

# 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: .....	5
Praca nad repozytorium: .....	6
Koniec .....	11

## Co to Git i Github:

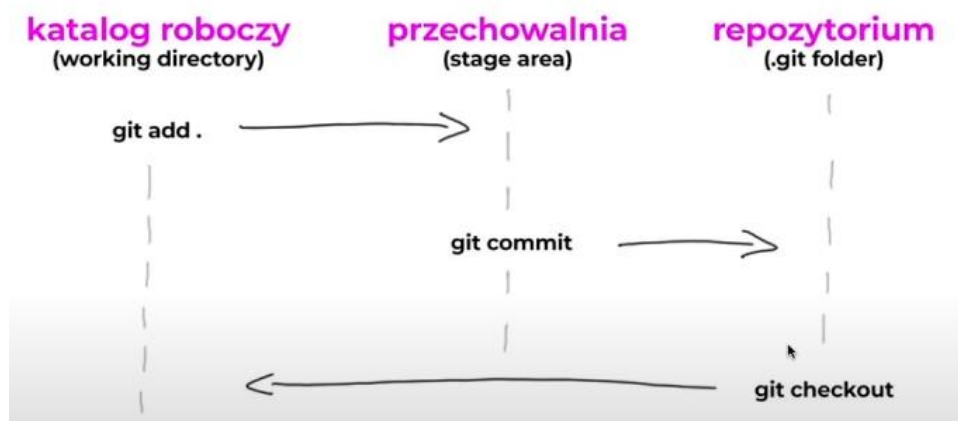
Git to rozproszony system kontroli wersji używany do śledzenia zmian w plikach i projektach.

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\*  
✗

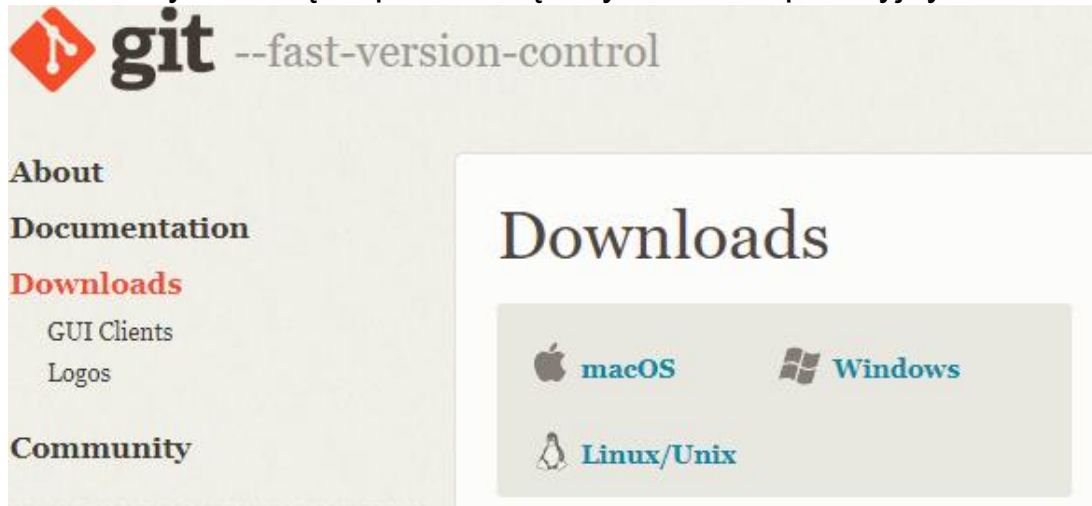
Enter a username\*  
→

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

The image shows a terminal window titled 'MINGW64:/c/Users/PC'. The prompt is 'PC@3vi10S20 MINGW64 ~'. The user enters '\$ git version' and the output is 'git version 2.42.0.windows.2'. The prompt then changes to '\$ :'.

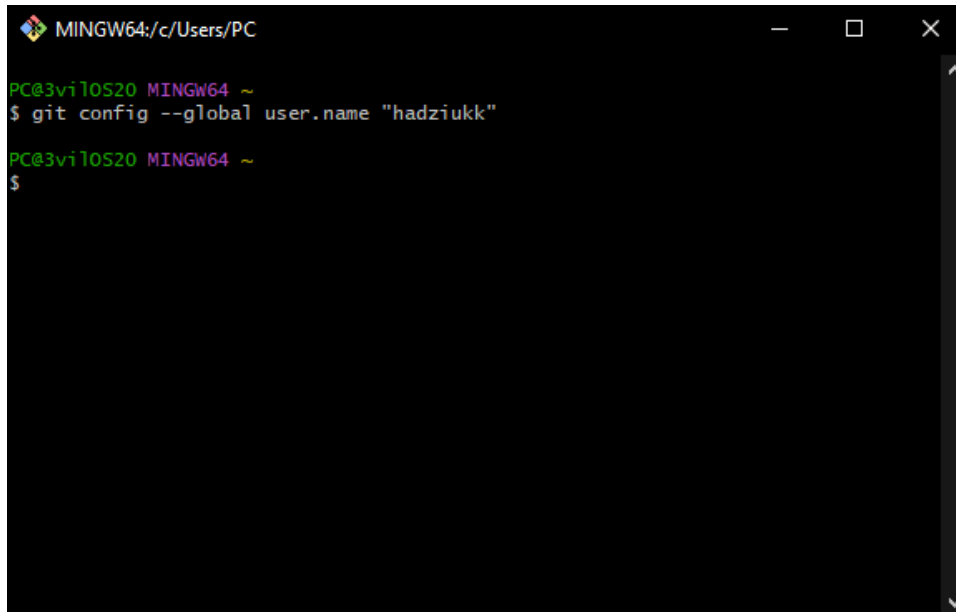
```
PC@3vi10S20 MINGW64 ~
$ git version
git version 2.42.0.windows.2
PC@3vi10S20 MINGW64 ~
$ :
```

## Konfiguracja:

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

Do konfiguracji nazwy użytkownika użyjemy komendy:

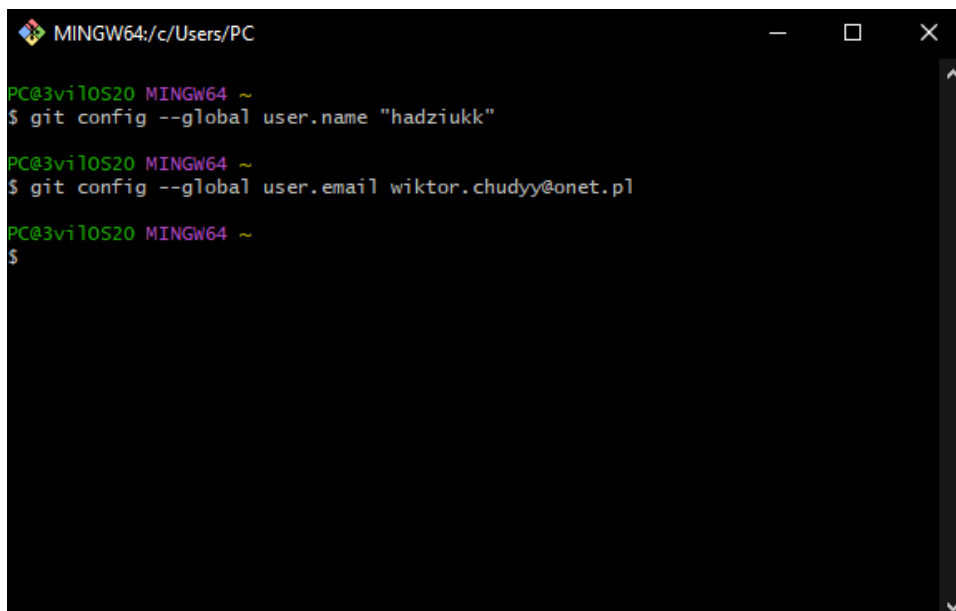
`git config --global user.name "nazwa"`

A terminal window titled 'MINGW64:/c/Users/PC' with standard window controls. The prompt is 'PC@3v10S20 MINGW64 ~'. The command '\$ git config --global user.name "hadziukk"' is entered and executed. The prompt returns to '\$'.

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

Do konfiguracji emaila użyjemy komendy:

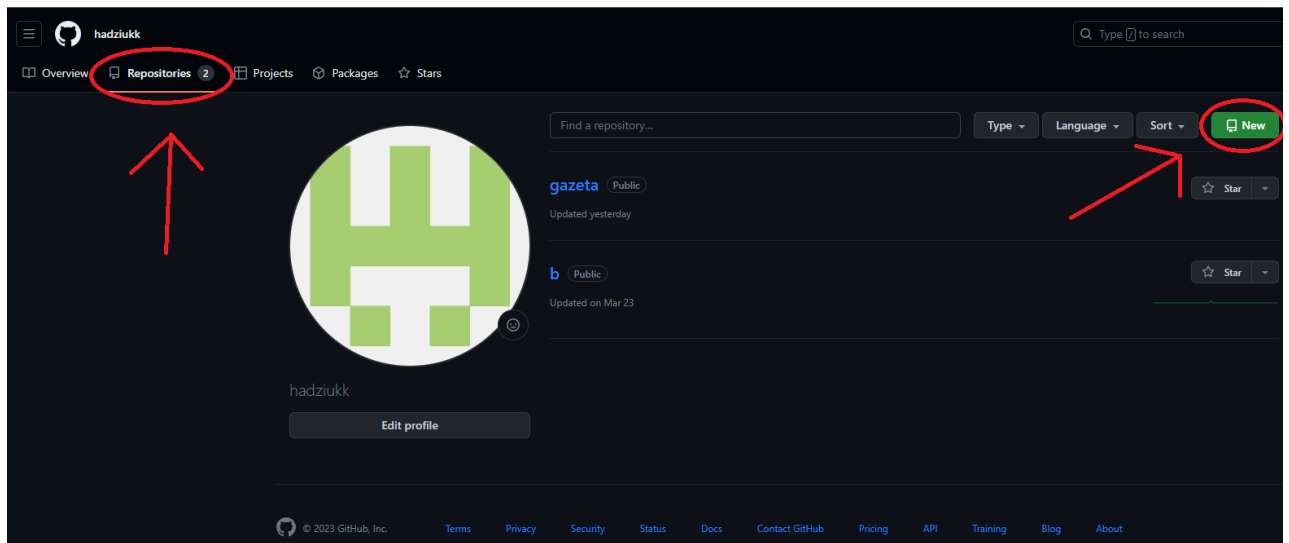
`git config --global user.email email`

A terminal window titled 'MINGW64:/c/Users/PC' with standard window controls. The prompt is 'PC@3v10S20 MINGW64 ~'. The command '\$ git config --global user.name "hadziukk"' is entered and executed. The prompt returns to '\$'. The second command '\$ git config --global user.email wikt0r.chudyy@onet.pl' is entered and executed. The prompt returns to '\$'.

```
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 \*

Repository name \*

hadziukk

/ 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 nad repozytorium:

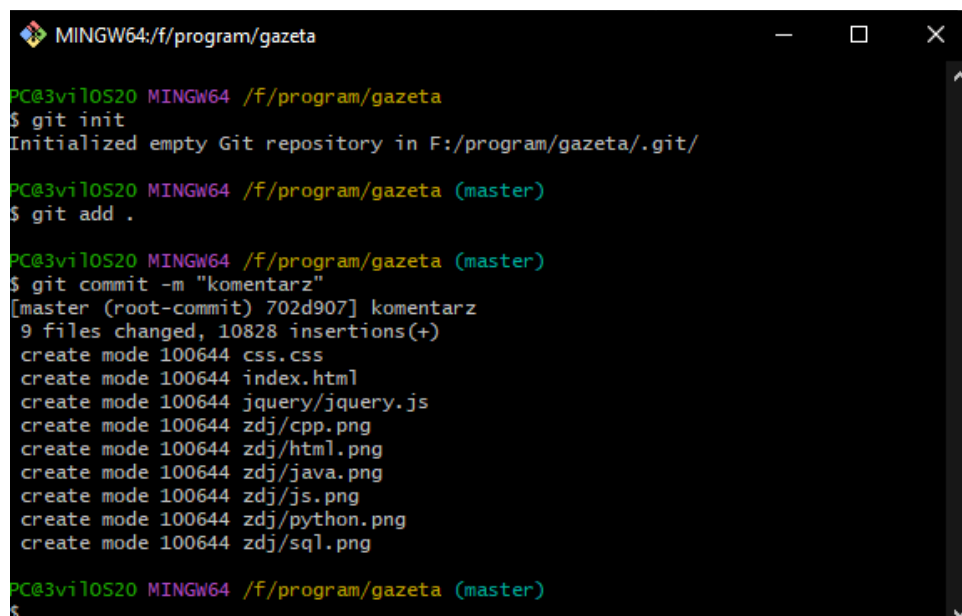
Aby dodać pliki do repozytorium należy użyć następujących komend w wybranym folderze:

`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

A screenshot of a terminal window titled 'MINGW64:/f/program/gazeta'. The terminal shows the following commands and output:

```
PC@3vil0S20 MINGW64 /f/program/gazeta
$ git init
Initialized empty Git repository in F:/program/gazeta/.git/

PC@3vil0S20 MINGW64 /f/program/gazeta (master)
$ git add .

PC@3vil0S20 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@3vil0S20 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@3vil0S20 MINGW64 /f/program/gazeta (master)
$ git branch -M main

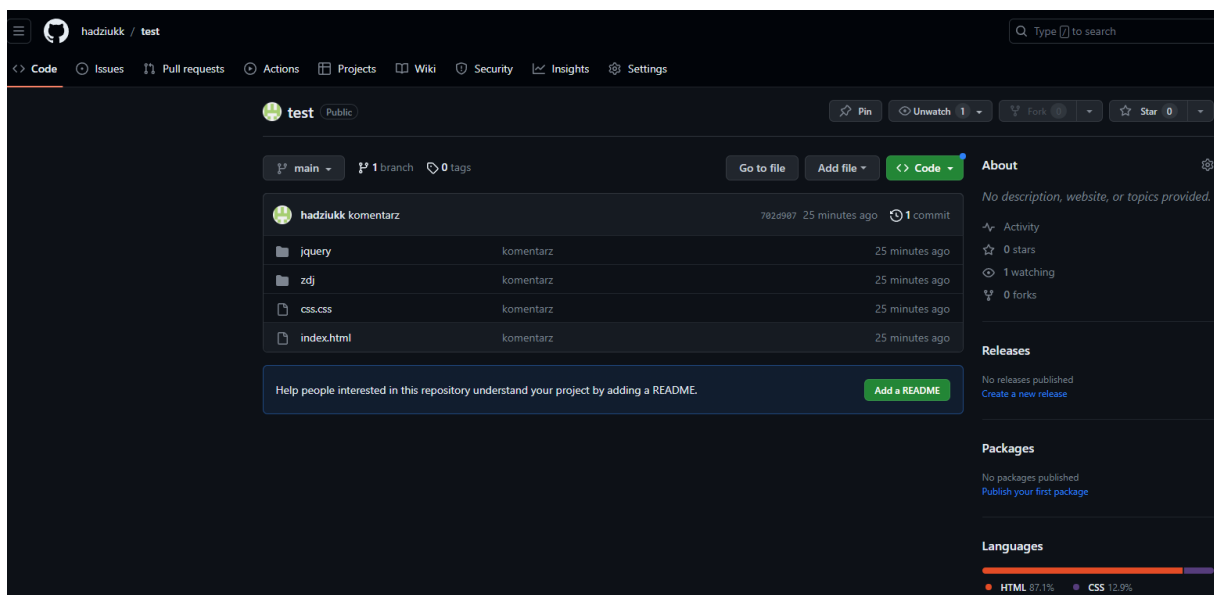
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$

PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git remote add origin https://github.com/hadziukk/test.git

PC@3vil0S20 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@3vil0S20 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.

```
MINGW64:/f/program/gazeta

PC@3v10S20 MINGW64 /f/program/gazeta (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    loga.png

nothing added to commit but untracked files present (use "git add" to track)
PC@3v10S20 MINGW64 /f/program/gazeta (main)
$ |
```

W celu dodania tego pliku do przechowalni naleŹ użyć komendy

`git add loga.png`

```
MINGW64:/f/program/gazeta

PC@3v10S20 MINGW64 /f/program/gazeta (main)
$ git add loga.png

PC@3v10S20 MINGW64 /f/program/gazeta (main)
$ |
```

A Źeby 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

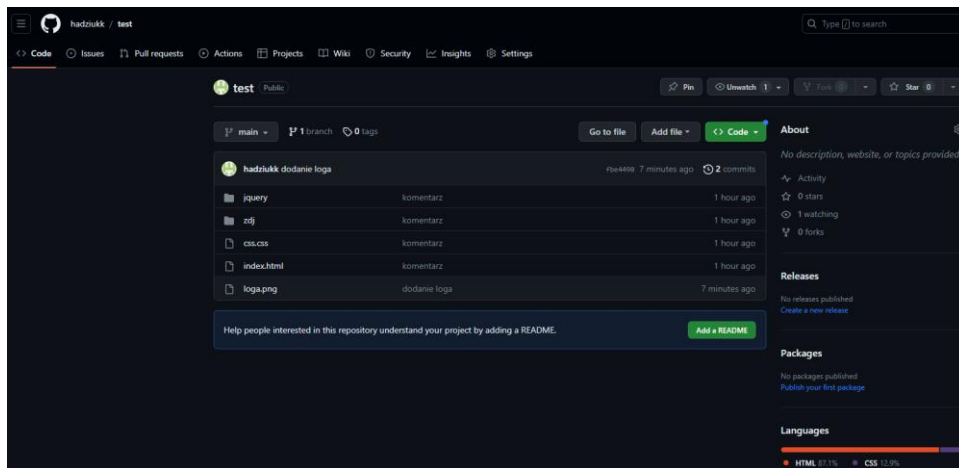
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ |
```

```
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..fbe4490  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 loga.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"
```

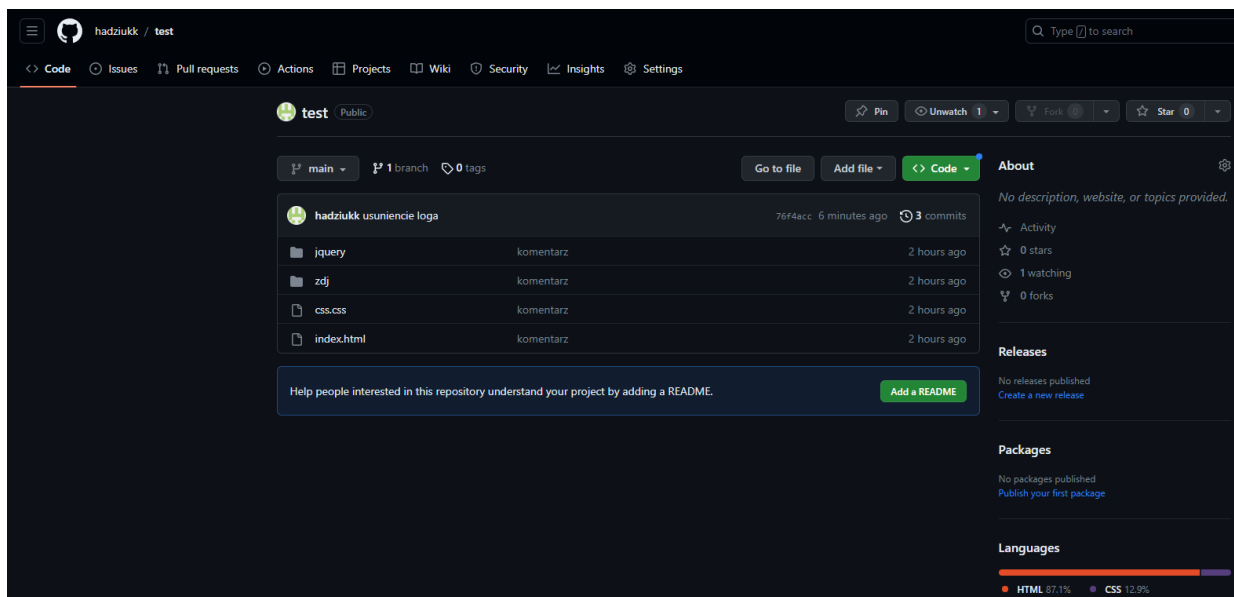
```
git push -u origin main
```

```
PC@3vil0S20 MINGW64 /f/program/gazeta (main)
$ git commit -m "usuniecie loga"
[main 76f4acc] usunięcie 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



Koniec