

Instalacja, praca z gitem i githubem

Co to Git i Github:	1
Zakładanie konta na githubie:	1
Instalacja gita:	2
Konfiguracja:	3
Repozytoria:	4
Opisy komend:	6
Praca z Gitem:	6
Koniec	14

Co to Git i Github:

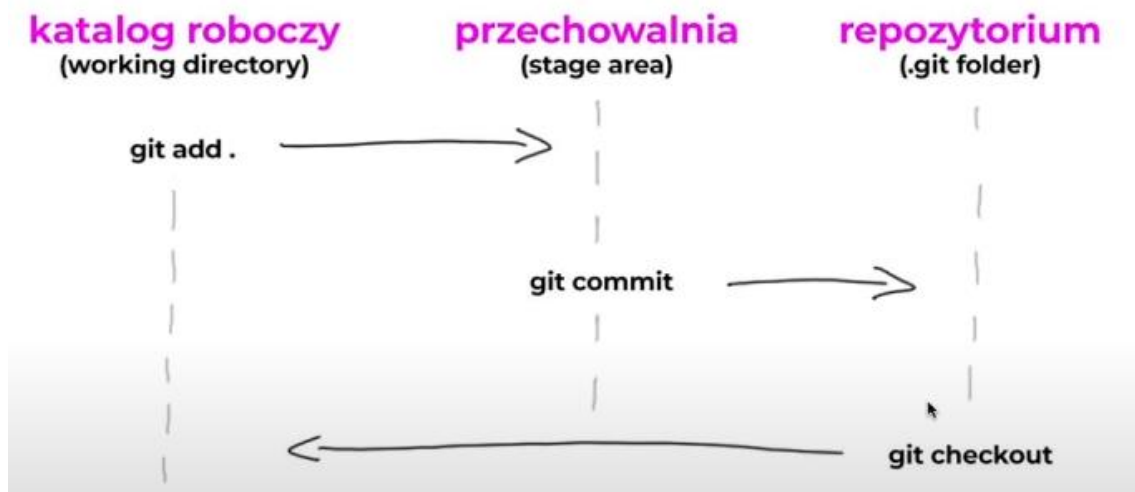
Git to rozproszony system kontroli wersji używany do śledzenia zmian w plikach i projektach.

Jest to narzędzie, które umożliwia programistom, zespołom deweloperskim i innym osobom zarządzanie historią zmian w kodzie źródłowym oraz współpracę nad projektem.

Daje dużą swobodę pracy nad kodem, nie martwiąc się o konflikty z kodem innych osób.

Natomiast GitHub to internetowa platforma hostingowa dla projektów opartych na systemie kontroli wersji Git.

Jest jednym z najpopularniejszych serwisów do przechowywania i współpracy nad kodem źródłowym projektów programistycznych



Zakładanie konta na githubie:

Żeby utworzyć konto należy udać się na stronę <https://github.com/signup> i wykonać prosty proces rejestracji.

Welcome to GitHub!
Let's begin the adventure

Enter your email*
✗

Create a password*
✗

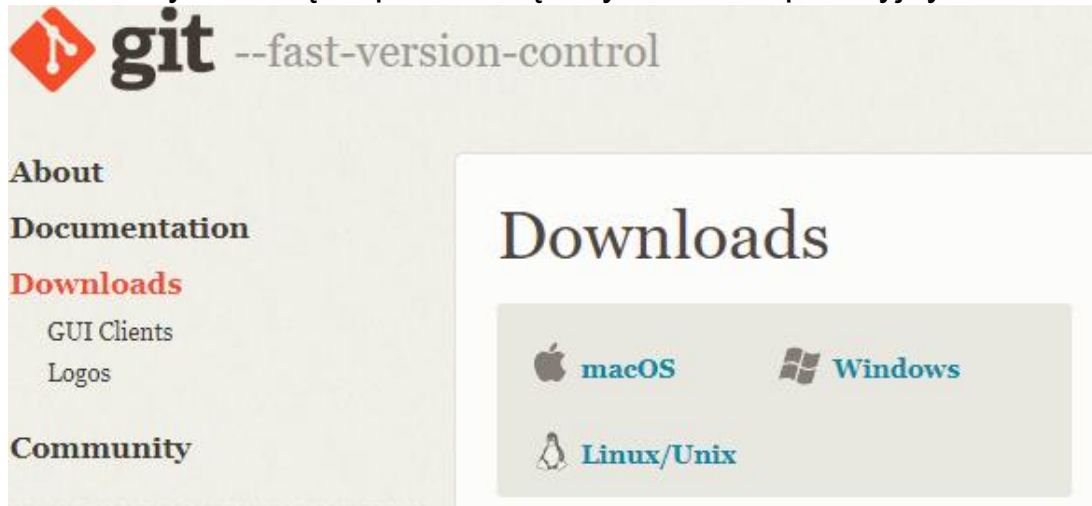
Enter a username*
→

Continue

Instalacja gita:

Aby zainstalować Gita:

1. Wchodzimy na stronę <https://git-scm.com/downloads>
2. Pobieramy instalkę odpowiednią z systemem operacyjnym



3. Pobieramy zachowując defaultowe ustawienia
4. Sprawdzamy czy wszystko się pobrało

```
MINGW64:/c/Users/PC

PC@3vil0S20 MINGW64 ~
$ git version
git version 2.42.0.windows.2

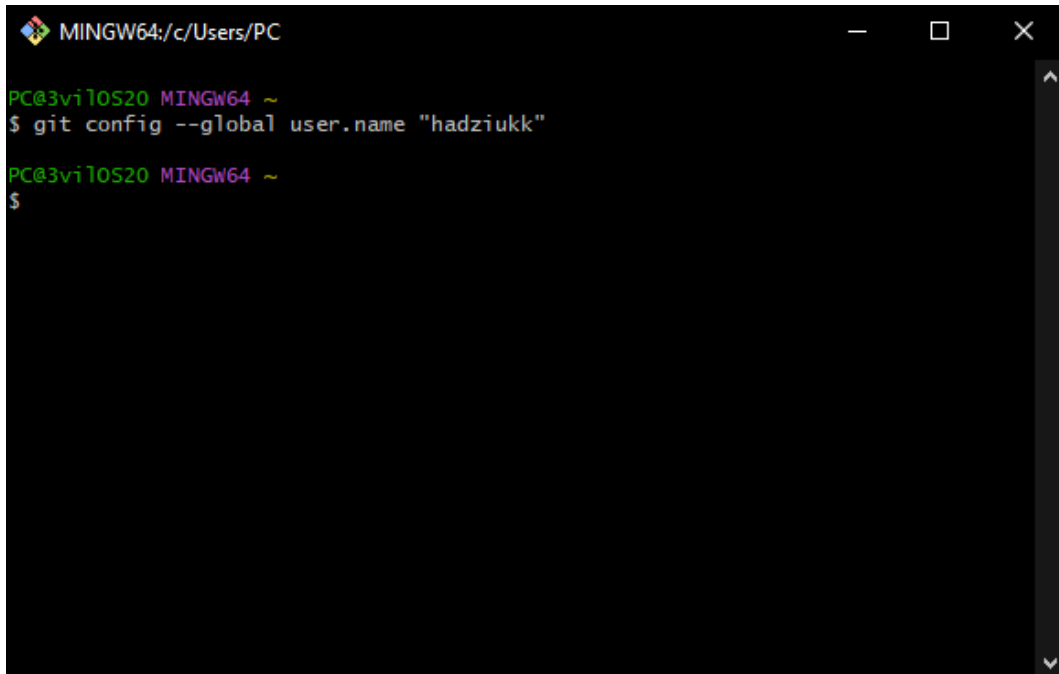
PC@3vil0S20 MINGW64 ~
$ .....
```

Konfiguracja:

Odpalamy git Bash'a w celu konfiguracji emailu oraz nazwy użytkownika.

Do konfiguracji nazwy użytkownika użyjemy komendy:

`git config --global user.name "nazwa"`

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/Users/PC'. The terminal content shows a prompt 'PC@3v10S20 MINGW64 ~' followed by the command '\$ git config --global user.name "hadziukk"'. The prompt returns to '\$' on the next line. A vertical scrollbar is visible on the right side of the terminal window.

```
MINGW64:/c/Users/PC
PC@3v10S20 MINGW64 ~
$ git config --global user.name "hadziukk"
PC@3v10S20 MINGW64 ~
$
```

Do konfiguracji emaila użyjemy komendy:

`git config --global user.email email`

```
MINGW64:/c/Users/PC

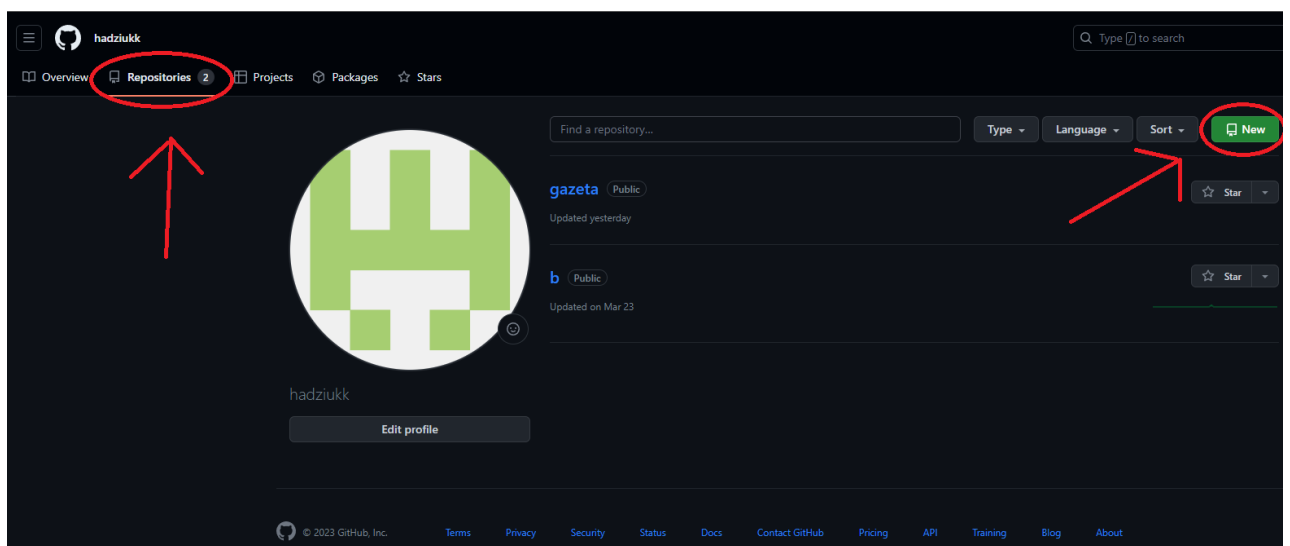
PC@3v10S20 MINGW64 ~
$ git config --global user.name "hadziukk"

PC@3v10S20 MINGW64 ~
$ git config --global user.email wikt0r.chudyy@onet.pl

PC@3v10S20 MINGW64 ~
$
```

Repozytoria:

Żeby utworzyć repozytorium na Githubie należy, otworzyć swój profil następnie wejść w zakładkę Repositories orak kliknąć [New](#)



Po czym uzupełnić nazwę repozytorium i kliknąć [Create repository](#)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * hadziukk / Repository name * test

✔ test is available.

Great repository names are short and memorable. Need inspiration? How about [miniature-succotash](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

[Create repository](#)

Opisy komend:

Branch- służy do zarządzania gałęziami

Commit- służy do zatwierdzania zmian w repozytorium Git, po wykonaniu polecenia zmiany zostaną wprowadzone do historii repozytorium.

Push- służy do wysyłania zmian z lokalnego repozytorium Git do zdalnego repozytorium.

Praca z Gitem:

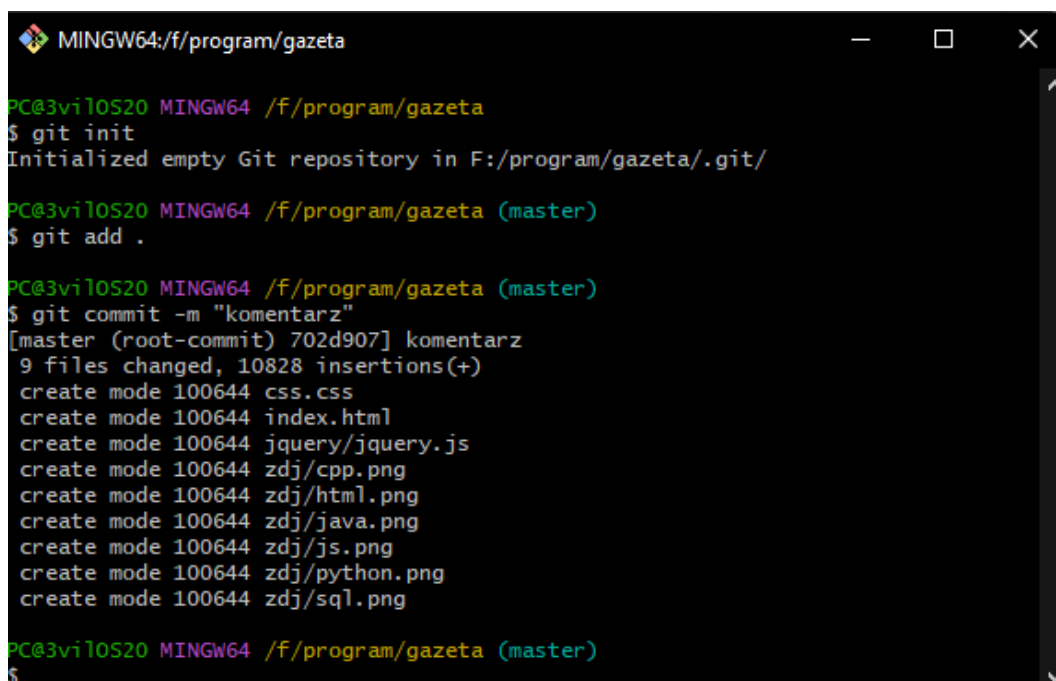
Aby dodać pliki do repozytorium należy użyć następujących komend w wybranym folderze:

`git init`

`git add .`

`git commit -m "komentarz"`

pierwsze 3 linie inicjują repozytorium, dodają wszystkie pliki do przechowalni(kropka oznacza wszystkie pliki, zamiast niej można użyć nazwy pliku), zatwierdza zmiany



```
PC@3vil0520 MINGW64 /f/program/gazeta
$ git init
Initialized empty Git repository in F:/program/gazeta/.git/

PC@3vil0520 MINGW64 /f/program/gazeta (master)
$ git add .

PC@3vil0520 MINGW64 /f/program/gazeta (master)
$ git commit -m "komentarz"
[master (root-commit) 702d907] komentarz
9 files changed, 10828 insertions(+)
create mode 100644 css.css
create mode 100644 index.html
create mode 100644 jquery/jquery.js
create mode 100644 zdj/cpp.png
create mode 100644 zdj/html.png
create mode 100644 zdj/java.png
create mode 100644 zdj/js.png
create mode 100644 zdj/python.png
create mode 100644 zdj/sql.png

PC@3vil0520 MINGW64 /f/program/gazeta (master)
$
```

`git branch -M main`

`git remote add origin https://github.com/hadziukk/test.git`

`git push -u origin main`

```
MINGW64:/f/program/gazeta

PC@3vi10S20 MINGW64 /f/program/gazeta (master)
$ git branch -M main

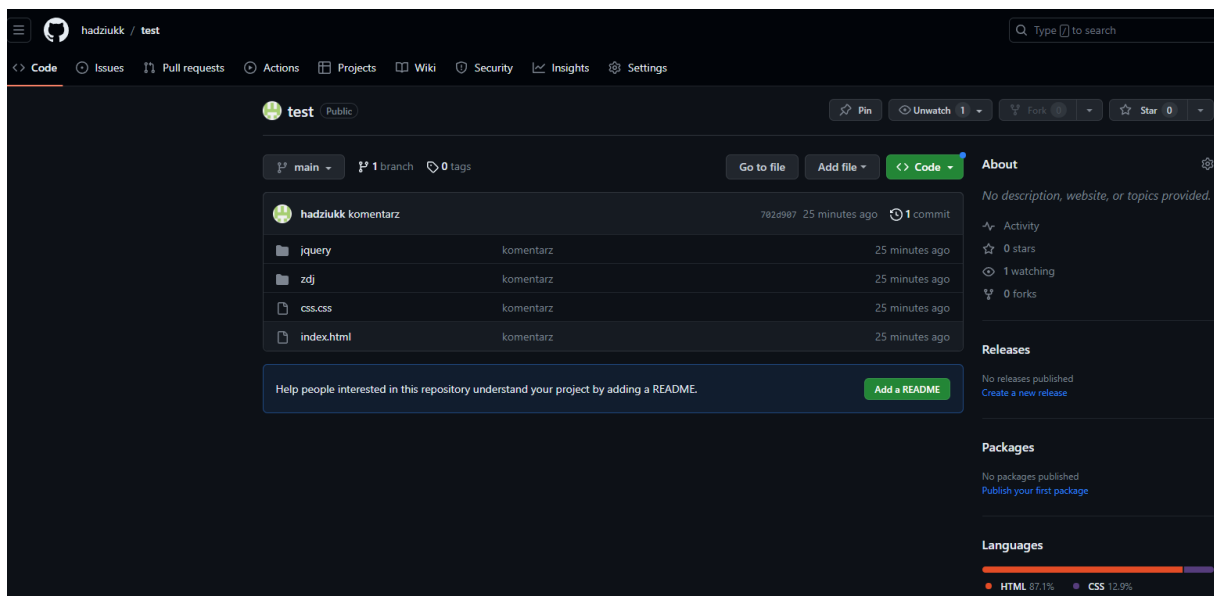
PC@3vi10S20 MINGW64 /f/program/gazeta (main)
$

PC@3vi10S20 MINGW64 /f/program/gazeta (main)
$ git remote add origin https://github.com/hadziukk/test.git

PC@3vi10S20 MINGW64 /f/program/gazeta (main)
$ git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 14 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 355.34 KiB | 23.69 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/hadziukk/test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

PC@3vi10S20 MINGW64 /f/program/gazeta (main)
$
```

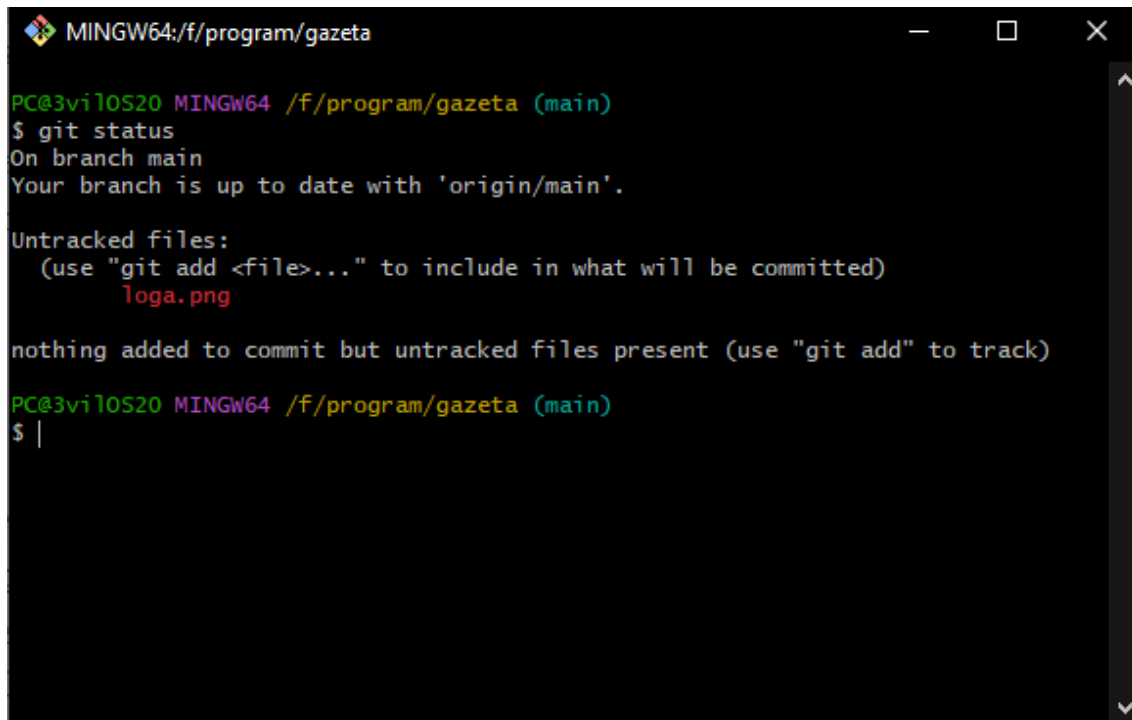
W taki sposób dodaliśmy pliki do repozytorium



git status

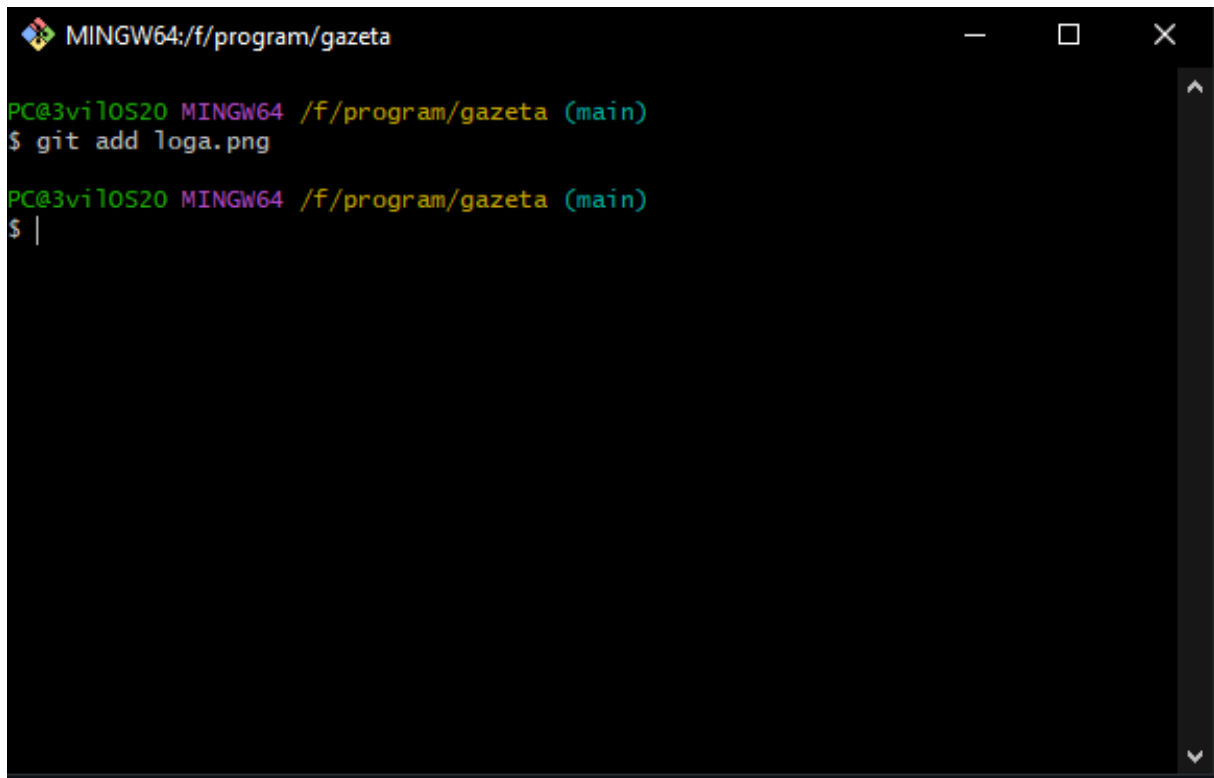
Wyświetla stan katalogu roboczego i przechowalni.

Screen został wykonany po dodaniu kolejnego pliku do tego folderu.

A screenshot of a terminal window titled 'MINGW64:/f/program/gazeta'. The prompt is 'PC@3v10S20 MINGW64 /f/program/gazeta (main)'. The user has entered '\$ git status'. The output shows 'On branch main', 'Your branch is up to date with 'origin/main'.', and 'Untracked files: (use "git add <file>..." to include in what will be committed)' followed by 'loga.png' in red. A final message says 'nothing added to commit but untracked files present (use "git add" to track)'. The prompt is now '\$ |'.

W celu dodania tego pliku do przechowalni należy użyć komendy

`git add loga.png`



```
MINGW64:/f/program/gazeta

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git add loga.png

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ |
```

A żeby zapisać go na githubie użyjemy

`git commit -m "dodanie loga"`

`git push -u origin main`



```
MINGW64:/f/program/gazeta

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git add loga.png

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git commit -m "dodanie loga"
[main fbe4490] dodanie loga
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 loga.png

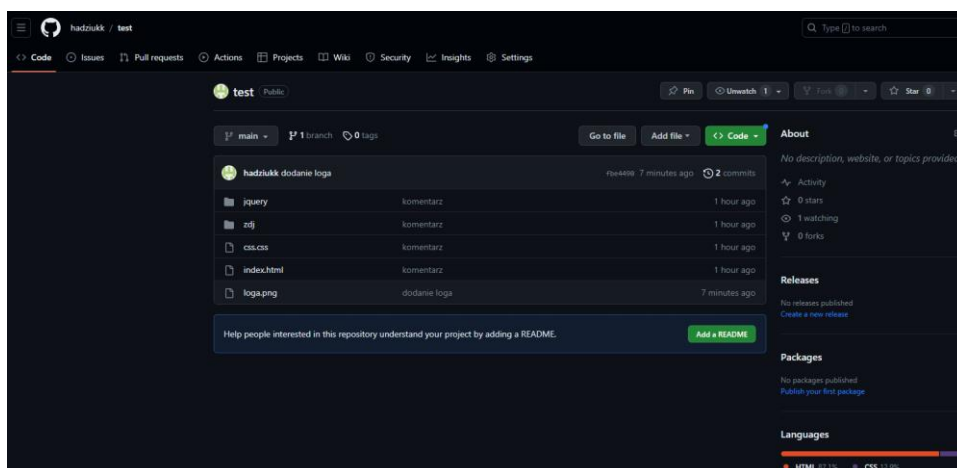
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ |
```

```
MINGW64:/f/program/gazeta

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 14 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.27 KiB | 1.27 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hadziukk/test.git
 702d907..fbc4490  main -> main
branch 'main' set up to track 'origin/main'.

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$
```

Spowodowało to dodanie pliku do repozytorium w githubie



Aby usunąć ten plik należy użyć

`git rm logo.png`

```
MINGW64:/f/program/gazeta

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git rm loga.png
rm 'loga.png'
```

Po czym oczywiście użyć commita i pusha

git commit -m "usuniecie loga"

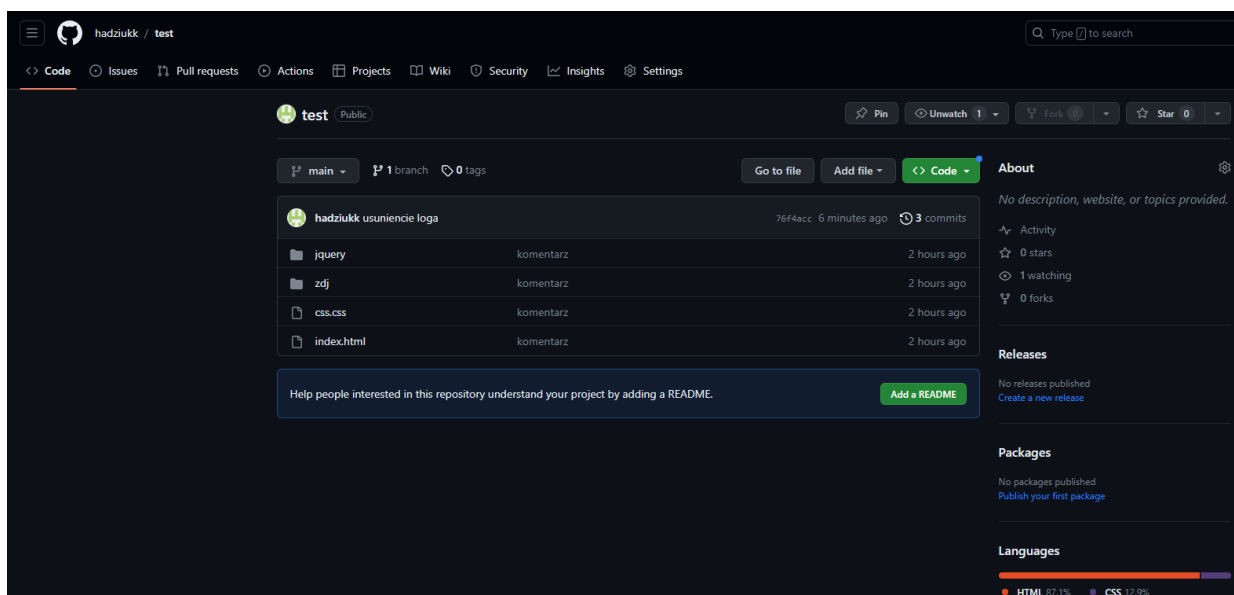
git push -u origin main

```
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git commit -m "usuniecie loga"
[main 76f4acc] usuniecie loga
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 loga.png

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 14 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 226 bytes | 226.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hadziukk/test.git
 fbe4490..76f4acc main -> main
branch 'main' set up to track 'origin/main'.

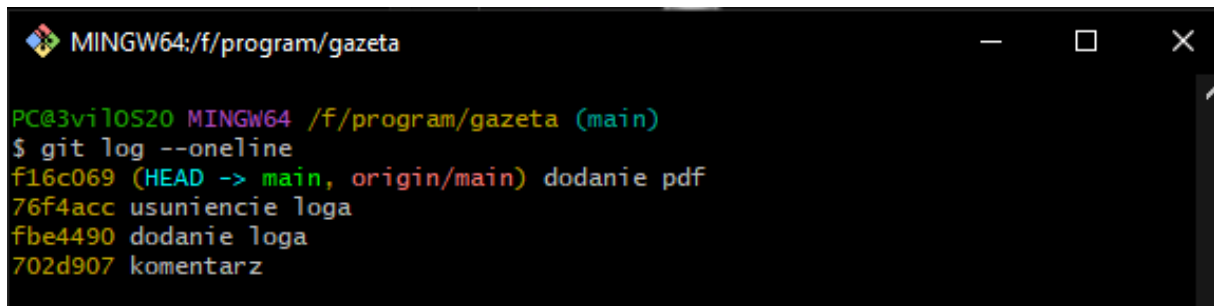
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ |
```

Otrzymamy



Do sprawdzenia historii commitów użyjemy polecenia:

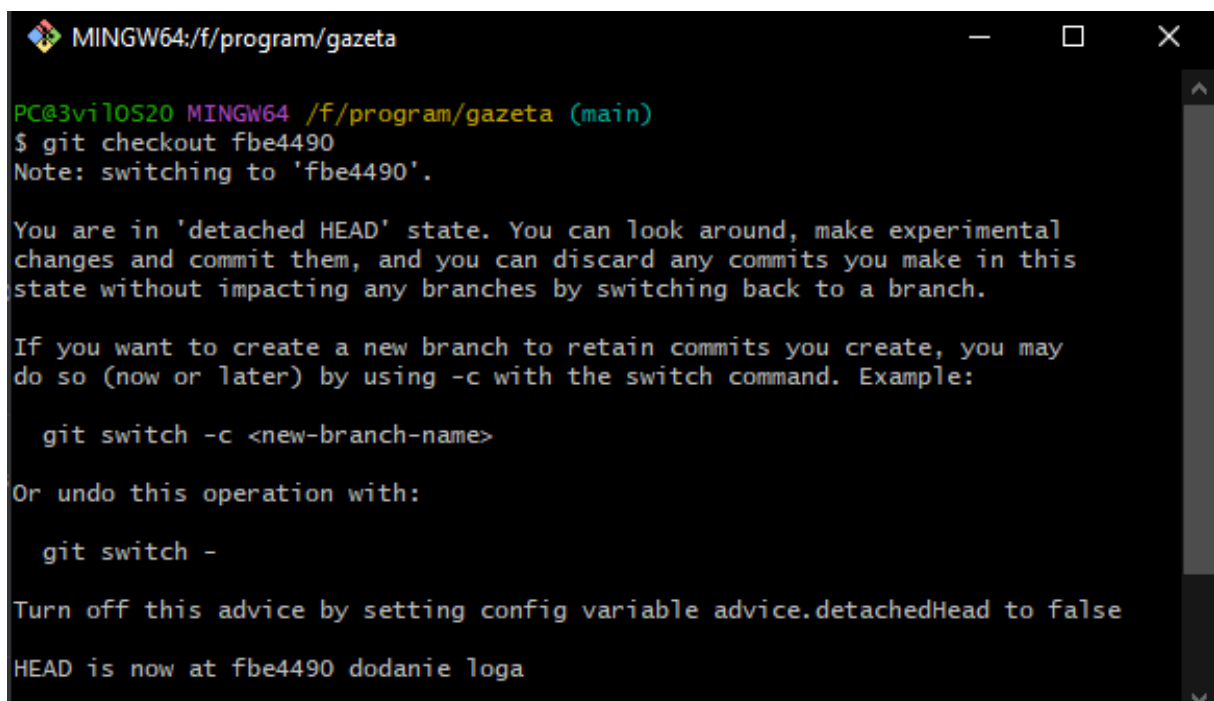
`git log --oneline`

A terminal window titled 'MINGW64:/f/program/gazeta' showing the output of the 'git log --oneline' command. The output lists four commits: f16c069 (HEAD -> main, origin/main) dodanie pdf, 76f4acc usunięcie loga, fbe4490 dodanie loga, and 702d907 komentarz.

```
MINGW64:/f/program/gazeta
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git log --oneline
f16c069 (HEAD -> main, origin/main) dodanie pdf
76f4acc usunięcie loga
fbe4490 dodanie loga
702d907 komentarz
```

Aby cofnąć wersję kopiujemy kod znajdujący się koło commita który nas interesuje i wykonujemy komendę:

`git checkout fbe4490`

A terminal window titled 'MINGW64:/f/program/gazeta' showing the output of the 'git checkout fbe4490' command. The output indicates switching to 'fbe4490' and provides information about the 'detached HEAD' state, including instructions on how to create a new branch or undo the operation.

```
MINGW64:/f/program/gazeta
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git checkout fbe4490
Note: switching to 'fbe4490'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>







Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at fbe4490 dodanie loga
```

Spowodowało to cofnięcie się do momentu w którym mamy jeszcze logo

Ten komputer > Nowy (F:) > program > gazeta		
Nazwa	Typ	Rozmiar
 .git	Folder plików	
 jquery	Folder plików	
 zdj	Folder plików	
 css.css	Dokument Arkusz ...	1 KB
 index.html	Chrome HTML Do...	5 KB
 loga.png	Plik PNG	2 KB

Koniec