

Web aplikacija za vođenje bilješki na projektima

Kuterovac, Luka

Undergraduate thesis / Završni rad

2023

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:200:040175>

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

Download date / Datum preuzimanja: **2025-03-31**

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**

Sveučilišni studij

**WEB APLIKACIJA ZA VOĐENJE BILJEŠKI NA
PROJEKTIMA**

Završni rad

Luka Kuterovac

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 11.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Luka Kuterovac
Studij, smjer:	Programsko inženjerstvo
Mat. br. Pristupnika, godina upisa:	R4523, 27.07.2020.
OIB Pristupnika:	72988377521
Mentor:	izv. prof. dr. sc. Mirko Köhler
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Web aplikacija za vođenje bilješki na projektima
Znanstvena grana rada:	Obradba informacija (zn. polje računarstvo)
Zadatak završnog rad:	ZAUZETO ZA: Luka Kuterovac. Zadatak završnog rada je izrada web aplikacije koja ima mogućnost prijave korisnika, kreiranja projekata te kreiranje i uređivanje bilježaka za svaki zasebni projekt. Bilješke se sastoje od natuknica koje funkcioniraju poput zadataka koje korisnici mogu označiti izvršenima. Glavni korisnik ima mogućnost dodavati druge korisnike koji mogu surađivati na projektu. Korisnici koji nisu
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	11.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 25.09.2023.

Ime i prezime studenta:

Luka Kuterovac

Studij:

Programsko inženjerstvo

Mat. br. studenta, godina upisa:

R4523, 27.07.2020.

Turnitin podudaranje [%]:

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za vođenje bilješki na projektima**

izrađen pod vodstvom mentora izv. prof. dr. sc. Mirko Köhler

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 studenta:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. PREGLED PODRUČJA TEME	2
2.1. Notion	2
2.2. Jira	3
2.3. Monday.com	4
2.4. Trello	4
3. KORIŠTENE TEHNOLOGIJE I ALATI	6
3.1. Django	6
3.1.1. MVT arhitektura	6
3.1.2. Python	7
3.1.3. Pip	7
3.1.4. SQLite	8
3.2. Bootstrap	8
3.3. Visual Studio Code	8
3.4. GitHub	9
4. IZRADA APLIKACIJE	10
4.1. Dizajn aplikacije	10
4.2. Kreiranje virtualne okoline	12
4.3. Kreiranje aplikacije	13
4.4. Postavljanje projekta	15
4.5. Modeli	16
4.6. Pogledi	17
4.7. Predlošci	19
4.8. Signali	19
4.9. Forme	20
5. IZGLED I RAD APLIKACIJE	22

6. ZAKLJUČAK.....	27
LITERATURA	28
SAŽETAK.....	29
ABSTRACT	30
ŽIVOTOPIS.....	31
DODATAK	32

1. UVOD

U današnje vrijeme ljudi surađuju na projektima raznih veličina i područja. Kako bi projekt bio uspješan jako je važno dobro organizirati projekt i same zadatke, kao i voditi računa o tome što je obavljeno, a što nije. Za dobru organizaciju potrebno je puno znanja i iskustva. No kako bi sve to bilo uspješno, potrebno je na nekakav način taj projekt i strukturirati, bilo na papiru ili u nekakvom alatu. Bez dobre organizacije i strukture projekta može doći do raznih problema poput obavljanja već riješenog zadatka, zanemarivanja zadataka koji nisu zabilježeni, gubljenja sredstava ili vremena i mnogih drugih problema. Na internetu se mogu pronaći mnogi alati koji služe svrsi upravljanja i organiziranja projekata. Cilj ovog završnog rada je izraditi web aplikaciju koja korisnicima omogućava da na lak način kreiraju projekte, te da u njima zapisuju zadatke koji trebaju biti obavljani. S obzirom na to da najčešće na projektima sudjeluje više ljudi, web aplikacija omogućava i dodavanje drugih korisnika s kojima kreator projekta surađuje.

Za svrhu izrade web aplikacije za ovaj završni rad koristit će se Django. Django je besplatni web okvir (engl. *web framework*) otvorenog koda (engl. *open source*) koji omogućuje kreiranje korisničkog sučelja, razvoj serverskog dijela aplikacije i modeliranje baze podataka. Django razvija i održava Django Software Foundation. Aplikacija je pisana u programskom jeziku Python kojeg razvija i održava Python Software Foundation. Aplikacija je razvijena u uređivaču teksta Visual Studio Code kojeg je razvio Microsoft.

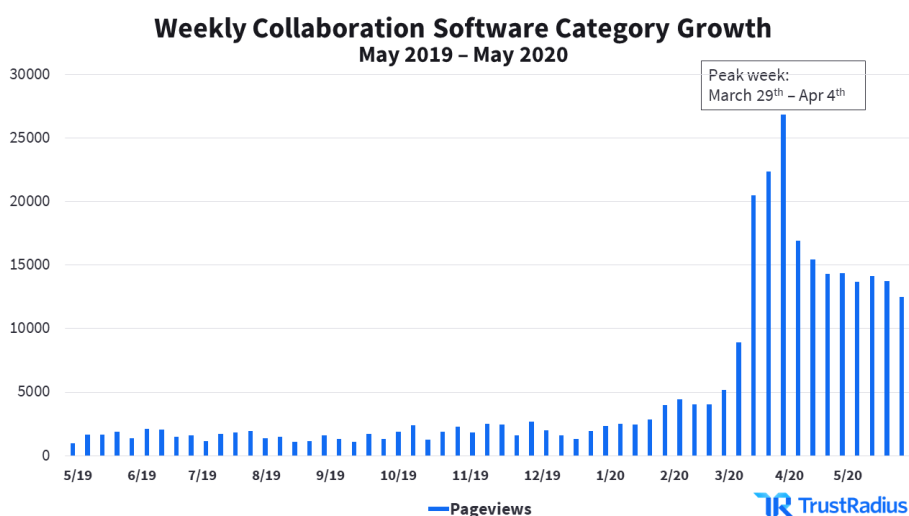
U drugom poglavlju ovog završnog rada bit će pokazana već dostupna rješenja koja rješavaju isti problem, te kako izgleda njihovo sučelje. U trećem poglavlju navedene su i detaljnije objašnjene tehnologije i alati koji su bili korišteni za razvoj ove web aplikacije. U četvrtom poglavlju bit će opisan postupak izrade aplikacije, a u petom poglavlju bit će prikazano sučelje same aplikacije, te njene funkcionalnosti.

1.1. Zadatak završnog rada

Zadatak završnog rada je izrada web aplikacije koja ima mogućnost prijave korisnika, kreiranja projekata, te kreiranje bilježaka za svaki zasebni projekt. Bilješke funkcioniraju poput zadataka koje korisnici mogu označiti izvršenima i mogu sadržavati privitke. Korisnik koji kreira projekt ima mogućnost dodavati druge korisnike koji mogu surađivati na projektu. Korisnici koji nisu dodani na projekt ne mogu te projekte vidjeti niti im pristupiti.

2. PREGLED PODRUČJA TEME

Vođenje bilješki i praćenje zadataka na projektima bitno je kako bi se postigli što bolji rezultati i kako bi izvođenje zadataka bilo uspješnije. Stoga se za projekte koriste razni alati kako bi se to ostvarilo. Bez obzira na veličinu projekta ili tima koji radi na samom projektu, za svaki projekt bi bilo poželjno koristiti alat koji pruža mogućnost vođenja bilješki na projektu. Pojavom bolesti COVID-19 i ograničenja dolazaka na radna mjesta znatno je porasla potreba za alatima za kolaboraciju. Na slici 2.1. nalazi se graf koji prikazuje broj posjeta na web stranicama alata za kolaboraciju. Vidljivo je da su posjete takvim web stranicama porasle u mjesecima kada se pojavila bolest i kada su uvedena ograničenja.

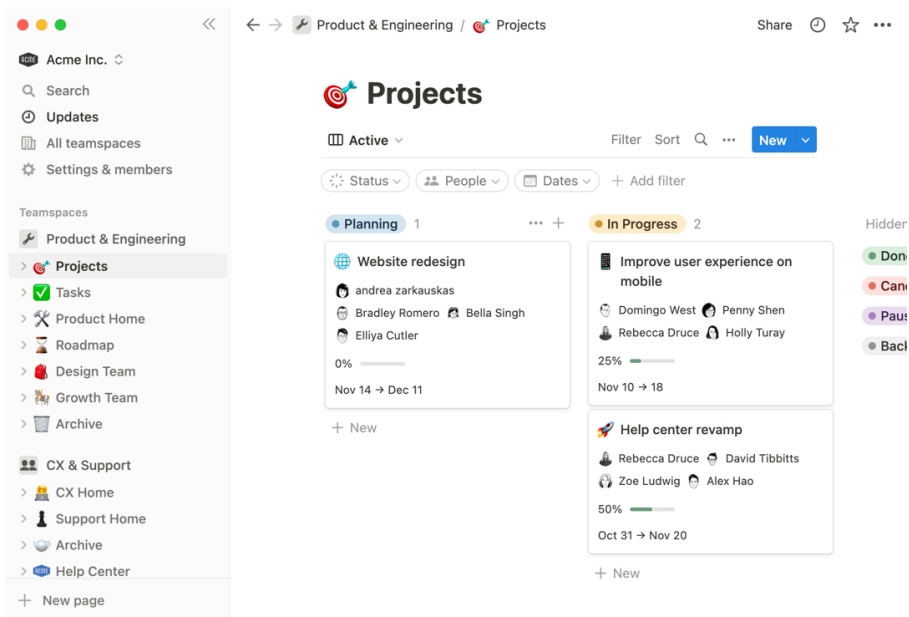


Slika 2.1. Graf posjeta web stranicama alata za kolaboraciju [12]

U nastavku će biti prikazan prvi komercijalni softver i trenutno najkorišteniji alati koji se koriste u svrhu raspodjele zadataka i vođenja bilješki.

2.1. Notion

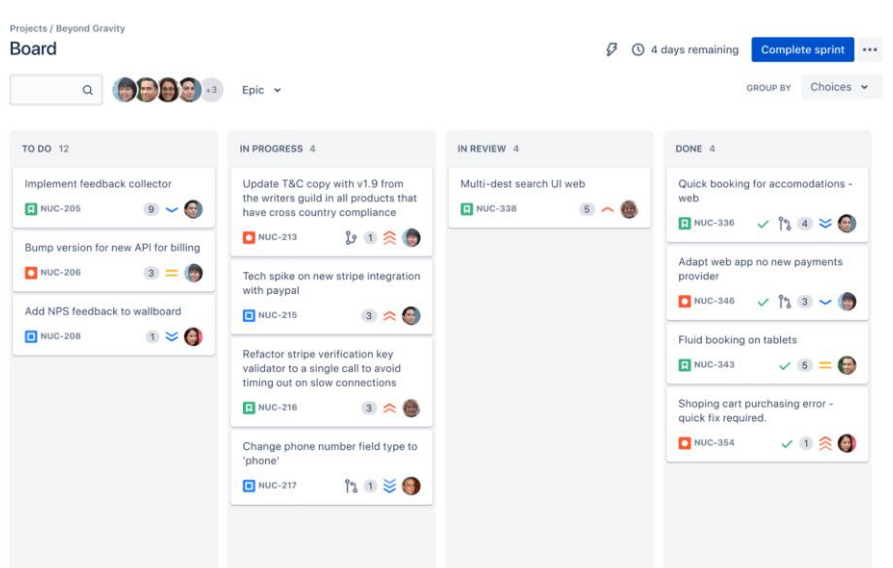
Prva aplikacija koja će biti prikazana naziva se Notion. Notion nudi razne alate za organiziranje, upravljanje i praćenje projekata i zadataka. Moguće je kreirati jednostavne liste zadataka koje su namijenjene za upravljanje individualnim projektima, kompleksnije projekte i zadatke na kojima je moguće dodati sudionike i naprednije upravljati projektom i zadacima, a nudi i mnoge druge funkcionalnosti i integriranje s drugim alatima. Aplikacija je besplatna za upotrebu grupama do 10 ljudi, ali postoje i plaćene opcije za veće grupe ili organizacije i moguće ju je koristiti unutar web preglednika, a postoje i aplikacije za Windows i macOS operativne sustave kao i mobilne aplikacije za Android i iOS. [1] Na slici 2.2. prikazano je korisničko sučelje aplikacije.



Slika 2.2. Prikaz korisničkog sučelja aplikacije Notion

2.2. Jira

Druga aplikacija koja će biti prikazana naziva se Jira. Poput aplikacije Notion, Jira nudi razne alate za organizaciju projekata, praćenje zadataka, kao i integraciju s drugim alatima. Aplikacija je besplatna za korištenje grupama do 10 korisnika, dok za veće grupe postoje plaćene opcije. Aplikaciju je moguće koristiti u web pregledniku ili putem desktop aplikacije za Windows, Linux i macOS operacijske sustave ali ne postoji mobilna aplikacija. [2] Na slici 2.3. prikazano je korisničko sučelje aplikacije.



Slika 2.3. Prikaz korisničkog sučelja aplikacije Jira

2.3. Monday.com

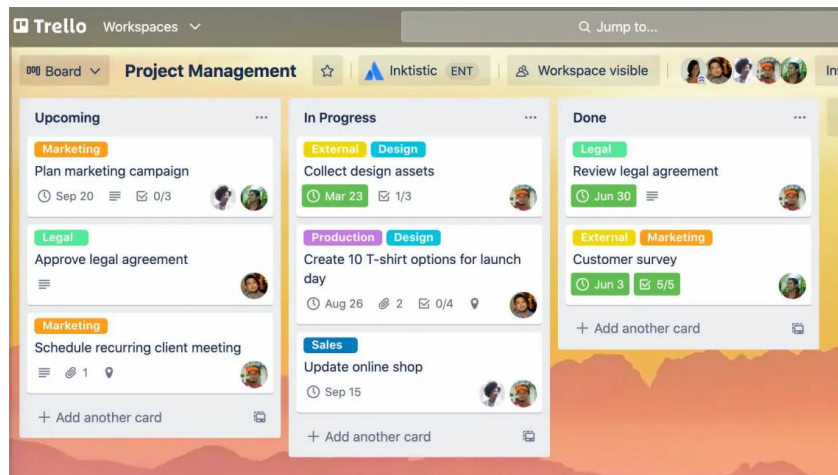
Sljedeći alat naziva se Monday.com. Za razliku od aplikacije Notion koja nije primarno namijenjena upravljanja projektima, aplikacija Monday.com sličnija je aplikaciji Jira kojima je glavni zadatak omogućiti vođenje projekata. Dok je Jira više namijenjena za razvoj softvera, Monday.com omogućuje vođenje projekata u raznim područjima poput marketinga, ljudskim resursima i drugima. Integriranje s drugim alatima moguće je kao i kod prethodno spomenutih aplikacija. Aplikacija je besplatna za korištenje za najviše 2 korisnika u grupi, dok za veće grupe postoje plaćene opcije. Aplikaciju je moguće koristiti unutar web preglednika, kao desktop aplikaciju na Windows ili macOS operativnom sustavu ili na Android ili iOS uređajima. [3] Na slici 2.4. prikazano je korisničko sučelje aplikacije.

Q3 project overview					
Main table Timeline Kanban Dashboard					
Integrate Automate / 2					
This month					
	Owner	Status	Timeline	Due date	Priority
Finalize kickoff materials		Done	<div style="width: 100%;"></div>	Sep 15	★★★★★
Refine objectives		Working on it	<div style="width: 75%;"></div>	Sep 19	★★★★★
Identify key resources		Stuck	<div style="width: 25%;"></div>	Sep 22	★★★★★
Test plan		Done	<div style="width: 100%;"></div>	Sep 26	★★★★★
Next month					
	Owner	Status	Timeline	Due date	Priority
Update contractor agreement		Done	<div style="width: 100%;"></div>	Oct 10	★★★★★
Conduct a risk assessment		Working on it	<div style="width: 75%;"></div>	Oct 13	★★★★★
Monitor budget		Stuck	<div style="width: 25%;"></div>	Oct 19	★★★★★
Develop communication plan		Done	<div style="width: 100%;"></div>	Oct 22	★★★★★

Slika 2.4. Prikaz korisničkog sučelja aplikacije Monday.com

2.4. Trello

Posljednja aplikacija koja će biti prikazana naziva se Trello. Aplikacije Jira i Trello razvija tvrtka Atlassian. Za razliku od aplikacije Jira, Trello nije namjenjen specifičnoj grupi korisnika. Omogućuje kreiranje ploča s zadacima i integracije s drugim alatima kao i prethodni alati. Besplatna opcija omogućuje korištenje individualnim korisnima ili grupama kreiranje manji broj ploča dok plaćene opcije dodaju napredne mogućnosti uz neograničen broj ploča. Dostupne su aplikacije za web preglednike, Windows i macOS operacijske sustave i Android i iOS uređaje. [4] Na slici 2.5. prikazano je korisničko sučelje aplikacije Trello.



Slika 2.5. Prikaz korisničkog sučelja aplikacije Trello

3. KORIŠTENE TEHNOLOGIJE I ALATI

U ovom poglavlju bit će navedene sve tehnologije i alati koji su korišteni u svrhu izrade web aplikacije za ovaj završni rad. Za izradu je korišten uređivač teksta Visual Studio Code, web aplikacija izrađena je u Python programskom jeziku koristeći Django web okvir, te je za opisivanje korisničkog sučelja korišten Bootstrap CSS okvir (engl. *CSS framework*).

3.1. Django

Django je web okvir otvorenog koda napisan u programskom jeziku Python koji omogućuje kreiranje web aplikacija. Tim koji je zadužen za razvoj i održavanje Django-a naziva se Django Software Foundation. Django omogućuje brzi razvoj skalabilnih aplikacija nudeći veliki izbor dodatnih alata poput:

- autentifikacije korisnika,
- administraciju sadržaja,
- dinamičko generiranje HTML-a.

Nadalje, Django sadrži veliki broj sigurnosnih alata koji sprječavaju najčešće sigurnosne propuste kao što su:

- zaštite od SQL ubrizgavanja,
- uporabu SSL/HTTPS enkripcije,
- zaštitu od krivotvorenja zahtjeva,
- sustava za upravljanje korisničkim računima i lozinkama . [5]

Django aplikacije temelje se na model view template (MVT) arhitekturi koja će biti objašnjena u nastavku. U svrhu izrade aplikacije u ovom završnom radu korištena je verzija Django 4.

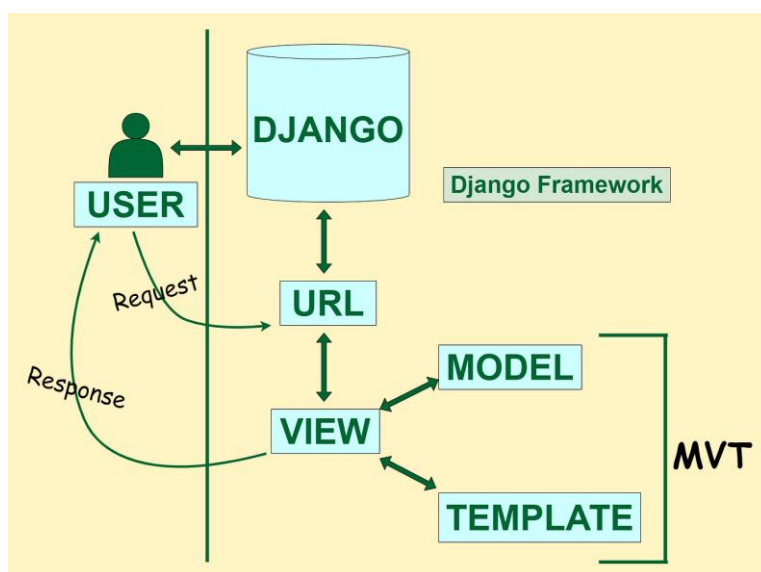
3.1.1. MVT arhitektura

Django web aplikacije izrađene su koristeći model view template (MVT) arhitekturu. Korištenje obrazaca arhitekture (engl. *architectural pattern*) prilikom izrade omogućava razvoj aplikacija prateći dobre prakse, daje aplikacijama strukturu i definira glavne komponente, što čini pronalaženje i ispravljanje greški i razumijevanje koda lakšim, čini aplikacije skalabilnima i održivima. MVT arhitektura sastoji se od tri komponente:

- Model
- Pogled (engl. *view*)

- Predložak (engl. *template*)

Model predstavlja sučelje podataka Django aplikacija koje opisuje kako su podatci predstavljeni u aplikaciji i bazi podataka, kao i logiku upravljanja podacima. Pogled je zaslužan za primanje, obradu i odgovaranja na korisničke zahtjeve. Unutar pogleda se nalazi logika zaslužna za upravljanje modelima i dohvaćanjem dinamičkog sadržaja koji će biti predstavljen u predlošku. Predložak se sastoji od statičkog HTML sadržaja i posebne sintakse Django template jezika koja opisuje gdje i kako će biti prikazan dinamički sadržaj i kontrolira logiku unutar pogleda. Na slici 3.1. prikazana je shema model view template arhitekture. [6]



Slika 3.1. Shema MVT arhitekture [11]

3.1.2. Python

Python je objektno-orientirani programski jezik otvorenog koda. Razvoj i održavanje Python-a obavlja tim pod nazivom Python Software Foundation. Prilikom izrade aplikacije u ovom završnom radu korištena je verzija Python 3.

3.1.3. Pip

Upravitelj paketa (engl. *package manager*) programskog jezika Python naziva se pip. Pomoću njega moguće je instalirati razne pakete koji olakšavaju razvoj programa desktop ili web aplikacija, pronalaženjem grešaka, formatiranjem koda prema standardima i pružaju mnoge druge mogućnosti.

3.1.4. SQLite

SQLite je laka (engl. *lightweight*) baza podataka koja ne zahtijeva posebnog poslužitelja već podatke sprema na lokalni disk. [7] Django po zadanim postavkama koristi SQLite za bazu podataka što je korisno za proces razvoja aplikacije kada nema potrebe za pohranjivanjem velikih količina podataka. Također omogućava prijenos koda na veće baza podataka poput PostgreSQL ili Oracle koje se također mogu koristiti za Django aplikacije.

3.2. Bootstrap

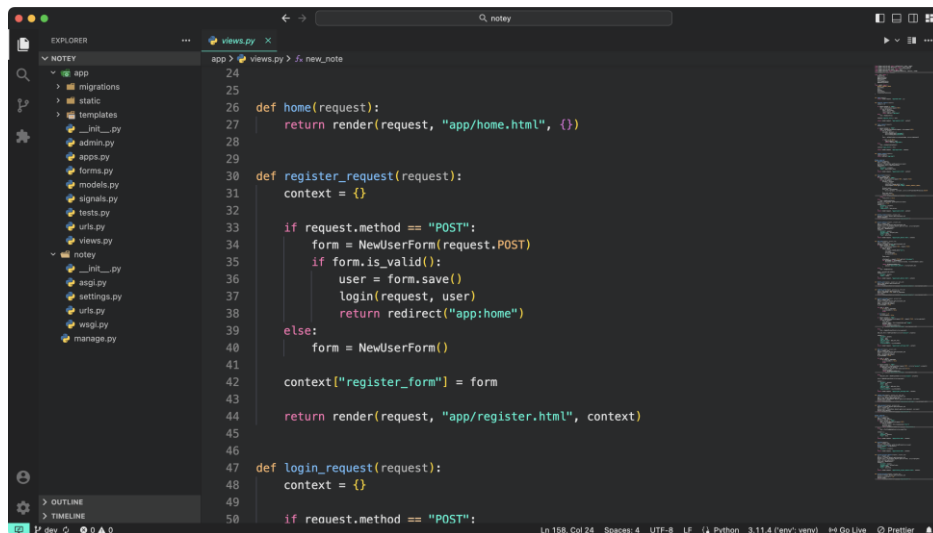
Bootstrap je Cascading Style Sheets (CSS) okvir otvorenog koda namijenjen lakšoj izgradnji korisničkih sučelja web stranica pružajući gotove klase i komponente. Nadalje, pruža mogućnost izgradnje responzivnih web stranica što omogućuje prilagodbu raznim veličinama ekrana. Dodavanje Bootstrap-a u bilo koji projekt je moguće uključivanjem Cascading Style Sheets (CSS) i JavaScript (JS) datoteka potrebnih za funkcioniranje Bootstrap-a unutar HTML datoteka. [8] Za potrebe izgradnje korisničkog sučelja korištena je verzija Bootstrap 5.

3.3. Visual Studio Code

Visual Studio Code je besplatan uređivač teksta koji pruža razne značajke koje olakšavaju razvoj aplikacija. Neke od tih značajki su:

- ugrađeni naredbeni redak,
- integracija Git alata za verzioniranje koda,
- veliki broj proširenja (engl. *extensions*),
- podršku za veliki broj programskih jezika,
- otkrivanje pogrešaka (engl. *debugging*)

i mnoge druge. [9] Dostupne su desktop aplikacije za Windows, Linux i macOS operacijske sustave. Na slici 3.2. prikazano je korisničko sučelje aplikacije.



Slika 3.2. Sučelje aplikacije Visual Studio Code

3.4. GitHub

GitHub je besplatna platforma koja omogućava pohranjivanje koda, verzioniranje i kolaboraciju s drugim programerima. Kako bi pohranili kod na GitHub potrebno je kreirati lokalni repozitorij pomoću Git alata za verzioniranje, koji se tada može povezati s GitHub repozitorijem na kojemu je onda moguće pohranjivati promjene. GitHub je koristan ukoliko više programera radi na jednom projektu, jer omogućava dohvaćanje promjena koje su drugi programeri napravili koristeći jednostavne naredbe u naredbenom retku.

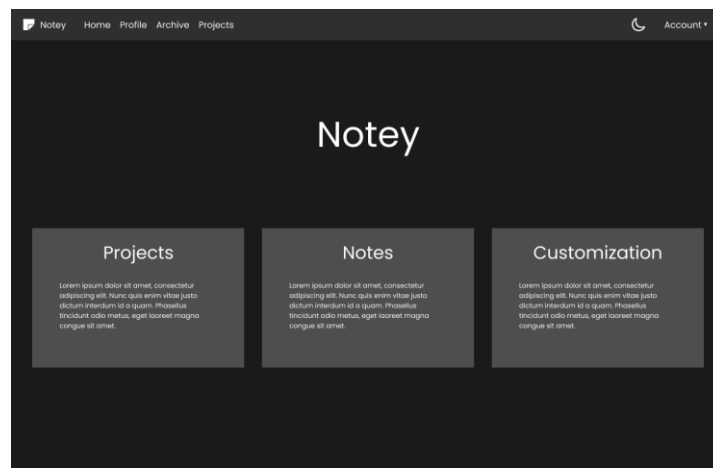
4. IZRADA APLIKACIJE

U ovom poglavlju bit će opisan postupak izrade web aplikacije.

4.1. Dizajn aplikacije

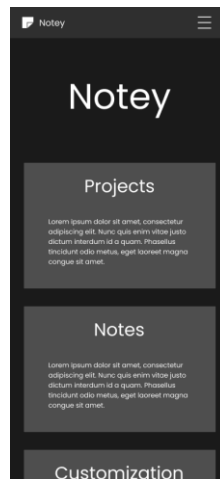
Prije početka izrade aplikacije, dobra je ideja kreirati maketu dizajna aplikacije kako bi lakše bilo izgraditi korisničko sučelje prilikom izrade aplikacije. Za potrebu izrađivanja makete korišten je besplatan alat Figma koji se može koristiti u web pregledniku, a nudi i desktop aplikacije za Windows, macOS i Linux. Korištenje Figma i sličnih alata omogućuje prikazivanje rasporeda sadržaja, veličine fontova, korištene boje i mnoge druge pojedinosti korisničkog sučelja koje se žele definirati prije početka izrade aplikacije. U nastavku će biti prikazan dizajn aplikacije izrađen u alatu Figma.

Izrađena maketa aplikacije sastoji se od tri različita ekrana kao i desktop i mobilnom verzijom. Na slici 4.1. prikazana je desktop verzija početnog zaslona aplikacije. Početni zaslon sastoji se od tri kartice koje detaljnije opisuju mogućnosti aplikacije. Također, na vrhu ekrana može se vidjeti navigacijska traka koja sadrži poveznice na druge zaslone, padajući izbornik za korisnički račun i gumb za mijenjanje teme aplikacije.



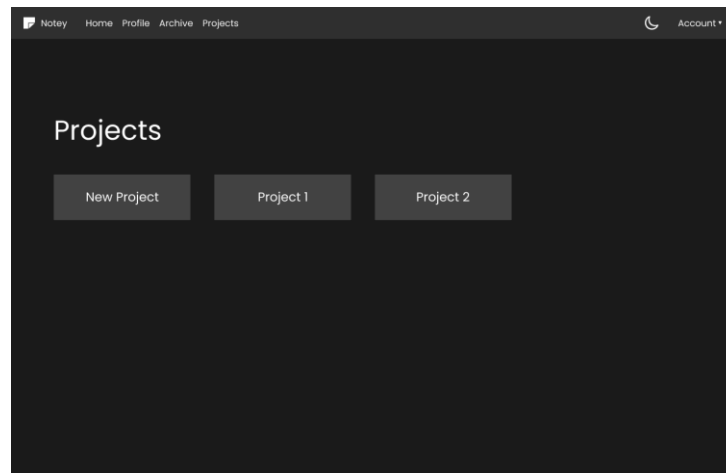
Slika 4.1. Maketa početnog zaslona

Ukoliko se aplikacija koristi na mobilnom uređaju, sučelje početnog zaslona izgledati će kao što je prikazano na slici 4.2..



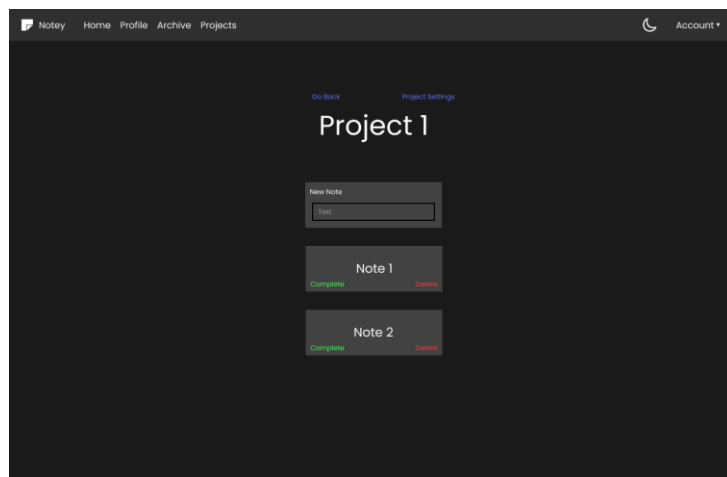
Slika 4.2. Maketa početnog zaslona na ekranu mobilnog uređaja

Prelaskom na zaslon za projekte prikazat će se sučelje kao na slici 4.3.. Ono sadrži sve projekte koje je korisnik kreirao ili na koje je dodan, kao i gumb za kreiranje novog projekta. Klikom na odabrani projekt korisniku će biti prikazan zaslon projekta.



Slika 4.3. Maketa zaslona s projektima

Pristupanjem na zaslon određenog projekta korisniku se prikazuje sučelje kao na slici 4.4.. Gumbovi za završavanje, brisanje, polje za unos teksta bilješke ili postavke projekta bit će prikazani ovisno o tome je li korisnik kreirao projekt ili o ovlastima koje su dodijeljene korisniku, ukoliko je dodan na projekt.



Slika 4.4. Maketa zaslona bilješki na projektu

4.2. Kreiranje virtualne okoline

Upravitelj paketa sve pakete koje preuzme stavlja u isti direktorij, što može izazvati probleme ukoliko se nakupi veći broj paketa, jer može doći do konflikata između paketa koji ovise o drugim paketima. Kako bi se to spriječilo potrebno je kreirati virtualnu okolinu (engl. *virtual environment*) što omogućuje Python-ov ugrađeni modul venv.

Python virtualna okolina predstavlja strukturu direktorija koja sadrži sve što je potrebno za korištenje lake (engl. *lightweight*) i izolirane Python okoline. [10] Nakon kreiranje virtualne okoline potrebno ju je aktivirati kako bi ju koristili, te je onda moguće preuzeti potrebne pakete za izradu aplikacije. Na slici 4.5. prikazan je postupak kreiranja virtualne okoline s imenom env i aktiviranja okoline.

```
~/Documents/zavrzni-rad
> python -m venv env
~/Documents/zavrzni-rad
> source env/bin/activate
~/Documents/zavrzni-rad  zavrzni-rad
>
```

Slika 4.5. Kreiranje virtualne okoline

Kada je virtualna okolina aktivirana najčešće se prije direktorija u naredbenom retku nalazi ime okoline, s obzirom da je korišten naredbeni redak s modifikacijama. Na slici je s desne strane vidljiva ikona Python jezika i naziv direktorija u kojemu se nalazi virtualna okolina.

Ukoliko korištenje virtualne okoline više nije potrebno, moguće ju je deaktivirati naredbom prikazanom na slici 4.6. čime nestaje oznaka aktivirane virtualne okoline.

```
~/Documents/zavrnsni-rad
> deactivate

~/Documents/zavrnsni-rad
>
```

Slika 4.6. Deaktiviranje virtualne okoline

4.3. Kreiranje aplikacije

Kada je aktivirana virtualna okolina potrebno je preuzeti i instalirati Django paket koristeći pip upravitelj paketa. Slika 4.7. prikazuje preuzimanje i instalaciju Django paketa unutar virtualne okoline.

```
~/Documents/zavrnsni-rad
> pip install django
Collecting django
  Downloading Django-4.2.4-py3-none-any.whl (8.0 MB)
    ----- 8.0/8.0 MB 9.6 MB/s eta 0:00:00
Collecting asgiref<4,>=3.6.0 (from django)
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1 (from django)
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
    ----- 41.2/41.2 kB 4.4 MB/s eta 0:00:00
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.4 sqlparse-0.4.4

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: pip install --upgrade pip
```

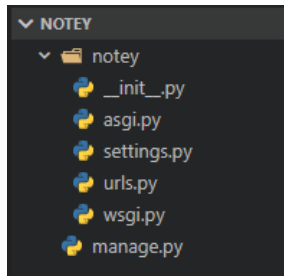
Slika 4.7. Instalacija Django paketa

Nakon preuzimanja paketa potrebno je kreirati Django projekt koji će predstavljati web aplikaciju. Unutar projekta moguće je kreirati aplikacije koje predstavljaju dijelove glavnog projekta pomoću kojih se odvajaju razne funkcionalnosti. S obzirom na zahtjeve aplikacije koja se izrađuje u završnom radu bit će kreirana samo jedna aplikacija. Django projekt je moguće kreirati koristeći naredbu *django-admin startproject project-name*. Kreiranjem projekta kao što je prikazano na slici 4.8. kreira se novi direktorij, koji će unutar sebe sadržavati direktorij projekta i datoteke potrebne za projekt.

```
~/Documents/zavrnsni-rad
> django-admin startproject notey
```

Slika 4.8. Kreiranje Django projekta

Na slici 4.9. prikazana je struktura direktorija projekta kao i sve datoteke koje se nalaze unutar projekta.

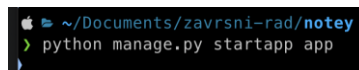


Slika 4.9. Struktura Django projekta

Svaki Django projekt sadrži sljedeće datoteke:

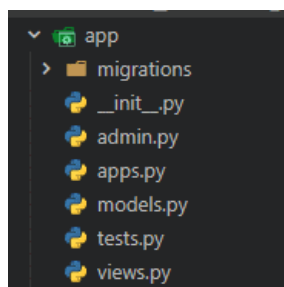
- *manage.py* datoteka predstavlja alat pomoću kojega se upravlja projektom i ona se jedina ne nalazi unutar direktorija projekta.
- *__init__.py* je prazna datoteka koja direktorij u kojemu se nalazi Python-u predstavlja kao modul.
- *asgi.py* predstavlja ulaznu točku za ASGI kompatibilne web servere.
- *settings.py* datoteka sadrži postavke Django projekta.
- *urls.py* datoteka sadrži deklaracije URL-ova za projekt.
- *wsgi.py* predstavlja ulaznu točku za WSGI kompatibilne web servere. [5]

Nadalje, potrebno je kreirati Django aplikaciju kako je prikazano na slici 4.10..



Slika 4.10. Kreiranje Django aplikacije

Kreiranjem aplikacije dobiva se direktorij sa strukturom datoteka kao što je prikazano na slici 4.11..



Slika 4.11. Struktura Django aplikacije

Svaka Django aplikacija sadrži sljedeće:

- *migrations* direktorij sadrži datoteke sa promjenama u modelima po kojima se mijenja shema baze podataka.

- `__init__.py` je prazna datoteka koja direktorij u kojemu se nalazi Python-u predstavlja kao modul.
- `admin.py` datoteka služi za konfiguriranje administratorskog sučelja. Unutar nje je moguće podesiti koji modeli će biti vidljivi u administratorskom sučelju web aplikacije.
- `apps.py` datoteka služi za konfiguraciju aplikacije.
- `models.py` datoteka služi za definiranje modela aplikacije. Po uzoru na modele se kreira baza podataka.
- `tests.py` datoteka sadrži kod za testove koji ispituju ispravnost aplikacije.
- `views.py` datoteka sadrži definicije pogleda koji opisuju kako se aplikacija ponaša za zahtjev koji primi.

Unutar app direktorija potrebno je još dodati direktorij naziva templates s poddirektorijem app u kojemu će se nalaziti sve HTML datoteke predložaka korištene u aplikaciji.

4.4. Postavljanje projekta

Unutar settings.py datoteke potrebno je dodati prethodno kreiranu aplikaciju kako bi ju mogli povezati s projektom. Na slici 4.12. prikazan je način povezivanja aplikacije s projektom, također na slici su vidljivi i dodatni paketi koji su dodani, a služe za uređenje obrazaca (engl. *forms*) koristeći Bootstrap okvir.

```

32 # Application definition
33
34 INSTALLED_APPS = [
35     "django.contrib.admin",
36     "django.contrib.auth",
37     "django.contrib.contenttypes",
38     "django.contrib.sessions",
39     "django.contrib.messages",
40     "django.contrib.staticfiles",
41     "app",
42     "crispy_forms",
43     "crispy_bootstrap5",
44 ]
45

```

Slika 4.12. Povezivanje projekta i aplikacije

Osim dodavanja aplikacije u settings.py datoteci, aplikaciju je potrebno povezati i unutar urls.py datoteke koja sadrži URL rute koje se povezuju s pogledima. Na slici 4.13. prikazan je dio urls.py datoteke koja sadrži kod za povezivanje aplikacije s projektom.

```

20
21 urlpatterns = [
22     path("admin/", admin.site.urls),
23     path("", include("app.urls")),
24 ]
25

```

Slika 4.13. URL putanje projekta

Nakon povezivanja aplikacije s projektom moguće je započeti s kreiranjem aplikacije. U nastavku će biti opisan postupak dodavanja funkcionalnosti u aplikaciju kreiranjem modela, kako bi se opisali podaci korišteni u aplikaciji, opisivanjem pogleda, kako bi se povezali podaci s korisničkim sučeljem i definiranjem predložaka koji opisuju statičke i dinamičke dijelove korisničkog sučelja.

4.5. Modeli

U Django okviru modeli predstavljaju podatke koji će se pohranjivati i biti korišteni unutar aplikacije. Kako bi kreirali model, potrebno je unutar datoteke `models.py` kreirati novu klasu, koja je podklasa (engl. *subclass*) klase `django.db.models.Model`. Unutar klase modela moguće je definirati njene attribute kao polja pomoću predefiniраниh Django klasa, kojima je moguće postaviti dodatne opcije poput zadane vrijednosti, maksimalne vrijednosti i slično. Na slici 4.14. prikazan je primjer modela projekta koji je korišten u aplikaciji.

```

16 class Project(models.Model):
17     creator = models.ForeignKey(User, on_delete=models.SET_NULL, null=True)
18     name = models.CharField(max_length=128, unique=True)
19     image = models.ImageField(
20         upload_to="images/projects/", default=DEFAULT_PROJECT_IMAGE
21     )
22     is_archived = models.BooleanField(default=False)
23
24     def __str__(self) -> str:
25         return f"[{self.creator.username}]{self.name}"
26
27     def get_notes(self):
28         return Note.objects.filter(project=self)
29
30     def get_users(self):
31         return ProjectUser.objects.filter(project=self)
32

```

Slika 4.14. Model projekta

Model Project sadrži sljedeća polja i metode:

- Polje *creator* koje sadrži strani ključ (engl. *foreign key*) na korisnika koji je kreirao projekt.
- Polje *name* sadrži ime projekta.
- Polje *image* sadrži sliku, odnosno putanju na sliku spremljenu lokalno, koja se prikazuje u kartici projekta.
- Polje *is_archived* sadrži Booleovu vrijednost koja govori je li projekt arhiviran.

- Metodu `__str__` koja vraća detalje o projektu u obliku niza znakova (engl. *string*).
- Metodu `get_notes` koja vraća niz bilješki zapisanih u projektu.
- Metodu `get_users` koja vraća niz objekata modela `ProjectUser` koje povezuju korisnike i projekte na koje su dodani.

Kreirane modele moguće je dodati u administratorsko sučelje koje omogućava pregled svih entiteta pohranjenih u bazi podataka. Kako bi se model dodao u administratorsko sučelje potrebno ga je uvesti iz `models.py` datoteke i registrirati koristeći metodu `register`. Dodavanje modela u administratorsko sučelje prikazano je na slici 4.15..

```

2
3 from .models import Profile, Project, ProjectUser, Note, Attachment
4
5 admin.site.register(Profile)
6 admin.site.register(Project)
7 admin.site.register(ProjectUser)
8 admin.site.register(Note)
9 admin.site.register(Attachment)
10

```

Slika 4.15. Dodavanje modela u administratorsko sučelje

Nakon definiranja modela također je potrebno napraviti migracije na bazi podataka. Migracija za sve kreirane modele pravi tablice unutar baze podataka ili preslikava promjene napravljene unutar modela na bazu podataka. Prva naredba koju je potrebno izvršiti je `python manage.py makemigrations` koja kreira migracije ovisno o promjenama na modelima. Nakon toga je potrebno izvršiti naredbu `python manage.py migrate` koja kreirane migracije primjenjuje na bazu podataka.

4.6. Pogledi

Pogled predstavlja funkciju koja za web zahtjev (engl. *web request*) vraća web odgovor (engl. *web response*). Odgovor može biti sadržaj HTML web stranice, preusmjerenje (engl. *redirect*) ili greška (engl. *error*). Funkcije pogleda sadržavaju logiku potrebnu za vraćanje odgovora. Osim u obliku funkcija, poglede je moguće predstaviti kao klase što omogućava strukturiranje pogleda korištenjem nasljeđivanja (engl. *inheritance*) i mixina. [5] Za izradu ovog završnog rada korišteni su funkcijski pogledi. Na slici 4.16. vidljiv je primjer pogleda koji se koristi kada korisnik pokuša pristupiti zaslona s projektima.

```

75 @login_required
76 def projects(request):
77     user = request.user
78     projects = ProjectUser.getProjects(user=user)
79     new_project_form = NewProjectForm()
80     context = {
81         "projects": projects,
82         "form": new_project_form,
83     }
84     return render(request, "app/projects.html", context)
85

```

Slika 4.16. Pogled zaslona s projektima

Unutar pogleda nalazi se varijabla `context` koja služi za sakupljanje dinamičkih dijelova podataka i predaju istih predlošku. Predstavljena je rječnikom (engl. *dictionary*) koji za ključ (engl. *key*) sadrži nekakvu vrijednost. Unutar predložaka moguće je pristupiti dinamičkim podacima iz `context` varijabla pomoću ključeva koji su postavljeni za vrijednosti. Oznaka `@login_required` iznad definicije funkcije je dekorater koji preusmjerava korisnika na zaslon za prijavu ukoliko nije prijavljen ili na željeni zaslon ukoliko je prijavljen. Ovo je potrebno zato što se na početnom zaslonu aplikacije nalazi poveznica koja vodi na zaslon s projektima ili na zaslon profila korisnika, a neprijavljeni korisnik tom zaslonu ne bi trebao moći pristupiti. Osim toga je potrebno unutar postavki projekta dodati putanju na pogled za prijavu (engl. *login*) kao što je prikazano na slici 4.17..

```

138
139 LOGIN_URL = "app:login"
140

```

Slika 4.17. URL putanja za prijavu

Nakon kreiranja pogleda pomoću funkcija potrebno ih je povezati s njihovim URL putanjama (engl. *path*). Kako bi se povezali pogledi i URL putanje potrebno je unutar `urls.py` datoteke dodati putanje unutar liste `urlpatterns`. Svaka putanja definira se pomoću funkcije `path` koja za argumente prima rutu, pogled i naziv. Na slici 4.18. prikazan je dio putanja aplikacije.

```

5 app_name = "app"
6 urlpatterns = [
7     path("", views.home, name="home"),
8     path("register", views.register_request, name="register"),
9     path("login", views.login_request, name="login"),
10    path("logout", views.logout_request, name="logout"),
11    path("projects", views.projects, name="projects"),
12    path("archive", views.archive, name="archive"),
13    path(
14        "archive/<int:project_id>",
15        views.archive_project_details,
16        name="archive_project_details",
17    ),
18    path("profile", views.profile, name="profile"),
19 ]

```

Slika 4.18. URL putanje aplikacije

Odlaskom na određenu URL putanju aplikacija poziva funkciju povezanu s tom putanjom koja je definirana unutar argumenata. Argument name služi kada se unutar predložaka kreiraju gumbi koji vode do određenog dijela aplikacije, kako bi Django znao na koju URL putanju odvesti korisnika.

4.7. Predlošci

Kako bi se prikazao sadržaj pogleda potrebno je napraviti predložak koji je povezan s istim pogledom. Kao što je prethodno objašnjeno, predlošci se smještaju u direktorij templates unutar direktorija aplikacije. Predlošci su način na koji Django dinamički generira HTML sadržaj. Za opisivanje dinamičkog sadržaja unutar datoteka predložaka korišten je Django template language. Unutar predložaka mogu se nalaziti varijable koje se zamjenjuju s vrijednostima kada se predložak evaluira ili oznake koje služe za upravljanje logikom predloška. [5] Osim toga omogućuje i produživanje predložaka na način da se kreira osnovni predložak koji sadrži sadržaj vidljiv u svakom dijelu web aplikacije. Osnovni predložak obično se imenuje base.html. Unutar osnovnog predloška dodaje se blok prikazan na slici 4.19. koji će biti zamijenjen sa sadržajem zasebnih predložaka.

```
231 </div>
232 <div class="my-auto">{% block content %} {% endblock %}</div>
233 </div>
```

Slika 4.19. Blok za sadržaj unutar osnovnog predloška

Na slici 4.20. prikazan je predložak ekrana za prijavljivanje korisnika.

```
1 {% extends 'app/base.html' %} {% block content %} {% load crispy_forms_tags %}
2 <div class="container py-5 col col-md-6 col-lg-4">
3   <h1>Login</h1>
4   <form method="POST">
5     {% csrf_token %} {{ login_form|crispy }}
6     <button class="btn btn-primary px-4 py-2" type="submit">Login</button>
7   </form>
8 </div>
9 {% endblock %}
```

Slika 4.20. Predložak zaslona za prijavljivanje

4.8. Signali

S obzirom na to da je u aplikaciji bilo potrebno proširiti korisnički model s dodatnim poljima, a u aplikaciji nije mijenjan osnovni korisnički model koji dolazi predefiniiran u Django-u, kreiran je model koji se sastoji od stranog ključa i dodatnih polja. Prilikom kreiranja korisničkog računa potrebno je stvoriti i povezati novi model s istim korisničkim računom, što je postignuto korištenjem signala. Signali u Django-u omogućavaju da pošiljatelj (engl. *sender*) obavijesti

primatelje (engl. *receiver*) da se nešto dogodilo unutar aplikacije i nalaze se u datoteci `signals.py`. [5] Na slici 4.21. prikazane su funkcije signala korištene za kreiranje modela koji proširuje osnovni korisnički model.

```
8 @receiver(post_save, sender=User)
9 def create_profile(sender, instance, created, **kwargs):
10     if created:
11         Profile.objects.create(user=instance)
12
13
14 @receiver(post_save, sender=User)
15 def save_profile(sender, instance, **kwargs):
16     instance.profile.save()
```

Slika 4.21. Funkcije signala

Funkcije sa slike 4.21. pozivaju se prilikom kreiranja ili ažuriranja User modela i mijenjaju instancu Profile modela koja je povezana s korisničkim računom.

4.9. Forme

Forma je skup elemenata, a nalazi se između oznaka za definiranje forme koji omogućuju korisnicima unos teksta, izbor između različitih opcija, prijenos datoteka i slično. Za rad sa formama postoje metode GET i POST. Metoda POST prikupljene podatke šalje na poslužitelja i čeka odgovor, dok metoda GET prikupljene podatke stavlja u URL na koji se šalje zahtjev. Zbog toga je poželjnije koristiti POST metodu za slanje privatnih podataka, poput lozinke prilikom registriranja ili prijavljivanja.

Forme u Django-u se definiraju kao klase unutar `forms.py` datoteke. Moguće je kreirati forme nasljeđivanjem osnovne klase `Form` ili drugih klasa, poput `ModelForm` koje olakšavaju kreiranje formi. `ModelForm` klasa omogućuje kreiranje formi po uzoru na polja modela, kako ne bi bilo potrebno definirati polja u formi za svaki od polja u modelu. [5] Na slici 4.22. prikazana je forma za stvaranje novog projekta.

```
31
32 class NewProjectForm(ModelForm):
33     class Meta:
34         model = Project
35         fields = ["name", "image"]
36         exclude = ["creator"]
37
```

Slika 4.22. Forma za kreiranje projekta

Na slici 4.23. prikazan je dio predloška za prikazivanje prethodne forme.

```

<form
  action="{% url 'app:new_project' %}"
  method="post"
  enctype="multipart/form-data"
>
  {% csrf_token %}
  <div class="modal-body">{{ form|crispy }}</div>
  <div class="modal-footer">
    <a
      class="btn btn-secondary px-4 py-2"
      href="{% url 'app:projects' %}"
    >
      Close
    </a>
    <button type="submit" class="btn btn-primary px-4 py-2">
      Create
    </button>
  </div>
</form>

```

Slika 4.23. Korištenje forme unutar predloška

Kako bi se forme zaštitile od krivotvorenja zahtjeva, između oznaka za definiranje forme potrebno je dodati oznaku za Cross Site Request Forgery (CSRF) token. Osim toga na slici je vidljiva oznaka dodatnog paketa Django Crispy Forms koji se koristi za lakše uređivanje izgleda formi.

Osim kreiranja forme i prikazivanja u predlošku, potrebno je unutar pogleda na koji se forma poziva definirati što se događa ukoliko je zahtjev koji je poslan dobro odrađen ili sadrži greške. Na slici 4.24. prikazan je pogled za prethodno definiranu formu.

```


1 def new_project(request):
2     user = request.user
3     if request.method == "POST":
4         form = NewProjectForm(request.POST, request.FILES)
5         if form.is_valid():
6             project = Project(
7                 creator=user,
8                 name=form.cleaned_data["name"],
9                 image=request.FILES.get("image", DEFAULT_PROJECT_IMAGE),
10            )
11            project.save()
12            proj_user = ProjectUser(
13                project=project, user=user, permission=ProjectUserPermission.DELETE
14            )
15            proj_user.save()
16            form_errors = True
17            return HttpResponseRedirect(reverse("app:projects"))
18        else:
19            form_errors = False
20    else:
21        form = NewProjectForm()
22        projects = ProjectUser.getProjects(user=user)
23        form_errors = True
24        context = {
25            "projects": projects,
26            "form": form,
27            "form_errors": form_errors,
28        }
29    return render(request, "app/projects.html", context)
30

```

Slika 4.24. Pogled koji sadrži formu

5. IZGLED I RAD APLIKACIJE

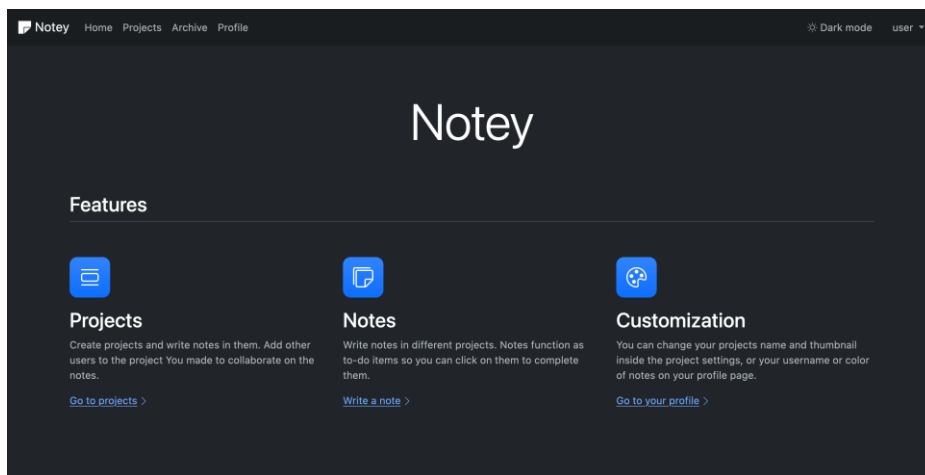
Kako bi se aplikacija mogla koristiti ili testirati potrebno je pokrenuti lokalni poslužitelj pomoću `manage.py` datoteke, kako bi mogli putem web preglednika pristupiti aplikaciji. Pokretanje lokalnog poslužitelja za aplikaciju prikazano je slikom 5.1..

```
~/Documents/zavrzni/notey-django/notey on  dev
> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 29, 2023 - 16:09:15
Django version 4.1.7, using settings 'notey.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

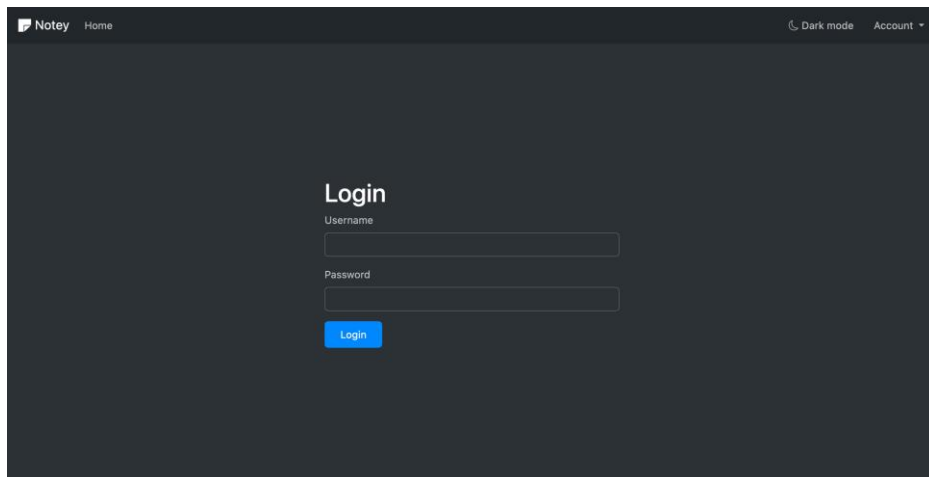
Slika 5.1. Pokretanje aplikacije

Pristupanjem aplikaciji putem web preglednika korisniku se prikazuje naslovna strana aplikacije prikazana na slici 5.2..



Slika 5.2. Početni zaslon aplikacije

Na vrhu stranice vidljiva je navigacijska traka (engl. *navigation bar*) koja omogućuje korisniku navigaciju do drugih zaslona, poput zaslona s projektima, arhive i profila, ukoliko je korisnik prijavljen. S desne strane se nalazi padajući izbornik (engl. *dropdown menu*) koji omogućuje korisniku prijavu ili registraciju ukoliko korisnik nije prijavljen i odjavljivanje ukoliko je prijavljen, kao i prekidač za tamni način rada (engl. *dark mode toggle*). Na sredini se nalaze tri dijela koja opisuju mogućnosti aplikacije, ujedno s poveznicama koje vode na zaslone vezane za mogućnost. Ukoliko korisnik nije prijavljen, poveznice korisnika odvedu na zaslon za prijavu. Na slici 5.3. prikazan je zaslon za prijavljivanje korisnika.

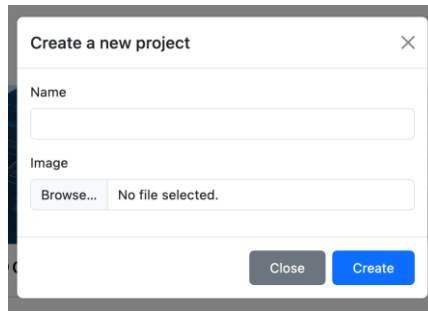


Slika 5.3. Zaslón za prijavu korisnika

Klikom na poveznicu Projects na navigacijskoj traci prikazuje se zaslón prikazan na slici 5.4.. Ovaj zaslón sadrži kartice svih projekata koje je korisnik kreirao ili na koje je dodan, kao i karticu za kreiranje novog projekta. Klikom na karticu za kreiranje novog projekta otvara se iskočni prozor s obrascem za kreiranje projekta prikazan na slici 5.5.. Na sljedećim slikama aplikacija je prikazana u svijetlom načinu rada.

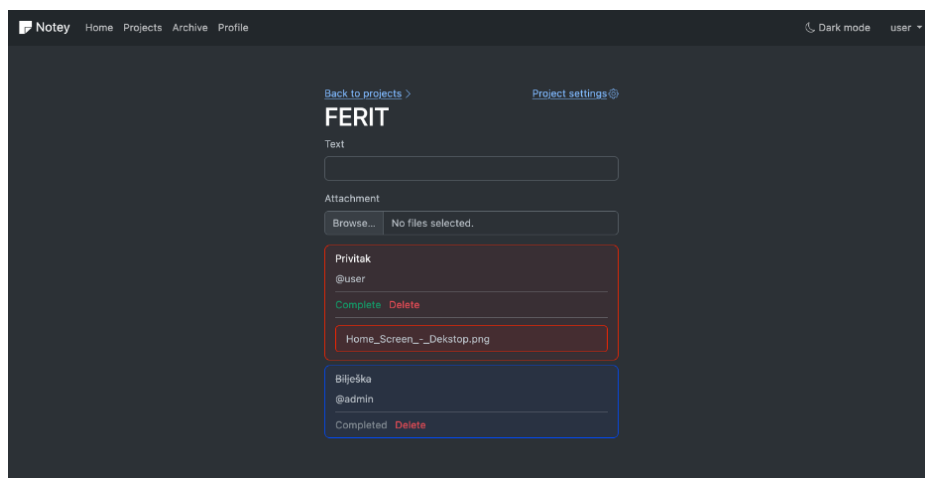


Slika 5.4. Zaslón s projektima



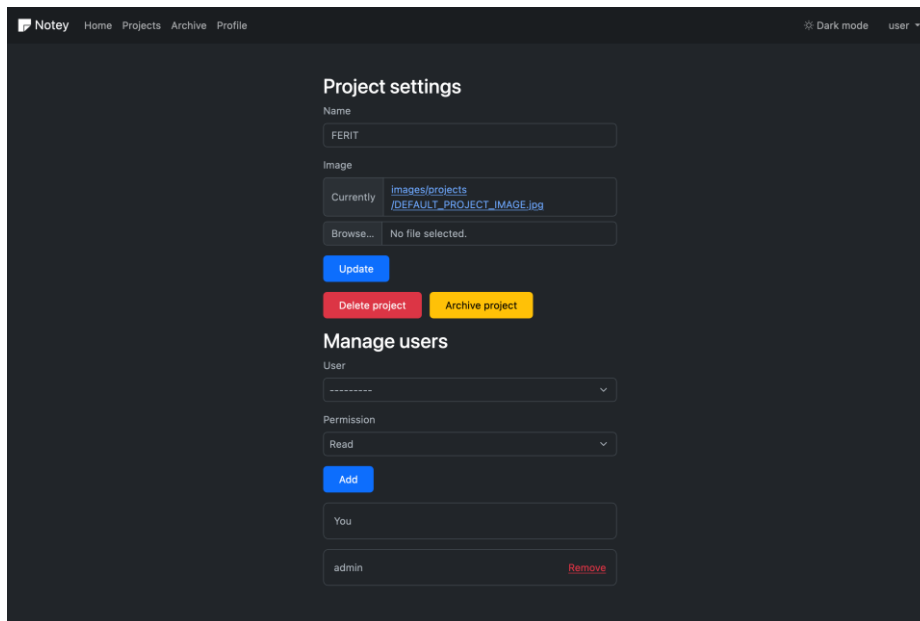
Slika 5.5. Iskočni prozor za kreiranje projekta

Nakon kreiranja projekta na zaslonu se stvara kartica koja omogućuje ulazak u projekt. Klikom na karticu prikazuje se zaslon prikazan na slici 5.6.. Iznad imena projekta nalazi se poveznica za povratak na zaslon sa svim projektima, kao i poveznica za postavke projekta koja je vidljiva samo korisniku koji je kreirao projekt. Ispod imena projekta nalazi se polje za unos teksta bilješke i polje za dodavanje privitaka na bilješku, a ispod polja nalaze se sve bilješke. Svaka bilješka sadrži korisničko ime autora, gumb za označavanje bilješke dovršenom, kao i gumb za brisanje bilješke. Ukoliko je bilješka označena kao dovršena, tekst bilješke biti će sive boje i precrtan.



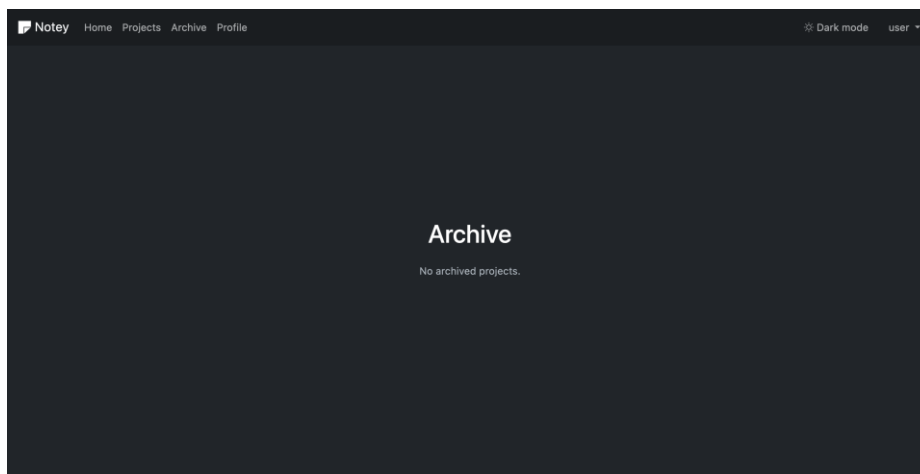
Slika 5.6. Zaslon projekta s bilješkama

Klikom na poveznicu za odlazak na postavke projekta prikazuje se zaslon prikazan na slici 5.7.. Unutar postavki projekta moguće je promijeniti ime samog projekta, sliku projekta, obrisati projekt, arhivirati projekt ukoliko su sve bilješke označene kao izvršene i dodati nove korisnike na projekt s odabranim dopuštenjima ili ukloniti postojeće korisnike. Zavisno od dopuštenja koja se dodijele korisniku prilikom dodavanja, korisniku se možda neće prikazati polja za zapisivanje bilješki, gumb za izvršavanje bilješki ili gumb za brisanje bilješki.



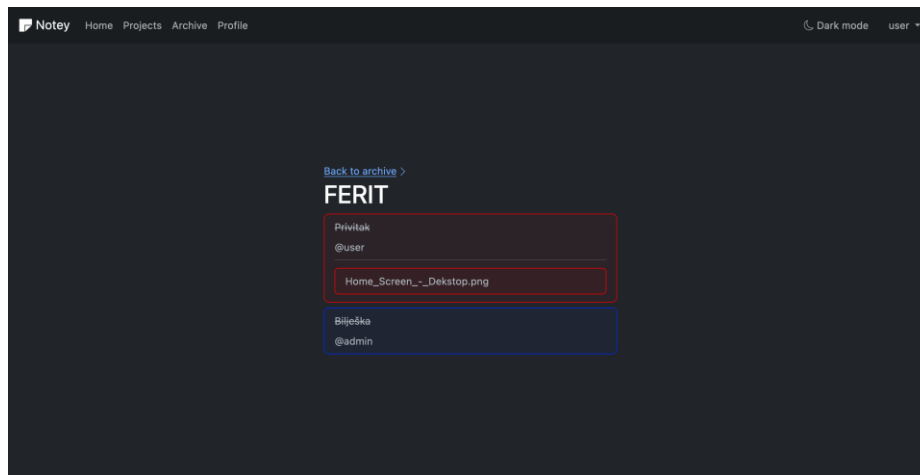
Slika 5.7. Postavke projekta

Na slici 5.8. prikazan je zaslon arhive projekata koji prikazuje sve projekte na kojima je korisnik sudjelovao koji su arhivirani ili poruku ukoliko nema arhiviranih projekata kao što je vidljivo na slici.



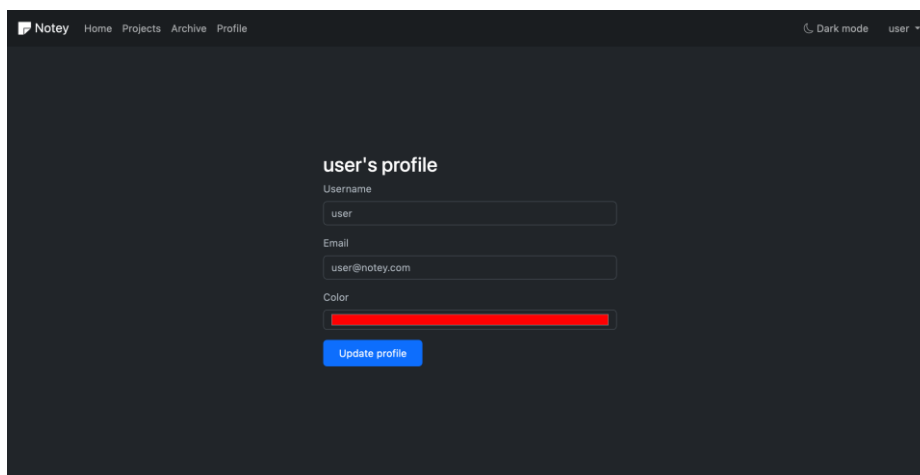
Slika 5.8. Zaslon arhive projekata

Ukoliko postoje arhivirani projekti, njima je moguće pristupiti i pregledati sve bilješke zapisane na tom projektu bez mogućnosti brisanja i dodavanja novih. Na slici 5.9. prikazan je izgled arhiviranog projekta.



Slika 5.9. Arhivirani projekt

Na navigacijskoj traci također postoji poveznica koja korisnika vodi na njegov profil. Korisnik na profilu može promijeniti korisničko ime, mail adresu povezanu s računom i boju bilješki koje je korisnik napisao. Na slici 5.10. prikazan je zaslon korisničkog profila.



Slika 5.10. Zaslon korisničkog profila

6. ZAKLJUČAK

U ovom radu opisan je problem vođenja projekata i bilješki na projektu, kao i izrada web aplikacije koja rješava takav problem. Aplikacija je izrađena tako da korisnicima bude jednostavna za korištenje. Korisnici mogu lako kreirati nove projekte i pregledavati arhivirane projekte. Za izradu aplikacije korištene su razne tehnologije i alati koji omogućuju brzi razvoj aplikacija koje su sigurne i prilagođavaju se na sve veličine ekrana, što osigurava dobro korisničko iskustvo prilikom korištenja. Za kreiranje aplikacije korišten je web okvir Django koji omogućuje brz razvoj složenih web aplikacija korištenjem programskog jezika Python za izradu. U radu su opisani dijelovi Django aplikacija i prikazani primjeri iz izrađene aplikacije. Moguća je i nadogradnja aplikacije dodavanjem novih značajki. Neki od primjera značajki koji bi se mogli dodati su mogućnost označavanja statusa zadatka, dodavanje roka za izvršavanje zadatka, uređivanje postojećih bilješki i sl.

LITERATURA

- [1] Notion: Your connected workspace for wiki, docs & projects, <https://www.notion.so/>, pristup 20.8.2023.
- [2] Jira | Issue & Project Tracking Software - Atlassian, <https://www.atlassian.com/software/jira>, pristup 20.8.2023.
- [3] Monday.com, <https://monday.com/>, pristup 22.9.2023.
- [4] Trello: Manage Your Team's Projects From Anywhere, <https://trello.com/>, pristup 22.9.2023.
- [5] Django: The web framework for perfectionists with deadlines, <https://www.djangoproject.com>, pristup 22.8.2023.
- [6] Django Project MVT Structure, <https://www.geeksforgeeks.org/django-project-mvt-structure/>, pristup 22.8.2023.
- [7] Python Documentation, <https://docs.python.org/3/library/sqlite3.html>, pristup 22.8.2023.ž
- [8] Bootstrap, <https://getbootstrap.com/docs>, pristup 22.8.2023.
- [9] Visual Studio Code - Code Editing. Redefined, <https://code.visualstudio.com/>, pristup 22.8.2023.
- [10] Python Virtual Environments: A Primer, <https://realpython.com/python-virtual-environments-a-primer/>, pristup 22.8.2023.
- [11] The MVT Design Pattern of Django, <https://python.plainenglish.io/the-mvt-design-pattern-of-django-8fd47c61f582>, pristup 22.8.2023.
- [12] Collaboration Software Statistics and Trends, <https://www.trustradius.com/vendor-blog/collaboration-statistics-and-trends>, pristup 22.9.2023.

SAŽETAK

U ovom završnom radu prikazan je postupak izrade web aplikacije čiji je cilj olakšati organizaciju projekata i vođenja bilješki. Aplikaciju je moguće koristiti za osobne projekte, ali i za projekte sa više suradnika. Osim toga, aplikacija omogućuje davanje dopuštenja čitanja, pisanja ili izvršavanja zadataka suradnicima na projektu, što poboljšava korisničko iskustvo prilikom korištenja aplikacije. Završene projekte moguće je arhivirati što omogućuje korisnicima da pregledavaju projekte nakon završavanja bez potrebe za brisanjem projekata. Web aplikacija izrađena je koristeći Django web okvir i Python programski jezik za razvoj serverskog dijela aplikacije, a za izgradnju korisničkog sučelja korišten je HTML uz Django template language i Bootstrap CSS okvir.

Ključne riječi: Bootstrap, Django, projekt, vođenje bilješki, web aplikacija

ABSTRACT

Web application for note taking on projects

This final paper presents the process of creating a web application which aims to make project organization and note taking easier. The application can be used for personal projects as well as for projects with multiple collaborators. In addition to that, the application allows giving permissions for reading, writing or completing tasks for other collaborators on the project which enhances the user experience while using the application. Completed projects can be archived which allows users to view the projects after completeing without the need to delete them. The web application is made using the Djano web framework and Python programming language for developing the application backend, and HTML with the Django template language and Bootstrap CSS framework for building the frontend of the application.

Keywords: Bootstrap, Django, note taking, project, web application,

ŽIVOTOPIS

Luka Kuterovac rođen je u Vinkovcima 15. lipnja 2001. godine. Osnovno obrazovanje započinje 2008. godine u Osnovnoj školi Vladimira Nazora u Vinkovcima. Nakon stjecanja osnovnog obrazovanja 2016. godine, započinje školovanje u Tehničkoj školi Ruđera Boškovića u Vinkovcima, smjer tehničar za mehatroniku. Završetkom srednje škole 2020. godine upisuje preddiplomski smjer računarstva, smjer programsko inženjerstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

Potpis autora

DODATAK

Poveznica na GitHub repozitorij aplikacije: <https://github.com/lukakuterovac/notey-django/tree/dev>