Kratki vodič za korišćenje git interfejsa

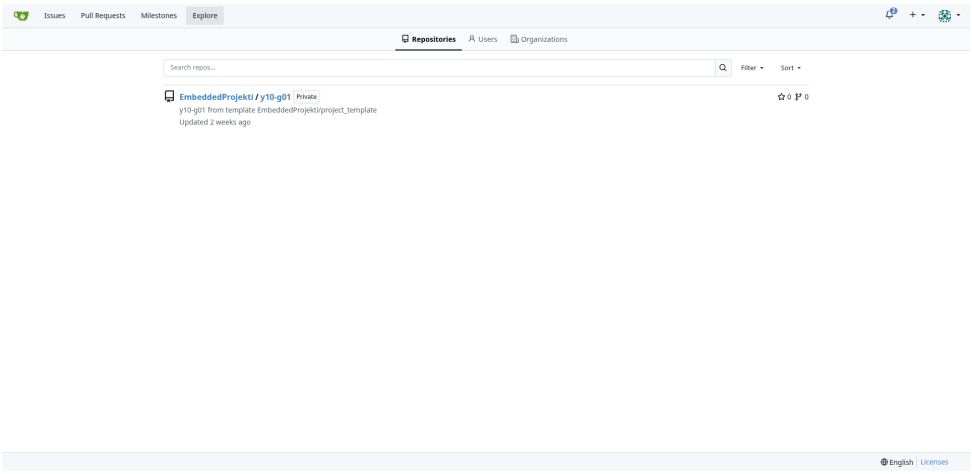
Ovaj vodič je namenjen studentima koji prvi put koriste git kao veb interfejs za rad sa Git repozitorijumima (projekat, domaći zadaci, laboratorijske vežbe). Fokus je na osnovnim git operacijama, radu sa granama `main`, `esl`, `rtl`, `fvh`, `eos`, Pull Request-ovima (PR) i prijavljivanju pitanja putem Issues.

1 Prijava i osnovni izgled git interfejsa

1. Otvorite adresu fakultetskog git servera u browseru:
   `embedded-elektronika.ftn.uns.ac.rs`
2. Ulogujte se sa korisničkim imenom i lozinkom koje ste dobili od nastavnika ili asistenta.

Posle prijave videćete glavnu stranicu sa listom repozitorijuma kojima imate pristup.
Glavni elementi su:

- gornji meni: pristup profilu, listi repozitorijuma, organizacijama;
- dugme "+" ili "New Repo" (ako imate pravo da kreirate repozitorijume);
- lista repozitorijuma: naziv projekta, opis, informacije o poslednjem commitu.



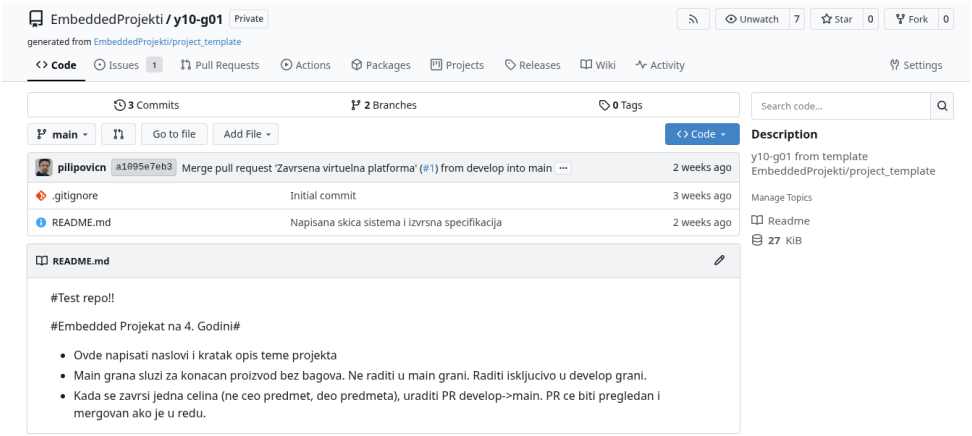
Slika 1: Početni ekran

Za rad na projektu dobijate pristup konkretnom repozitorijumu od nastavnika. Klikom na naziv repozitorijuma otvarate njegovu stranicu.

2 Pregled repozitorijuma i grane

Kada uđete u repozitorijum, videćete:

- listu fajlova i direktorijuma u repozitorijumu,
- padajući meni sa granama (npr. `main`, `esl`, `rtl`, `fvh`, `eos`),
- informacije o poslednjem commitu (poruka, autor, vreme),
- tabove: **Code**, **Issues**, **Pull Requests**, **Releases** itd.



Slika 2: Izgled repoa

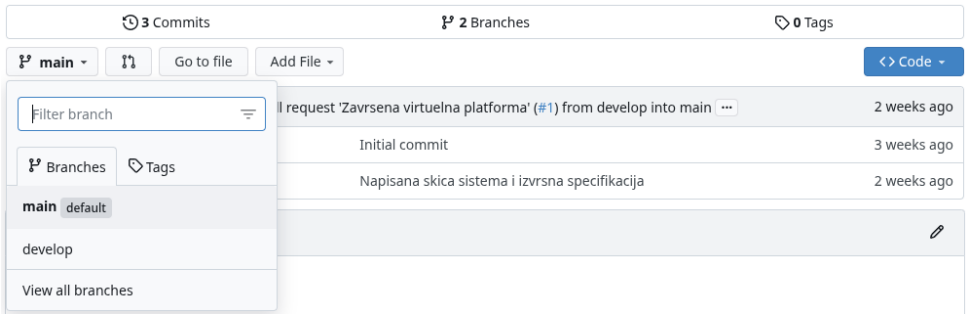
2.1 Grananje: `main` i radne grane

Tipičan scenario:

- `main` – stabilna verzija koda (ono "što radi" i ne možete direktno da menjate bez PR-a);
- `esl`, `rtl`, `fvh`, `eos` – radne grane, po jedna za svaki predmet, u kojima radite na projektu, dodajete svoje izmene, debagujete, pre nego što eventualno odu na `main`.

U web interfejsu izbor grane se radi iz branch padajućeg menija (obično piše `main` i ima malu strelicu).

1. Kliknite na padajući meni sa imenom grane.
2. Izaberite radnu granu (npr. `rtl`) da biste videli razvojnu verziju koda.
3. Vratite se na `main` na isti način kada želite.



Slika 3: Biranje grane

3 Kloniranje repozitorijuma i osnovne git komande

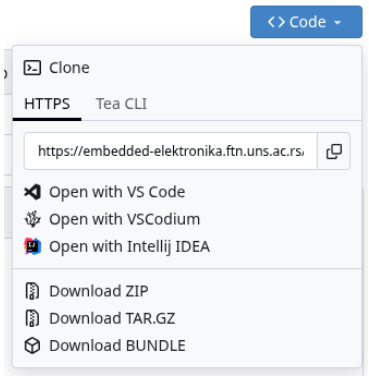
Za rad na kodu potrebno je da repozitorijum preuzmete lokalno koristeći Git (u terminalu ili preko VS Code / nekog GUI klijenta).

Na stranici repozitorijuma u tabu **Code** na vrhu liste fajlova postoji URL repozitorijuma (HTTPS/SSH) i dugme **Copy**.

3.1 Kloniranje repozitorijuma

U terminalu ukucajte:

```
git clone <URL_REPOZITORIJUMA>
cd <IME_PROJEKTA>
```



Slika 4: Clone meni

Primer:

```
git clone https://embedded-elektronika.ftn.uns.ac.rs/EmbeddedProjekti/y10-g01.git
cd y10-g01
```

Ovim ste preuzeli repozitorijum i ušli u lokalni folder.

3.2 Prebacivanje na radnu granu

Proverite koje grane postoje:

```
git branch -a
```

Prebacivanje na radnu granu, npr. `rtl`:

```
git checkout rtl
```

4 Tipičan radni tok za studente

Jednostavan scenario rada:

1. Ažurirate lokalnu kopiju `main` i radne grane.
2. Napravite nove izmene u radnoj grani (npr. `rtl`).
3. Napravite commit.
4. Gurnete (push) izmene na server.
5. Napravite Pull Request (nakon završenog nekog dela) za spajanje u `main`.

4.1 Ažuriranje lokalne kopije (pull)

Pre početka rada obično uradite:

```
git pull origin main      # Azurirate main granu
git checkout rtl          # Prelazite na radnu granu
git pull                  # povuci poslednje izmene rtl grane
```

Ovo osigurava da radite nad svežom verzijom koda.

4.2 Dodavanje izmena i commit

Posle izmena u kodu:

```
git status           # proveri koje su datoteke promenjene
git add .            # dodaj sve izmene (ili selektivno fajlove)
git commit -m "Kratak opis izmene"
```

Poruka commita treba da bude jasna (npr. `"Dodata funkcija za racunanje proseka"`).

4.3 Slanje izmena na server (push)

```
git push
```

Ako radite na grani `rtl`, izmene idu na udaljenu `rtl` granu na git serveru.
Ako kreirate novu granu (npr. `feature/vp_dma`), prvo je kreirajte:

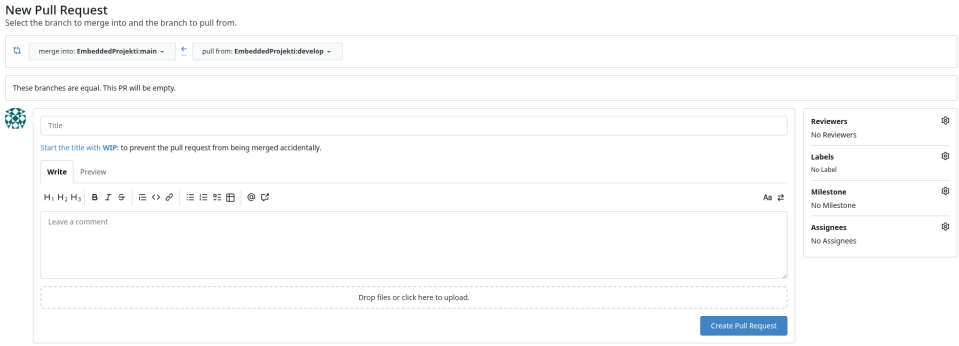
```
git checkout -b feature/vp_dma
# uradite izmene...
git push --set-upstream origin feature/vp_dma
```

5 Kreiranje Pull Request-a (PR)

Pull Request (PR) je zahtev da se vaše izmene sa jedne grane spoje u drugu (npr. iz `feature/vp_dma` u `rtl`, ili iz `rtl` u `main`). PR pravite samo za spajanje u konačnu `main` granu. Za merge u radne grane i vaše proizvoljne grane, imate dozvolu da sami radite merge.

5.1 Kako otvoriti PR

1. Gurnite (push) svoju granu na server (`git push`).
2. Otvorite git interfejs u browseru i otvorite repozitorijum.
3. Iz taba **Pull Requests** kliknite na **New Pull Request**.



Slika 5: Kreiranje PR-a

Zatim:

1. Izaberite baznu granu (**merge into**) u koju želite da spojite izmene (u nasem slucaju `main`).
2. Izaberite granu koju želite spojiti (**pull from**) sa vašim izmenama (npr. `develop` iz slike iznad).
3. Proverite listu commit-ova i promenjenih fajlova.
4. U polje **Title** upišite jasan naslov (npr. “Napravljen RTL model”).
5. U polje **Description** ukratko opišite šta PR radi, šta je urađeno, koje su izmene, i na šta treba obratiti pažnju.
6. Kliknite na dugme **Create Pull Request**.

5.2 Pregled, komentarisanje i spajanje PR-a

- Nastavnici ili asistenti mogu da ostavljaju komentare direktno na linije koda ili na ceo PR.
- Vi možete odgovarati na komentare i gurati dodatne commit-ove u istu granu – oni će se automatski pojaviti u PR-u.
- Kada je sve u redu, asistent će uraditi **Merge** PR-a u ciljni main branch.

6 Pisanje Issues (pitanja oko projekta)

Issues se koriste za svu diskusiju o projektu, zadacima, pitanja. Sva diskusija se radi ovde.

6.1 Kako kreirati Issue

1. Otvorite svoj repozitorijum u web interfejsu.
2. Pređite na tab **Issues**.
3. Kliknite na dugme **New Issue**.

Zatim popunite:

- **Title** – kratak naslov (npr. "Nedoumica oko HLS alata");
- **Description** – detaljno objasnite problem, pitanje, predlog, i slicno.
- Po potrebi spomenite relevantnog asistenta kako bi dobio brze obavestenje koristeci @.

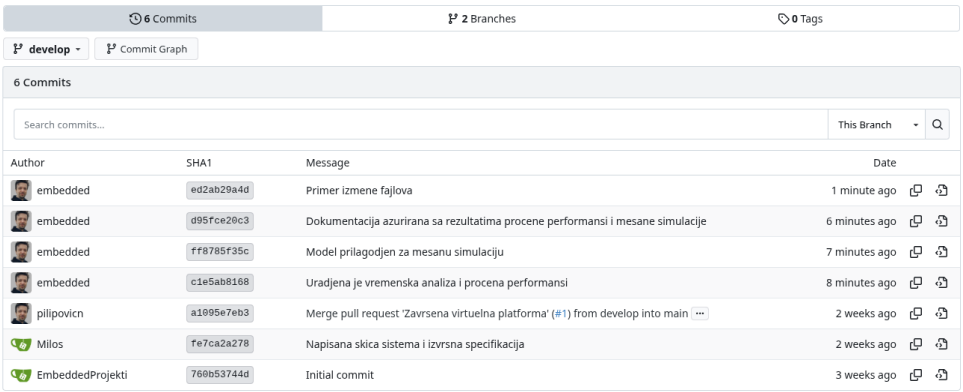
Nakon slanja, Issue će se pojaviti na listi svih Issues u repozitorijumu.

7 Pregled istorije izmena i razlika

7.1 Pregled commit istorije

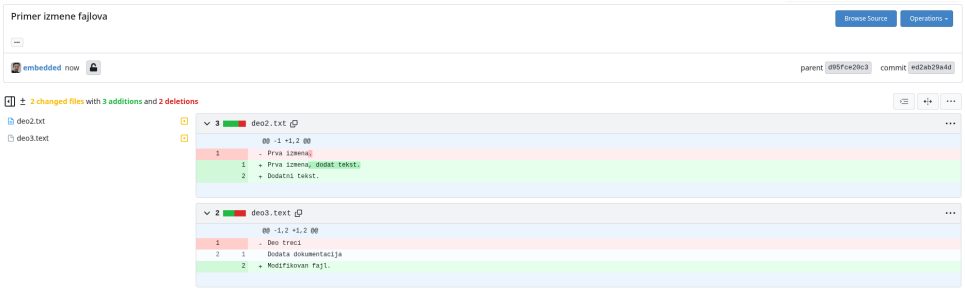
U tabu **Code** klikom na **Commits** možete videti istoriju commit-ova:

- autor,
- poruku commita,
- datum i vreme,
- hash commita.



Slika 6: Lista commitova

Klikom na commit otvarate detaljan pregled izmenjenih fajlova i diff prikaz (zelene linije – dodato, crvene – obrisano).



Slika 7: Diff pogled

8 Dobre prakse za studente

- **Često pull-ujte** pre rada, da biste izbegli konflikte.
- Pravite **male, česte commit-ove** sa smislenim porukama.
- Radite u **posebnim granama** (npr. `feature/ime_dela`) umesto direktno u radnu granu.
- Dajte **jasan opis PR-ova** – šta ste tačno uradili i kako testirati.
- Koristite **Issues** za bilo kakva pitanja u vezi projekta, umesto usmenog dogovora. Usmeni dogovor se često zaboravi ili dođe do nesporazuma. Koristite Issues da ostavimo pisani trag za svaki dogovor oko projekta.

9 Autentifikacija

Podrazumevano, git za operacije pull i push, traži kredencijale korisnika. Ako želite da izbegnete da vas svaki put autentifikuje, možete ga podesiti da pamti kredencijale.

```
git config --global credential.helper store
```

Ovo čuva kredencijale beskonačno dugo na disku. Imajte na umu da ovime dobijate udobnost a oduzimate sigurnost. U slučaju kompromitovanja vaše mašine, kredencijali su nezaštićeni i vidljivi na disku. Za nešto sigurniju opciju možete koristiti:

```
git config --global credential.helper 'cache --timeout=3600'
```

Ovo omogućava privremeno čuvanje kredencijala u RAM-u, gde se nakon vremena od 3600s zaboravljaju.

Podešavanja su globalna za sve repoe. Ako želite pamćenje šifre samo za ovaj repo, uklonite flag `--global`.
Autentifikacija javnim ključevima trenutno nije moguća.