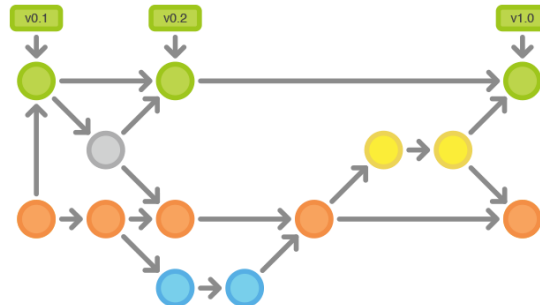




GIT & Bitbucket

Системи за верзионирање на код

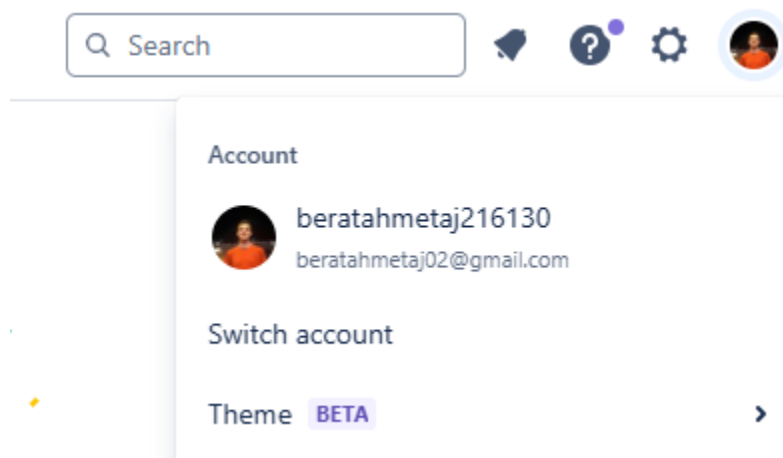


Разгледајте ги можностите на GIT и порталот Bitbucket, извршете ги барањата подолу и под секое од нив ставете слика (PrintScreen) од соодветната веб форма, прозорец или напишете ги потребните наредби за GIT. На крај, генерирајте pdf од документот и прикачете го на Courses во формата за лабораториската вежба.

Забелешка: Кога ќе креирате корисничка сметка треба да биде во формат име, презиме индекс. Доколку на сликите не се гледа вашето име, презиме и индекс, вежбата ќе ви биде оценета со 0 поени.

Забелешка: За да ги извршите барањата, потребно е да инсталирате GIT на Вашата локална машина. Може да се користи терминал или GUI алатки, по Ваш избор.

1. Да се креира корисничка сметка на <https://bitbucket.org/>.
Слика: Избришете ја сликата, па прикачете ја својата. Оваа слика стои само за пример како треба да изгледа.



2. Креирајте нов проект на Вашиот локален компјутер (или искористете некој од Вашите тековни проекти) и иницијализирајте го GIT репозиторот за него.



Управување со ИКТ проекти

Create a new repository

[Import repository](#)

Workspace beratahmetaj216130

Project name*

Repository name*

Access level ☒ Private repository
Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README? Yes, with a tutorial (for beginne...

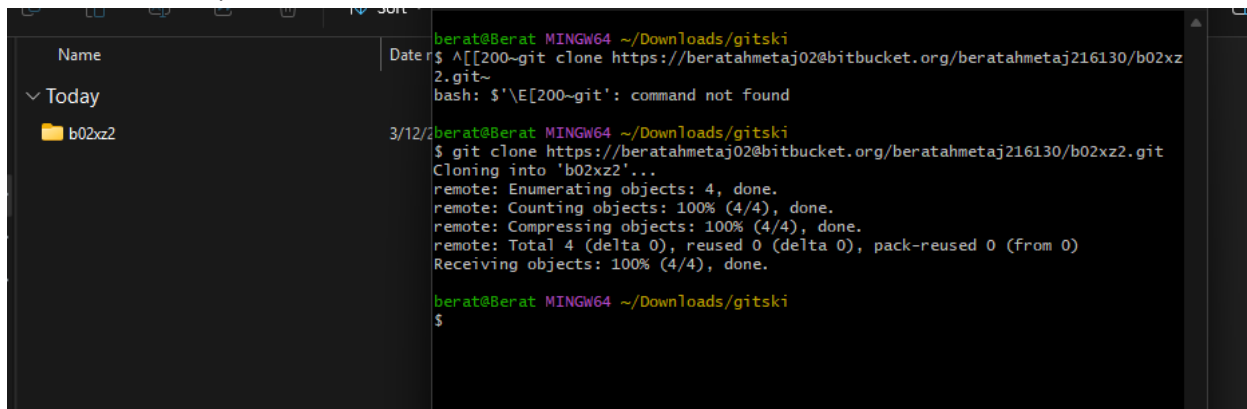
Default branch name

Include .gitignore? Yes (recommended)

[Advanced settings](#)

3.

Слика или GIT наредби:



```
beratahmetaj216130
beratahmetaj02 MINGW64 ~/Downloads/gitski
$ git clone https://beratahmetaj02@bitbucket.org/beratahmetaj216130/b02xz2.git
Cloning into 'b02xz2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

4. Креирајте нов проект на Bitbucket, додадете ја неговата адреса во Вашиот локален GIT репозитор (git remotes) и прикачете го (push) Вашиот локален проект на Bitbucket.
- Слика и/или GIT наредби:



```
berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (main)
$ git commit -m "changes to structure"
[main 96c8962] changes to structure
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 folderino/test.txt

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 367 bytes | 367.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://bitbucket.org/beratahmetaj216130/b02xz2.git
   3608474..96c8962  main -> main

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (main)
$ |
```

5. Направете нова гранка во локалниот репозитор и додадете неколку промени на неа, а потоа прегледајте ги и зачувајте ги (commit) направените промени со помош на GIT.

Слика или GIT наредби:

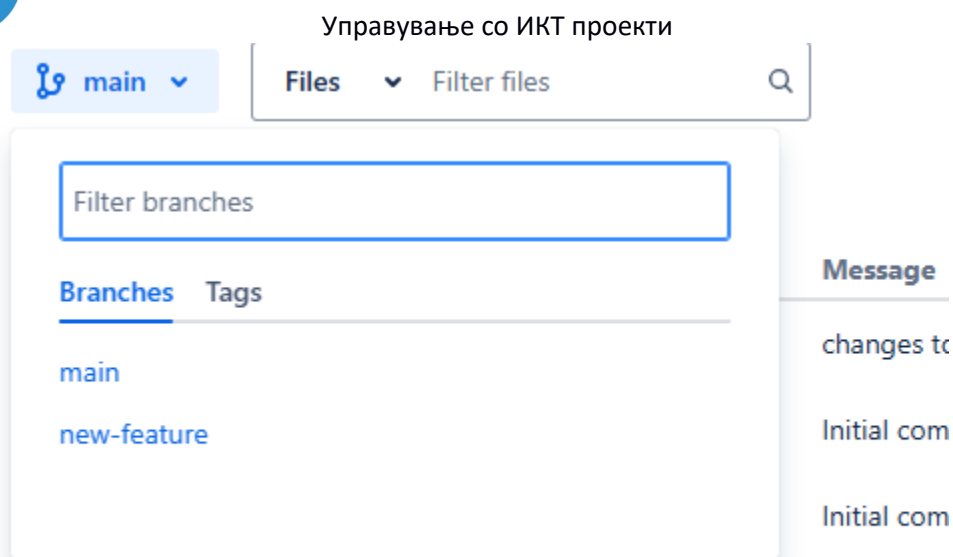
```
berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (main)
$ git checkout -b new-feature
Switched to a new branch 'new-feature'

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (new-feature)
$ git add *

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (new-feature)
$ git commit -m "new branch new feature who dis"
[new-feature 4fb9b0b] new branch new feature who dis
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 folderino/new feature.txt

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (new-feature)
$ git push origin new-feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create pull request for new-feature:
remote:   https://bitbucket.org/beratahmetaj216130/b02xz2/pull-requests/new?source=new-feature&t=1
remote:
To https://bitbucket.org/beratahmetaj216130/b02xz2.git
 * [new branch]      new-feature -> new-feature

berat@Berat MINGW64 ~/Downloads/gitski/b02xz2 (new-feature)
$
```



6. Прикачете ја новата гранка на Вашиот проект на Bitbucket.

Слика или GIT наредби:

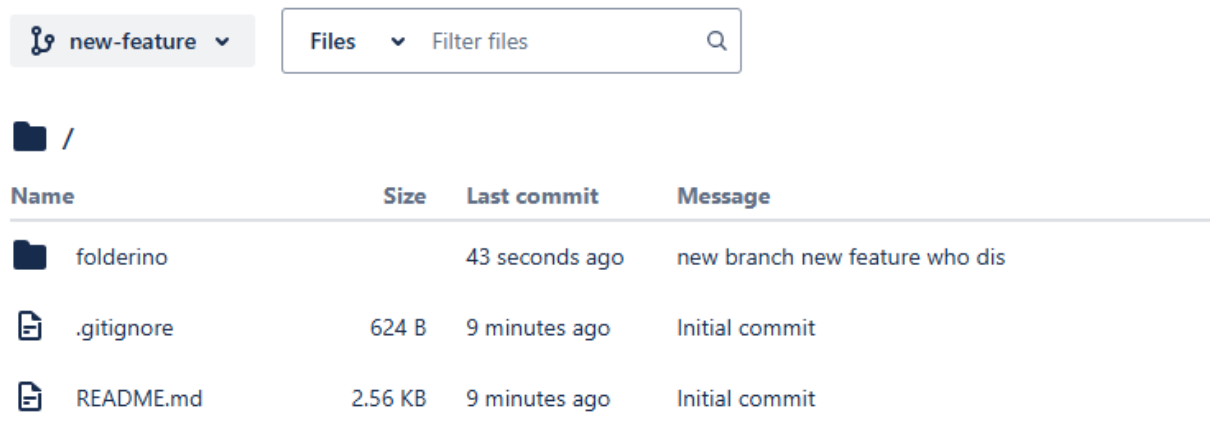
b02xz2

Invite

Clone



Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).



7. Споделете го Вашиот проект на Bitbucket со Ваш колега и побарајте од него да направи нова гранка со неколку промени и да ја прикачи (push) на Bitbucket. Потоа прегледајте ги промените на Bitbucket (на Веб).

Слика:



Add users or groups

stefan.angelovski@students.finki.ukim.mk

8. Преземете ги промените на Вашиот колега во Вашиот локален GIT репозитор.

Слика или GIT наредби:

```
cd ~/Downloads/gitski/b02xz2
```

```
git fetch
```

```
git checkout new-changes
```

```
git checkout main
```

```
git merge new-changes
```

```
git push origin main
```

9. Погледнете ги промените, што ги направил Вашиот колега и направете нова гранка во Вашиот локален репозитор и додадете неколку промени во истиот код каде што менувал и Вашиот колега. Потоа спојте ги (merge) гранките и разрешете ги настанатите конфликти.

Слика или GIT наредби:

```
cd ~/Downloads/gitski/b02xz2
```

```
git fetch
```

```
git checkout main
```

```
git checkout -b my-changes
```

```
git status
```

```
git add .
```

```
git commit -m "Додадени промени во истиот код што го менуваше колегата"
```

```
git push origin my-changes
```

```
git checkout my-changes
```

```
git merge new-changes
```

```
git push origin my-changes
```

10. Додадете маркер (tag) на гранката, што ја добивте со спојување.

Слика или GIT наредби:

```
git tag -a v1.0 -m "Означување на првата стабилна верзија по спојување"
```

```
git push origin v1.0
```



git push --tags

11. Направете неколку промени, прегледајте ги и поништете ги.

Слика или GIT наредби:

git add .

git reset HEAD file1.txt

git reset HEAD file2.txt

git commit -m "Some changes"

git reset --soft HEAD~1

git reset --hard HEAD~1

git push --force origin main

12. Означете делови од проектот (датотеки/фолдери) кои нема да се верзионираат (.gitignore) и проверете дали GIT ги игнорира по означувањето.

Слика или GIT наредби:

git add .gitignore

git commit -m "Додаден .gitignore за исклучување на несакани датотеки"

git push origin main

-Да, gitignore ги игнорира тие фајлови или екстензии кои се внесени во фајлот

13. Која GIT наредба/можност не беше наведена/потребна во претходните барања, а Вие сметате дека е корисна?

Слика или GIT наредби:

Повеќето се внесени, океј беше