## Univerzitet u Kragujevcu

## Fakultet Tehničkih nauka u Čačku



## Smer:Informacione Tehnologije Predmet:Softversko Inženjerstvo

**Git na primeru – lokalne i remote grane, primer projekta i rada više članova u timu**

## Profesor: dr. Marija Blagojević Studenti:

## Asistenti:Nemanja Vićović, Milan Miletić 53/2018

## Mladen Janjić Igor Tanović 54/2018

## Čačak, 2021.godine

## Sadržaj

1. [Uvod 3](#_bookmark0)
2. [Grananje u gitu 4](#_bookmark1)
3. [Lokane grane 5](#_bookmark2)
4. [Remote grane 9](#_bookmark3)

5.[Primer projekta 12](#_bookmark4)

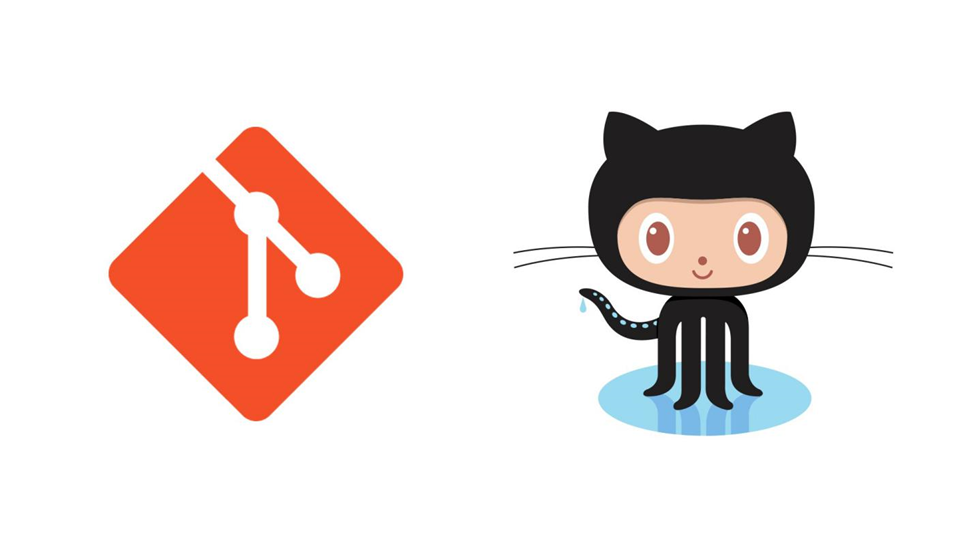
6.[Zaljučak 13](#_bookmark5)

7.[Literatura 14](#_bookmark6)

# UVOD

Git je sistem za verzionisanje koji se koristi za praćenje promena u računarskim fajlovima i koordinisanje rada više ljudi na tim fajlovima. On se prvenstveno koristi za upravljanje izvornim kodom u razvoju softvera, ali može biti korišćen za praćenje promena u bilo kojem kompletu fajlova. Kao distribuirana kontrola revizije, usmeren je na brzinu, celovitost podataka i podršku za distribuirane, nelinearne radne tokove (workflows).

**Ukratko:** Git prati promene fajlova na kojima radite i onemogućava da se obriše ili promeni tuđi kod, odnosno da se on "pregazi". Izmedju ostalog omogucava da se kreiraju lokalni repozitorijumi, klonirati postojece sa nekog servera, commitovanje (cuvanje promena u lokalnom repozitorijumu), itd...



# GRANANJE U GITU

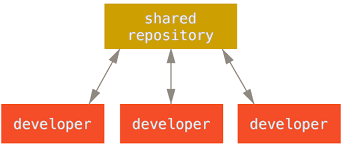
Skoro svaki VCS ima neku vrstu podrške za grananje. Grananje znači da se divergira od glavne linije razvitka programa i da se nastavi rad bez uticaja na tu glavnu liniju. U mnogim VCS alatima, ovo je nešto skuplji proces, jer se često zahteva pravljenje nove kopije direktorijuma sa izvornim kodom, što zna da potraje za veće projekte.

Neki ljudi Gitov model grananja zovu prvoklasnom osobinom, i zaista, mogućnost koje nudi odvaja Git od ostalih VCS-ova. Ali po čemu je poseban? Način na koji se Git grana je neverovatno jednostavan za obradu, što čini da se operacije grananja izvrše skoro u istom trenutku kada se zadaju, a skakanje s jedne na drugu granu je podjednako brzo. Za razliku od mnogih drugih VCS-ova, Git ohrabruje korišćenje tokova rada koji se često granaju i spajaju, čak i nekoliko puta tokom jednog dana. Razumevanje i savladavanje ove tehnike predstavljaće snažno i unikatno oruđe koje bi u potpunosti moglo da promeni način na koji razvijate svoj proizvod.

Da bismo stvarno razumeli kako Git barata grananjem, moramo da se odmaknemo korak unazad i da istražimo kako Git čuva podatke.

Git ne čuva podatke kao seriju skupa promena ili razlika, već kao seriju snimaka.

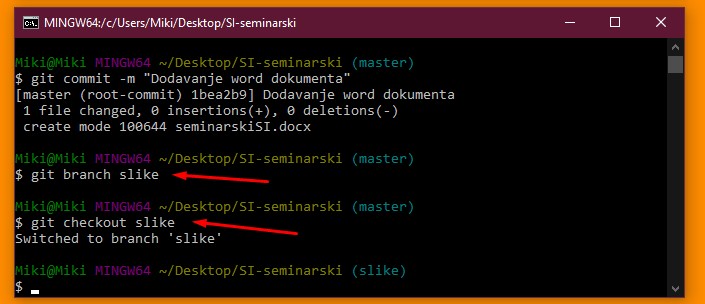
Kada napravite komit, Git čuva komit-objekat koji sadrži pokazivač na snimak objekta koji je stejdžovan. Ovaj objekat sadrži i autorovo ime i mejl, poruku koja je ukucana, i pokazivače na komit ili komitove koji su direktno došli pre ovog komita (roditelja ili roditelje): nula roditelja na inicijalni komit, jednog roditelja za normalni komit, i više roditelja za komit koji je rezultat spajanja dveju ili više grana.



**Slika 1.** Grananje

# LOKALNE GRANE

Lokalne grane su grane koje samo lokalni korisnik moze da vidi i one postoje samo na njegovoj lokalnoj mašini.



**Slika 2.** Kreiranje lokalne grane

# REMOTE GRANE

Remote grana je grana na udaljenoj lokaciji. Novu remote granu povezemo sa udaljenom lokacijom i tada svi ostali korisnici mogu da je prate.

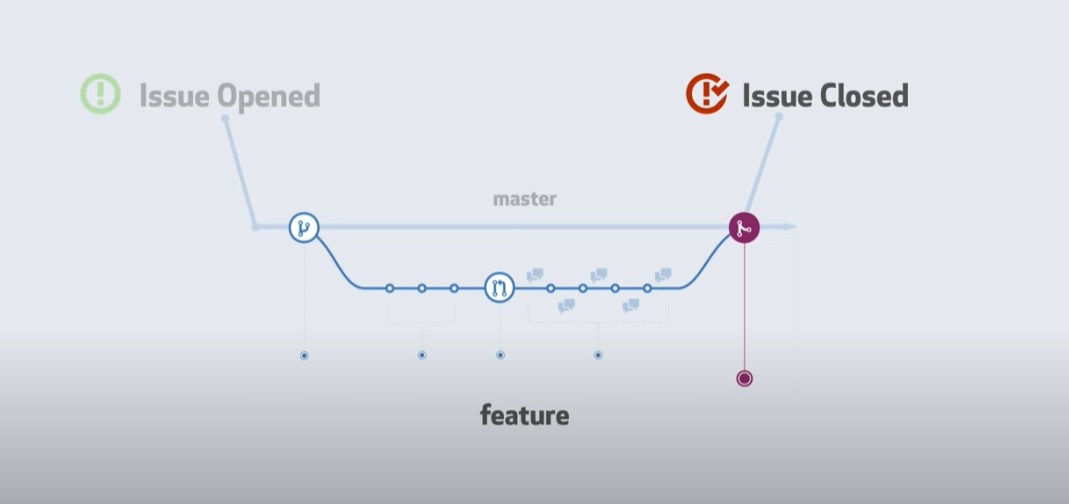
Remote tracking grana je lokalna kopija remote grane.

Udaljene grane su reference (pokazivači) u vašim udaljenim repozitorijumima, uključujući grane, tagove, i tako dalje. Možete da pogledate celu listu udaljenih referenci eksplicitno sa git ls-remote <remote>, ili git remote show <remote> za udaljene grane kao i za više informacija. Ipak, češći način je korišćenje udaljenih praćenih grana (**remote-tracking branches**).

Udaljene praćene grane su reference na stanja udaljenih grana. One su lokalne reference koje ne možete da pomerite; one se pomeraju automatski kada radite bilo kakav vid komunikacije preko mreže. Udaljene praćene grane rade kao obeleživači koji treba da vas podsete gde su grane u vašim repozitorijumima na daljinu bile kada ste se poslednji put povezali sa njima.

Uzimaju formu (remote)/(branch). Na primer, ako želite da pogledate kako je master grana na origin udaljenom repozitorijumu izgledala poslednji put kada ste stupili u komunikaciju sa njom, proverili biste origin/master granu. Ako ste radili na tiketu sa partnerom koji je slao izmene na grani iss53, vi imate svoju lokalnu granu iss53; ali grana na serveru pokazuje na komit origin/iss53.

Ovo je možda malo zbunjujuće, zato hajde da pogledamo primer. Recimo da imate Git server na svojoj mreži na git.ourcompany.com. Ako klonirate odavde, Gitova komanda clone će to automatski nazvati origin, povući će sve podatke, napraviće pokazivač na mesto gde je master grana i nazvati je origin/master lokalno. Git vam takođe daje vlastitu master granu koja počinje na istom mestu kao i origin-ova master grana, tako da imate odakle da krenete da radite.



**Slika 3.** Grafički prikaz remote grane

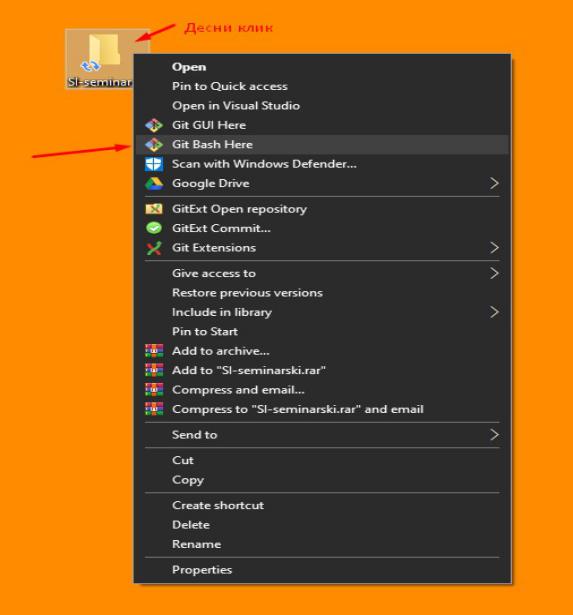
# 5. PRIMER PROJEKTA

Prvo se kreira folder koji ce biti git repozitorijum ili se koristi vec postojeci folder.

****

**Slika 4.** Kreiranje foldera

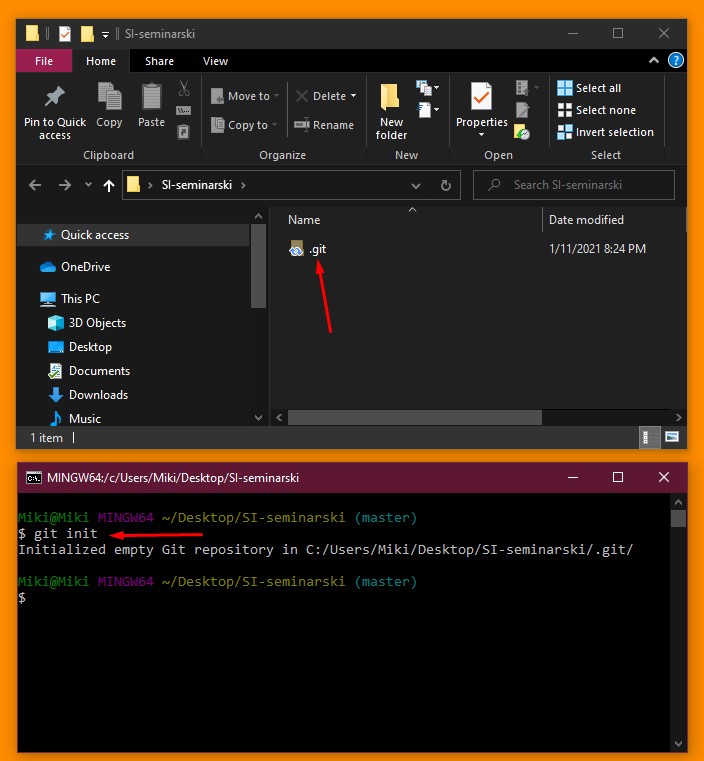
Desnim klikom na folder izabrati opciju *Git Bash Here* koja ce iskočiti samo ako je instaliran Git.



**Slika 5.** Biranje opcije Git Bash Here

Sledece sto treba uraditi jeste da se komandom *$* *git init* taj folder učini Git repozitorijumom.

Ako je potrebno napraviti novi folder koji je Git repozitorijum to se može učiniti komandom *$ git init noviRepozitorijum*.



**Slika 6.** Komanda *$ git init*

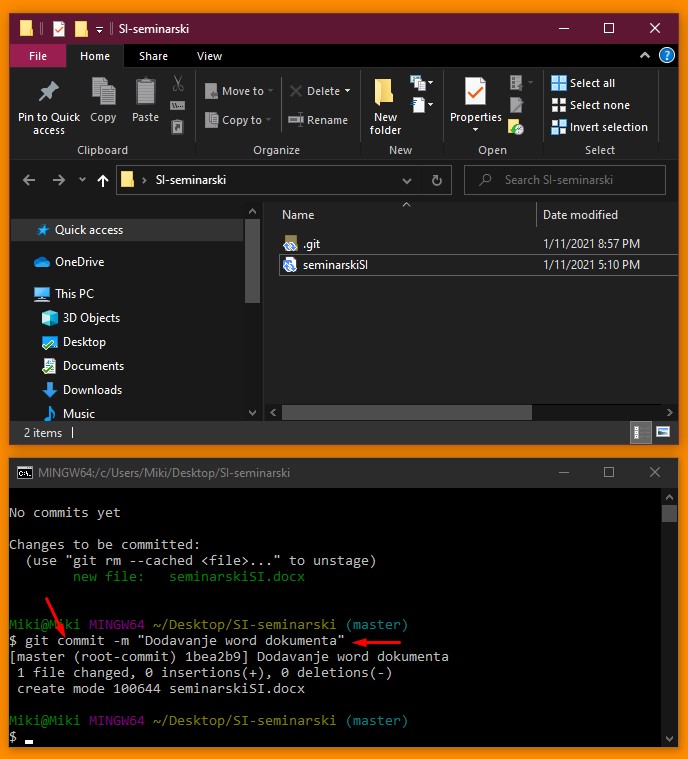
Komandom *$ touch seminarskiSI.docx* kreira se word dokument u folderu.

Sad postoji repozitorijum koje je prazan ali komandom *$ git add seminarskiSI.docx* dodajemo fajl*.* Da bi se sve dodalo koristi se komanda ,,*$ git add .”.*

gitAdd

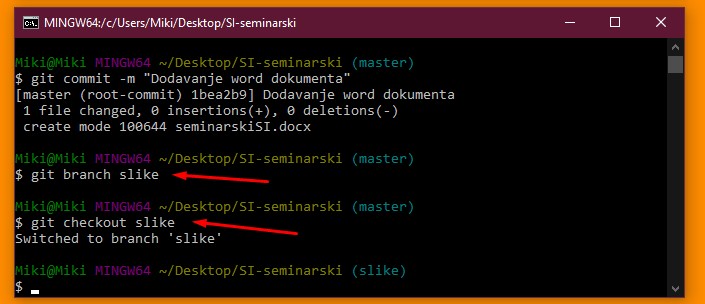
**Slika 7.** Komanda *$ git add*

Kada je fajl dodat, sada se tek skladište u Git repozitorijum. To se radi komitovanjem i za svaki komit trebalo bi da se napiše naziv.



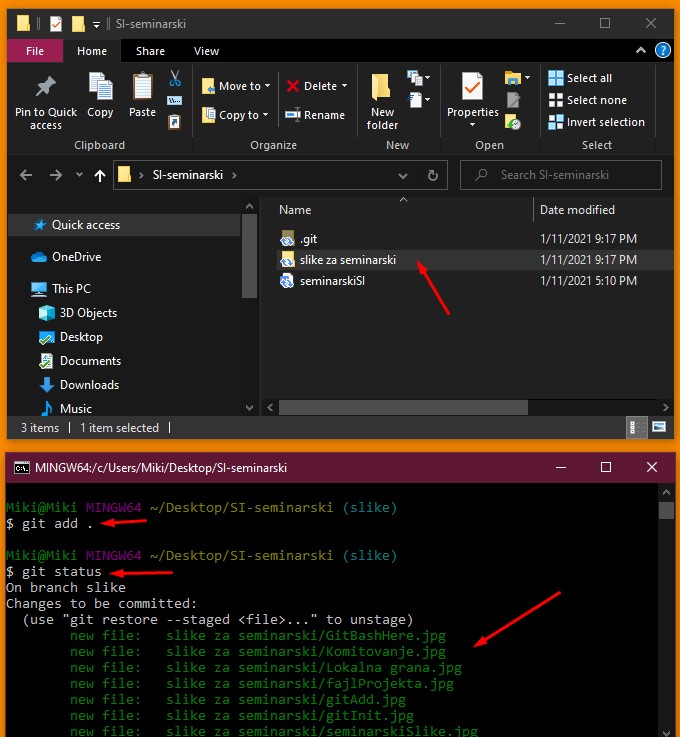
**Slika 8.** Komanda *$ git commit -m*

Ako je potrebno uraditi neku promenu ali ne baš sa glavnom granom onda može da se kreira nova lokalna grana komandom *$ git branch.* U ovom primeru je kreirana lokalna grana *slike.*



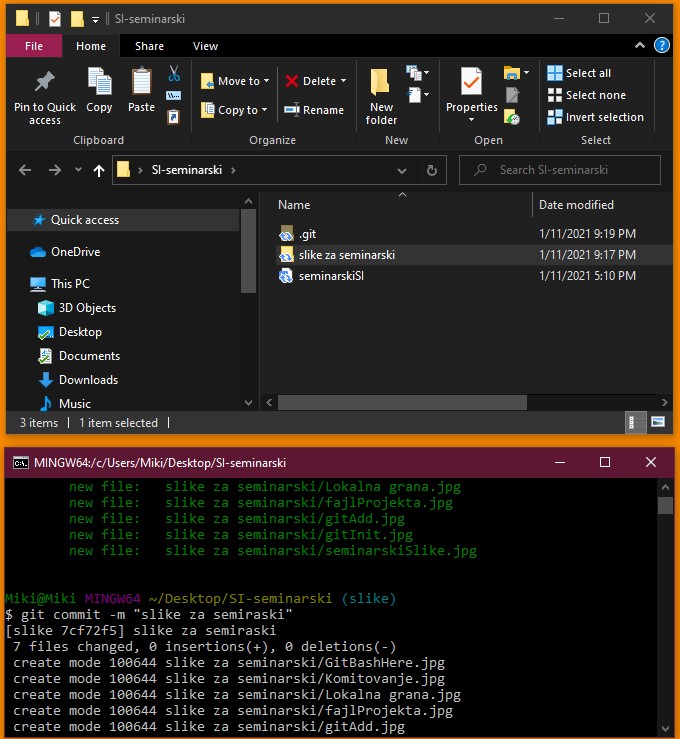
**Slika 9.** Kreiranje lokalne grane komandom *$ git branch*

Dodat je folder *slike za seminarski* i komandom *$ git status* se vidi da su prihvaćeni novi fajlovi za komitovanje.



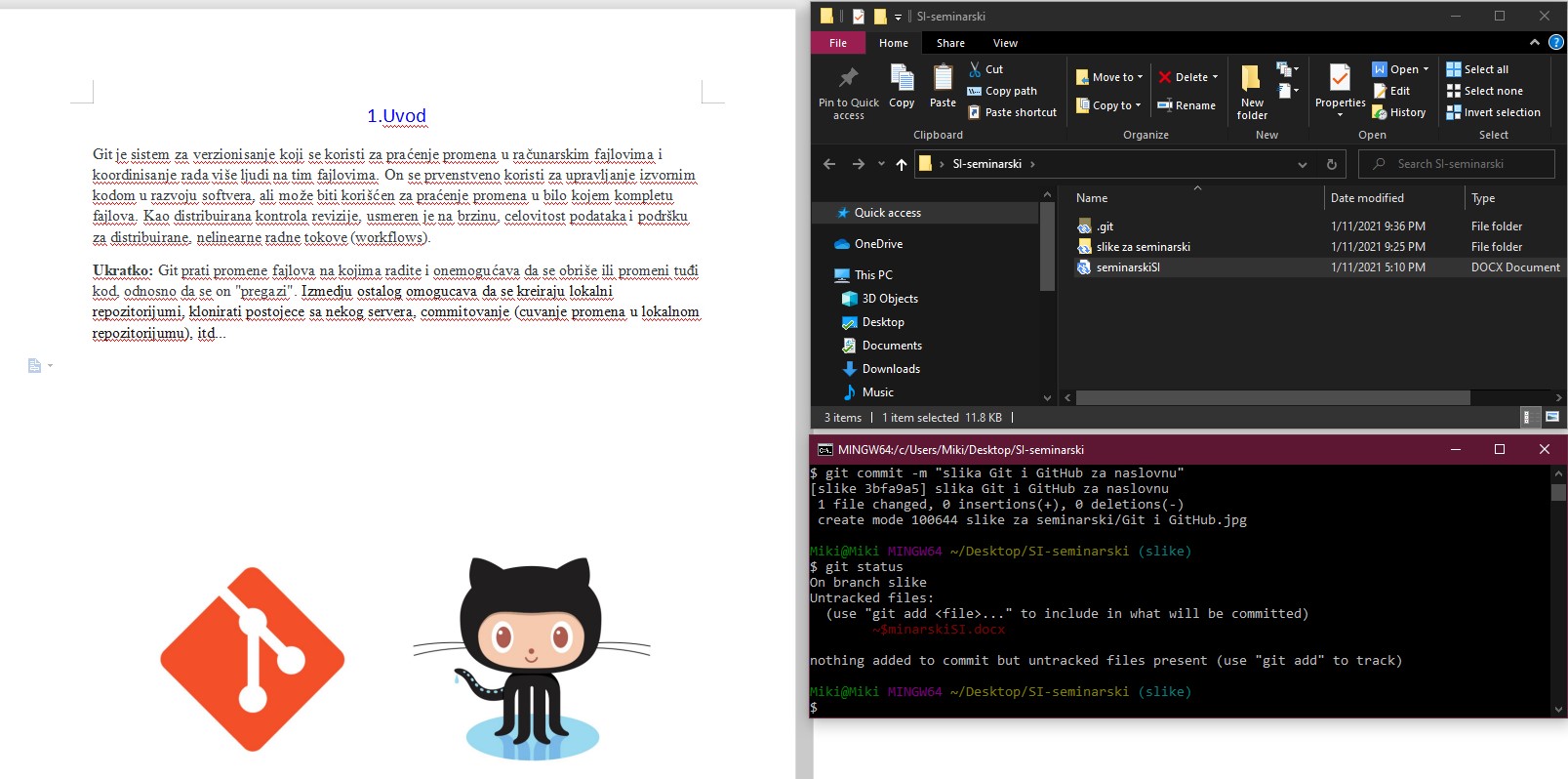
**Slika 10.** Dodavanje svih fajlova i komanda *$ git status*

Sada je folder *slike za seminarski* spreman da bude komitovan.



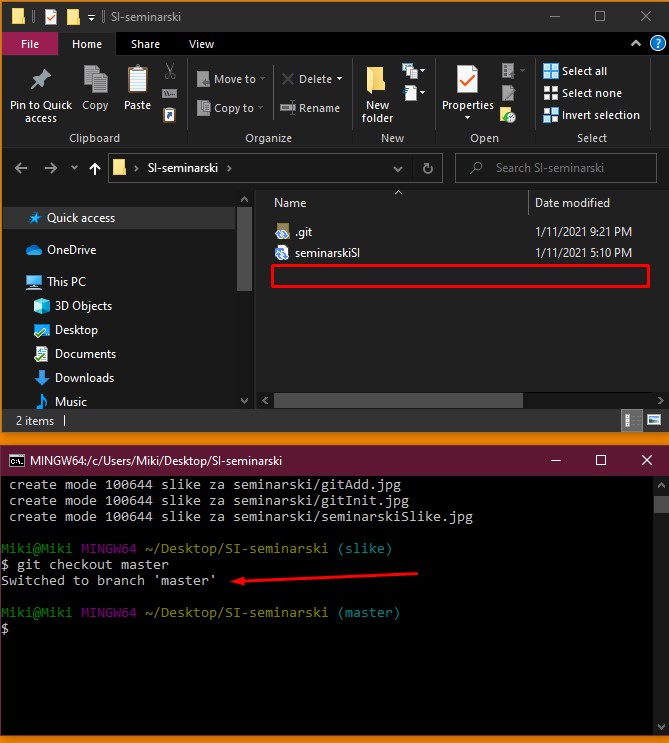
**Slika 11. Komitovanje foldera *slike za seminarski***

Pošto se sve ovo obavlja na grani slike, to znači da se može menjati i word dokument *seminarskiSI.docx* a da se na *master* ili *main* grani ne dešavaju nikakve promene, to i jeste svrha kreiranja grana. Tako da ako se u word dokumentu izmeni uvod to se neće odraziti na word dokument *seminarskiSI.docx* na *master* ili *main* grani.



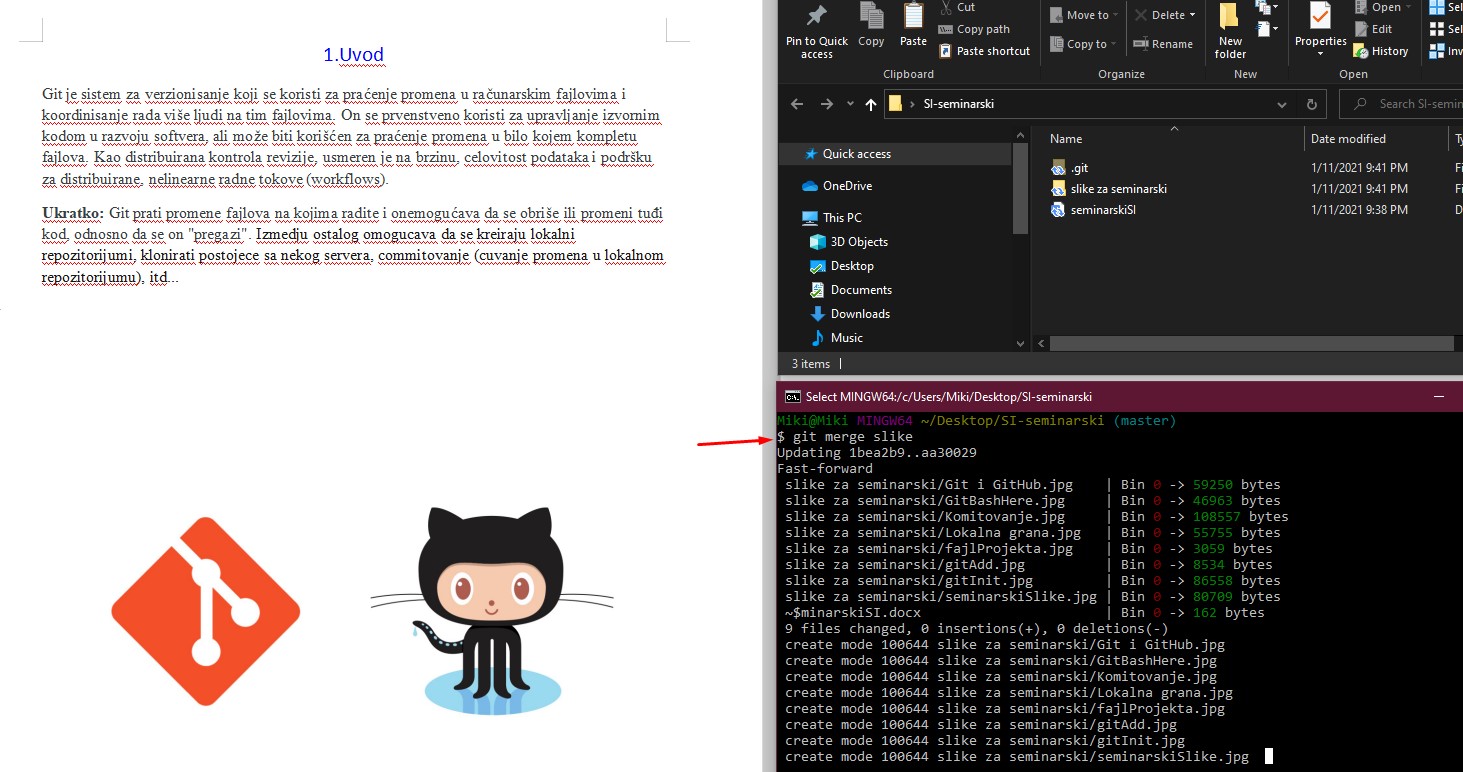
**Slika 12.** Uredjivanje uvoda na grani *slike*

Komandom *$ git checkout master* prebacuje se na master tj. glavnu granu na kojoj može da se vidi da nema foldera *slike za seminarski*, a takođe u word dokumentu *seminarskiSI.docx* uvod neće biti izmenjen.



**Slika13.** Prebacivanje na glavnu granu

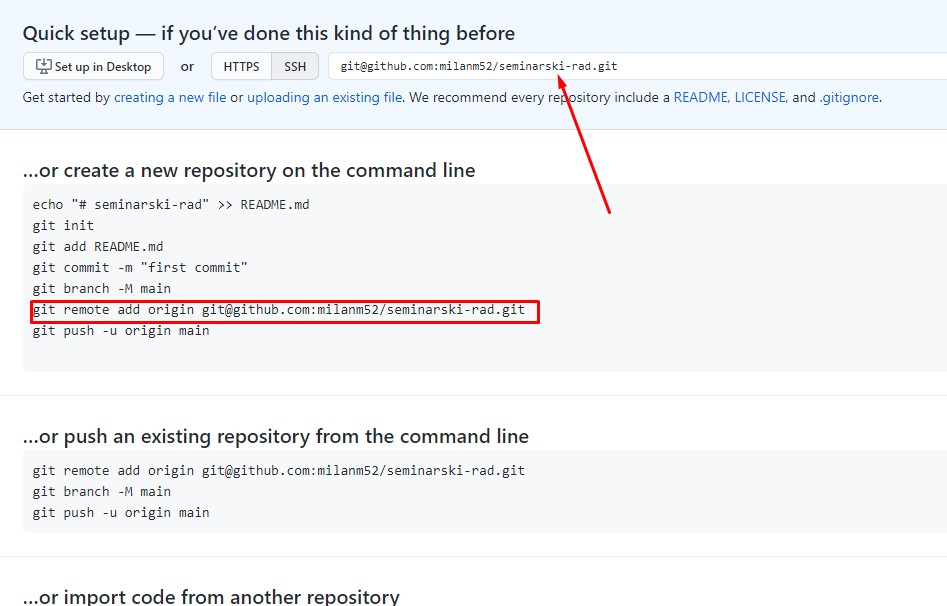
Kada je rad na drugim granama, u ovom slučaju na grani slike, završen i sagledan onda se vrši merdzovanje ili spajanje grana sa glavnom granom komandom *$ git merge slike.* Tada će se pojaviti folder *slike za seminarski* i uvod u wordovom dokumenti *seminarskiSI.docx* će biti uređen kao na grani *slike*.



**Slika 14.** Merge - spajanje grana

**Povezivanje sa GitHub-om**

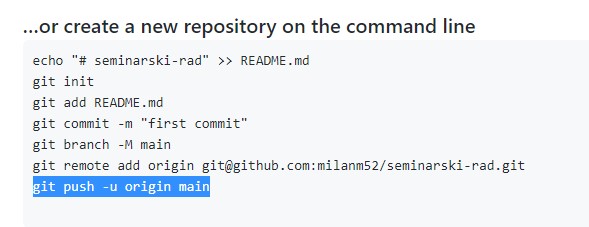
Na GitHub-u se kreira novi repozitorijum sa nazivom *seminarski-rad* gde dobijemo komande koje će možda zatrebati kao što je komanda *$ git remote add origin* posle koje ide link od kreiranog repozitorijuma na GitHub-u.



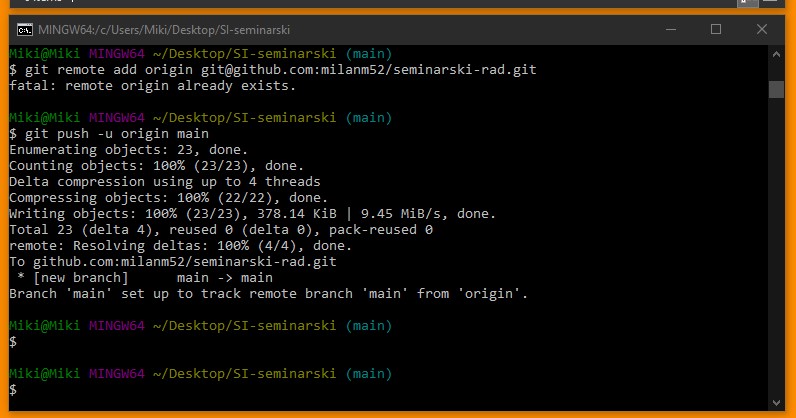
**Slika 15.** Kreiranje repozitorijuma na GitHub-u

Upisivanjem ove komande u konzolu lokalna mašina se povezuje sa repozitorijumu na GitHub-u.

Komandom ispod nje, *$ git push -u origin main* (master) repozitorijum na lokalnoj mašini se push-uje tj kopira u repozitorijum *seminarski-rad* na GitHub-u.

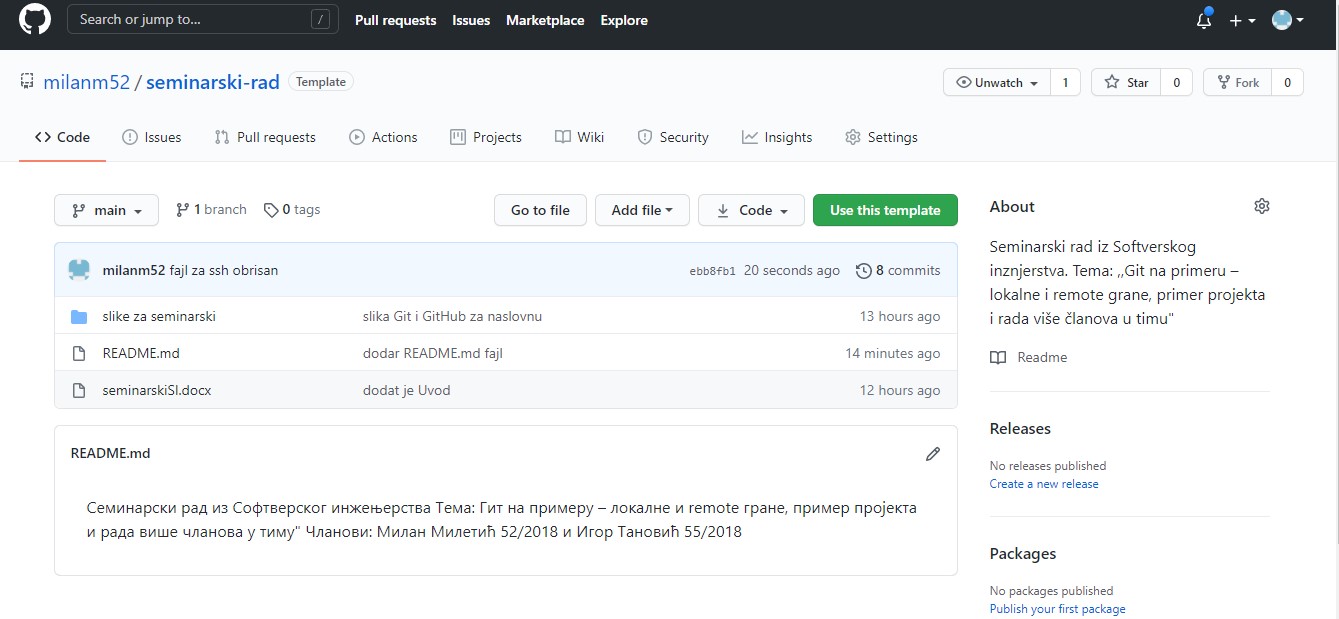


**Slika 16.** *Push* komanda



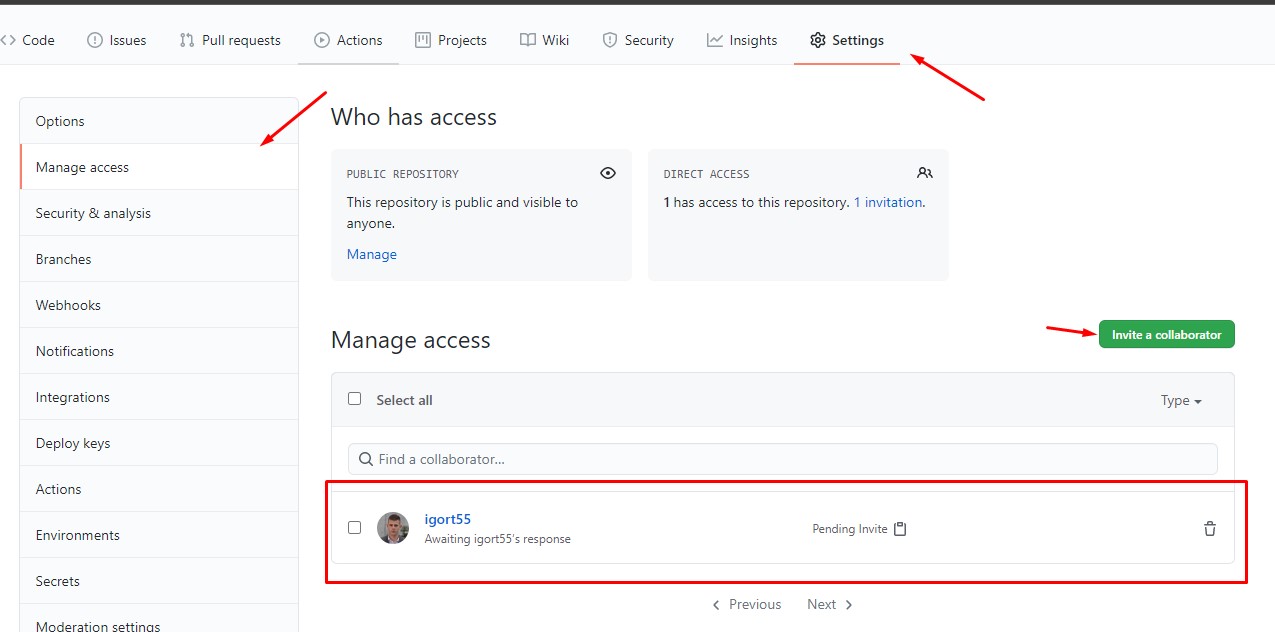
**Slika 18.** Push-ovanje na GitHub

I sada u repozitorijumu na GitHub-u se nalaze fajlovi iz repozitorijuma sa lokalne mašine.



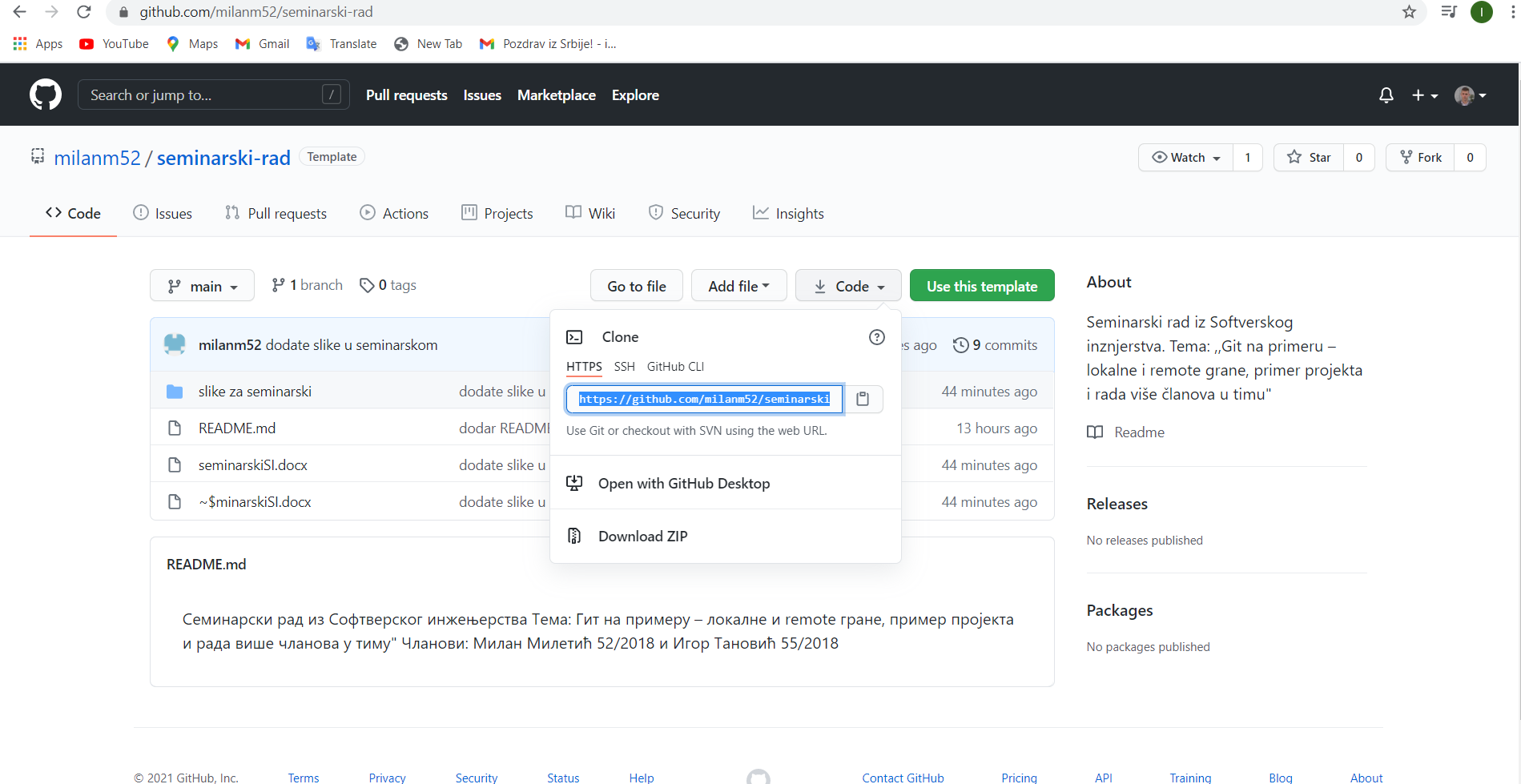
**Slika 19.** Repozitorijum na GitHub-u

Pošto je projekat *public* svi ga mogu videti. Klikom na **Settings** pa na **Manage access** vidimo ko sve ima pristup i takodje možemo da dodamo člana tima putem kolaboracije na dugme**Invite a collaborator**.

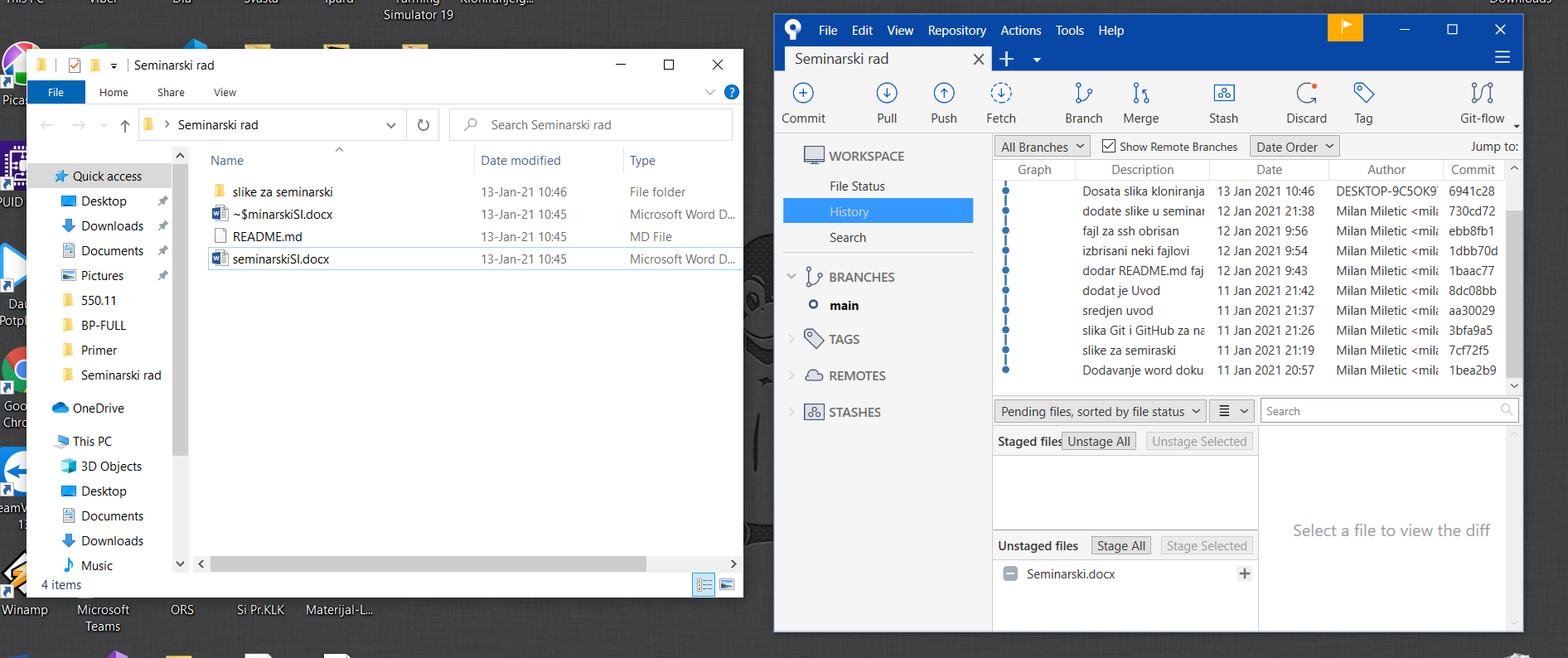


**Slika 20.** Kolaboracija

U sledećem koraku je prikaznano kako novi član klonira poijekat na svom racunaru.

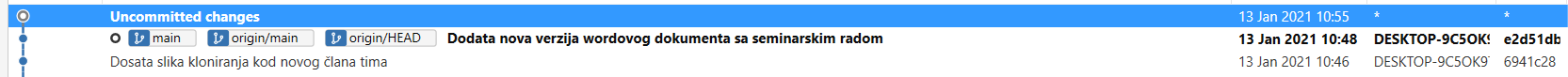


**Slika 21.**Git



**Slika22.** Kloniranje u novi folder na računaru novog korisnika

Na sledecoj slici su prikazani komitovi novog člana tima.



**Slika23.** Komit novog člana

6.ZAKLJUČAK

7.LITERATURA

1. https://www.atlassian.com/git/tutorials/syncing/git-fetch
2. https://git-scm.com/book/sr/v2/Grananje-u-Gitu-Grananje-ukratko/
3. <https://devconnected.com/how-to-push-git-branch-to-remote/>
4. <http://fantastic001.github.io/git/2017/07/04/git.html>