

Izrada My.ferit.hr Bloga primjenom Django razvojnog kostura

Lovin, Filip

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:846704>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-27**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Stručni prijediplomski studij Računarstvo

**IZRADA MY.FERIT.HR BLOGA PRIMJENOM DJANGO
RAZVOJNOG KOSTURA**

Završni rad

Filip Lovin

Osijek, 2024.

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	Filip Lovin
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	ARV 18, 24.09.2021.
JMBAG:	0165092545
Mentor:	doc. dr. sc. Hrvoje Leventić
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Krešimir Romić
Član Povjerenstva 1:	doc. dr. sc. Hrvoje Leventić
Član Povjerenstva 2:	Marin Benčević, univ. mag. ing. comp.
Naslov završnog rada:	Izrada My.ferit.hr Bloga primjenom Django razvojnog kostura
Znanstvena grana završnog rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada:	Izraditi blog platformu za više tipova korisnika koristeći Django. Istražiti i primijeniti razvojne okvire posebno prilagođene za blogove u Django okruženju, kao i u drugim programskim jezicima. Blog treba podržavati različite vrste sadržaja i korisničke profile. (Rezervirano: Filip Lovin, VK)
Datum ocjene pismenog dijela završnog rada od strane mentora:	23.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	9.10.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	09.10.2024.



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 09.10.2024.

Ime i prezime Pristupnika:

Filip Lovin

Studij:

Stručni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

ARV 18, 24.09.2021.

Turnitin podudaranje [%]:

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Izrada My.ferit.hr Bloga primjenom Django razvojnog kostura**

izrađen pod vodstvom mentora doc. dr. sc. Hrvoje Leventić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. BLOGOVI	2
2.1. WordPress.com	2
2.2. Wix	3
2.3. Medium	3
2.4. Blogger	5
3. IZRADA MY.FERIT.HR BLOGA PRIMJENOM DJANGO RAZVOJNOG KOSTURA	6
3.1. Python	6
3.2. Django	6
3.2.1. Sloj modela.....	7
3.2.2. Sloj pogleda.....	7
3.2.3. Sloj predložaka	7
3.2.4. Django forme.....	7
3.2.5. Django paginator.....	7
3.3. JavaScript i jQuery	8
3.4. TinyMCE	8
3.5. Tailwind	8
4. PRIMJENA TEHNOLOGIJA	9
4.1. Izgled stranice	9
4.2. Baza podataka i primjena sloja modela	11
4.2.1. Baza podataka.....	11
4.2.2. <i>Page</i> model.....	12
4.2.3. <i>Post</i> model.....	13
4.2.4. Model za oznake.....	14
4.2.5. Model za korisnike	14
4.3. Primjena sloja pogleda	15
4.3.1. Naslovnica	15
4.3.2. Registracija i prijava korisnika.....	15

4.3.3. Profil i postavke.....	15
4.3.4. Kreiranje novih stranica	16
4.3.5. Kreiranje, pregled i uređivanje objava.....	16
4.3.6. Pretraživanje objava i korisnika	17
4.4. Primjena sloja predloška	17
4.4.1. <i>Base</i> predložak.....	17
4.4.2. Predložak za navigaciju.....	17
4.4.3. Profil predložak	18
4.4.4. Predlošci za kreiranje, uređivanje i prikazivanje objava	18
4.4.5. Predlošci za registriranje i prijavu korisnika	20
4.4.6. Predložak za pretraživanje objava i korisnika	21
5. ZAKLJUČAK.....	23
LITERATURA	24
SAŽETAK.....	26
ABSTRACT	27

1. UVOD

Blogovi su web stranice koje se sastoje od raznih objava formatiranih prema želji autora. Blogovi mogu imati jednog ili više autora, a tema bloga može obuhvaćati bilo koji sadržaj o kojem autor želi pisati. Postoje brojne stranice koje poslužuju blogove, dok su u posljednje vrijeme više popularne stranice koje nisu centralno fokusirane na same blogove, već i na druge elemente kao što su rasprave ili kraće objave koje opisuju manje ozbiljne teme. Izrada ovog rada se fokusira na razvoj poslužiteljske stranice za blogove koja je orijentirana studentima za raspravu tema te koji se mogu dijeliti sa ostalim studentima ili spremati privatno za sebe. U drugom poglavlju dan je uvid u stranice koje se odnose na blogove. U detalje je prikazano kako su te stranice organizirane i kakve značajke imaju te su uspoređene sa stranicom u radu. U trećem poglavlju su opisane tehnologije koje koristimo u izradi stranice te je u detalje razrađeno kako te tehnologije pridonose samoj stranici. U četvrtom poglavlju je prikazano kako su te tehnologije iskorištene za izradu stranice, specifično kako su upotrijebljene tehnologije da bi se implementirale značajke koje su potrebne za stranicu u radu. U detalje su opisane te značajke i organizacija same stranice. Također su prikazani rezultati implementiranja ovih značajki te izgled same stranice koji je postignut primjenom ovih tehnologija.

1.1. Zadatak završnog rada

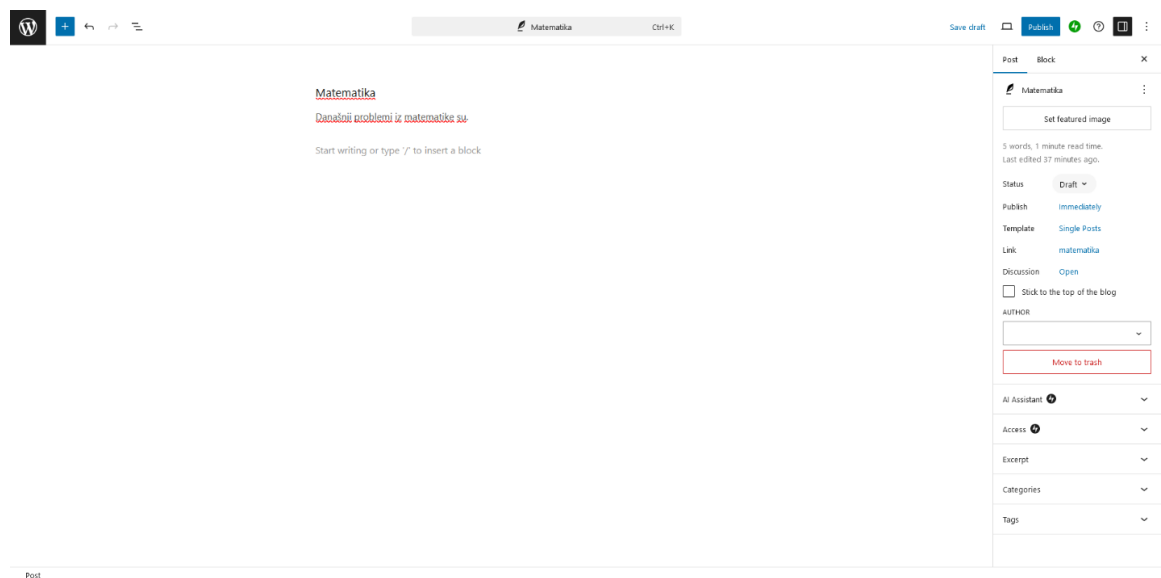
Zadatak završnog rada je razvoj web stranice za FERIT na kojoj će studenti moći objavljivati blogove o temama po svom izboru koristeći Django razvojni kostur. Osim objavljivanja, studenti bi trebali imati veliki broj mogućnosti za manipulaciju blog stranicama i svojih objava.

2. BLOGOVI

Postoji puno stranica koje služe za blogove. Dok većina imaju slične značajke svaka ima svoju specijalnost. Našoj stranici potrebne su određene značajke koje su usmjerene prema studentima i studentskoj zajednici. Stoga su pregledane neke od trenutno popularnih blog stranica, uključujući njihove značajke i njihov izgled. Ove stranice su opisane u ovom dijelu, zajedno s njihovim značajkama, a zatim su uspoređene sa značajkama koje su potrebne za našu stranicu.

2.1. WordPress.com

WordPress.com [1] je hostirana verzija softvera otvorenog koda, Wordpress. Pri prijavi na WordPress.com pojavljuje se mogućnost odabira domene što može biti besplatno ili plaćeno ovisno o domeni koju korisnik želi. Nakon odabira domene može se započeti blog i dodavati stranice na blog sa raznim izborima formatiranja uključujući font, boju teksta i ostale opcije formatiranja teksta kao *bold*, *italic* i slično.

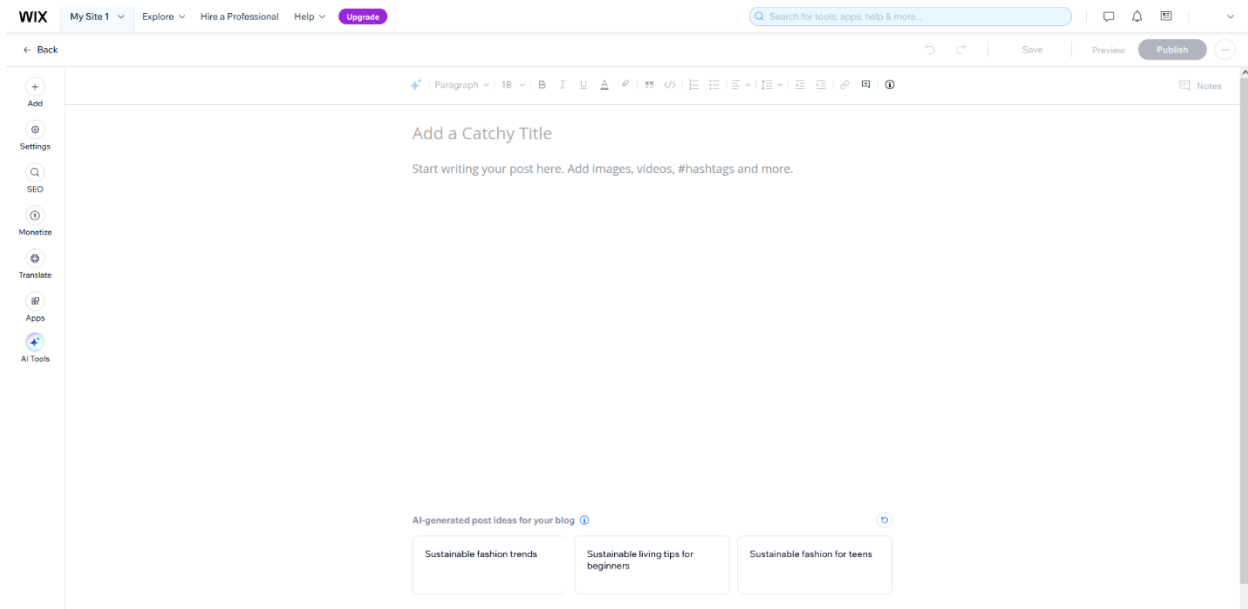


Sl. 2.1. Sučelje stranice WordPress.com

Naravno, WordPress.com je napravljen s brojnim planovima koji omogućuju razne opcije za kreiranje, izgradnju i održavanje stranica koje se ne odnose samo na blogove. Ipak, WordPress.com ostaje kao jedna od najpopularnijih opcija za pisanje blogova i ima puno značajki za pisanje i uređivanje blogova koje dijeli sa stranicom u ovom radu.

2.2. Wix

Wix [2] je poslužiteljska stranica koja omogućuje postavljanje raznih vrsta web stranica, uključujući i blogove. Pri odabiru opcije za kreiranje blogova korisnicima se nudi izbor domene. Nakon odabira domene prikazuje se sučelje za kreiranje samog bloga.



Sl. 2.2. Sučelje stranice Wix

Kao i WordPress.com, Wix nudi puno opcija za formatiranje teksta i izgleda stranice. Primjećuje se velika sloboda pri formatiranju te brojni savjeti i pomoć za korisnike. Ova stranica isto nije namijenjena za neku specifičnu svrhu, već za širi broj korisnika te se radi o jednoj blog stranici na koju se može dodati više objava. Za razliku od toga, stranica u ovom radu je orijentirana prema studentima i predstavlja centraliziranu verziju gdje su svi blogovi tog studenta na njegovom profilu i spremni za dijeljenje sa ostalim studentima.

2.3. Medium

Medium [3] je blog stranica koja je najsličnija našoj stranici, gdje je fokus na socijalni aspekt blogova. Pri registraciji i prijavi, korisnici se usmjeravaju na početnu stranicu koja nudi razne teme koje bi ih mogle zanimati. Omogućeno je praćenje autora, čime se na početnoj stranici korisnika prikazuju objave tih autora..

For you Following Technology Data Science Programming

Julian Frazier, PhD
Most People Don't Know the Difference Between "Feelings" and "Emotions"
 The single most cliché question in all of psychology is, "How does that make you feel". And yet I find that even my colleagues...
 Nov 22, 2022 2.5K 38

Annie Tanasugarn, PhD in Invisible Illness
The Effects of Unprocessed Trauma on Your Emotional Maturity
 3 subtle signs of emotional immaturity in your life.
 2d ago 903 16

Because you follow Data Science

Sumita Gollapudi
Advanced Tokenization in LLMs: Usage of Advanced Graph Techniques
 Recently I was trying my hands on some rudimentary tokenization techniques on my LLM and in specific tried adaptiv...
 Apr 20 156 3

Staff Picks

Tony Stubblebine in The Medium Blog
Be part of a better internet

Evan Sharp
The Soul of Pinterest

Mary Rose
A Third of My Online College Students are AI-Powered Spambots. Now what?
 See the full list

Recommended topics

Self Improvement
 Machine Learning Writing
 Relationships Cryptocurrency
 Politics Productivity

See more topics

Who to follow

Kevin Beaumont
 Everything here is my personal work and opinions. Follow

Vishal Rajput
 Top writer in AI | Author: Ultimate NN Programming... Follow

Minko Gechev

Sl. 2.3. Naslovnica stranice Medium

Nakon toga, nalazi se odvojeni dio stranice gdje se mogu pisati objave. Kao i kod drugih stranica postoje velike mogućnost formatiranja teksta, dodavanja slika i sličnih aktivnosti koje su potrebne za pisanje bloga.

Medium Draft in Flip Publish

Title

Rich text editor icons: bold, italic, link, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, quote, code, undo, redo, help.

Select text to change formatting, add headers, or create links.

Rich text editor toolbar with icons for bold, italic, link, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, quote, code, undo, redo, help.

Writing on Medium

Sl. 2.4. Sučelje za pisanje blogova stranice Medium

Sučelje je elegantno i vrlo jednostavno, što omogućuje korisnicima da se fokusiraju na pisanje bloga bez da budu preplavljeni s previše informacija i izbora koji im nisu potrebni. Ovo je slično sučelju naše stranice, koje je minimalno kako bi studentima bilo omogućeno da se prijave i

odmah počnu pisati svoje objave bez puno razmišljanja o stvarima koje nisu povezane s blogom, ali s uključivanjem značajki koje su potrebne studentima.

2.4. Blogger

Blogger [4] je starija i jednostavnija stranica kojoj je svrha primarno blogovi. Kada se korisnik prijavi, omogućeno je kreiranje bloga, pri čemu se bira ime bloga i domena tog bloga. Nakon toga, na taj blog se mogu dodavati različite objave. U svakom trenutku se može napraviti novi blog na kojeg se mogu dodavati različite objave. Ovo je vrlo slično našoj stranici, no na našoj stranici je primijenjen centralizirani stil blogova gdje je svaki blog dostupan na profilu studenta u obliku blog stranica koje sadržavaju objave povezane s tom stranicom. Kao i na ostalim blog stranicama koje smo opisali, korisnicima je na raspolaganju uređivač teksta s brojnim opcijama za formatiranje i pisanje teksta. Ovo je zapravo standard za sve ove stranice, uključujući i našu.

3. IZRADA MY.FERIT.HR BLOGA PRIMJENOM DJANGO RAZVOJNOG KOSTURA

Cilj je bio razviti stranicu gdje su na naslovnici dostupni najnoviji javni blogovi od studenata, a korisnici klikom na njih mogu direktno otići na cjelokupni blog sa svim sadržajem. Blogove korisnici mogu vidjeti i na korisničkim profilima, gdje su dostupne sve javne blog stranice i njihove objave. Ako se student nalazi na svojoj profilnoj stranici, ima mogućnost dodavanja blog stranica. Nakon što se blog stranica napravi, ovisno o vrsti blog stranice može se dodati jedan ili više blogova. U pisanju blogova pomoću uređivača teksta TinyMCE dostupne su mnogobrojne mogućnosti za pisanje, formatiranje i obrađivanje teksta. Sa poslužiteljske strane korišten je Django razvojni kostur koji omogućuje jednostavan i širok spektar primjene. U daljnjem tekstu opisane su tehnologije korištene za ostvarenje navedenog.

3.1. Python

Jezik koji je korišten za poslužiteljski dio stranice je Python. Python je interaktivni [5], objektno-orijentirani programski jezik. Uključuje iznimke i dinamički *typing*, što znači da se tip varijable provjerava pri izvođenju, omogućujući promjenu tipa varijable tijekom izvršavanja programa. Osim toga, podržava dinamičke podatkovne tipove jako velike razine. Podržava više paradigmi programiranja osim samo objektno orijentiranog programiranja, kao što su proceduralno i funkcionalno programiranje. Python ima veliku moć sa jako jasnom sintaksom. Ima sučelja do puno sistemskim poziva i biblioteka. Python je prenosiv i može ga se pokrenuti na puno Unix varijanti uključujući Linux, macOS i Windows. Korišten je Python jer se za poslužiteljski dio stranice koristi Django razvojni kostur koji koristi Python i pomoću kojega su implementirane sve značajke koje su potrebne za stranicu sa poslužiteljske strane.

3.2. Django

Django je poslužiteljski web okvir koji je korišten na našoj stranici. Django je Python web razvojni kostur koji potiče brzi razvoj i čisti [6], pragmatički dizajn. Uključuje brojne alate za rukovanje čestim zadacima web razvoja kao npr. Autentikacija korisnika i administracija sadržaja. Veoma je siguran i štiti programera od čestih sigurnosnih greški. Django sadrži registar instaliranih aplikacija koji sadrži konfiguracije [7] i omogućava introspekciju. On također održava listu dostupnih modela. Ova web stranica je zapravo cjelina napravljena od više razvijenih Django aplikacija. Te aplikacije su iskorištene za implementaciju značajki koje nisu blisko povezane. Django je MTV (engl. *Model-Template-View*) okvir koji odvaja aplikaciju na tri dijela tj. Model,

predložak (engl. *Template*) i *pogled* (engl. *View*). Model definira strukturu podataka, pogled sadržava logiku koju će stranica koristiti te se povezuje s određenim URL-om. Dok predložak sadržava klijentsku stranu stranice, tj. U predlošku se oblikuje izgled stranice i svakakve značajke koje su potrebne s klijentske strane.

3.2.1. Sloj modela

Django pruža apstrakcijski sloj (modele) [8] za strukturiranje i manipulaciju podacima web aplikacije. Model je jedinstveni izvor informacija o našim podacima. Sastoji se od bitnih polja i ponašanja od podataka koji se spremaju. Općenito, svaki model je mapiran u jednu tablicu baze podataka.

3.2.2. Sloj pogleda

Django ima koncept pogleda za logiku koja je odgovorna za obradu korisničkih zahtjeva i za vraćanje odgovora [9]. Funkcija pogleda je Python funkcija koji prima web zahtjev i vraća web odgovor. Taj odgovor može biti HTML sadržaj stranice, preusmjerenje, slika ili bilo što drugo što je potrebno. Sam pogled sadrži logiku koja je potrebna za slanje tog odgovora.

3.2.3. Sloj predložaka

Sloj predložaka (engl. *template*) [10] koristi se za renderiranje informacija koje se žele prikazati korisniku. Django-u je potreban način da generira HTML dinamički. U tu svrhu koriste se predlošci koji sadrže statički dio HTML izlaza sa posebnom sintaksom koja opisuje kako se umeće dinamički sadržaj.

3.2.4. Django forme

U HTML-u, forma je kolekcija elemenata unutar oznaka za formu koja omogućuje posjetiteljima da obavljaju aktivnosti poput unosa teksta [11], odabira opcija, manipulacije ili kontrole objektima te omogućuje slanje tih informacija natrag serveru. Rukovanje formama je kompleksno, Django funkcionalnost formi mogu pojednostaviti i automatizirati većinu posla vezanog uz forme i to radi puno sigurnije nego što većina programera može kod pisanja vlastitog koda.

3.2.5. Django paginator

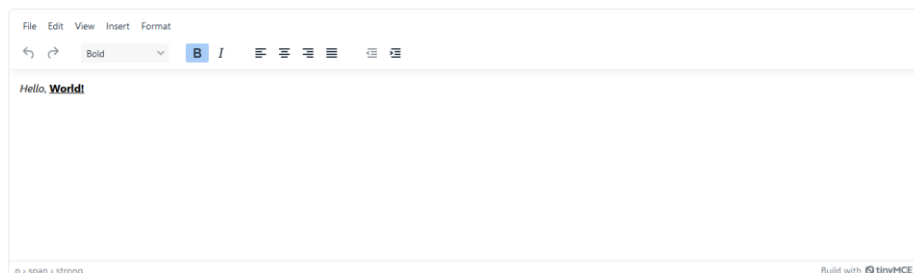
Django pruža par klasa koje pomažu u upravljanju paginiranim podacima. Paginirani podaci su podaci koji su razdvojeni na nekoliko stranica [12], pri čemu su sve stranice povezane u oba smjera. Ova funkcionalnost se specifično koristi za pretraživanje objava i korisnika kako bi se olakšao teret na serveru kada je riječ o velikim broju rezultata pri pretraživanju.

3.3. JavaScript i jQuery

JavaScript je lagani (engl. *lightweight*) interpretirani programski jezik koji je najpoznatiji kao skriptni jezik za web stranice [13], on je jedno-nitni dinamički jezik s više paradigmi koji podržava objektno orijentirano i funkcionalno programiranje. JavaScript biblioteka jQuery koristi se zbog svoje brzine [14], laganosti i mnoštvo značajki. Ona olakšava prolazak i manipulaciju HTML dokumentima, rukovanje događajima, te korištenje Ajax-a sa API-em koji je jednostavan za korištenje i koji radi na više preglednika. Alpine je minimalni alat za pisanje značajki direktno u HTML-u [15]. Koristi se kada je potreban jednostavan JavaScript tako da se ne mora raspisivati JavaScript kao inače nego da se unutar HTML elemenata može koristiti jedan od njegovih 15 atributa ili 6 svojstava.

3.4. TinyMCE

TinyMCE je uređivač teksta (engl. *rich text editor*) odabran za pisanje i formatiranje bloga. TinyMCE pruža sučelje za uređivanje teksta koje nudi brojne opcije za formatiranje i oblikovanje teksta na način koji je potreban. Izlaz koji se generira pomoću TinyMCE-a je HTML5 i može uključivati liste [16], tablice i slične elemente. Postoji opcija dodavanja dodataka koji omogućuju poboljšanje značajki uređivača teksta. Također je moguće prilagoditi izgled sučelja kako bi bolje odgovarao aplikaciji.



Sl. 3.1. TinyMCE sučelje za uređivanje teksta

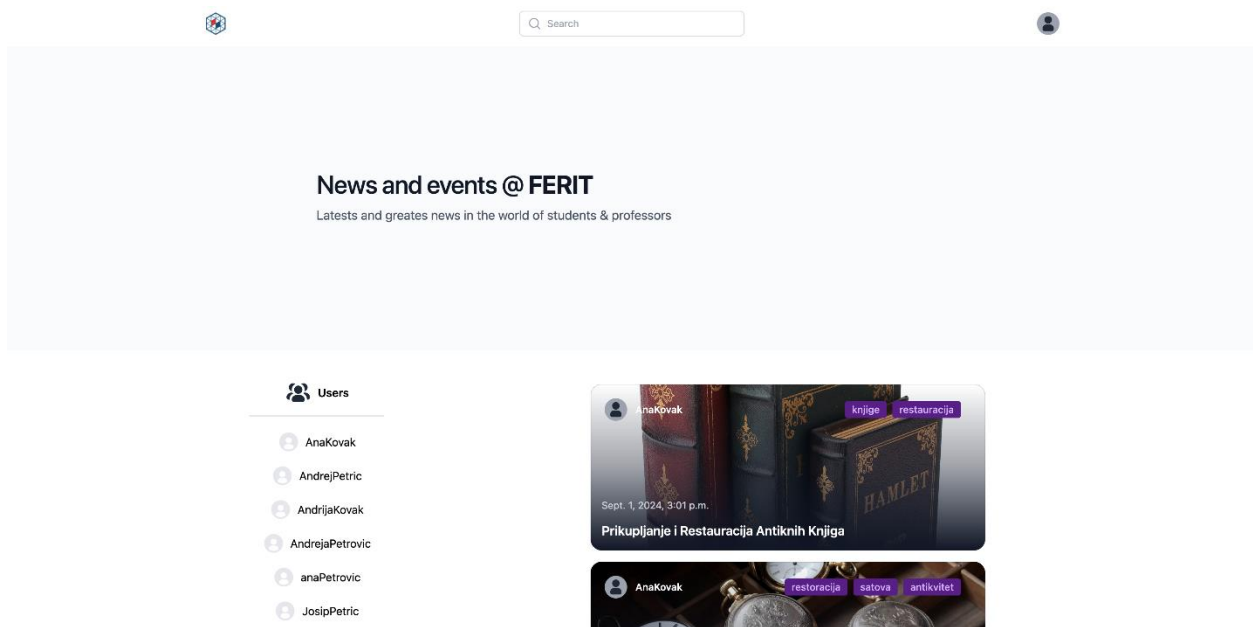
3.5. Tailwind

Tailwind je CSS razvojni kostur koji omogućava lagano i učinkovito pisanje CSS koda [17] za razvoj izgleda stranice. Funkcionira na način da skenira HTML datoteke i pomoću imena HTML klasa generira stilove koje zapisuje u statičku CSS datoteku. Ovo omogućuje vrlo brz razvoj izgleda stranice koristeći klase u HTML datoteci.

4. PRIMJENA TEHNOLOGIJA

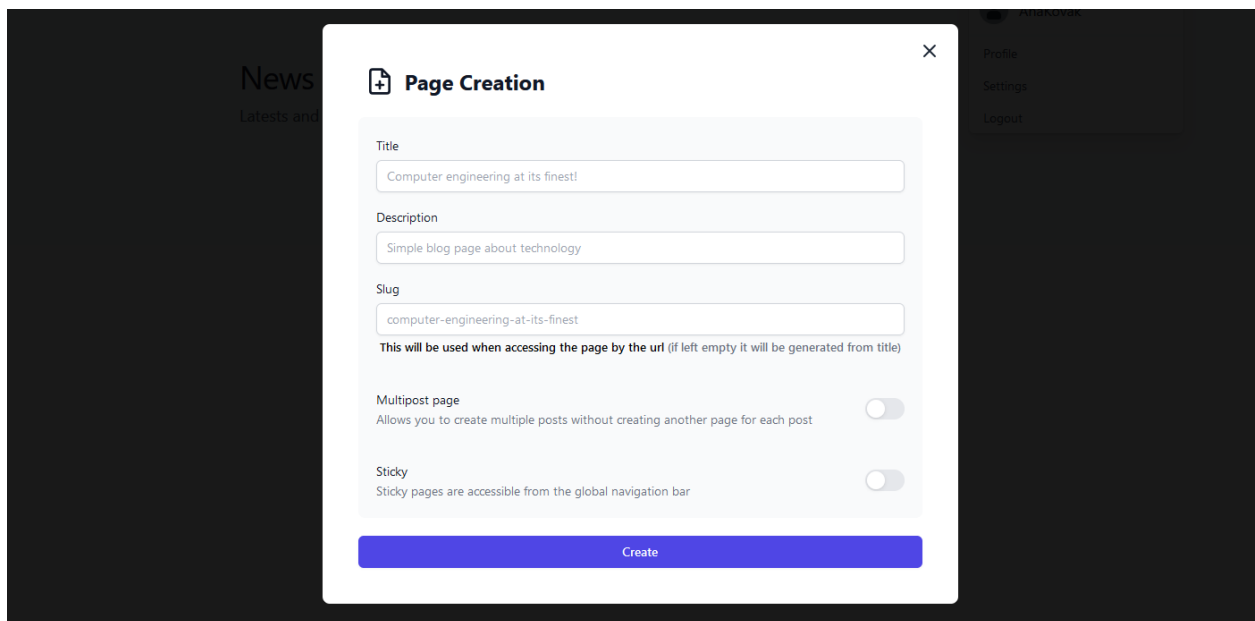
Kao što je navedeno, za poslužiteljski dio stranice koristi se Django razvojni kostur kod kojeg sloj modela omogućava strukturiranje i manipulaciju podacima. Nakon što su podaci manipulirani sa poslužiteljske strane, koriste se za prikazivanje potrebnih informacija korisniku sa klijentske strane. Ovu funkciju omogućava Django sloj predložaka koji dopušta korištenje klijentskih skriptnih tehnologija za prilagodbu izgleda stranice i ostvarivanje drugih značajki potrebnih sa klijentske strane. Izgled stranice i važnije Django funkcionalnosti koje se koriste za ostvarivanje značajki su daljnje opisani.

4.1. Izgled stranice



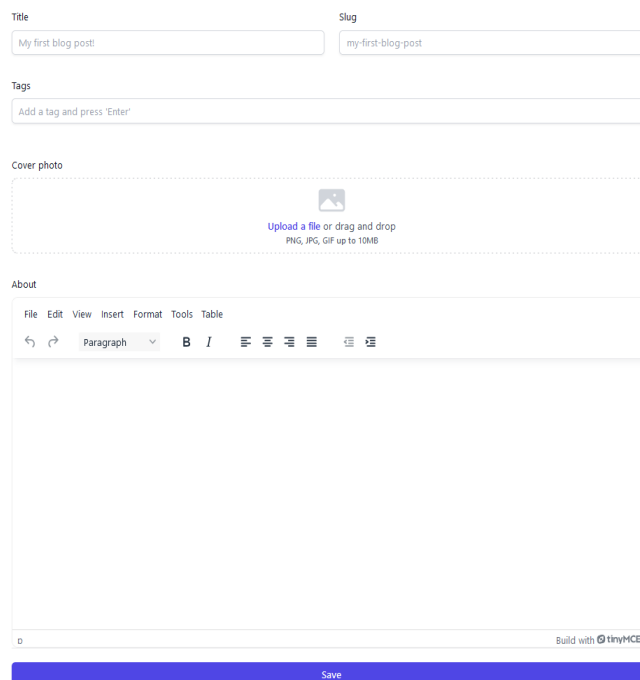
Sl. 4.1. Naslovnica My.ferit.hr stranice

Na naslovnici se prikazuju najnovije objave korisnika, slike koje predstavljaju objave te oznake koje opisuju sadržaj objava. Klikom na objave korisnik se preusmjerava na blog stranicu koja sadrži tu objavu i druge objave, ako se radi o blog stranici s opcijom više objava. Osim objava, prikazuju se i najnoviji korisnici koji su se registrirali. U navigacijskoj traci nalazi se opcija prijavljivanja ili registriranja pomoću Django autentikacijskog modela. Također postoji mogućnost pretraživanja objava i korisnika koji postoje na stranici.



Sl. 4.2. Modal za kreiranje blog stranice

Modal za kreiranje stranica pojavljuje se nakon što korisnik odabere izradu nove blog stranice dok je prijavljen. U modalnom prozoru pruža se mogućnost postavljanja naslova, opisa stranice i slug-a. *Slug* je jedinstveni identifikator koji je zapravo tekst na kraju poveznice blog stranice. Korisnik pomoću njega može odabrati dodatak na poveznicu koja vodi na njegov blog. Naravno ima i opciju odabira „*multipost*“ stranice i odlučivanja hoće li stranica biti javna.

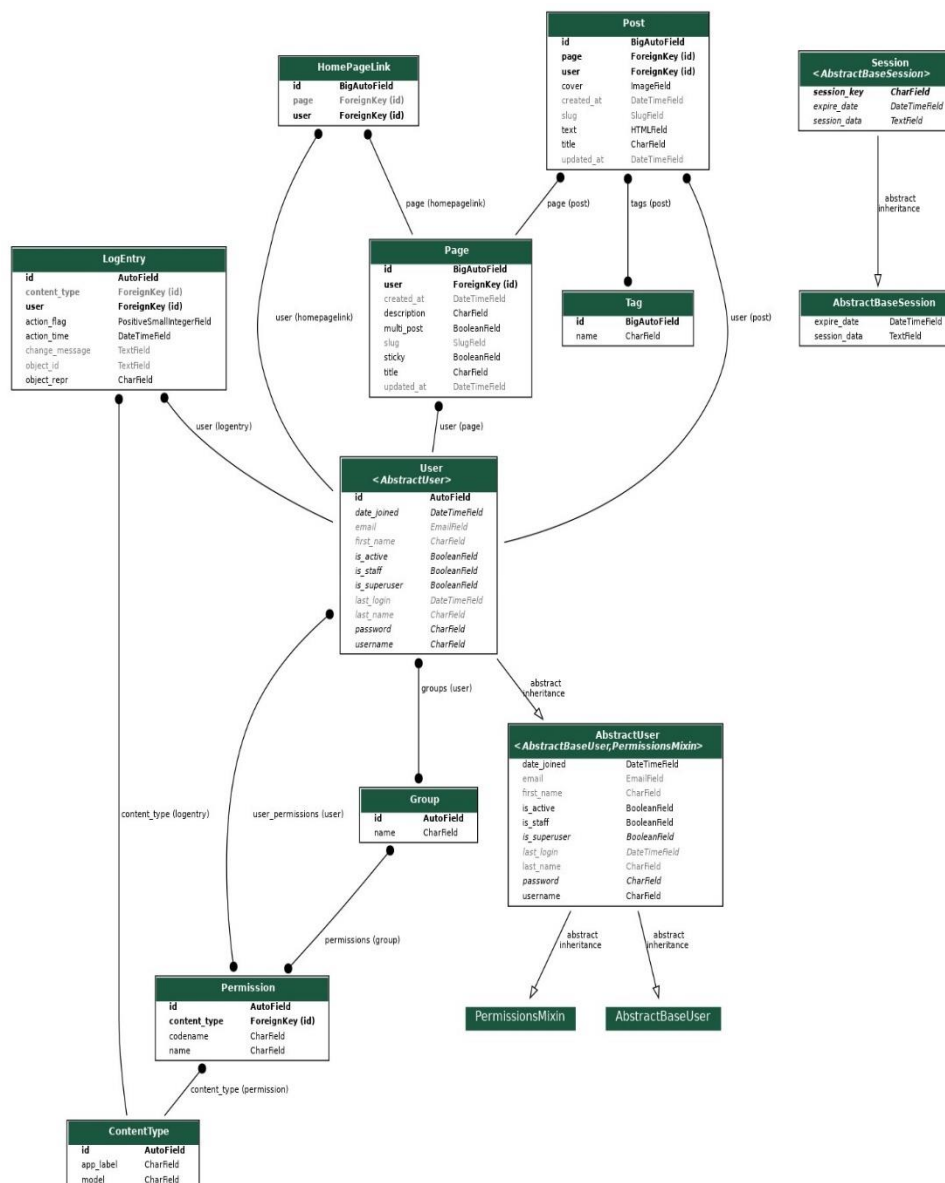


Sl. 4.3 Sučelje za kreiranje objave

Nakon kreiranja stranice postoji mogućnost kreiranja objave na stranici. Može se odabrati naslov, *slug* te naravno naslovna slika koja će predstavljati objavu. Pomoću TinyMCE je omogućeno formatiranje teksta i slika koje će se koristiti u objavi. Također se mogu dodavati oznake koje točno opisuju sadržaj objave. Pomoću tih oznaka moguće je pretraživanje objava i pružanje informacija o sadržaju objave. Oznake se mogu proizvoljno dodavati ili brisati u slučaju greške. Sve ove funkcionalnosti omogućene su primjenom slojeva koji će biti daljnje opisani.

4.2. Baza podataka i primjena sloja modela

4.2.1. Baza podataka



Sl. 4.4. Relacijski model baze podataka

Django koristi prethodno spomenuti sustav modela za predstavljanje podataka iz baze podataka. Model klasa predstavlja tablicu baze podataka [18], a instanca te klase predstavlja određeni zapis u tablici baze podataka. Prema slici 4.4. su vidljivi modeli stranice u relacijskom modelu baze podataka. Najvažniji prikazani modeli će daljnje biti objašnjeni, uključujući kako funkcioniraju i kako su korišteni.

4.2.2. Page model

Za uspostavljanje funkcionalnosti stranica na profilu koristi se Page model. Ovim modelom omogućuje se dodavanje stranica na profil, na koje se potom mogu dodavati objave, ovisno o odabiru jedne ili više objava. Iako je funkcionalnost samih objava realizirana na drugom modelu.

```
class Page(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    title = models.CharField(max_length=255)
    description = models.CharField(max_length=1024)
    slug = models.SlugField(max_length=255, unique=False, blank=True)

    multi_post = models.BooleanField(default=True)
    sticky = models.BooleanField(default=False)

    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def save(self, *args, **kwargs):
        if not self.slug:
            self.slug = slugify(self.title)

        super().save(*args, **kwargs)

    def __str__(self):
        return self.title

    @classmethod
    def get_sticky_pages(cls, user: User):
        return Page.objects.filter(user=user, sticky=True).order_by("created_at")
```

Sl. 4.5. Django model za blog stranicu i njegova polja

Na ovom modelu imamo 8 polja od kojih su važniji:

- *User* – polje povezuje korisnika pomoću stranog ključa u bazi podataka sa modelom što omogućava da svaki korisnik može napraviti svoju stranicu i da se ona može povezati sa korisničkim profilom.
- *Multi_post* – tip bloga koji se može napraviti, ovo je zapravo izbor između više objava ili samo jedne objave na toj jednoj stranici. Ako se izabere više objava onda se na stranicu može dodati koliko god objava je potrebno na profilu.
- *Title* – ovo polje je jednostavan naslov za stranicu koja će sadržavati blogove.
- *Slug* – polje koje sadrži *slug* stranice tj. Dio poveznice na kojoj će se stranica nalaziti.
- *Created_at* – datum kada je blog stranica napravljena

Osim toga, korištene su razne metode za postizanje funkcionalnosti pri radu s ovim modelom.

4.2.3. Post model

Za implementaciju blogova na stranici koristi se post model, kojim se omogućava dodavanje objava sa raznim značajkama i povezivanje objave sa stranicom kojoj pripada. Osim toga, sadržava relaciju koja omogućava dodavanje oznaka koje prikazuju o kakvom se sadržaju radi.

```
class Post(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    page = models.ForeignKey(Page, on_delete=models.CASCADE)

    title = models.CharField(max_length=255)
    slug = models.SlugField(max_length=255, unique=False, blank=True)

    cover = models.ImageField(upload_to="images/")
    text = HTMLField()

    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    tags = models.ManyToManyField(Tag, blank=True)

    def update_tags(self, tag_names: str):
        self.tags.clear()

        updated_tags = Tag.from_str(tag_names)
        if updated_tags == None:
            return

        for tag in updated_tags:
            self.tags.add(tag)

    @classmethod
    def get_page_posts(cls, page: Page):
        return Post.objects.filter(page=page).order_by("created_at")

    def save(self, *args, **kwargs):
        if not self.slug:
            self.slug = slugify(self.title)

        super().save(*args, **kwargs)

    def __str__(self):
        return self.title
```

Sl. 4.6. Django model za blog i njegova polja

U ovom modelu postoji 9 polja od kojih su važniji:

- *User* - ovo polje povezuje korisnike iz baze podataka sa objavom, to omogućava da svaki korisnik može kreirati svoju objavu i da dalje može uređivati i provoditi ostale akcije
- *Page* - polje koje povezuje objavu sa blog stranicom kojoj pripada
- *Title* - naslov objave definiran je kao polje znakova (engl. *CharField*) duljine 255, što omogućuje da naslov stranice bude dug do 255 znakova.
- *Slug* - kao i kod modela stranice ovo polje sadrži *slug* stranice tj. Dio poveznice na kojoj će se stranica nalaziti.
- *Cover* - polje koje sadrži sliku bloga. Sliku bloga autor odabire proizvoljno.

- *Text* - ovo polje sadrži cijeli sadržaj bloga uključujući formatirani tekst, poveznice na slike i slično.
- *Created_at* - polje datuma koje sadrži datum kada je blog napravljen.
- *Tags* – polje koje sadrži poveznice na više oznaka koje mogu pripadati objavi, ovo polje referencira model za oznake da bi uspostavilo poveznicu između njih

4.2.4. Model za oznake

Model za oznake dopušta autoru mogućnost dodavanja oznaka koje ukratko opisuju sadržaj objave. Ove oznake se nalaze na samoj objavi te se mogu dodavati i brisati po želji korisnika.

```
class Tag(models.Model):
    name = models.CharField(max_length=40)

    @classmethod
    def from_str(cls, string: str) -> list[Self]:
        if len(string) == 0:
            return None

        tag_names = string.split(",")

        tags: list[Self] = []
        for tag_name in tag_names:
            tag, _ = cls.objects.get_or_create(name=tag_name)
            tags.append(tag)

        return tags

    def __str__(self):
        return self.name
```

Sl. 4.7. Django model objave i njegova polja

U ovom modelu nalazi se jedno polje i to je njegovo ime. Međutim sadrži važnu metodu koja se koristi za razdvajanje oznaka koje su primljene od korisnika i njihovo dodavanje na objavu. Ovo se postiže putem metoda koje se nalaze na modelu za objave na način da se pozivaju na metodu iz ovog modela. Time je omogućena veća fleksibilnost pri dodavanju novih značajki na stranicu koje također koriste sustav oznaka, budući da se može koristiti već definirana metoda u ovom modelu.

4.2.5. Model za korisnike

Za autentikaciju korisnika koristi se Django model za korisnike, koji pruža sve što je potrebno za svrhe bloga. Korisnički objekt je najvažniji dio autentikacijskog sustava [19], taj objekt većinom predstavlja ljude koji se služe sa stranicom, i koristi se za ograničenje pristupa, registraciju korisničkih profila i za povezivanje korisnika sa nekim sadržajem. Samo jedna korisnička klasa postoji u Django autentikacijskom okviru dok su stvari kao administrator ili *superuser* samo korisnički objekti sa posebnim atributima.

Korisnički model ima 12 polja [20] od kojih je najvažnijih 6 za našu stranicu:

- *Username* – polje koje sadržava korisničko ime za registraciju i prijavu korisnika na stranicu, maksimalna duljina korisničkog imena je 150 znakova.
- *Password* – sadržava lozinku za registraciju i prijavu korisnika, ne sadržava identično lozinku nego *hash* lozinke.
- *Email* – sadržava email adresu korisnika.
- *First_name* – polje sadržava ime korisnika, u ovom slučaju to je ime studenta.
- *Last_Name* – polje sadržava prezime korisnika, u ovom slučaju to je prezime studenta.
- *Last_login* – pohranjuje datum kada se zadnji put korisnik prijavio na stranicu

Naravno osim ovih polja postoje i polja vezana za dopuštenja, grupe i administracijske ovlasti. Administracijske ovlasti dodjeljuje korisnicima vlasnik stranice što im daje pristup admin stranici gdje mogu kontrolirati razne stvari vezane uz modele koje smo naveli, kao što su dodavanje, brisanje, modifikacija i slično.

4.3. Primjena sloja pogleda

Sloj pogleda je zapravo taj koji pruža poslužiteljsku funkcionalnost većini stranica kao što su naslovnica, profil, pretraživanje korisnika. Taj sloj je odgovoran za logiku same stranice.

4.3.1. Naslovnica

Na naslovnici su prikazani najnoviji javni blog postovi od svih korisnika. Klikom na te blogove može se pročitati cijela verzija na stranici kojoj taj blog pripada. Ovu funkcionalnost omogućava pogled tako što sortira sve javne blogove te ih prikazuje od najnovijeg prema najstarijem, a zatim uspostavlja poveznicu na stranicu na kojoj se blog nalazi..

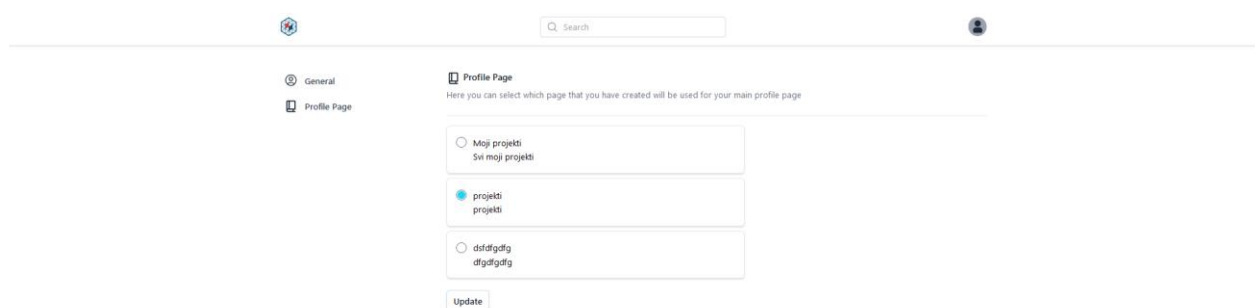
4.3.2. Registracija i prijava korisnika

Rukuje se s Django korisničkim modelom i s Django formama kako bi se procesirali podaci koje se dobivaju s klijentske strane pri registraciji, a potom se pomoću spomenutih Django funkcionalnosti taj korisnik registrira u bazu podataka. Pri prijavi se provjerava odgovaraju li korisničko ime i lozinka podacima u bazi podataka. Ako odgovaraju, korisnik se prijavljuje i preusmjerava na lokaciju gdje se omogućavaju potpune funkcionalnosti stranice..

4.3.3. Profil i postavke

Osim toga postoji pogled za korisnički profil koji ovisno o čijem je profilu riječ vraća profil predložak koji odgovara tom profilu. Ako korisnik nije postavio svoju glavnu blog stranicu onda

će pogled vratiti profil sa porukom da stranica nije postavljena. Vlasniku profila omogućava se postavljanje glavne stranice na koju želi usmjeriti korisnike koji kliknu na njegov profil. Također, pogled za postavke pruža opciju promjene glavne stranice na koju se želi usmjeriti korisnike klikom na profil.



Sl. 4.8. Postavke za odabir glavne stranice profila

4.3.4. Kreiranje novih stranica

Prijavljeni korisnik iz navigacijske trake uvijek ima mogućnost kreiranja nove stranice. Klikom na svoju profilnu sliku otvara se izbornik koji nudi poveznice na profil, postavke i mogućnost odjave sa stranice. Također, dostupna je opcija za kreiranje nove stranice. Klikom na opciju za kreiranje nove stranice otvara se modal za kreiranje stranice gdje se, prema modelu, postavlja ime stranici, odabire tip stranice i ostale potrebne postavke. Ovisno o tipu stranice korisnik može dodati jednu ili više objava po stranici. Ovo je sve omogućeno pomoću pogleda za kreiranje stranica koji, nakon što se ispuni Django *form* s odgovarajućim informacijama, obrađuje podatke i zapisuje ih u bazu podataka gdje se stranica kreira i dodjeljuje korisniku.

4.3.5. Kreiranje, pregled i uređivanje objava

Ove tri funkcije su omogućene sa odgovarajućim pogledima. Za kreiranje objava koriste se odgovarajuće varijable i Django metode da bi bili primljeni podaci sa korisničke strane. Putem tih metoda i Django formi, podaci se upisuju u bazu podataka i dodaje se nova objava koja pripada specifičnom korisniku i stranici. Pogled za prikaz vraća predložak s blog stranicama i pripadnim objavama tog korisnika na jednom mjestu. Ako se radi o korisniku koji pregledava vlastiti profil, bit će mu dostupne sve njegove stranice i blogovi. Također će imati mogućnost uređivanja, brisanja i dodavanja novih objava ako se radi o stranici s opcijom više objava. U slučaju da korisnik pristupa profilu koji mu ne pripada, bit će mu dostupne samo stranice i blogovi koji su javni. U izborniku za dodavanje objave korisnik dodjeljuje ime objavi i koristi uređivač teksta koji nudi brojne mogućnosti za formatiranje teksta i slika. Nakon što završi i doda objavu na blog stranicu,

i dalje ima mogućnost uređivanja te objave. Stranica koja se pojavljuje pri pristupu na nečiji profil je stranica koju je korisnik odabrao u svojim postavkama.

4.3.6. Pretraživanje objava i korisnika

Napravljena je odvojena Django aplikacija za pretraživanje objava i korisnika. Putem Django pogleda je implementirano da korisnik dobije relevantne rezultate kada putem klijentske strane pretraži frazu koja ga zanima. Ta fraza se pretražuje u bazi podataka putem Django SQL implementacije. Natrag se dobiva set podataka koji sadrži pretraženu frazu i koji je poredan prema zatraženom kriteriju. U ovom slučaju, podaci su poredani od najnovijeg do najstarijeg. Sve je to realizirano uz pomoć Django paginator, koji omogućuje pretragu objava i korisnika razdvojenih po stranicama. Ovaj način pretrage putem paginacije smanjuje opterećenje na serveru kada je u pitanju velika količina rezultata te poboljšava izgled same stranice.

4.4. Primjena sloja predloška

Sloj predloška je zapravo ono što daje stranici njen izgled pomoću HTML, CSS i JavaScripta. Django omogućuje kreiranje tih predložaka, tj. HTML datoteka za svaku stranicu koja je potrebna, što se zatim povezuje pomoću sloja pogleda. U tim HTML datotekama razvija se izgled stranice i sve značajke koje su potrebne sa klijentske strane, poput funkcionalnosti koje se aktiviraju klikom na dugme ili općenitog dizajna i značajki stranice. Dok je pozadina za te značajke zapravo logika napisana za poslužiteljski dio, ono što povezuje te dvije komponente je upravo sloj predloška i skriptna logika u tim datotekama.

4.4.1. Base predložak

Ovaj predložak je većinom uključen u svaku stranicu pomoću Django sustava predložaka. On sadržava sustav poruka koji se koristi za obavještanje korisnika o potrebnim informacijama. Osim toga, ovdje je inicijaliziran TinyMCE uređivač teksta.

4.4.2. Predložak za navigaciju

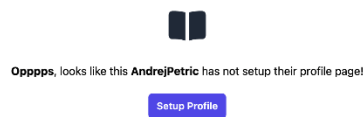
Koristi se predložak koji sadržava HTML, CSS i JavaScript za naslovnicu te daje izgled i određene funkcije naslovnoj stranici. Ovisno o tome radi li se o naslovnoj stranici, prikazuju se blogovi globalnih korisnika zajedno s navigacijskom trakom. Ako se radi o naslovnoj stranici, prikazani su blogovi sortirani po datumu te povezani s odgovarajućim blog objavama. U slučaju da se radi o drugoj stranici, prikazuje se samo navigacijska traka iz ovog predloška. Sve to je izvedeno u suradnji s pogledom koji stranici omogućuje pristup podacima o blogovima preko modela.



Sl. 4.9. Navigacijska traka stranice

4.4.3. Profil predložak

Datoteka koja sadržava HTML, CSS i JavaScript koji je zadužen za profilnu stranicu i daje izgled profilnim stranicama. Ovaj predložak se prikazuje kada korisnik nema namještenu glavnu blog stranicu koja bi se trebala prikazivati prilikom klika na njegov profil..



Sl. 4.10. Profil stranica kada nije namještena glavna blog stranica


4.4.4. Predlošci za kreiranje, uređivanje i prikazivanje objava

Tri odvojena predloška koji sadržavaju HTML, CSS i JavaScript zadužen za kreiranje, uređivanje i prikazivanje objava. Svaki od njih, putem Django pogleda, obavlja svoju funkciju. Kod kreiranja objave postoji mogućnost odabira naslova, slug-a i naslovne slike objave. Osim toga, mogu se dodati oznake na sam post i naravno pisati te formatirati sadržaj objave koja će biti objavljena.

Title Slug

Tags

Cover photo




Upload a file or drag and drop
PNG, JPG, GIF up to 10MB

About

File Edit View Insert Format Tools Table

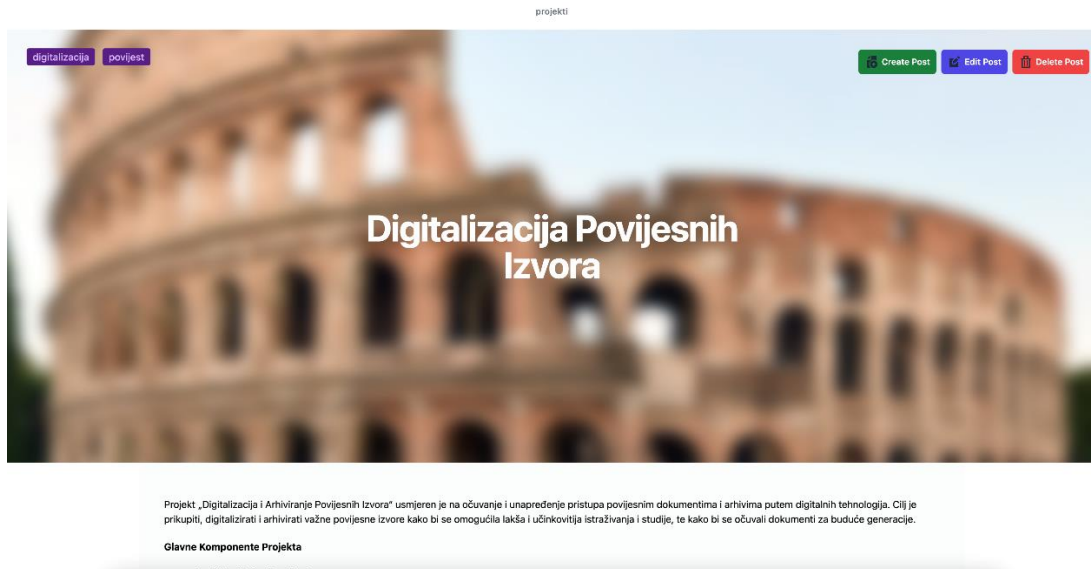
↶ ↷ Paragraph **B** *I* [List icons] [Link icon] [Unlink icon]

Build with  tinyMCE

Save

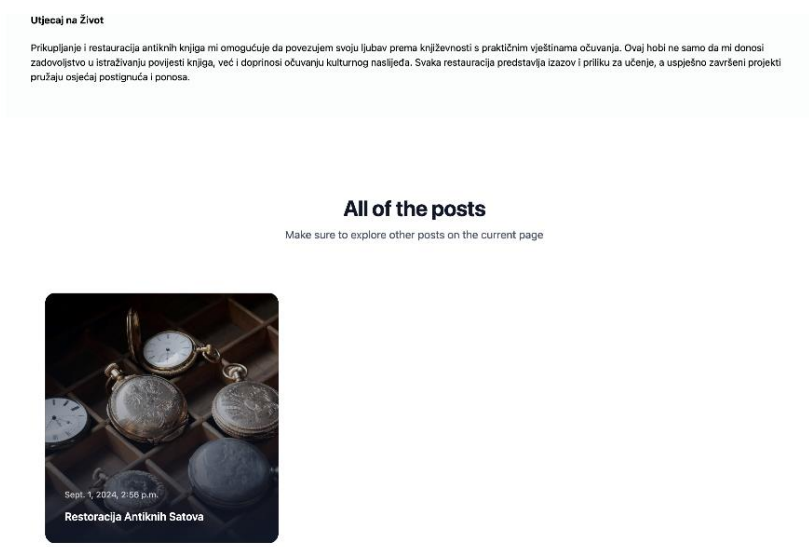
Sl. 4.11. Sučelje za kreiranje objava

Predložak za prikazivanje objava se sastoji od naslovne slike sa naslovom koji se nalazi na sredini te slike. U gornjem lijevom kutu slike prikazane su oznake koje je autor dodao objavi kako bi opisao njezin sadržaj. Klik na te oznake vodi na stranicu za pretraživanje, koja prikazuje sve objave koje odgovaraju toj oznaci. Osim toga, ako stranica pripada trenutnom korisniku, pružene su mu dodatne opcije za kreiranje nove objave, uređivanje ili brisanje postojeće. Ispod toga prikazan je sadržaj objave koju je autor napisao i formatirao. Naravno, postoji i predložak za uređivanje objava, koji izgleda gotovo isto kao predložak za prikazivanje, ali omogućuje uređivanje sadržaja objave te dodavanje ili brisanje oznaka koje pripadaju objavi.



Sl. 4.12. Blog objava iz perspektive autora

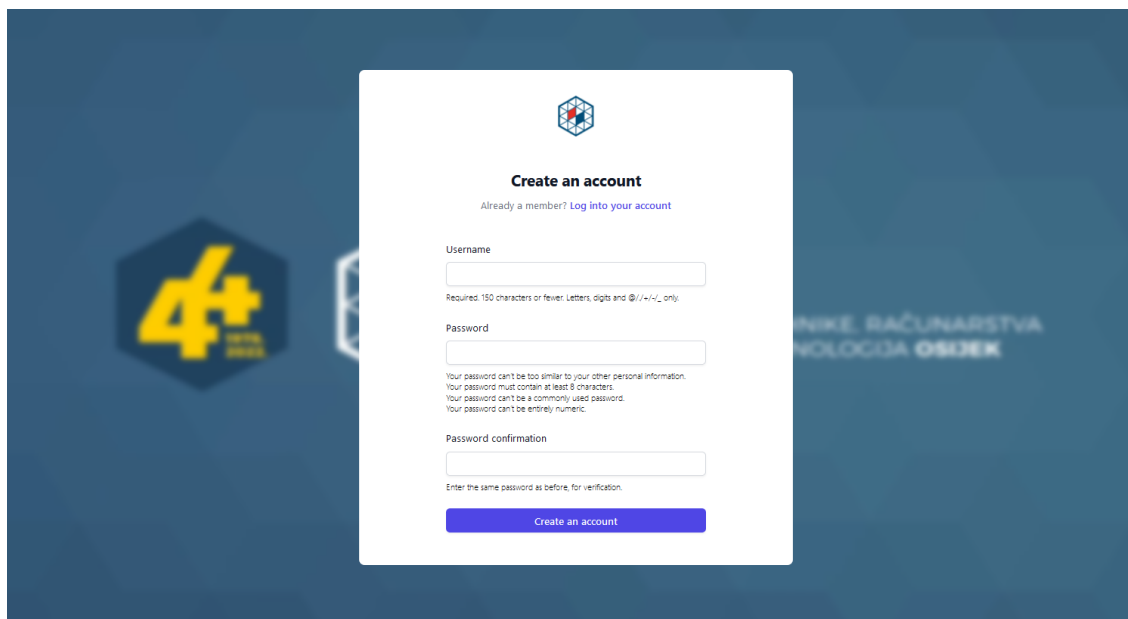
Ako je u pitanju blog stranica sa opcijom više objava po stranici onda se na dnu stranice nalazi lista svih ostalih objava.



Sl. 4.13. Ostale objave koje pripadaju blog stranici

4.4.5. Predlošci za registriranje i prijavu korisnika

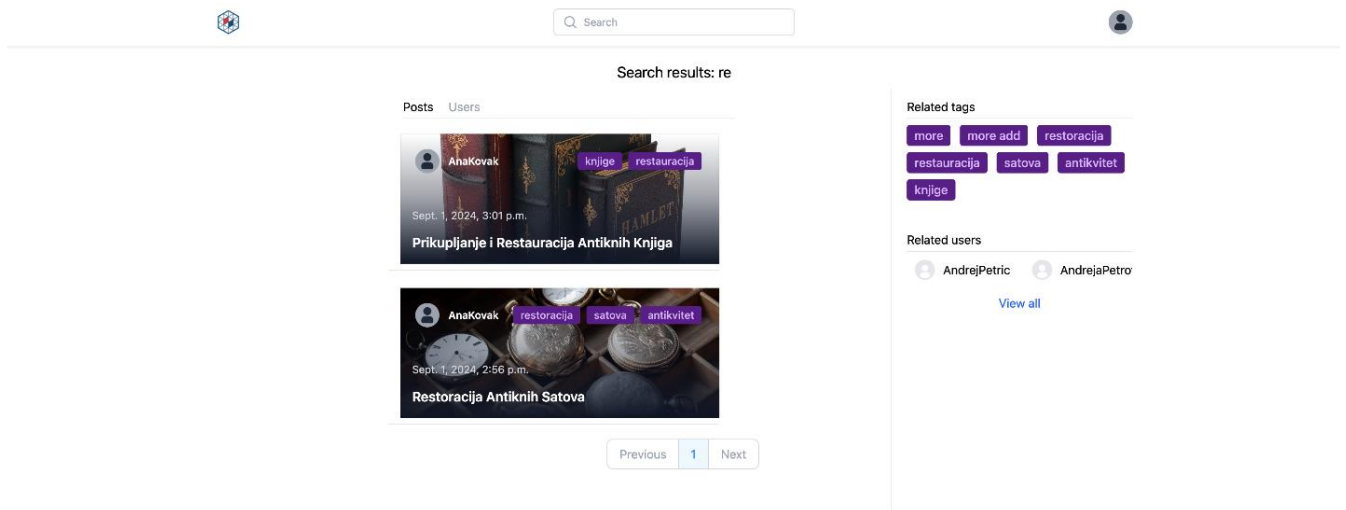
Django aplikacija za registraciju i prijavu korisnika se bavi tim pitanjem pomoću prije spomenutog Django korisničkog modela i odgovarajućih pogleda za registraciju i prijavu. Predlošci pružaju funkcionalnosti za ove procese s klijentske strane koristeći HTML, CSS i JavaScript. Prilikom registracije, korisnik unosi korisničko ime i lozinku koja će kasnije biti korištena za prijavu. Preko pogleda se korisnički račun registrira u bazu podataka, ako svi podaci odgovaraju i korisnik već ne postoji.



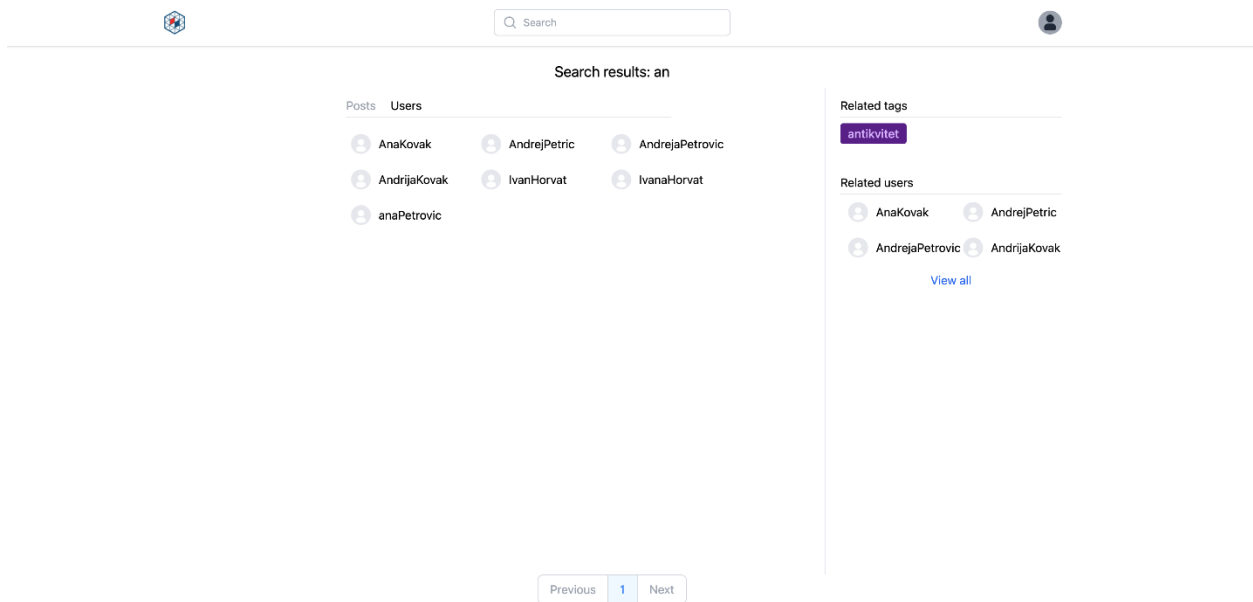
Sl. 4.14. Stranica za registraciju korisnika

4.4.6. Predložak za pretraživanje objava i korisnika

Ovo je klijentska strana prethodno opisanog pogleda gdje se koriste HTML, CSS i JavaScript za implementiranje pretraživanja objava i korisnika. Korisnik unosom fraze ili klikom na oznake može pretražiti traženi pojam. Pomoću informacija iz baze podataka i Django paginator objekta, stranično su prikazane sve objave i korisnici koji sadrže pretraženu frazu. Osim toga, sa strane je prikazan broj povezanih oznaka i korisnika. Kao rezultat pretrage, prvo su prikazane objave. Nakon toga, na sekciju za korisnike može se kliknuti kako bi se dobili rezultati za sve korisnike koji odgovaraju pretraženoj frazi. Ovisno o broju objava i korisnika, stranice sadrže određeni broj rezultata te se kroz njih može prolaziti pomoću izbornika na dnu stranice.



Sl. 4.15. Stranica za pretraživanje objava



Sl. 4.16. Stranica za pretraživanje korisnika

5. ZAKLJUČAK

Cilj ovog rada bio je razviti web aplikaciju za blogove usmjerenu studentima FERIT fakulteta. Pregledane su postojeće blog stranice te je detaljno proučen njihov izgled, funkcije i značajke. Značajke koje su odgovarale našoj stranici su usvojene, dok su razlike specifične za našu stranicu istaknute u usporedbi. Stranica je sa poslužiteljske strane napravljena pomoću Python razvojnog jezika, specifično koristeći Django web razvojni kostur, koji je omogućio jednostavan i učinkovit razvoj stranice. S klijentske strane korištena je kombinacija tehnologija kao što su JavaScript, Tailwind CSS i TinyMCE. Ove klijentske tehnologije su omogućile oblikovanje izgleda i funkcija naše stranice u točno ono što odgovara studentima za potrebe pisanja blogova. Stranica također ima socijalni aspekt, gdje naslovnica prikazuje sve javne blogove, od najnovijeg do najstarijeg. Na profilu korisnika su prikazane sve njegove javne blog stranice. Korisnik u svakom koraku ima potpunu kontrolu nad svojim blog stranicama te objave može dodavati, uređivati ili brisati. Stranici bi se moglo dodati još više socijalnih značajki, kao što su praćenje drugih korisnika da bi vidjeli njihove objave na našoj naslovnici ili dopuštanje određenim korisnicima da pregledavaju naše privatne objave. Osim toga izgled same stranice bi se uvijek mogao poboljšati da bi više odgovarao korisničkim potražnjama.

LITERATURA

- [1] WordPress.com, WordPress naslovna stranica, WordPress.com, 2024, dostupno na: <https://wordpress.com/> [18.06.2024.]
- [2] Wix, Wix naslovna stranica, Wix, 2024, dostupno na: <https://www.wix.com/> [18.06.2024.]
- [3] Medium, Medium naslovna stranica, Medium, 2024, dostupno na: <https://medium.com/> [18.06.2024.]
- [4] Blogger, Blogger naslovna stranica, Blogger, 2024, dostupno na: <https://www.blogger.com> [18.06.2024.]
- [5] Python, Python general FAQ, Python, 2024, dostupno na: <https://docs.python.org/3/faq/general.html> [10.05.2024.]
- [6] Django, Django naslovna stranica, Django, 2024, dostupno na: <https://www.djangoproject.com/> [22.01.2024.]
- [7] Django, Django documentation: Applications, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.1/ref/applications/> [13.08.2024.]
- [8] Django, Django documentation: Models, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.0/topics/db/models/> [15.06.2024.]
- [9] Django, Django documentation: Writing views, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.0/topics/http/views/> [15.06.2024.]
- [10] Django, Django documentation: Templates, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.0/topics/templates/> [15.06.2024.]
- [11] Django, Django documentation: Working with forms, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.1/topics/forms/> [13.08.2024.]
- [12] Django, Django documentation: Pagination, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.1/topics/pagination/>
- [13] MDN contributors, JavaScript, 2024, dostupno na: <https://developer.mozilla.org/en-US/docs/Web/JavaScript#reference> [10.06.2024.]
- [14] jQuery, jQuery naslovna stranica, jQuery, 2024, dostupno na: <https://jquery.com/> [20.06.2024.]

[15] C.P, Porzio, Alpinejs naslovna stranica, Alpinejs, 2024, dostupno na: <https://alpinejs.dev/> [08.08.2024]

[16] TinyMCE, TinyMCE documentation: TinyMCE 7 Documentation, TinyMCE, 2024, dostupno na: <https://www.tiny.cloud/docs/tinymce/latest/> [01.08.2024.]

[17] Tailwind Labs Inc., Get started with Tailwind CSS, Tailwind Labs Inc., 2024, dostupno na: <https://tailwindcss.com/docs/installation> [25.08.2024.]

[19] Django, Django documentation: Making queries, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.1/topics/db/queries/> [08.09.2024.]

[19] Django, Django documentation: Using the Django authentication system, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.0/topics/auth/default/> [18.06.2024.]

[20] Django, Django full API documentation, Django, 2024, dostupno na: <https://docs.djangoproject.com/en/5.0/ref/contrib/auth/#django.contrib.auth.models.User> [18.06.2024.]

SAŽETAK

U ovom završnom radu prikazana je izrada my.ferit.hr bloga za studente. Stranica omogućuje dodavanje blog stranica te brojne opcije za manipulaciju samim blogovima, što studentima omogućuje da na jednom mjestu imaju sve svoje blogove bez obzira na temu. U prvom dijelu rada su opisane stranice koje imaju slične značajke kao naša stranica i uspoređene su s uporabom i značajkama naše stranice. Dalje su opisane tehnologije koje su korištene za realizaciju stranice te je prikazano kako su te tehnologije primijenjene. Za realizaciju same stranice je korišten Django razvojni kostur i Python programski jezik za poslužiteljsku stranu, a s klijentske strane Tailwind CSS i JavaScript za izgled i značajke stranice. Korisnicima je omogućeno uređivanje svojih objava prema vlastitim željama, uz brojne opcije formatiranja pomoću TinyMCE uređivača teksta. Na kraju rada je prikazan izgled stranice, izvedba značajki te način primjene tehnologija. U zaključku su razmatrani način izvedbe stranice i mogućnosti za poboljšanje stranice.

Ključne riječi: Blog, Django, Python, studenti, web stranica.

ABSTRACT

This final paper demonstrates the development of the my.ferit.hr blog for students. The site allows us to add blog sites and then gives a large number of options concerning the manipulation and use of the blog pages. It also lets students have all of their blogs in one place regardless of the topic. In the first part of the paper, sites with similar features to our site are described and compared regarding their use and functionality. The technologies used for the implementation of the site are also detailed, demonstrating how these technologies were applied. For the realization of the site itself, the Django development framework and the Python programming language are used for the server side. On the client side Tailwind CSS and JavaScript are used for the appearance and features of the site. Users can edit their posts however they want with a wide variety of formatting options using the TinyMCE rich text editor. At the end of the paper, the appearance of the site is shown and the features of the site are discussed including the way we used certain technologies. In the conclusion, the methods of site implementation and potential improvements are discussed.

Keywords: Blog, Django, Python, students, website.