

ČETVRTA LABORATORIJSKA VJEŽBA

1. Ulogujete se preko PuTTY-ja na server **89.188.32.136**. U vašem home direktorijumu kreirajte folder **LAB4**. Pozicionirajte se u direktorijum **LAB4**, a zatim u njemu kreirajte fajl **README** u koji treba upisati tekst „Moj prvi projekat“.

```
mkdir ~/LAB4
cd ~/LAB4
touch README
echo Moj prvi projekat > README
```

2. Konfigurirate vaš git sa sljedećim podacima: vaše user name, password i editor.

```
git config --global user.name "Ime i prezime"
git config --global user.email email@edu.ac.me
git config --global core.editor nano
```

Trenutna konfiguracija se može provjeriti na sljedeći način:

```
git config --list
```

Informacije koje ste unijeli se čuvaju u fajlu **~/.gitconfig**, što se može provjeriti komandom:

```
cat ~/.gitconfig
```

3. Pozicionirajte se u direktorijum **LAB4**, kreirajte novi repozitorijum (**git init**), a zatim dodajte sve fajlove iz LAB4 u repozitorijum (**git add .**, trenutno samo postoji fajl README). Na kraju commit-ujte promjene u repozitorijum.

```
cd LAB4
git init
git add .
git commit -m "Novi projekat"
```

Informacije o repozitorijumu se čuvaju u direktorijumu **.git**, što se može provjeriti komandom:

```
ls -l .git
```

4. Napravite promjene u README fajlu, a zatim provjerite stanje git-a (**git status**). Odraditi ponovo commit, opet provjerite stanje git-a i prikažite log fajl (**git log**).

```
git status
nano README
git status
git add README
git status
git commit -m "Updated README"
git status
git log
```

5. Kreirajte nalog na GitLab:

<http://89.188.32.141/>

Unesite isti user i password koji koristite na Linux serveru sa vježbi (radi lakšeg pamćenja). Takođe koristite **edu.ac.me** mail. Nakon toga kreirajte prvi projekat pod nazivom **LAB4** (slug: **lab4**), i u podešavanjima postavite da je projekat **javan** i uklonite check sa opcije „Initialize repository with a README“.

Project name
Lab4

Project URL
http://89.188.32.141/zarkoz/

Project slug
lab4

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.

Internal
The project can be accessed by any logged in user except external users.

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

Nakon ovoga vaš projekat će biti javno dostupan na adresi:

http://89.188.32.141/vas_user/lab4

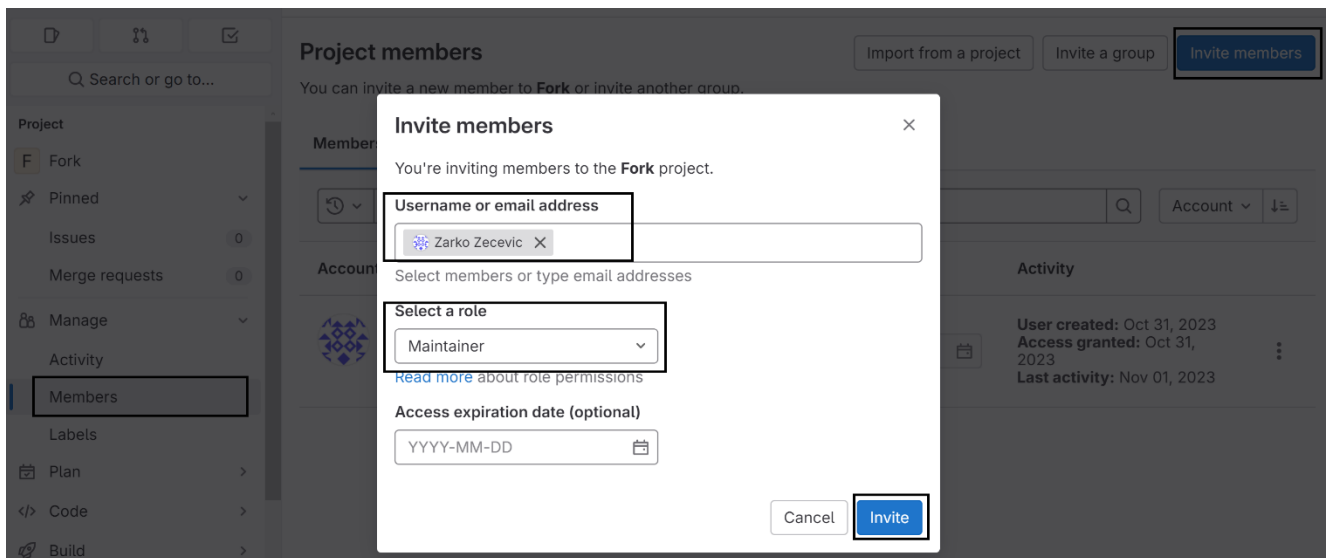
Istražite malo korisničko okruženje na GitLAB-u. Pronađite kako da dodate članove tima na projekat i kako da obrišete tekući projekat.

6. Ponovo se pozicionirajte u direktorijum **LAB4** (ukoliko već niste), a zatim se povežite (**git remote**) na glavnu razvojnu granu (**origin**) vašeg projekta čija je glavna verzija nalazi na GitLAB serveru (http://89.188.32.141/vas_user/lab4.git). Uploadujte na server promjene (**git push**) koje ste napravili u lokalnom repozitorijumu.

```
cd ~/LAB4
git remote add origin http://89.188.32.141/vas_user/lab4.git
git push -u origin master
```

Napomena: pomoću komande **git remote -v** provjeravate URL adresu repo-a koji ste podesili, dok pomoću komande **git remote set-url origin novi_url** možete da promijenite URL.

7. Provjerite na GitLab-u da li su promjene uspješno uploadovane. Nakon toga u meniju sa lijeve strane pronađite **Manage->Members**, a zatim za člana tima dodajte nalog **zarkoz** (uloga **Maintainer**).



8. U nastavku će biti „simulirano“ kako članovi tima rade na istom projektu. Vi ćete zapravo mijenjati projekat, ali sa drugog git naloga. Najprije u vašem home-u treba kreirati direktorijum **LAB44** i u njega povući projekat sa servera:

```
mkdir ~/LAB44
cd ~/LAB44
gitclone http://89.188.32.141/vas_user/lab4.git
cd lab4
```

Nakon toga treba prekopirati ove dvije linije koda:

```
git config user.name "Zarko Zecevic"
git config user.email "zarkoz@ac.me"
```

Ovo zapravo znači da se za ovaj projekat ne koriste globalna podešavanja iz `~/.gitconfig` fajla, već lokalna podešavanja – tj. user zarkoz@ac.me.

U sljedećem koraku je potrebno kreirati novi fajl **index.html**, dodati ga u repozitorijum (commit) i napisati odgovarajući komentar.

```
touch index.html
git add index.html
git commit -a -m "Dodat index.html"
```

Konačno, sve izmjene treba uploadovati na server:

```
git push
```

Koristiti username **zarkoz** i password **soksoksok**.

9. U zadnjem koraku se opet pozicionirajte u početni direktorijum ~/LAB4. Povucite novu verziju repozitorijuma. Refreshujte i opet odradite commit.

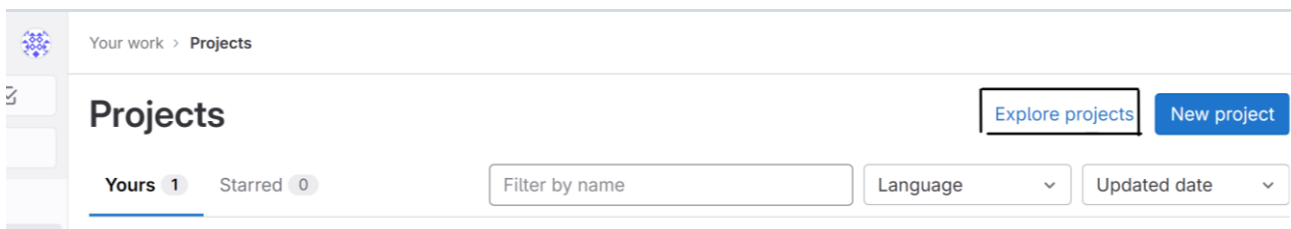
```
git pull
touch >> index.html
git commit -a -m "Izmijenjen fajl index.html"
git push
```

Na kraju otvorite vaš projekat u browser-u:

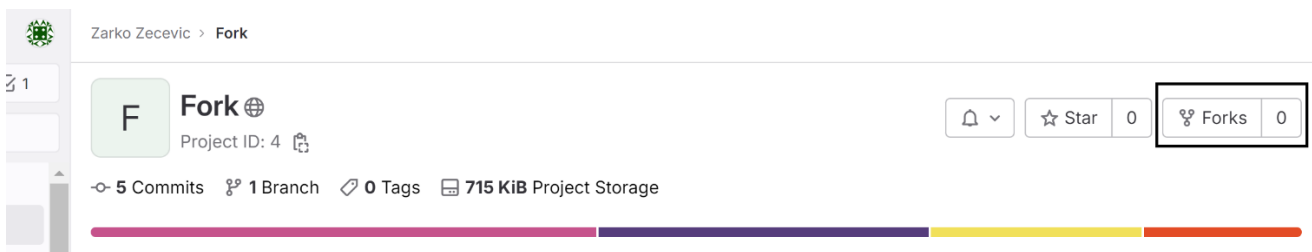
http://89.188.32.141/vas_user/lab4

Provjerite koji sve fajlovi postoje i ko je unio promjene.

10. Na početnoj strani GitLab-a pronađite link **Explore project** i kliknite na njega, a zatim pronađite projekat pod nazivom „Fork“ koji je kreirao korisnik Žarko Zečević (username zarkoz).



Otvorite projekat i u desnom gornjem uglu kliknite na **Forks**. Ova opcija se koristi za importovanje repozitorijuma drugog korisnika u vaš repozitorijum, nakon čega nezavisno možete da razvijate svoju verziju. Kada izmijenite i završite vašu verziju, možete da predložite originalnom vlasniku da prihvati (merguje) vašu verziju.



Popunite podatke koji se traže od vas (vaš username, u ime projekta takođe stavite naziv fork i nakon toga kliknite na dugme **Fork**, kao što je prikazano na slici ispod.



Fork project

A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.

Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

Project slug

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

Project description (optional)

Visibility level ?

 Private

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

 Internal

The project can be accessed by any logged in user.

 Public

The project can be accessed without any authentication.

Projekat zapravo predstavlja jednu statičku web stranicu (**index.html**) kojoj se može pristupiti preko linka:

<http://89.188.32.136/~zarkoz/fork/>

Vaš zadatak je da izmijenite html **article tag** koji se otvara klikom na dugme **About**. Primijetićete da se u prozoru koji se otvara pojavljuje siva slika. U **index.html** fajlu pronađite naziv slike, a zatim u vašem GitLab folderu pod istim nazivom uploadajte sliku <https://i.imgur.com/cUuu62D.jpeg> i sačuvajte promjene. GitLab neće dozvoliti upload slike pod istim nazivom, dok prethodno ne obrišete originalnu sliku. Nakon uploada slika, pritisnite kombinaciju tastera shift+F5 (ili shift+Fn+F5) da obrišete keš (u slučaju da se nova slika nije učitala).

The screenshot shows the GitLab repository interface for a forked project. The left sidebar contains navigation options: Project, Fork, Pinned, Issues (0), Merge requests (0), Manage, Plan, Code (selected), Merge requests (0), Repository (highlighted), Branches, Commits, and Tags. The main content area shows the repository path 'fork / images /' with a dropdown menu and a left arrow. Below the path, it indicates 'Forked from Zarko Zecevic / Fork' and 'Up to date with the upstream repository.' A table lists files with their names, last commit status, and last update time:

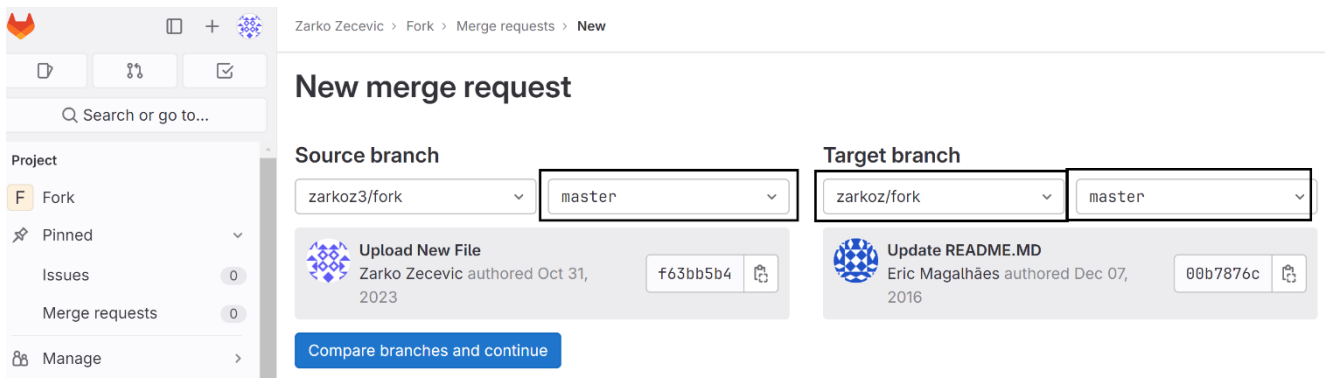
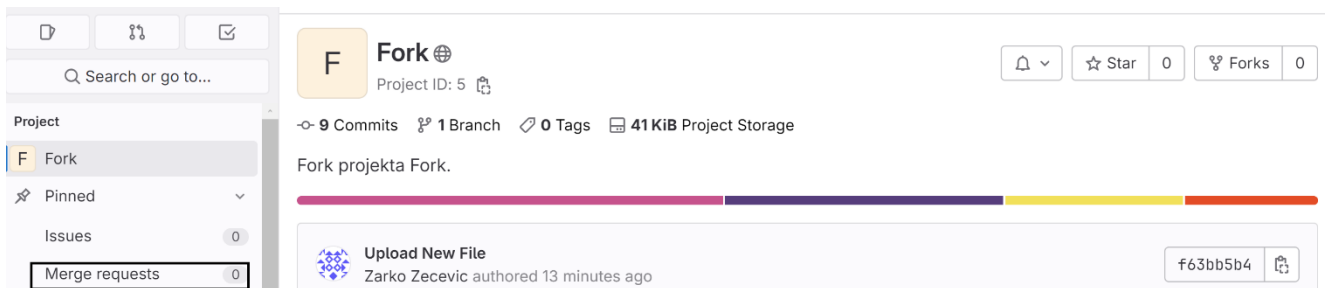
Name	Last commit	Last update
..		
bg.jpg	initial commit	6 years ago
overlay.png	initial commit	6 years ago
pic01.jpg	initial commit	6 years ago
pic02.jpg	initial commit	6 years ago
pic03.jpg	initial commit	6 years ago

Nakon ovoga koraka, ulogujte se na server i u `public_html` direktorijum klonirajte vaš repozitorijum fork. Provjerite da li se stranica i uploadovana slika uspješno otvaraju putem linka

http://89.188.32.136/~vase_korisniko_ime/fork/

```
cd ~/public_html
git clone http://89.188.32.141/vas_user/fork.git
```

Na kraju, otvorite opet Fork projekat na GitLabu i kliknite **Merge Requests**. Pošaljite zahtev za spajanje **vaše master** grane i **master** grane **originalnog fork** projekta.



11. U browser-u ukucajte sljedeći URL:

<https://github.com/LotfarKaes/CRUD-with-PHP-and-MySQL>

Pročitajte dokumentaciju o CRUD aplikaciji.

Pozicionirajte se u vaš `public_html` direktorijum, a zatim u njega povucite sadržaj repozitorijuma (na GitHub-u pronađite URL repo-a):

<https://github.com/LotfarKaes/CRUD-with-PHP-and-MySQL.git>

Komandom `ls` provjerite da li postoji direktorijum `CRUD-with-PHP-and-MySQL`, a zatim promijenite njegov naziv u `CRUD`.

U fajl `config/config.php` unesite podatke o vašoj bazi podataka, a zatim importujte sql fajl `userdata.sql`.

```
git clone https://github.com/LotfarKaes/CRUD-with-PHP-and-MySQL.git
mv CRUD-with-PHP-and-MySQL CRUD
nano ~/public_html/CRUD/config/config.php
mysql -uproba -pproba proba < ~/public_html/CRUD/userdata.sql
```

Konačno, otvorite link:

http://89.188.32.136/~vas_user/CRUD/crud.php

da bi provjerili da li je sve dobro podešeno. Na ovaj način smo se spremili za naredne vježbe.