



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Koroška cesta 46
2000 Maribor, Slovenija



Naloga 02

Orodja za vodenje razvoja

oktober, 2021

Aleksander Grobelnik,
FERI RIT 2. letnik

Kazalo vsebine

1	Meistertask.....	1
1.1	Kreiranje projekta.....	1
1.2	Vedra.....	2
1.3	Opravila.....	3
2	Mega.....	4
2.1	Funkcionalnost.....	4
3	Discord.....	5
3.1	Kreiranje strežnika.....	5
3.2	Funkcionalnosti.....	6
4	Github.....	9
4.1	Kreiranje projekta.....	9
4.2	Dodajanje funkcionalnosti.....	10
5	Git odjemalec.....	11
5.1	Instalacija.....	11
5.2	Primer uporabe.....	12

Kazalo slik

Slika 1:	Zavihek Projekti.....	1
Slika 2:	Pojavno okno 2.....	1
Slika 3:	Pojavno okno 1.....	1
Slika 4:	Naš novi projekt.....	1
Slika 5:	Povabilo za priključitev.....	1
Slika 6:	Dodajanje vedra.....	2
Slika 7:	Dodajanje vedra.....	2
Slika 8:	Poimenovanje vedra.....	2
Slika 9:	Spreminjanje lastnosti veder.....	2
Slika 10:	Dodajanje opravila in dodajanje dolžnosti.....	3
Slika 11:	Nadzor opravila.....	3
Slika 12:	Dodatne možnosti opravila.....	3
Slika 13:	Možnost prenašanja datotek.....	4
Slika 14:	Dodajanje datotek oziroma map.....	4
Slika 15:	Izbiranje datoteke z diska.....	4
Slika 16:	Možnost delanja nad datotekami.....	5
Slika 17:	Prenos zip datoteke.....	5
Slika 18:	Nalaganje datotek.....	5
Slika 19:	Brisanje datotek.....	5
Slika 20:	Dodajanje strežnika.....	5
Slika 21:	Kreiranje projekta.....	6
Slika 22:	Povabljanje ljudi v strežnik.....	6
Slika 23:	Kreiranje strežnika.....	6
Slika 24:	Kreiranje strežnika.....	6
Slika 25:	Osnovna stran strežnika.....	7
Slika 26:	Klic v glasovnem kanalu.....	7
Slika 27:	Kanal Nov.....	8
Slika 28:	Kreiranje novega kanala.....	8
Slika 29:	Primer prikazovanja zaslona med klicom.....	8
Slika 30:	Pregled strežnika.....	8
Slika 31:	Deljenje zaslona.....	8

Slika 32: Začetna stran githuba.....	9
Slika 33: Kreiranje repozitorija.....	9
Slika 34: Vlečna zahteva.....	10
Slika 35: Dodajanje problema.....	10
Slika 36: Zavitek s problemi.....	10
Slika 37: Nastavitve github.....	11
Slika 38: git install.....	11
Slika 39: git check.....	11
Slika 40: ssh-keygen.....	12
Slika 41: ssh-add.....	12
Slika 42: Glavni git ukazi.....	12
Slika 43: git clone.....	13
Slika 44: git commit.....	13
Slika 45: vidnost na githubu.....	13
Slika 46: git branch.....	13

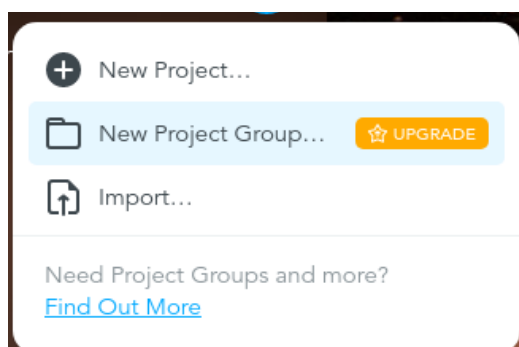
1 Meistertask

1.1 Kreiranje projekta

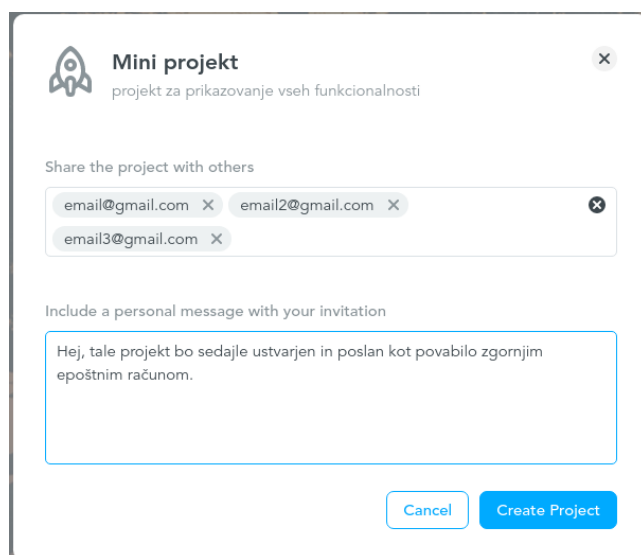
Po opravljeni registraciji in klik na gumb za potrditev e-poštnega računa se nam odpre začetni zaslon, kjer opazimo ogromno stvari in med njimi tudi PROJECTS zavihek, kjer kliknemo na plus znak v krogu. Odpre se nam pojavno okno s tremi opcijami. Izberemo najustreznejšo, kar v našem primeru pomeni Nov projekt, in sledimo zahtevam novega pojavnega okna, ki od nas zahteva ime projekta, kratek opis, e-poštne naslove vseh dodanih uporabnikov, ki dobijo tudi e-pošto, in sporočilo, ki ga bodo prejeli prijemniki. Lahko bi tudi kreirali Novo gručo projektov, vendar tega v zastoj verziji ni, oziroma lahko bi izbrali uvoz že kreiranih projektov v drugih okoljih. Odpre se nam novo kreirani projekt s predpripravljenimi vedri, ki jih bomo v naslednjem sklopu urejali.



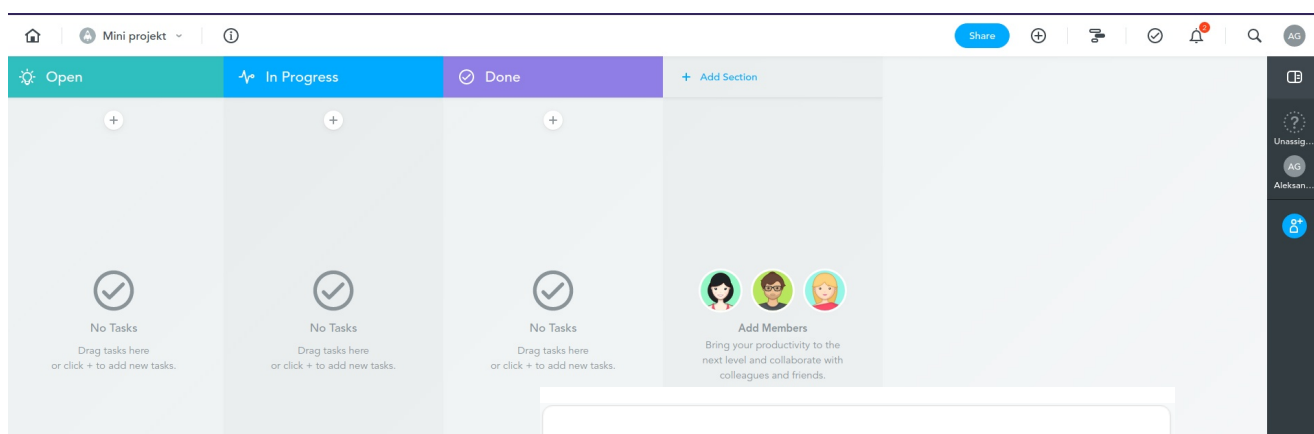
Slika 1: Zavihek Projekti



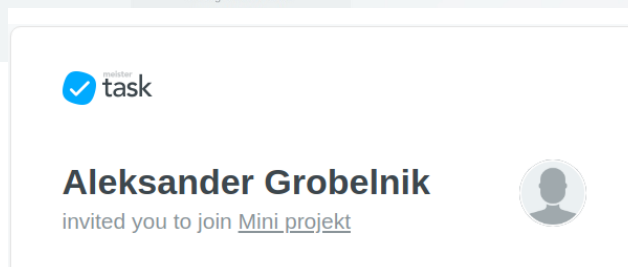
Slika 3: Pojavno okno 1



Slika 2: Pojavno okno 2



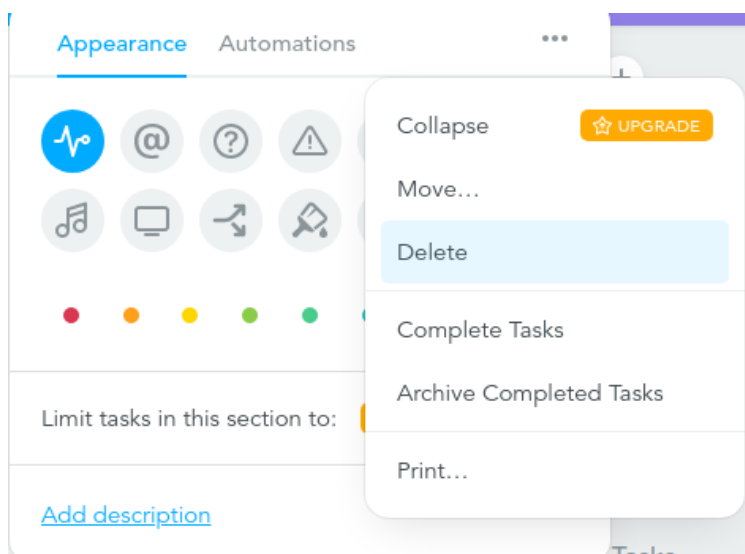
Slika 4: Naš novi projekt



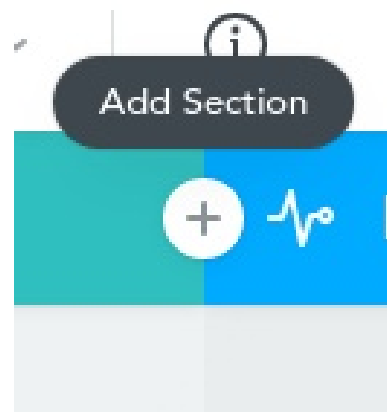
Slika 5: Povabilo za priključitev

1.2 Vedra

Če želimo dodajati vedra, lahko to storimo na začetnem zaslonu izbranega projekta. Postavimo se med dva vedra oziroma kliknemo na zadnji stolpec, ki je privzeto dodajanje novega vedra. Po pritisku na gumb oziroma privzeto vedro se nam pojavi besedilno polje za vnos imena sekcije. Izberemo ustrezno ime in pritisnimo na potrditveni gumb (oz. Angl. “Enter”). Vedro lahko sedaj kadarkoli preimenujemo, spreminjamo logotip, barvo, krajši opis, premik, brisanje in opravljanje z opravili. Seveda lahko tudi natisnemo vsebino vedra, ki ga nato avtomatično kreira v pdf dokument.



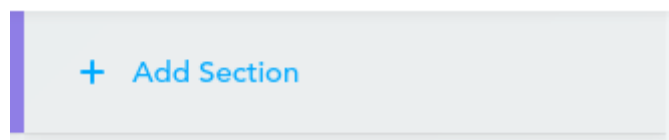
Slika 9: Spreminjanje lastnosti vedra



Slika 6: Dodajanje vedra



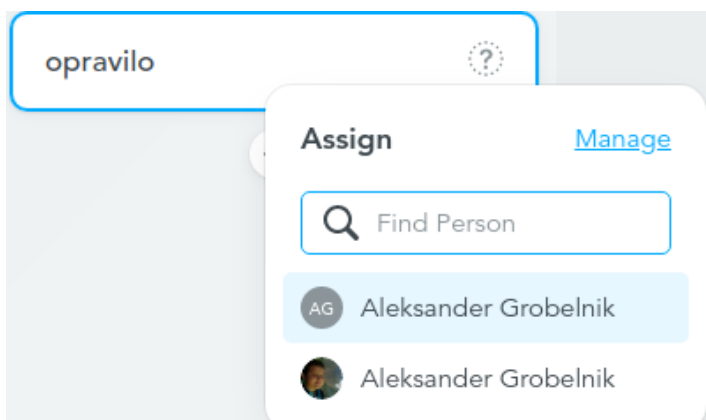
Slika 8: Poimenovanje vedra



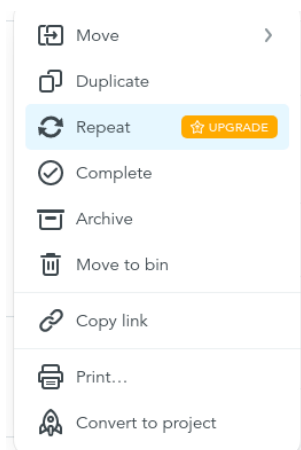
Slika 7: Dodajanje vedra

1.3 Opravila

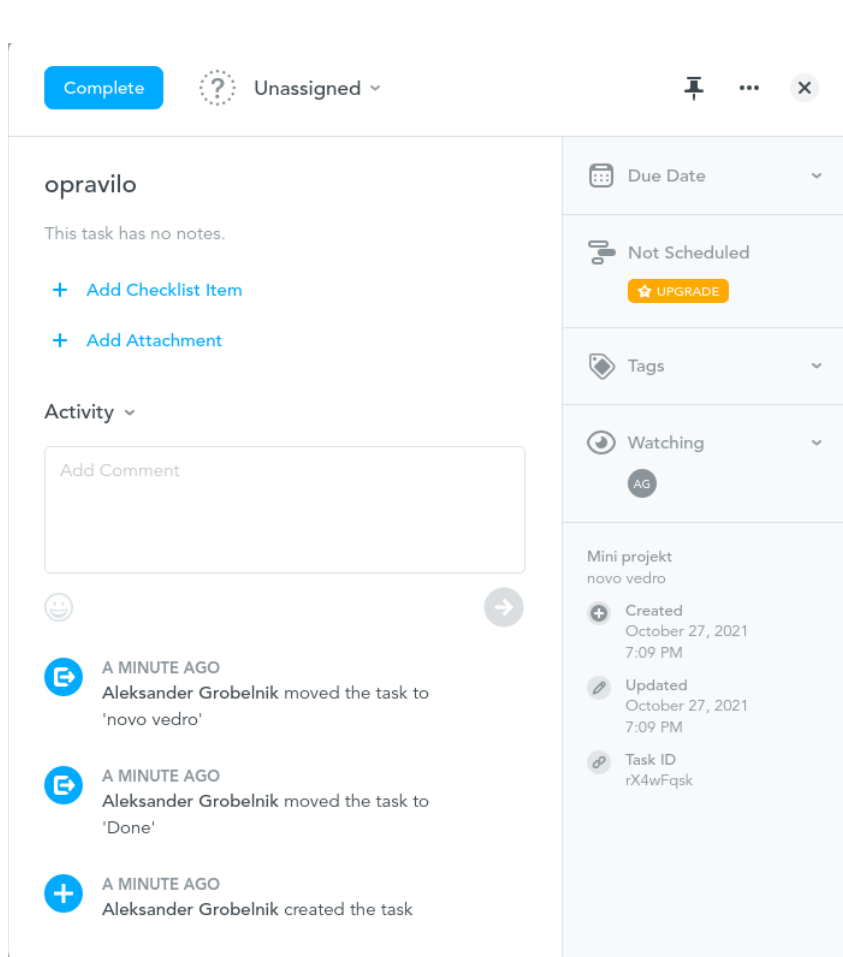
Če želimo dodajati opravila lahko to storimo s pritiskom na + v posameznem vedru. Po pritisku se odpre novo polje, ki je namenjeno poimenovanju opravila. Ko poimenujemo in potrdimo s potrditvenim gumbom se nam kreira novo opravilo. Če želimo še dodatno funkcionalnost kliknemo nanj oziroma na vprašaj v krogu zraven, ki nam omogoča zastavljanje dolžnosti opravil. Če pa kliknemo nanj, pa se nam odpre novo pojavno okno z ogromno funkcijami. Nekatere izmed teh so: dodajanje zapiskov, kontrolnega seznama in en element le tega seznama, raznih datotek, komentarjev, zadnji rok izvedbe opravila in pomembni podatki kot so: čas kreiranja, čas premika v vedro, kreiranja vedra, task ID in zadnji čas ažuriranja v obliki časovnega žiga. Opravilo lahko tudi brišemo, premikamo, podvojimo, arhiviramo, natisnemo ali pa ga pribijemo na oglasno desko. Seveda, pa ne smemo pozabiti na najpomembnejšo funkcionalnost, ki pa je zaključevanje oziroma arhiviranje projekta ob zaključku z klikom na kumb Zaključi.



Slika 10: Dodajanje opravila in dodajanje dolžnosti



Slika 12: Dodatne možnosti opravila

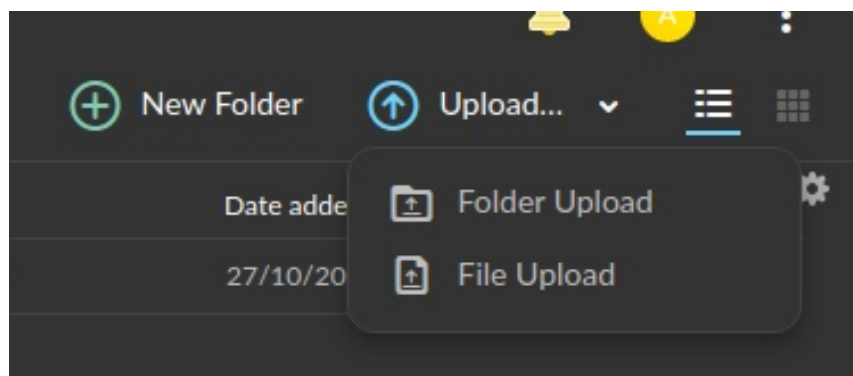


Slika 11: Nadzor opravila

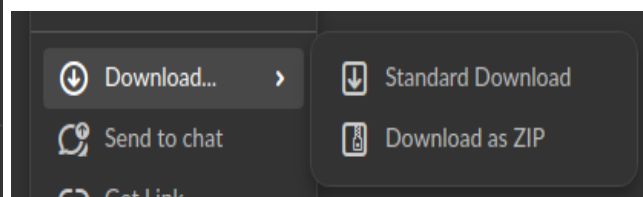
2 Mega

2.1 Funkcionalnost

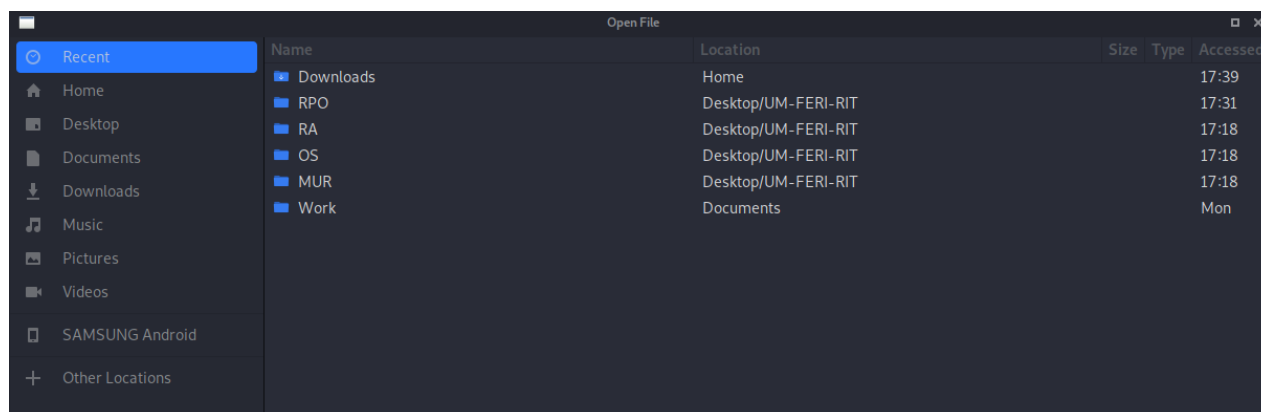
Po uspešni registraciji v sistem se nam pojavi prvo okno, v katerem lahko najdemo gumb Naloži in se postavimo z miško nad njim. Pojavijo se nam dve opciji, Naloži celo mapo oziroma Naloži posamezno datoteko. Po izbrani datoteki oziroma mapi se nam začne prenašati. Po zaključku prenašanja lahko opazimo novo naloženo datoteko na zaslonu. Če želimo z datoteko kaj početi kot so spreminjanje barve, pogled v koristne informacije, spreminjanje imena datoteke, premikanja, podvajanja, idp., lahko to storimo z desnim klikom na datoteko in izbira prej omenjenih opcij. Na seznamu je opcija Prenesi, ki nam da dve opciji in sicer, navade prenos oziroma prenos v stisjeni obliki ZIP. Po prenosu podatkov iz Mega strežnikov si lahko izberemo mesto shranjevanja. Kreiramo lahko tudi datoteke, v katere lahko dodajamo nove oziroma obstoječe datoteke s klikom na gumb Nova datoteka in ustreznim imenom. Če smo morda dodali napačno datoteko, oziroma jo zaradi drugih razlogov želimo odstraniti, izberemo Odstrani na posamezni datoteki oziroma mapi. Vpraša nas, če res želimo odstraniti datoteko oziroma jo prestaviti na stran v koš za odpad. Storitev je prostorsko omejena in moramo biti posebej pazljivi, da ne prekršimo meja uporabe.



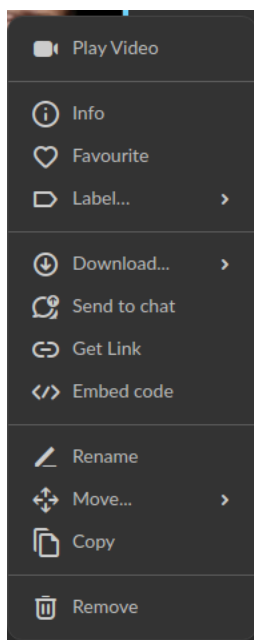
Slika 14: Dodajanje datotek oziroma map



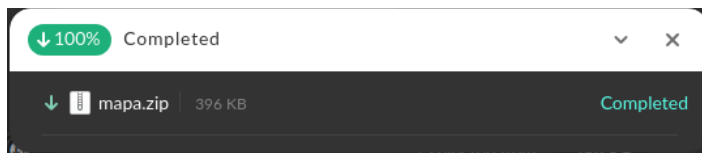
Slika 13: Možnost prenašanja datotek



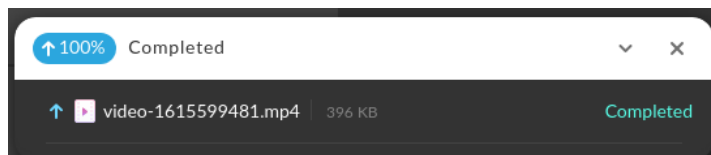
Slika 15: Izbiranje datoteke z diska



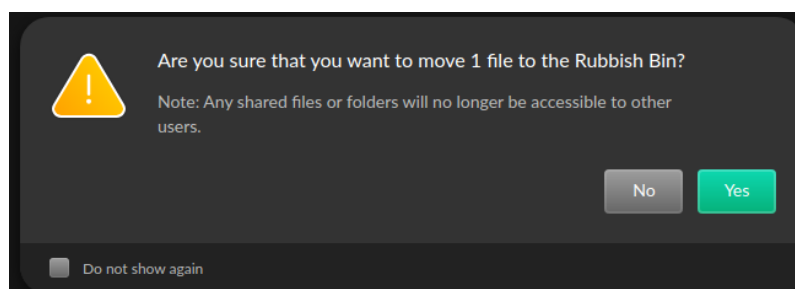
Slika 16: Možnost delanja nad datotekami



Slika 17: Prenos zip datoteke



Slika 18: Nalaganje datotek

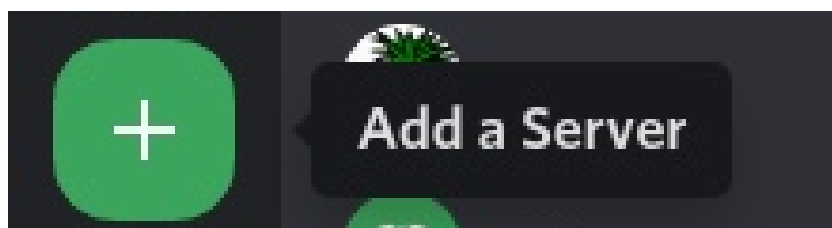


Slika 19: Brisanje datotek

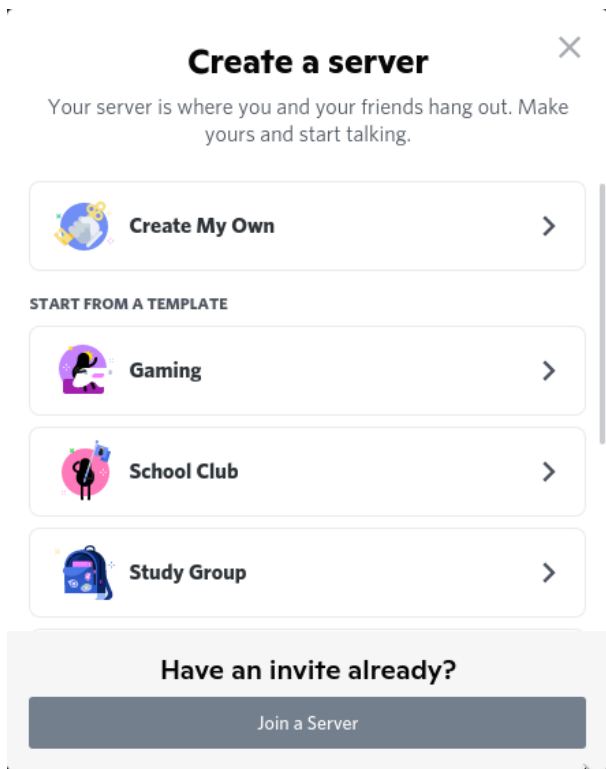
3 Discord

3.1 Kreiranje strežnika

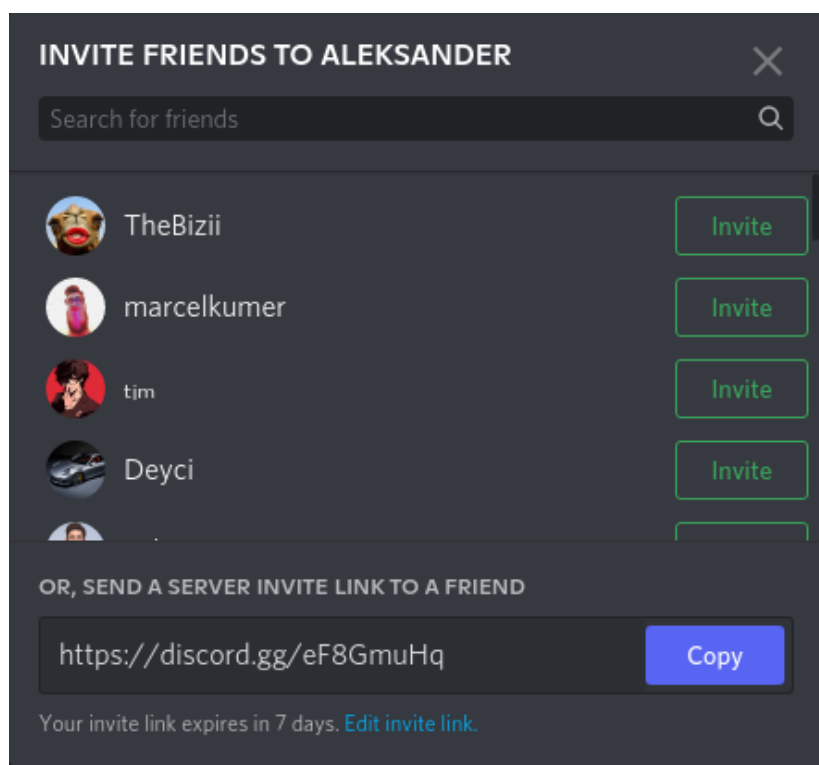
Če želimo uporabiti discord aplikacijo imamo več možnosti. Lahko uporabljamo spletno storitev discord.com, ki je dokaj podobna namizni aplikaciji. Jaz osebno ne morem povleciti dol discord aplikacije, brez uničevanja knjižnic, ki jih debian datoteka potrebuje in zato uporabljam spletno storitev. Če želimo kreirati strežnik najprej kliknemo na velik zelen plus znak se nam odpre novo belo podokno z naslovom Kreiraj strežnik. Na tem podoknu lahko zbiramo med predlogi, ki nam jih ponudi discord, vendar bomo vzeli Kreiraj svojega. Zberemo za koga kreiramo strežnik. V našem primeru je to za nas in naše prijatelje. Prišli smo do zadnjega koraka, kjer izbiramo ime strežnika in profilno sliko. Po izbrani odločitvi kreiramo strežnik.



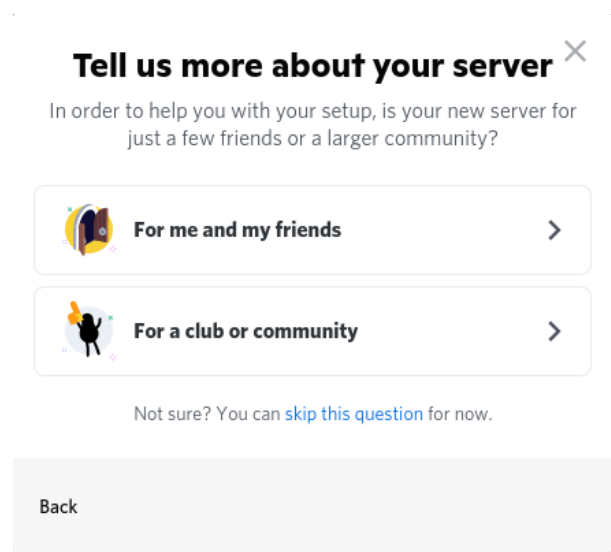
Slika 20: Dodajanje strežnika



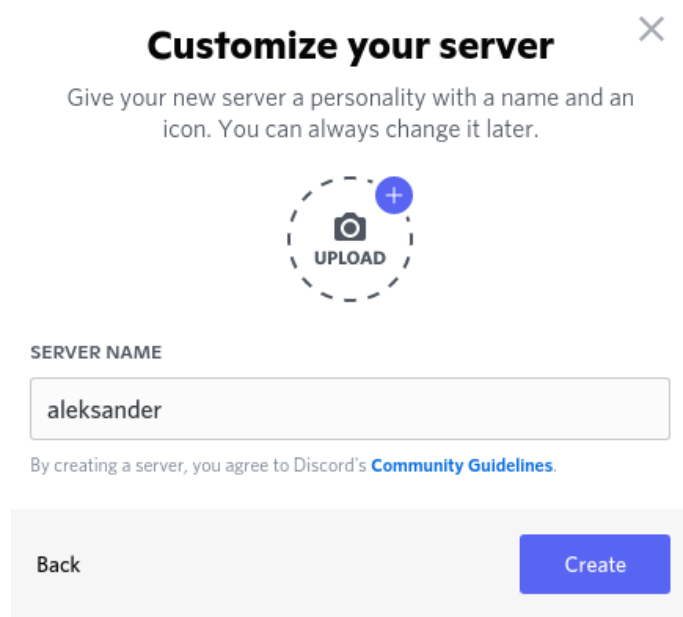
Slika 21: Kreiranje projekta



Slika 22: Povabljanje ljudi v strežnik



Slika 24: Kreiranje strežnika

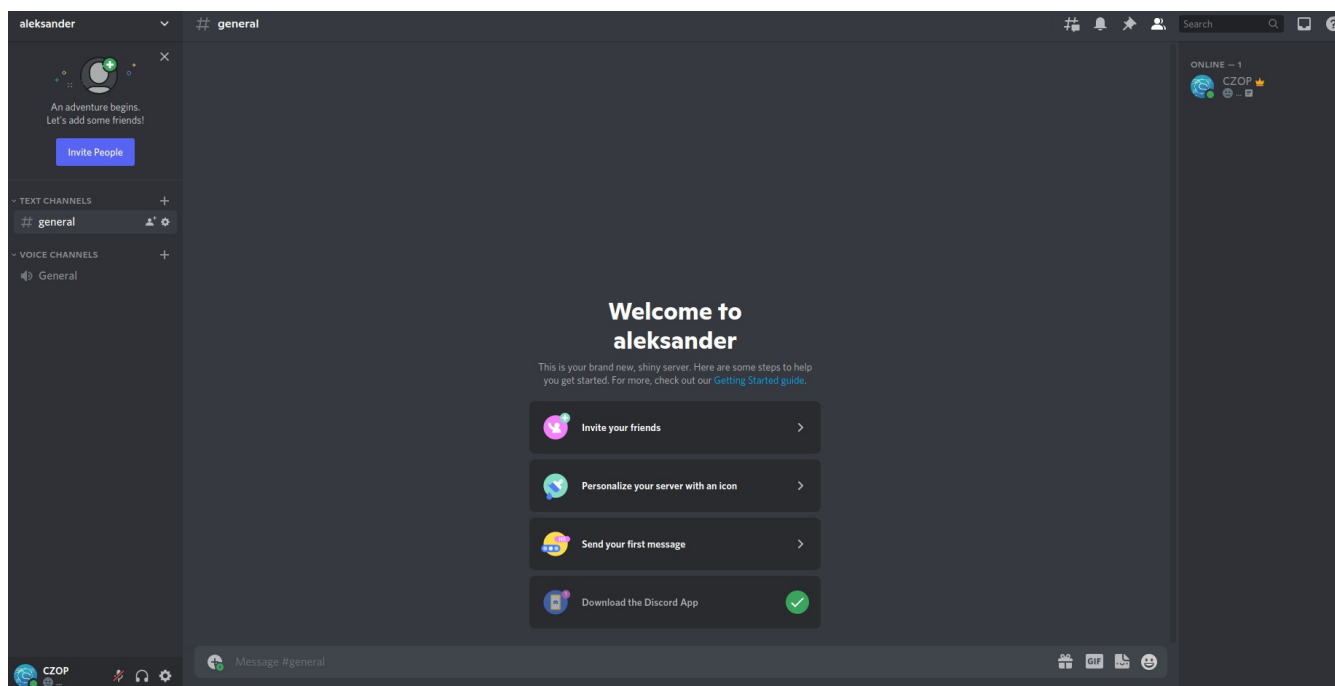


Slika 23: Kreiranje strežnika

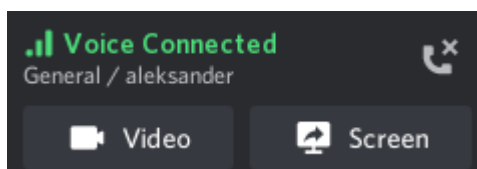
3.2 Funkcionalnosti

Znotraj sveže kreiranega strežnika lahko opazimo en kanal za pogovarjanje “general” in en pogovoren kanal prav tako poimenovan “general”. Opazimo lahko tudi na sredini pogovornega okna štiri pomagajoče gumbе. Prvi gumb je gumb za povabljanje nove prijatelje oziroma sodelavce. Povabimo jih preko url naslova oziroma jim pošljesh kar preko discorda, če jih imaš dodane pod prijatelje. Drugi je nastavitev slike, če je še slučajno nismo dodali pri kreiranju strežnika in pomoč za pošiljanje prvega sporočila. Kreirali

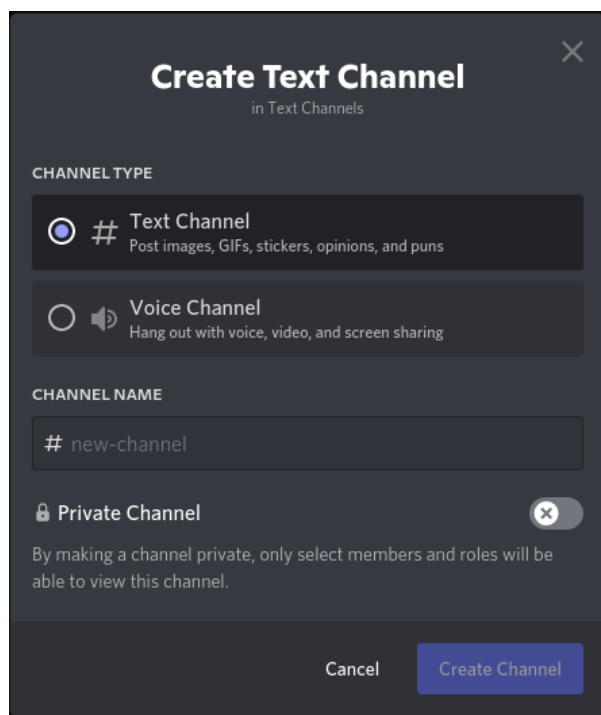
bomo novi kanal in ga poljubno poimenovali. V našem primeru smo kreirali kanal Nov, namenjen za besedilno pogovarjanje. Za glasovno pogovarjanje je postopek enak, le da spremenimo funkcijo kanala. Če imamo razdeljene vloge v strežniku pa lahko kreiramo privatni kanal in dodelimo pravice posameznim vlogam. V kanalu Nov sedaj napišemo sporočilo s klikom na spodnje vnosno polje in nekakšnim tekstom. Lahko se vključimo v zvočni klic s pritiskom na zbočnik plus ime kanala in začnemo oddajati naš zaslon vsem udeležencem v klicu. Kliknemo gumb Zaslon v levem spodnjem kotu in izberemo željeno deljeno vsebino. Če želimo končati klic pa pritisnemo gumb, ki zglada kot telefon s križcem. Če želimo upravljati z nastavitvami strežnika pa gremo v levi zgornji kot in izberemo nastavitve. Odpre se Pregled strežnika, ki nam omogoča vse mogoče funkcionalnosti, med njimi tudi zapustitev strežnika in izbris.



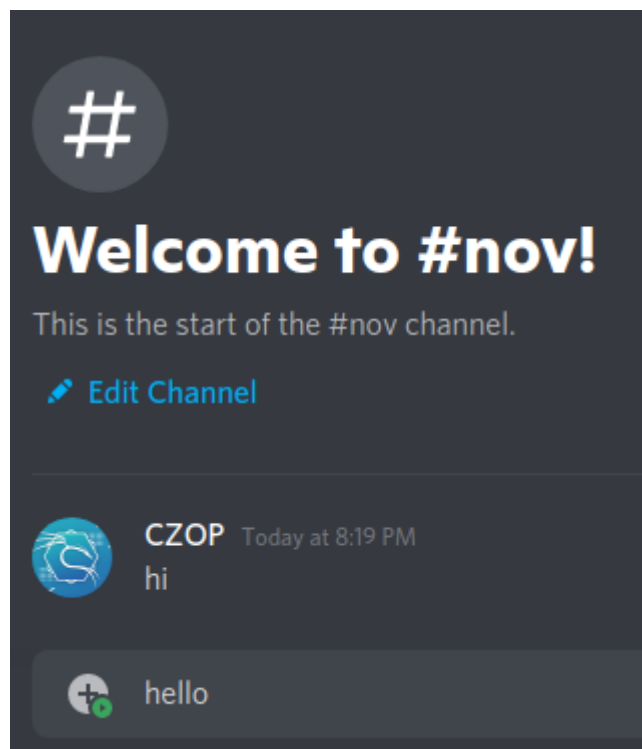
Slika 25: Osnovna stran strežnika



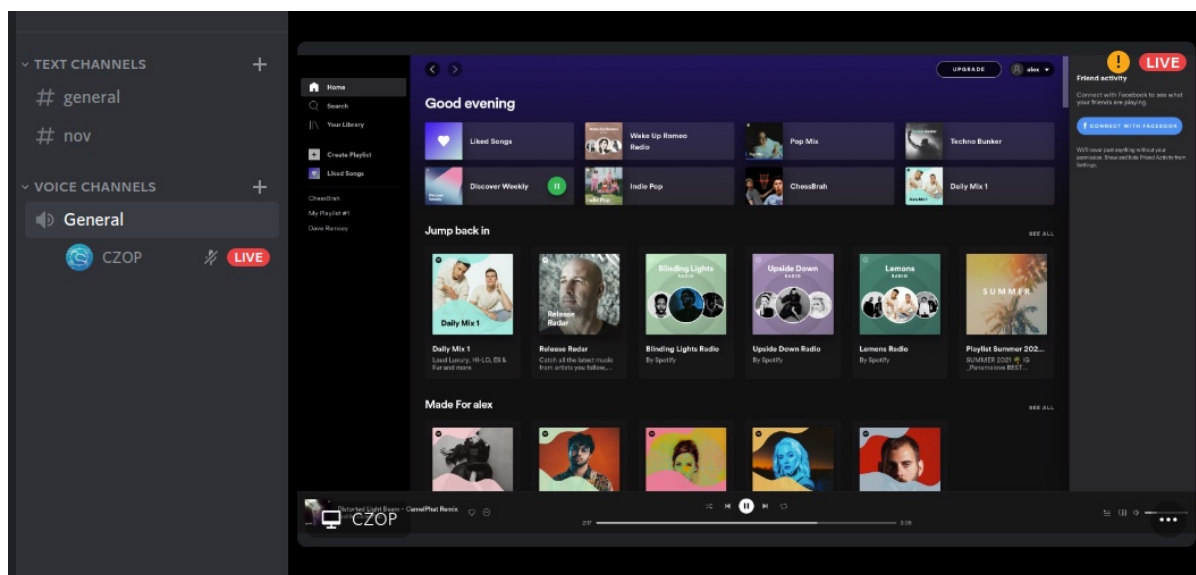
Slika 26: Klic v glasovnem kanalu



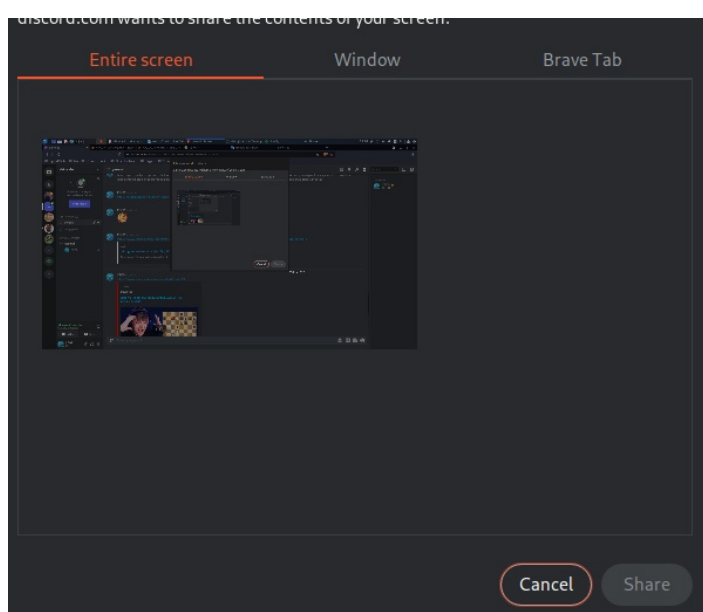
Slika 28: Kreiranje novega kanala



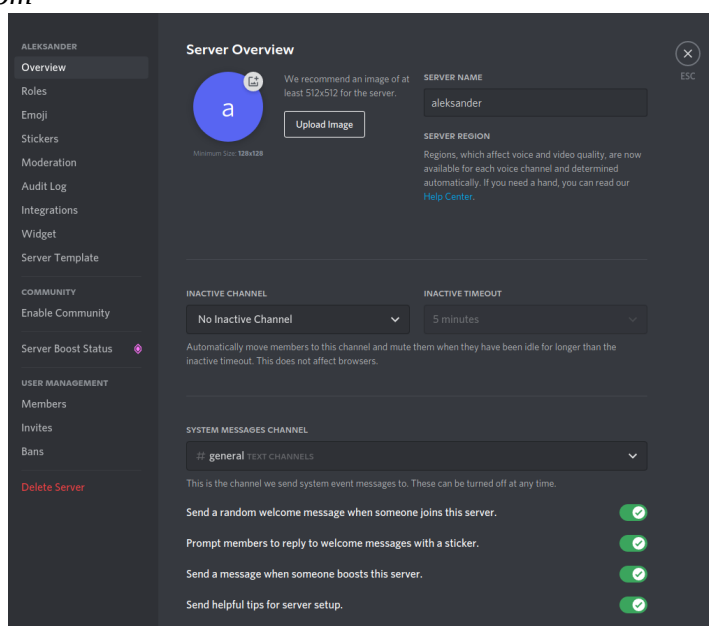
Slika 27: Kanal Nov



Slika 29: Primer prikazovanja zaslona med klikom



Slika 31: Deljenje zaslona

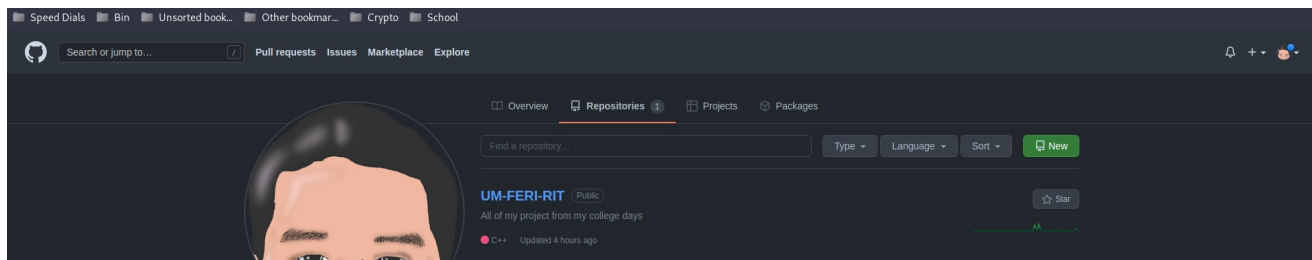


Slika 30: Pregled strežnika

4 Github

4.1 Kreiranje projekta

Ustrezno se prijavimo v spletno storitev Github. Po uspešni prijavi gremo v podokno Repozitoriji in kliknemo zelen gumb Novo. Odpre se podokno, kjer izpolnimo ime, opis, vidnost in poljubne opcijske dodatke. Projekt je ustvarjen, vendar je še popolnoma prazen. Kodo bomo dodajali pri GIT odjemalcih.



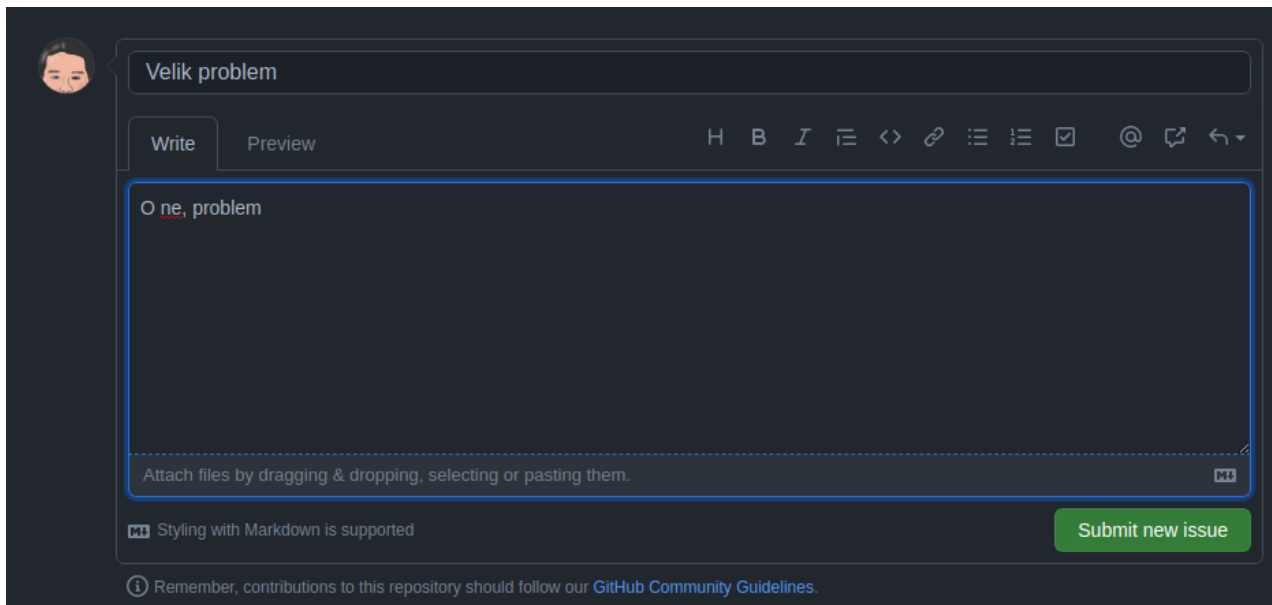
Slika 32: Začetna stran githuba

A screenshot of the 'Create a new repository' form on GitHub. The form is titled 'Create a new repository' and has a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' section shows a dropdown menu with 'profek10' selected. The 'Repository name' section shows a text input with 'test' and a green checkmark. Below these sections, there's a 'Description (optional)' text input. The 'Public' option is selected with a radio button, and the 'Private' option is unselected. Below the visibility options, there's a section 'Initialize this repository with:' with three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each checkbox has a description and a 'Learn more' link. At the bottom of the form, there's a green 'Create repository' button.

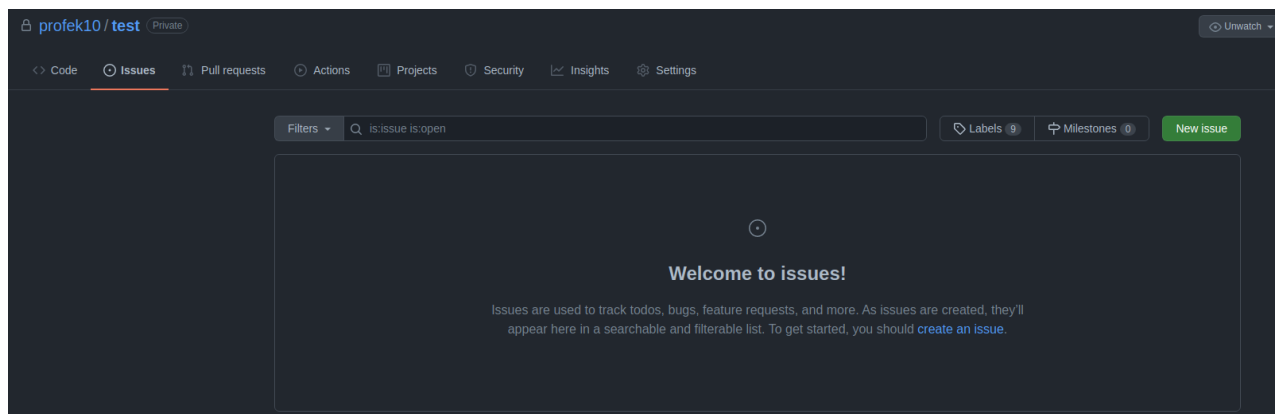
Slika 33: Kreiranje repozitorija

4.2 Dodajanje funkcionalnosti

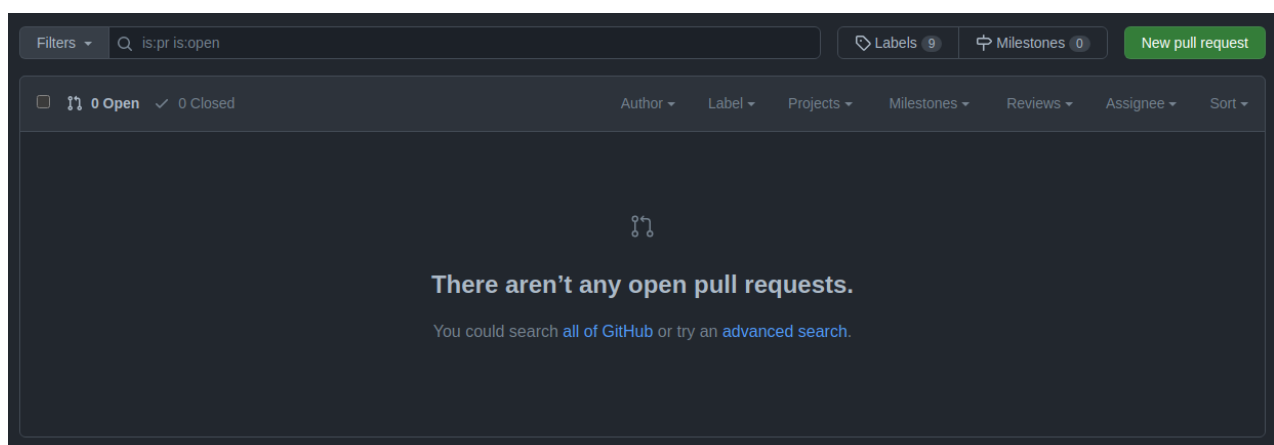
V podzavitku Problemi lahko kreiramo nov problem s pritiskom na Nov problem želimo kreirati nov projekt kliknemo “nov projekt”. Problem opišemo in dodamo naslov. Namen teh problemov je izpostavljjanje napak v programu in morebitih rešitvah. Ko smo se pogovorili o težavi jo zaključimo. Ustvarimo lahko tudi Vlečno zahtevo, ki lahko našo kodo potegnemo iz lokalnega repozitorija. Več pri GIT odjemalcih. V zavitku Nastavitve pa lahko opazimo in spreminjamo lastnosti kreiranega projekta. Tukaj lahko tudi brišemo projekt in mu spreminjamo vidljivost.



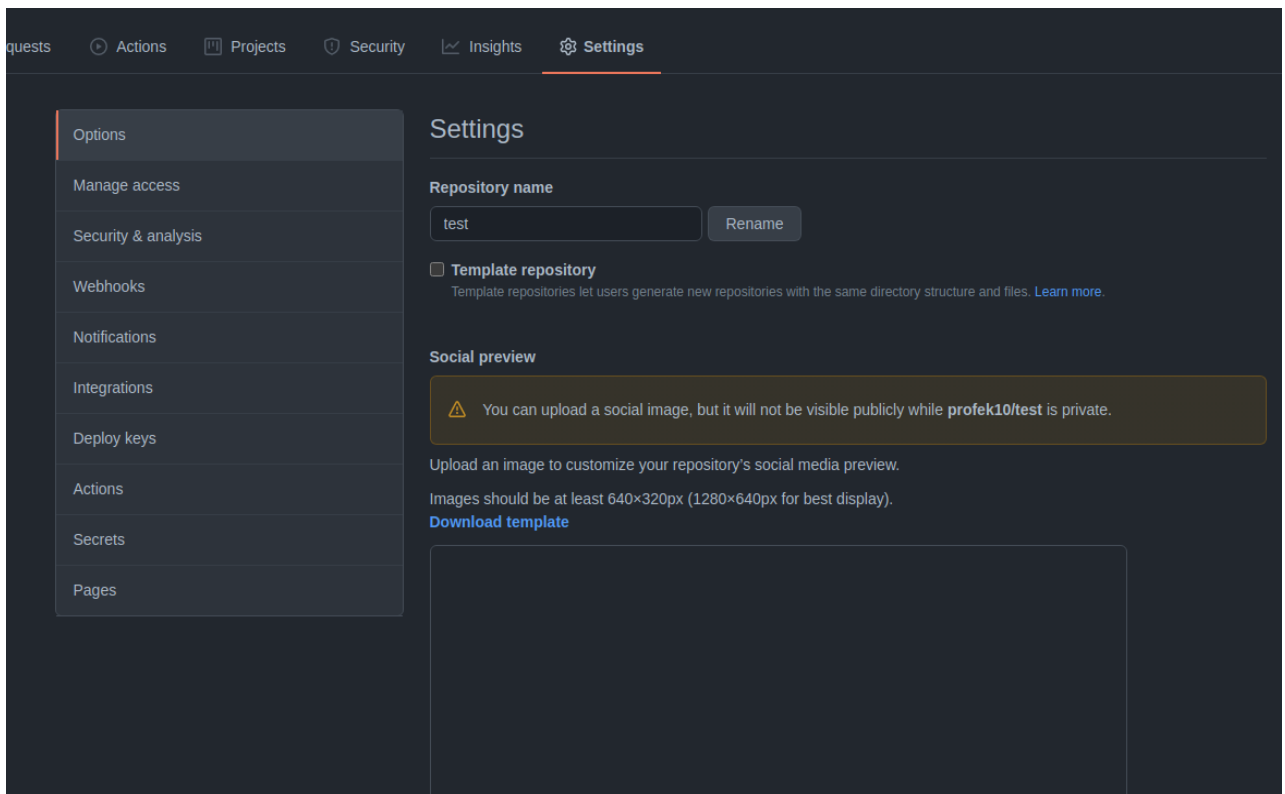
Slika 35: Dodajanje problema



Slika 36: Zavitek s problemi



Slika 34: Vlečna zahteva



Slika 37: Nastavitve github

5 Git odjemalec

5.1 Inštalacija

Na terminalnih, katerega uporabljam tudi jaz, je postopek sorazmerno hiter. Najprej pogledamo, če že imamo git klient inštaliran. Pogledamo z ukazom `git --help` in naložimo, če slučajno nimamo. Na linux debianih je to ukaz `sudo apt-get install git`. Zaženemo ukaz `git init`, da inicializiramo nov projekt, ki ga nato z ukazi „`git add <ime datotek>`“ in „`git commit -m „<sporočilo>“`“ dodamo na lokalni git repozitorij. Sedaj imamo pripravljene datoteke za push na git. Z naslednjim ukazom uredimo git strežnik, da bo Git vedel kam naj da datoteke. Ukaz: „`git remote add origin git@github.com:<User>/<UserRepo>.git`“. Seveda spremenimo vse lastnosti v manjše, večje znakovne oziroma drug ukaz za druge Git strežnike (primer je dan za GitHub). Poženemo še zadnji ukaz „`git push origin master`“ in smo uspešno dodali prvi projekt.

```
(aleks@aleks)-[~/Desktop/UM-FERI-RIT]
$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.33.0-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Slika 38: git install

```
$ git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
       [--exec-path<path>] [--html-path] [--man-path] [--info-path]
       [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
       [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
       [--super-prefix<path>] [--config-env<name>=<envvar>]
       <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index
  sparse-checkout  Initialize and modify the sparse-checkout

See 'git help' for more details.
```

Slika 39: git check

5.2 Primer uporabe

Preden začnemo uporabljati projekt je treba ustvariti ključ, ki ga bomo od sedaj naprej uporabljali za avtentikacijo med git odjemalcem in githubom. Kreiramo ključ s spodnjo komando, ki generira RSA ključ z dodanim geslom za zaščito. V githubu poskrbimo, da je dodan ime.pub ključ. Izvršimo še komando, ki doda ssh ključ z geslom, da lahko naslednjič napišemo samo golo komando. Sedaj želimo klonirati naš projekt zato napišemo komando git clone. Git nato uporabljamo s spodnjimi komandami git status/log/pull. Če želimo pa dodati datoteko nek.txt na github pa storimo slednje ukaze. Dodamo nov file z ukazom git add, dodamo sporočilo z ukazom git commit -m „sporočilo“ ter pošljemo na strežnik z ukazom git push. Reševanje napak med zavezami pa so težji problemi, ki se najpogosteje (ni edina) rešujejo s kreiranjem novih vej s slednjim ukazom git branch „ime“.

```
└─$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aleks/.ssh/id_rsa): /home/aleks/.ssh/profek10.github
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aleks/.ssh/profek10.github
Your public key has been saved in /home/aleks/.ssh/profek10.github.pub
The key fingerprint is:
SHA256:M0AKV105+Kd/L5szz4PpRUAZiTrhXV5IUTCNujWWqKA aleks@aleks
The key's randomart image is:
+---[RSA 3072]-----+
| . . . . . oB@. |
| o o .o ..*.o |
| . .o = +.o |
|      E+ = o *. |
| . .S. = .. |
| TROU . .o+ . . |
| . +   o. |
|      o  B+. |
|      ..o+B+. |
+---[SHA256]-----+
```

Slika 40: ssh-keygen

```
└─$ ssh-add .ssh/profek10.github
Enter passphrase for .ssh/profek10.github:
Identity added: .ssh/profek10.github (aleks@aleks)
```

Slika 41: ssh-add

```
(aleks@aleks) - [~/Desktop/UM-FERI-RIT/test]
└─$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

(aleks@aleks) - [~/Desktop/UM-FERI-RIT/test]
└─$ git log
commit 350e1af6fb8182f9f4370ee6e5f399e2a2c5b00e (HEAD → main, origin/main, origin/HEAD)
Author: Aleksander <aleksander.grobelnik1@gmail.com>
Date:   Wed Oct 27 22:17:19 2021 +0200

    Create README.md

(aleks@aleks) - [~/Desktop/UM-FERI-RIT/test]
└─$ git pull
Already up to date.
```

Slika 42: Glavni git ukazi


```

(aleks@aleks)-[~/Desktop/UM-FERI-RIT]
$ git clone git@github.com:profek10/test.git
Cloning into 'test' ...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

```

Slika 43: git clone

```

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ ls
nek.txt  README.md

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  nek.txt

nothing added to commit but untracked files present (use "git add" to track)

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git add nek.txt

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git commit -m "Added file nek.txt"
[main 308cd94] Added file nek.txt
1 file changed, 1 insertion(+)
create mode 100644 nek.txt

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git push
Warning: Permanently added the RSA host key for IP address '140.82.112.4' to the list of known hosts.
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 290 bytes | 290.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:profek10/test.git
 350e1af..308cd94  main -> main

```

Slika 44: git commit

The screenshot shows a GitHub commit page for the repository 'profek10/test'. The commit message is 'Added file nek.txt' and it was committed 2 minutes ago by user 'profek10'. The commit hash is 308cd946483388122bd2f2d4817ff0affbd6dc03. The page shows that 1 file was changed with 1 addition and 0 deletions. The diff view shows the addition of a file named 'neka.j'.

Slika 45: vidnost na githubu

```

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git branch veja

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git log
commit 308cd946483388122bd2f2d4817ff0affbd6dc03 (HEAD -> main, origin/main, origin/HEAD, veja)
Author: profek10 <aleksander.grobelnik1@gmail.com>
Date: Wed Oct 27 22:22:37 2021 +0200

    Added file nek.txt

commit 350e1af6fb8182f9f4370ee6e5f399e2a2c5b00e
Author: Aleksander <aleksander.grobelnik1@gmail.com>
Date: Wed Oct 27 22:17:19 2021 +0200

    Create README.md

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git switch veja
Switched to branch 'veja'

(aleks@aleks)-[~/Desktop/UM-FERI-RIT/test]
$ git status
On branch veja
nothing to commit, working tree clean

```

Slika 46: git branch