

PETA LABORATORIJSKA VJEŽBA

1. 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 -uvas_user -pvas_user vas_user < ~/public_html/CRUD/userdata.sql
```

Konačno, otvorite link:

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

S obzirom da ste unijeli promjene u fajl **config.php**, potrebno je odraditi commit:

```
git commit -am "Izmijenjen config.php"
```

2. Pozicionirajte se u **CRUD** direktorijum, a zatim provjerite sadržaj fajla **.git/config**. Uočite da je kao remote repozitorijum definisan github link sa kojeg ste preuzeli aplikaciju. Uklonite postojeći remote repozitorijum i postavite novi http://89.188.32.141/vas_user/crud.git. Na kraju uploadujete sve na remote repozitorijum.

```
cd ~/public_html/CRUD
cat .git/config
git remote remove origin
git remote add origin http://89.188.32.141/vas_user/crud.git
git push -u origin master
# Obratite pažnju da je korišćena opcija -u iz razloga što je nakon brisanja
# origin-a iz config fajla je obrisana i linija koja definiše na koju grane
# se po default-u vrši upload (push). Nakon korišćenja opcije -u informacija
# o grani master će biti dodata u .git/config i kasnije je dovoljno odraditi
# git push da bise odradio upload na ovu granu.
```

Unesite vaše username i password, a nakon toga provjerite stanje na serveru:

http://89.188.32.141/vas_user/crud.git

3. Kreirajte novu razvojnu granu projekta, pod nazivom **crudv1**, a zatim se pozicionirajte u tu granu.

```
git checkout -b crudv1
```

Gornja komanda zamjenjuje dvije komande:

```
git branch crudv1  
git checkout crudv1
```

Novokreirana grana sadrži kopiju fajlova iz master grane, jer smo se nalazili u master grani kada smo kreirali crudv1. Ova grana se na sličan način uploaduje na server:

```
git push -u origin crudv1
```

Iz jedne grane u drugu se možete prebacati pomoću komande git checkout:

```
git checkout master  
git checkout crudv1
```

4. Sada ćemo napraviti neke izmjene u grani **crudv1**. Ovaj dio je jednostavnije odraditi tako što ćete fajlove otvoriti pomoću FileZilla-e u Notepad++.

- Za početak promijenimo fajl **master.php**, koji zapravo predstavlja header u svakom ostalom fajlu. U headeru umjesto naslova „**PHP with MySQL fundamentals Training**“ postavite naslov „**Softver otvorenog koda**“, a umjesto „**Create, Retrieve, Update, Delete (CRUD)With PHP & MySQL**“ postavite „**Kreiranje, preuzimanje, ažuriranje i brisanje podataka pomoću PHP-a i MySQL-a**“. Takođe obrišite css kod **style="color:green;"** iz **h3** taga u kome je upisan ovaj tekst.
- U fajlu **css/stileshite.css** promijenite boju klasa **.headeroption** i **.fotteroption** iz **#69df2e** u **#A74D4D**. Takođe, pozadinsku boju klase **ul.b** postavite na **#A74D4D** (umjesto **#91df1c**). Primijete da se ul tag sa klasom **b** koristi za tekst „**Kreiranje, preuzimanje, ažuriranje...**“ u **master.php** fajlu. Ukoliko se promjene ne vide u browseru, obrišite keš.
- U fajlu **crud.php** promijenite sql upit tako da tabela bude sortirana prema sadržaju kolone **skill**

```
$query = "select * from tbl_user ORDER BY skill";
```

5. Odradite commit svih promjena i uploadujte ih na remote repozitorijum. Zatim otvorite stranicu:

<http://89.188.32.136/~zarkoz/CRUD/crud.php>

Prebacite se u granu, opet otvorite gornju stranicu i uočite razlike. Na kraju se vratite u granu crudv1. Provjerite na GilLab-u da li su uspješno uplaodvane promjene i da li se uočavaju dvije razvojne grane projekta

```
cd ~/public_html/CRUD  
git commit -m "Promjene u stilu"  
git push  
git checkout master  
git checkout crudv1
```

7. U vaš browser otvorite:

<http://89.188.32.136/nextcloud>

a zatim se ulogujte. Vaše korisničko ime i password su isti kao user na Linux sistemu. Kreirajte direktorijum **LAB5** i u njega uploadujte neki fajl.

Šerujte direktorijum **LAB5** sa korisnicima **zarkoz** i **lukam**, a zatim pronađite kako se pravi javni link na folder.

Na vašem mobilnom telefonu sa AppStore-a preuzmite aplikaciju **NextCloud** i ulogujte se na server. Screenshot aplikacije uploadujte u folder **LAB5**. Ukoliko ste zainteresovani, desktop klijent možete preuzeti sa linka:

<https://nextcloud.com/clients/>

8. NextCloud ima razvijen interfejs za pristup sa komandne linije ili iz PHP aplikacije. Na primjer, ako želite da napravite novi direktorijum, to možete odraditi pomoću komande:

```
curl u vas_user:vas_password -X -MKCOL  
"http://89.188.32.136/nextcloud/remote.php/dav/files/user/naziv_foldera"
```

dok se fajl može uploadovati na sljedeći način:

```
curl -u vas_user:vas_password -T putanja_do_fajla_na_serveru  
"http://89.188.32.136/nextcloud/remote.php/dav/files/user/folder/imefajla"
```

Kompletna dokumentacija se može naći na linku:

https://docs.nextcloud.com/server/15/user_manual/index.html

Da bi demonstrirali rad ovih komandi (koje ne treba pamtiti), u vašem home-u kreirajte prazan fajl **proba**, a zatim na serveru kreirajte direktorijum **BACKUP** i u njega uploadujte fajl **proba**.

```
touch proba  
curl -u user:password -X MKCOL  
"http://89.188.32.136/nextcloud/remote.php/dav/files/user/BACKUP"  
curl -u user:password -T ~/proba  
"http://89.188.32.136/nextcloud/remote.php/dav/files/user/BACKUP/proba.txt"
```

9. Editujete vaš crontab fajl na serveru tako da se (ne brisati prethodni sadržaj fajla):

- Svakog prvog dana u mjesecu, u 04.00h, pravi arhiva **~/html.tar.gz** u koju će biti upakovan sadržaj vašeg **public_html** foldera.
- Svakog prvog dana u mjesecu, u 04.15h, arhiva **html.tar.gz** uploaduje u folder **BACKUP** na NextCloud server.

```
crontab -e  
0 4 1 * * tar -cvzf ~/html.tar.gz ~/public_html  
15 4 1 * * curl -u user:password -T ~/html.tar.gz  
"http://89.188.32.136/nextcloud/remote.php/dav/files/user/BACKUP/"html.tar.gz"
```

10. Na GitLab-u otvorite vaš fork projekat sa prošlog časa. Adresa projekta je

http://89.188.32.141/vas_user/fork/,

ili ga pronađite među postojećim projektima. U repozitorijum dodajte fajl čiji je naziv **.gitlab-ci.yml**. Sadržaj fajla je dat ispod (ili ga kopirajte sa ovog [LINKA](#)). Obratite pažnju na liniju koda u kojoj treba da unesete vaše korisničko ime na Linux serveru (bez razmaka). **Ukoliko fajl editujete u WEB-IDE, izvršite commit!**

```
stages:
  - build
  - test
  - deploy

build-job:
  stage: build
  script:
    - echo "Hello, $GITLAB_USER_LOGIN!"

test-job:
  stage: test
  script:
    - ls -a
    - pwd
    - tidy -o korigovani.html -f greske.txt --drop-empty-elements no --
doctype omit --clean yes --tidy-mark no index.html
  artifacts:
    when: always
    paths:
      - greske.txt
      - korigovani.html

deploy-job:
  stage: deploy
  script:
    - usr=vas_user_name
    - cp -r ./* /home/$usr/public_html/git
```

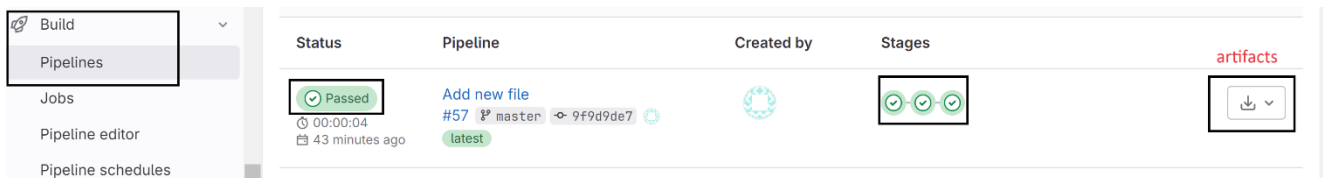
Gornji fajl definiše CI/CD pipeline koji se izvršava u tri faze: **build**, **test** i **deploy**. Faze se izvršavaju zdatim redosljedom, pri čemu se na sljedeću fazu prelazi ukoliko se poslovi u okviru prethodne faze uspješno završe. CI/CD pipeline se sastoji od tri posla: **build-job** (faza **build**), **test-job** (faza **test**) i **deploy-job** (faza **deploy**):

- U okviru posla **build-job** se zapravo ništa važno ne dešava. Jedino se na ekranu štampa vaše gitlab korisničko ime.
- U okviru posla **test-job** se izvršava par linux komandi (ls -a i pwd) i pokreće se softver **tidy** koji vrši validaciju html koda iz fajla **index.html**. Tidy program informacije o upozorenjima i greškama vraća u fajl **greske.txt**, dok se korigovani html fajl čuva pod nazivom **korigovani.html**. Ukoliko tidy program ne vrati upozorenja ili greške, ova faza će se uspješno završiti i preći će se na narednu fazu. Ključna riječ **artifacts** definiše fajlove koje će posao da vrati korisniku. U ovom slučaju to su fajlovi **greske.txt** i **korigovani.html** i ova dva fajla će uvijek biti vraćena, bez obzira na to da li je posao uspješno završen ili ne (**when: always**).

- U okviru posla **deploy-job** sadržaj repozitorijuma se kopira u folder **git** koji se nalazi u vašem **public_html** direktorijumu. Ovaj folder je već kreiran za sve studente, a ukoliko ne postoji treba odraditi sljedeće komandu:

```
mkdir -p ~/public_html/git && chmod 777 ~/public_html/git
```

Nakon što sačuvate YAML fajl, u **Build/Pipelines** provjerite da li su sve faze uspješno završene. U desnom dijelu moći ćete da preuzmete fajlove koje ste definisali u **artifacts**.



Klikom da neku od faza možete odabrati posao u okviru nje i prikazati rezultat izvršavanja skripte na gitlab runner-u.

```
24 $ pwd
25 /home/zarkoz/builds/mzhm6TBbj/0/Lukam/fork
26 $ tidy -o output.html -f greske.txt --drop-empty-elements no --doctype omit
--clean yes --tidy-mark no index.html
27 Uploading artifacts for successful job 00:01
28 Uploading artifacts...
29 Runtime platform arch=amd64 os=linux pid
=30113 revision=853330f9 version=16.5.0
30 greske.txt: found 1 matching artifact files and directories
31 output.html: found 1 matching artifact files and directories
32 Uploading artifacts as "archive" to coordinator... 201 Created id=190 resp
onseStatus=201 Created token=64_sti_S
33 Job succeeded
```

Ukoliko su sve faze uspješno završene, provjerite da li se izvršen deployment aplikacije

http://89.188.32.136/~vase_korisniko_ime/git/

Ukoliko je nastala greška u testnoj fazi, pokušajte da pronađete grešku u html kodu (ili prosto obrišite tidy komandu).

Napomena: Pipeline koji smo definisali se izvršava na gitlab-runner-u koji je instaliran na serveru 89.188.32.136. Repozitorijum se raspakuje u folder /home/zarkoz/builds/itd. i u tom folderu se izvršavaju sve skripte koje smo definisali. U zadnjoj fazi, sadržaj direktorijuma se kopira u folder git u vašem home-u (folder mora postojati i korisnik zarkoz mora imati dozvole za kopiranje).

ZA KRAJ:

U **index.html** izmijenite riječ Dimension u Dimenzija, a zatim provjerite da li će pipeline uspješno izvršiti i da li će se promjene odraziti na stranici http://89.188.32.136/~vase_korisniko_ime/git/.