

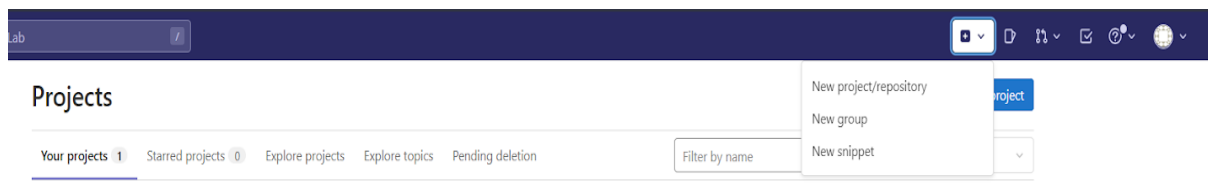
## Uputstvo za korišćenje GitLab repozitorijuma

### Version Control Systems VCS

- Programi koji se koriste kada je potrebno pratiti verzije projekta i/ili kada više učesnika radi na projektu
- Danas su nezaobilazni u procesu razvoja softvera
- Datoteke koje se prate zajedno sa svom istorijom promena se nalaze na udaljenom mestu — repozitorijum
- Svaka izmena koja se uradi, treba da se sačuva na repozitorijumu — commit
- Pored samih izmena čuva i informacije o tome ko je napravio izmenu, kad je izmena napravljena i šta je tačno izmenjeno

### Uputstvo za kreiranje repozitorijuma

1. Nakon registracije na [gitlab.com](https://gitlab.com) treba da kreirate repozitorijum za web dizajn projekat. Kliknete na ikonicu u gornjem desnom ćošku i izaberete opciju “New project/repository”.



2. Nakon toga, izaberete opciju **Create blank project**.
3. Sledeći korak predstavlja popunjavanje forme za inicijalizaciju projekta. Ime projekta neophodno je da bude u formatu **WD-SV-xx-yyyy**, gde xx i yyyy predstavljaju broj indeksa i godinu upisa. Osim navedenog, neophodno je da projekat bude **privatni**. Primer popunjavanja forme je dat na slici ispod.  
**Neophodno je da ime projekta bude u navedenom formatu!**

New project > Create blank project

Project name

WD-SR-999-2022

Project URL

https://gitlab.com/web-design-FTN/

Project slug

wd-sr-999-2022

Want to organize several dependent projects under the same namespace? [Create a group](#).

Project deployment target (optional)

Select the deployment target

Visibility Level [?](#)

☒ Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

☐ Public

The project can be accessed without any authentication.

Project Configuration

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more](#).

Create project

Cancel

4. Nakon kreiranja repozitorijuma, bićete preusmereni na stranicu na kojoj se nalazi vaš projekat.

5. Sa leve strane izaberite opciju *Project information/Members*, zatim u otvorenom prozoru izaberite opciju **Invite members**. U okviru otvorenog prozora dodajte nalog **web-design-FTN** i dodelite ulogu **Maintainer-a**.

**Ovaj korak je obavezan!**

WD-SR-999-2022 or invite another group.

Invite members

You're inviting members to the **WD-SR-999-2022** project.

Username or email address

web-design-FTN

Select members or type email addresses

Select a role

Maintainer

[Read more](#) about role permissions

Access expiration date (optional)

YYYY-MM-DD

Cancel Invite

## Instalacija sistema za kontrolu verzija - GIT

- Preuzeti instalaciju sa službene [web stranice](#)
- Prvobitna namena je korišćenje iz komandne linije
- Grafička okruženja za Git dostupna su [na linku](#)
- Provera uspešne instalacije: `git -version`
- VSCode ima integrisanu podršku

## Podešavanja

Nakon instalacije, dobra praksa je da se odmah podesi ime i email adresa koje će Git koristiti prilikom čuvanja izmena:

```
git config -- global user . name "Ime Prezime "  
git config -- global user . email mailadresa@nesto.com
```

Global će izvršiti podešavanja za svaki repozitorijum na računaru. Ukoliko se izostavi, podešavanja se vrše samo za onaj repozitorijum na koji smo trenutno pozicionirani iz komandne linije. Lista trenutno podešenih vrednosti se može dobiti komandom: `git config --list`

## Kreiranje lokalnog repozitorijuma

Postoje dva načina:

### 1. Inicijalizacijom novog (praznog) repozitorijuma

- `git init` je komanda koja će kreirati prazan lokalni repozitorijum
- Kreiraće se sakriveni folder `.git` unutar direktorijuma iz kojeg smo pokrenuli komandu (radnog direktorijuma)
  1. Pozicionirajte se u folder u kojem se nalaze fajlovi vašeg projekta
  2. U terminalu kucamo sledeće komande
    - A. **git init** (na ovaj način smo inicijalizovali git repozitorijum lokalno)
    - B. **git add .** (dodali smo sve fajlove i podfoldere trenutnog foldera za praćenje)
    - C. **git commit -m "opis šta ste odradili"** (svi fajlovi koji su dodati prethodno su sada spremni za slanje na repozitorijum, poruka koju stavite pod navodnike biće vidljiva na sajtu i predstavlja opis koda koji ste dodali)
    - D. **git remote add origin**  
<https://gitlab.com/web-design-FTN/wd-sr-999-2022.git>

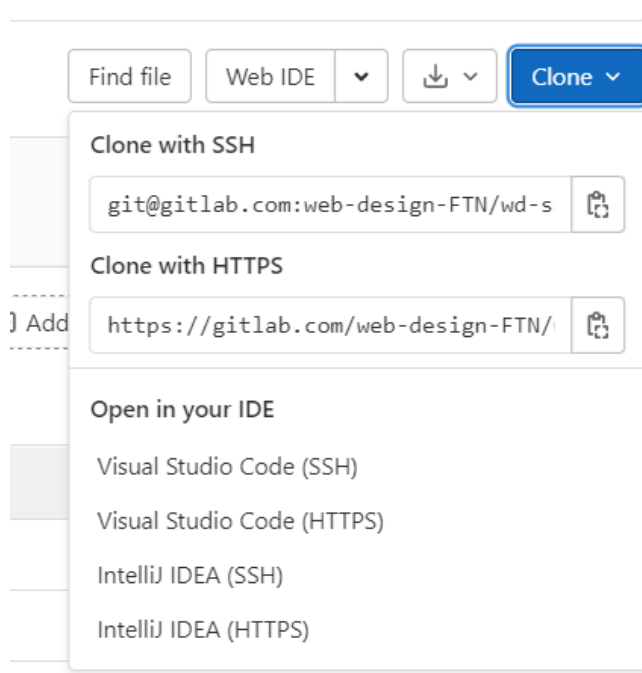
(nakon gitlab.com ide vaš username, pa naziv repozitorijuma koji ste kreirali)

### E. **git push --set-upstream origin main**

Nakon ovog koraka će vam tražiti kredencijale kojim ste se registrovali na gitlab.com. Ukoliko izvršite sve prethodne korake vaš prvi “commit” je uspešno postavljen na gitlab repozitorijum! Možete u browseru otići na vaš repozitorijum i tamo videti promene.

## 2. Kloniranjem postojećeg repozitorijuma sa udeljanog servera

- ***git clone [putanja do repozitorijuma] [lokalni direktorijum gde se klonira]*** je komanda koja će kreirati lokalni repozitorijum sa svim fajlovima i istorijom izmena sa udeljanog servera
- U okviru radnog direktorijuma kreira skriveni folder .git koji zapravo označava da je folder git repozitorijum.
- Primer: ***git clone https://gitlab.com/web-design-FTN/wd-sr-999-2022.git***  
*Link možete pronaći pod opcijom Clone u zaglavlju vašeg projekta*



Nakon ovog koraka, kao i u prethodnom slučaju možete dodati sve fajlove od interesa- *koraci za ovaj postupak dati su u sledećem naslovu*

*U svakom trenutku je moguće dobiti informacije o stanju repozitorijuma komandom **git status***

## Postavljanje izmena na repozitorijum

Nakon inicijalizacije lokalnog repozitorijuma i registracije na Gitlab platformi izmene je moguće poslati i na lokalni i na udaljeni repozitorijum.

- Operacija commit izmene šalje na lokalni repozitorijum
- Operacija push šalje lokalne izmene na udaljeni repozitorijum (može više commit-a)

Uobičajeni koraci:

***git add .***

***git commit -m " Tekst commit poruke "***

***git push origin main***

- Kreiranje datoteka u radnom direktorijumu neće navesti Git da prati izmene nad njima
- Pokretanje praćenja verzija nad nekom datotekom ili direktorijumom se vrši komandom ***git add [naziv datoteke]***
- Komanda git add se mora izvršiti za sve datoteke koje su nove, ali i za sve postojeće datoteke nad kojima su izvršene neke izmene (nalaze se u stanju modified)
- Možemo automatski dodati sve izmene komandom ***git add .***
- Naknadne izmene nad praćenim datotekama neće biti evidentirane automatski
- Prilikom svake izmene potrebno je ponoviti ***git add*** komandu

## Saveti oko commit-a

- Logički nezavisne izmene organizovati u odvojenim commit-ima
- Težiti ka davanju prefiksa opisu commit-a (add,fix,itd.) kako bi se ukazalo na tip izmene

## Git file status lifecycle

- Postavljanje jednog fajla u staging area ili Postavljanje svih fajlova u staging area  
***git add <ime\_fajla >*** ili ***git add .***
- Brisanje fajla iz fajl sistema i postavljanje te izmene u staging area  
***git rm <ime\_fajla >***
- Poništavanje izmena  
***git reset HEAD <ime\_fajla >***