, ker ne sme imeti program več uporabnikov z istim email naslovom. Ime pa uporabljamo za označbo kdo je posamezni uporabnik in da jih postavimo zraven določene objave. Ob pritisku gumba registracija se preverjo podatki in ob napaki nam na uporabniškem vmesniku javi ustrezno napako. Ob uspešni registraciji nas preusmeri na stran prijave, kjer se lahko prijavimo in uporabljamo spletno stran.



Slika 12: Registracija

Ko se uspešno registriramo se nam pokaže pojavno okno, ki nam sporoči da smo registrirani in se lahko prijavimo.



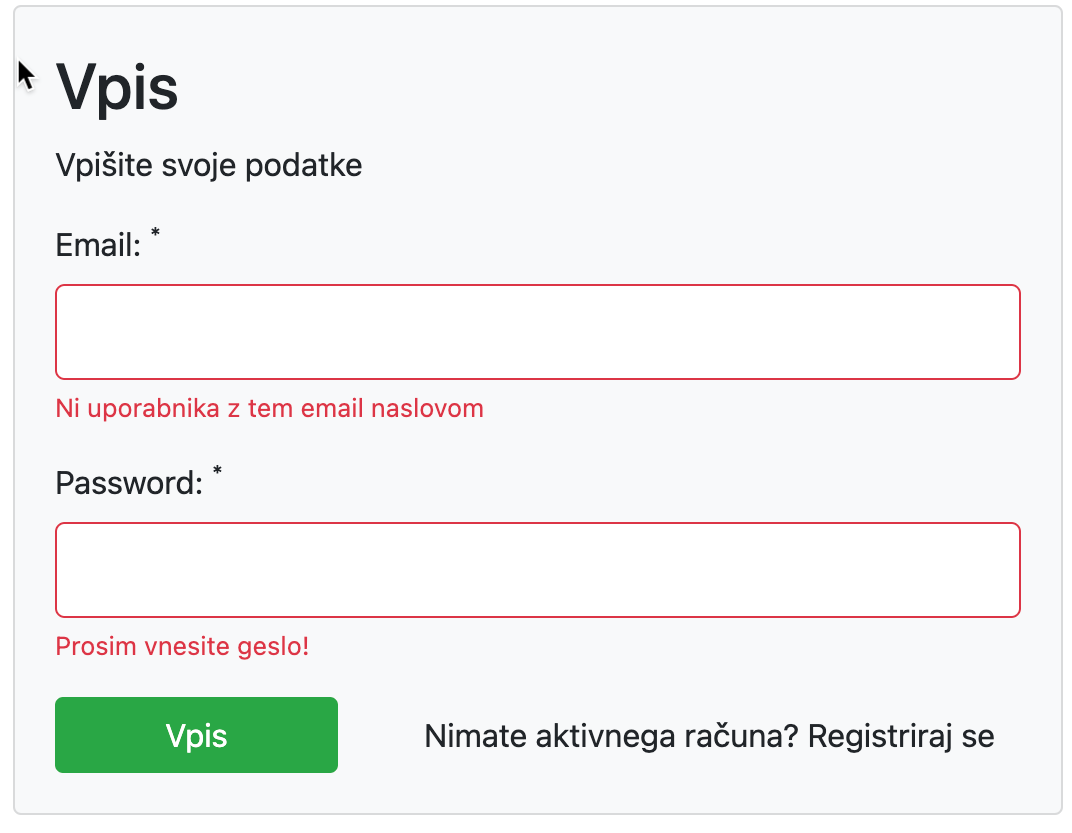
Slika : Prijava

# PRIJAVA

V račun se prijavimo z uporabniškim imenom in geslom. Ob prijavi se preverijo

podatki in se uporabniku prikažejo ustrezne napake. Če je prijava uspešna, se v

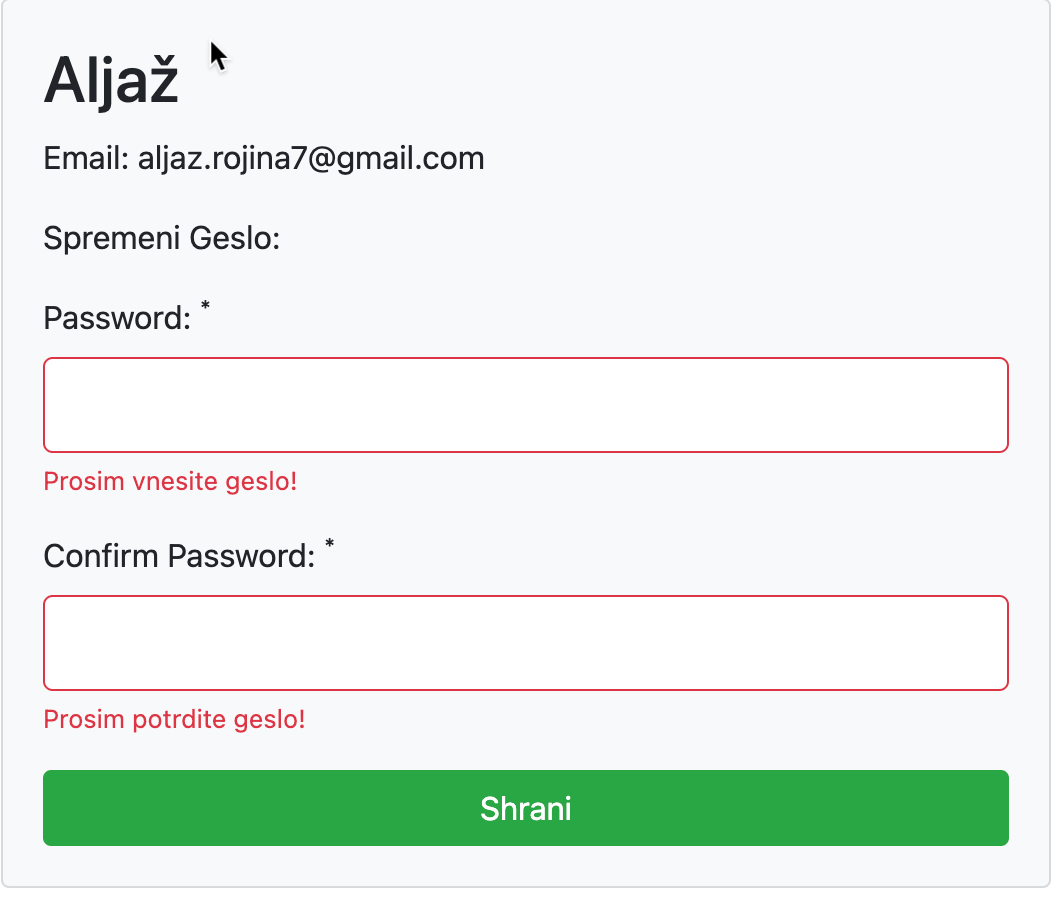
PHP sejo shranita uporabniško ime in zakodirano geslo in nato se prikaže stran z objavami drugih uporabnikov.



Slika : Vpis

# PONASTAVITEV GESLA

Uporabniku je podana možnost ponastavitve gesla. Ko smo prijavljeni, lahko poleg gumba odjava najdemo gumb z imenom uporabnika. Ko ga pritisnemo, nas preusmeri na stran, ki nam da opcijo spreminjanje gesla.



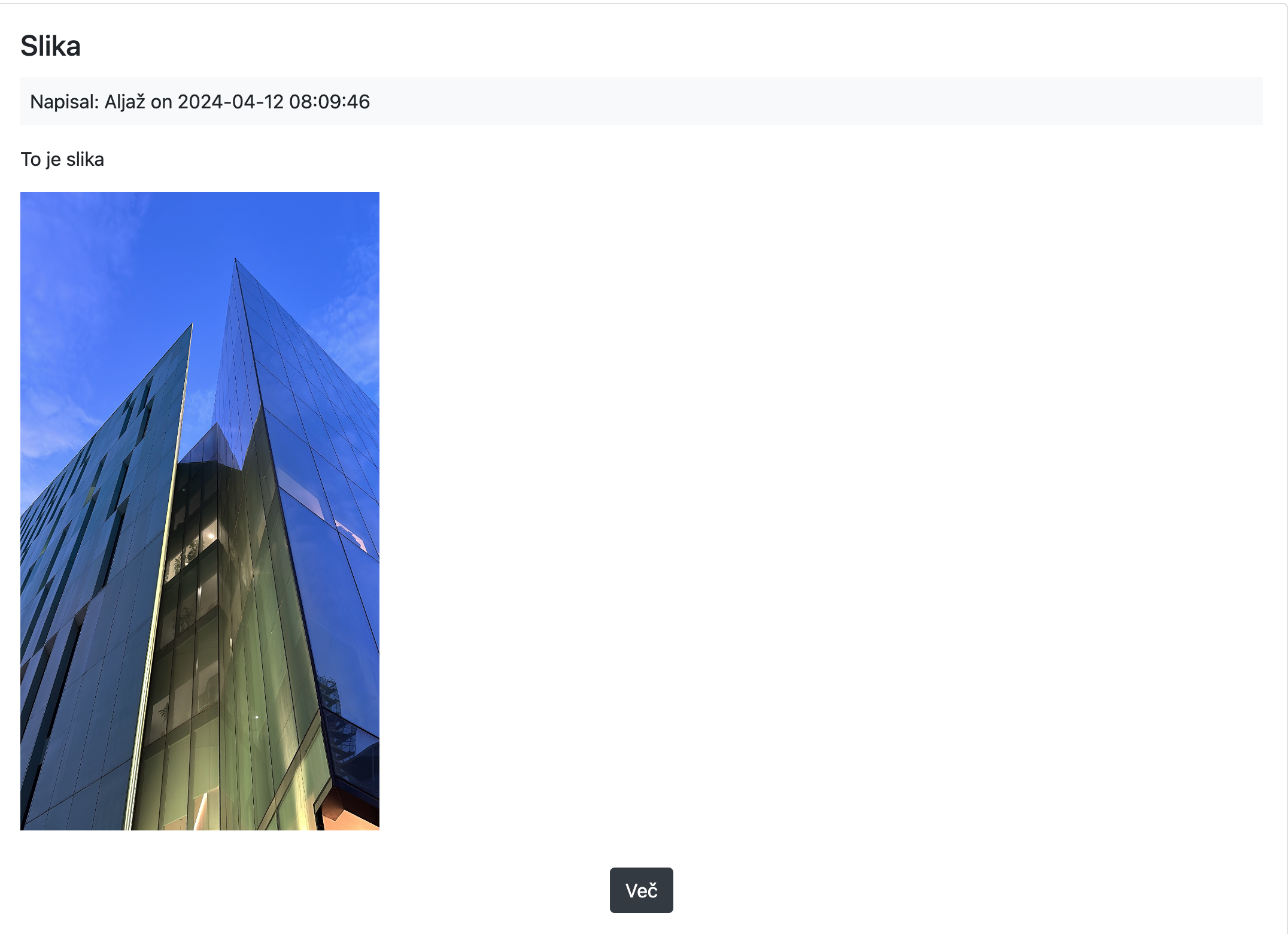
Slika : Ponastavitev gesla

# ADMINISTRATIVNI UPORABNIK

Administrativni uporabnik je bil dodan tako, da v phpmyadmin smo dodelili vrednost »admin« v vrstico »role«. Ta uporabnik lahko briše in ureja objave tudi drugih uporabnikov. Poleg dodeljevanja vrednosti admin v stolpec v bazi podatkov lahko uporabimo tudi druge metode, kot so ustvarjanje nove tabele, uporaba zunanjih knjižic ali ogordij za uporabniške pravice. Pomembno je tudi zagotoviti ustrezne varnostne mehanizme kot pa so šifriranje, gesla, preverjanje prisotnosti, da se tako lahko prepreči nepooblaščen dostop do administrativnih funkcij.

# OBJAVE

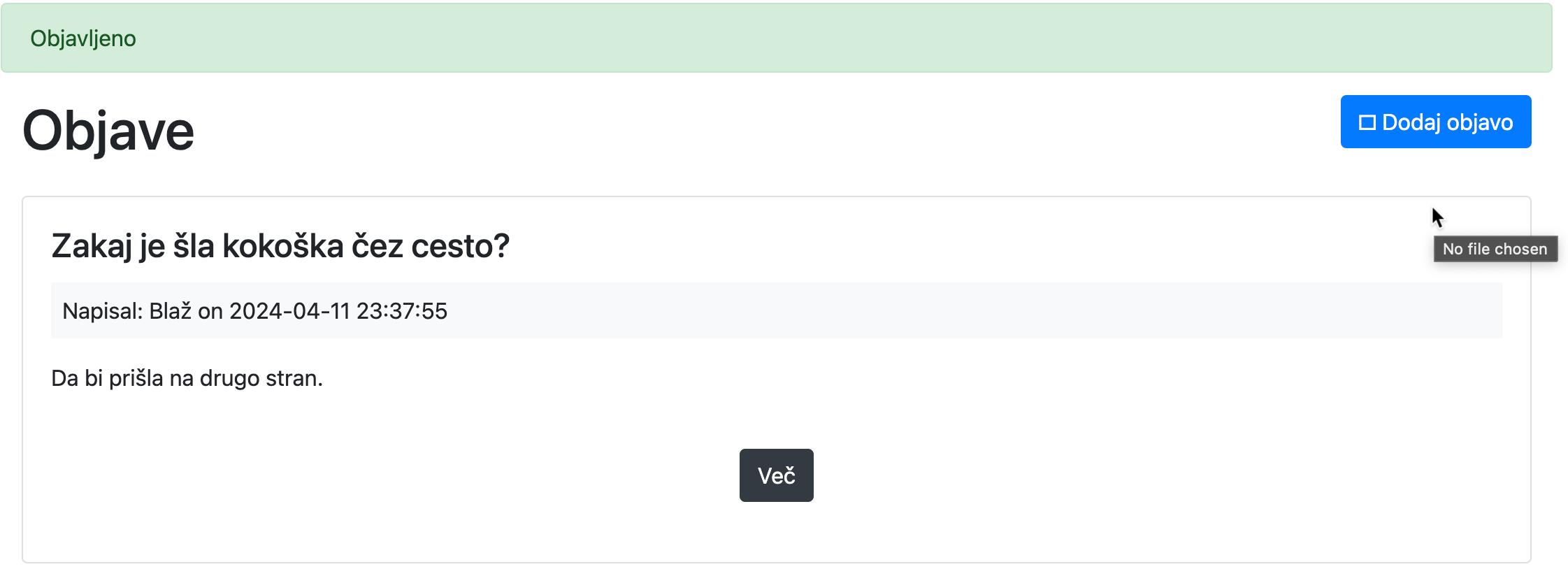
Stran objave prikazuje vse objave uporabnikov, ki obstajajo v bazi.Urejene so od najnovejše objave na vrhu do najstarejše na koncu strani. Vsaka objava vsebuje naslov, kdo je napisal objavo in kdaj, ter vsebino objave. To so obvezni elementi vsake objave. Neobvezna pa je slika poleg objave. To lahko pri objavi dodamo ali ne. Pod vsako objavo imamo še gumb »Več«, ki nam pokaže vsako objavo posebej v novi strani.



Slika : Objave

# DODAJANJE NOVE OBJAVE

Ko dodamo novo objavo se nam pokaže na vrhu strani Objave pokaže pojavno okno, ki nam tako potrdi da res ni prišlo do napake pri dodajanju objave v podatkovno bazo.



Slika : Dodajanje nove objave

# PRIKAZ OBJAVE

Ko kliknemo gumb »Več« pod objavo na domači strani, se nam odpre nova povezava. Nova stran vsebuje samo posamezno objavo. Če uporabnik odpre svojo objavo, jo lahko zbriše ali uredi, če pa odpre objavo ki ni njegova, ima pravico samo da pregledovanja. Pravice za brisanje in urejanje objave ima tudi administrator. Če stisnemo tipko nazaj, nas preusmeri nazaj na domačo stran. Na isto stran nas preusmeri, tudi če kliknemo gumb »Zbriši«, le da tukaj še odstrani objavo iz glavne strani in jo tudi zbriše ven iz podatkovne baze. Ko se to zgodi dobimo še eno pojano okno na domači strani, ki nam zagotovi uspeh postopka.

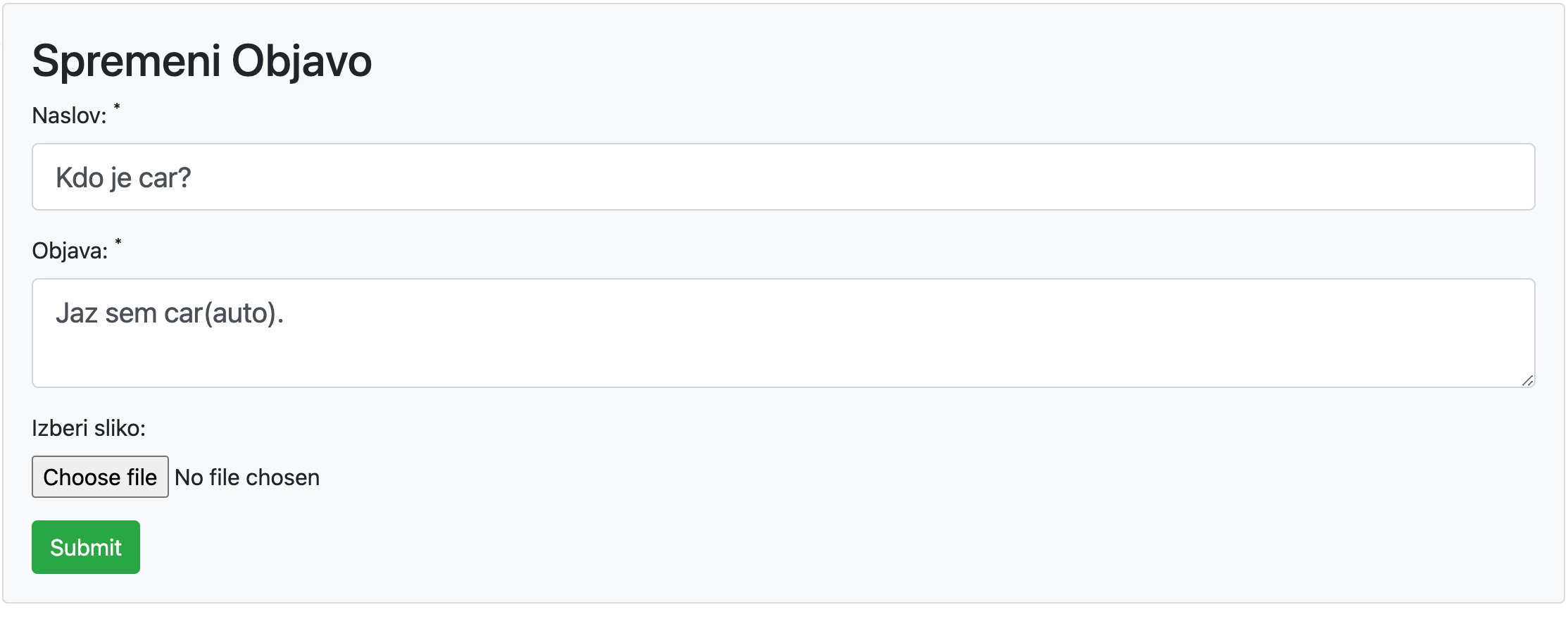




Slika : Prikaz objave

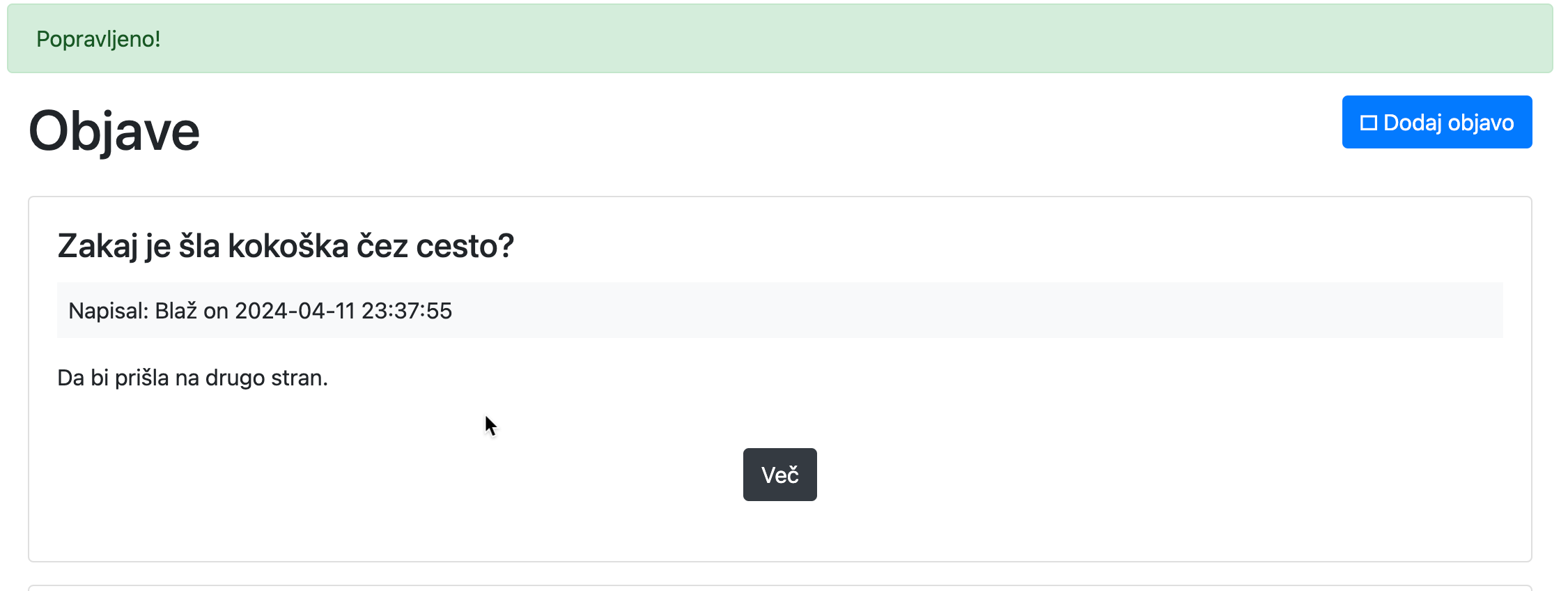
# UREJANJE OBJAV

Zadnja tipka #Uredi« nam ostane na strani, ki pokaže posamezno objavo, nas preusmeri na stran za urejanje objave. Tam lahko popravimo naslov, vsebino in sliko objave.



Slika : Sprememba objave

Ko popravimo objavo kot uporabnik ali kot administrator, nas preusmeri na stran z objavami in dobimo pojavno okno, da potrdi uspeh popravljanja objave.



Slika : Poprava objave

# ZAKLJUČEK

Pri izdelavi maturitetnega izdelka mi je uspelo realizirati zastavljene cilje in jih tudi nekoliko nadgraditi. Zavedam pa se da ima program še veliko potenciala za nadgradnjo. Skozi potek izdelave se je moje razumevanje za programske jezike PHP, SQL in HTML definitivno izboljšalo. Največja nadgradnja je bila pri jeziku PHP. Naučil sem se da ni težko programirati, vendar pa odpravljati napake, ki jih narediš ko nisi pozoren. Priprava enostavnega socialnega omrežja z centraliziranim dostopom se je izkazala za izjemno in koristno rešitev za podjetja in organizacije, katere potrebujejo učinkovito in varno komunikacijsko platformo. S tem projektom sem uspel zagotoviti enostaven dostop do ključnih informacij ter pa zmanjšanje stroškov, kar je lahko še posebej koristno za manjša podjetja z omejenimi proračuni. Med razvojem projekta sem se osredotočil na uporabnost in nizke stroške, kar se je odražalo v uporabi tehnologij, kot so MariaDB, PHP, Visual Studio Code in XAMPP. Uporaba MVC načrta je omogočila boljšo organizacijo in ločevanje funkcij aplikacije, kar je prispevalo k preglednosti in enostavnosti vzdrževanja. Poleg tega sem za zagotavljanje varnosti uporabil različne tehnike, kot so šifriranje gesel in preverjanje uporabniških pravic. Kljub uspehom projekta pa je treba omeniti tudi nekatere omejitve. Na primer, implementacija dodatnih funkcij, kot so zasebne objave ali napredno upravljanje s podatki, bi lahko zahtevala dodatno delo in čas. Prav tako je pomembno nenehno spremljati in posodabljati varnostne mehanizme, da se zagotovi zaščita pred morebitnimi varnostnimi ranljivostmi. V celoti lahko rečemo, da je izdelava socialnega omrežja z centraliziranim dostopom lahko učinkovito orodje za izboljšanje komunikacije in izmenjavo informacij znotraj podjetij. S pravilno načrtovano in izvedeno implementacijo lahko takšno omrežje postane ključno sredstvo za povečanje produktivnosti in uspešnosti poslovanja. Poleg tega je treba omeniti tudi pomembnost nadaljnjega razvoja in prilagajanja platforme glede na potrebe uporabnikov. Spreminjajoče se poslovno okolje zahteva prilagodljivost in inovativnost, zato je ključno, da se socialno omrežje nenehno posodablja in izboljšuje v skladu z novimi zahtevami in trendi. V prihodnosti bi lahko nadgradili platformo z dodatnimi funkcijami, kot so napredna analitika uporabniških podatkov za boljše razumevanje vedenja uporabnikov, možnost integracije s tretjimi storitvami za še večjo prilagodljivost, in razvoj mobilne aplikacije za enostavnejši dostop do omrežja tudi na mobilnih napravah.

Skupaj z uporabniki in skupnostjo razvijalcev bi lahko ustvarili platformo, ki bi postala nepogrešljivo orodje za poslovno komunikacijo in izmenjavo informacij v današnjem digitalnem svetu. Z močjo inovacij in nenehnega izboljševanja lahko socialno omrežje postane ključni dejavnik za uspeh in konkurenčnost podjetij v prihodnosti.

# VIRI IN LITERATURA:

<https://www.youtube.com/@traversymedia> 10.4.2024

<https://www.udemy.com/> 10.4.2024

[https://fontawesome.com/icons 10.4.2024](https://fontawesome.com/icons%2010.4.2024)

# IZJAVA O AVTORSTVU

Izjavljam, da je stokovno poročilo BookFace v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

12.4.2024

Aljaž Rojina