

Menedżerska Akademia Nauk Stosowanych w Warszawie

Imię i nazwisko studenta: Mateusz Plata

Imię i nazwisko prowadzącego: Jakub Słowik

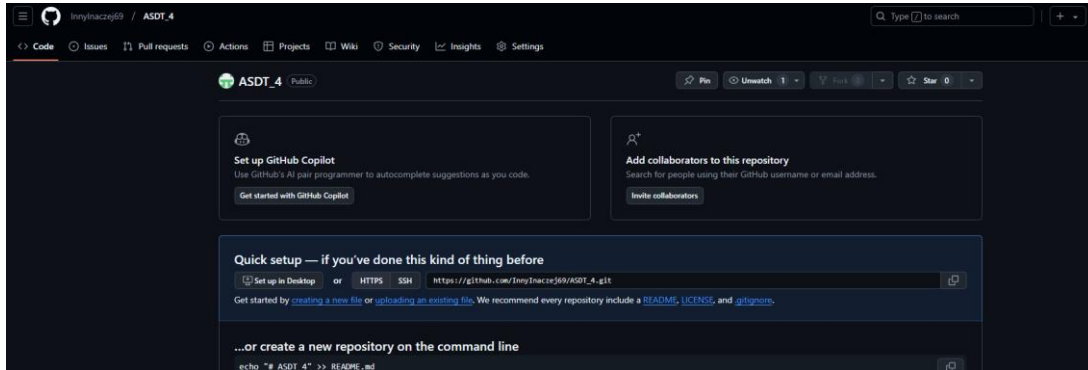
**Nazwa przedmiotu: Zaawansowane techniki tworzenia
oprogramowania**

Numer grupy: 55DP

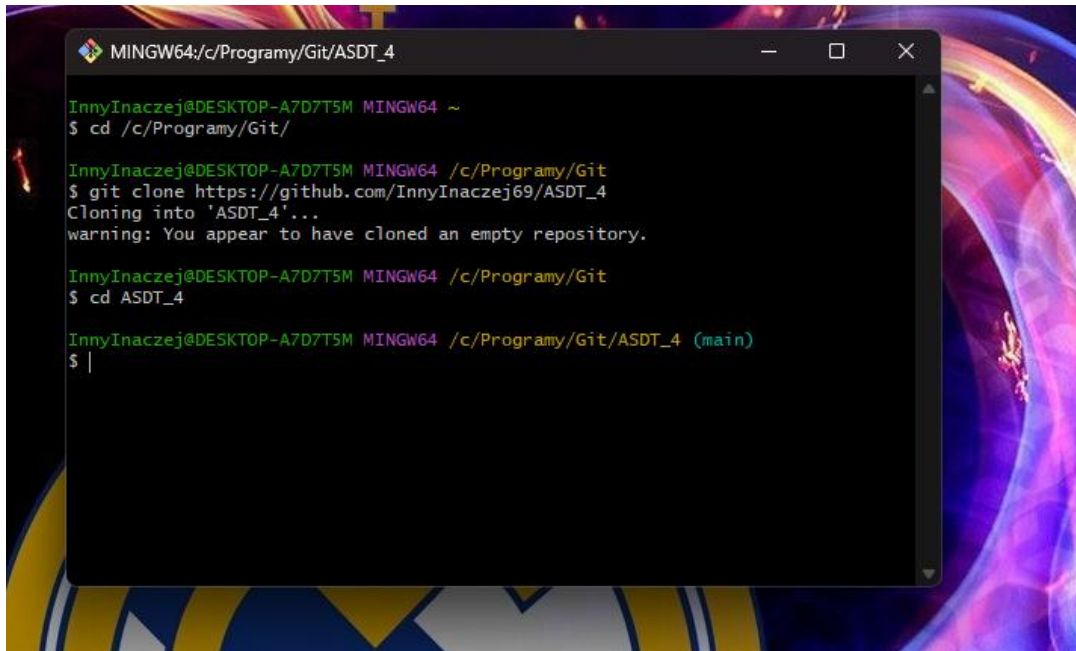
Data: 13.12.2024

1. Utworzenie prywatnego repozytorium na GitHubie

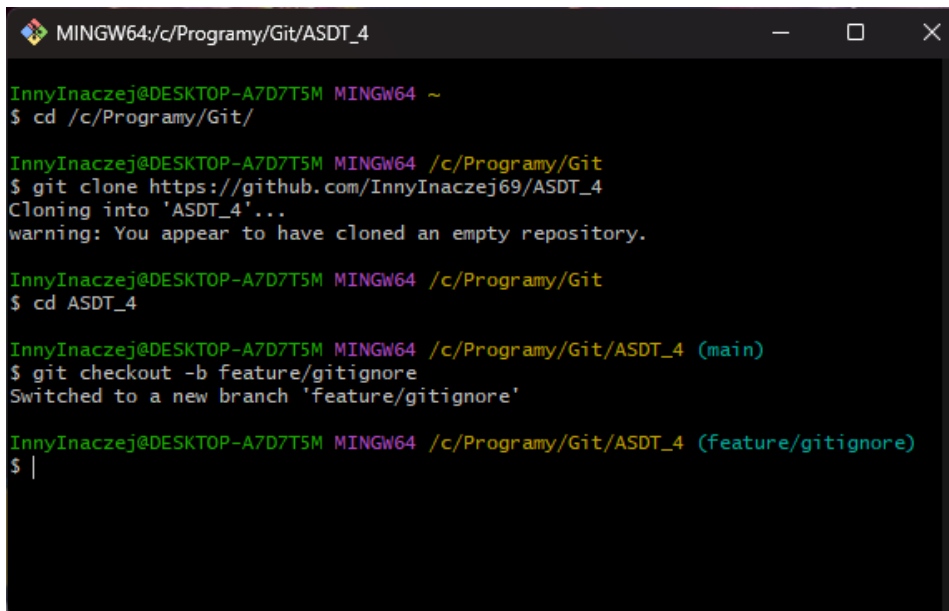
Pierwszym krokiem było utworzenie prywatnego repozytorium o nazwie **ASDT_4** na platformie GitHub. Repozytorium to zostało utworzone w trybie prywatnym, aby było dostępne tylko dla upoważnionych użytkowników.



2. Sklonowałem nowe repozytorium lokalnie.



3. Utworzyłem nową gałąź o nazwie feature/gitignore poleceniem `git checkout -b feature/gitignore`.



```
MINGW64:/c/Programy/Git/ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 ~
$ cd /c/Programy/Git/

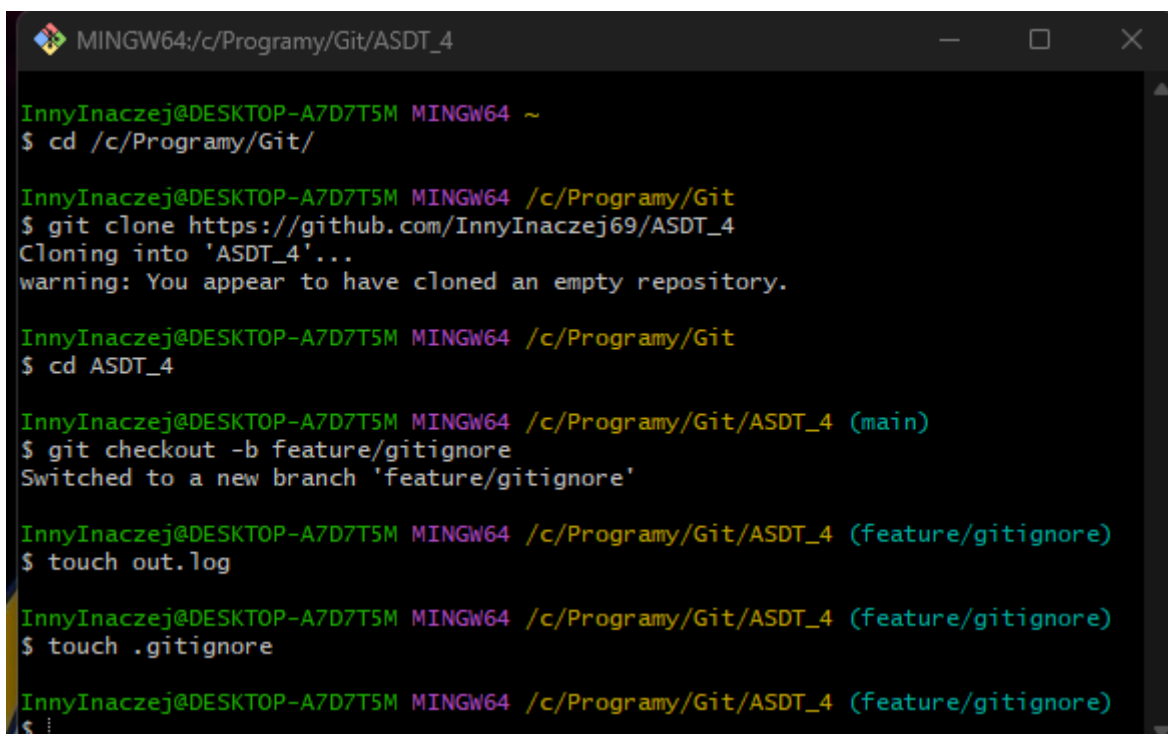
InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ git clone https://github.com/InnyInaczej69/ASDT_4
Cloning into 'ASDT_4'...
warning: You appear to have cloned an empty repository.

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ cd ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (main)
$ git checkout -b feature/gitignore
Switched to a new branch 'feature/gitignore'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ |
```

4. W repozytorium stworzyłem nowy plik `out.log`, używając komendy `touch out.log`.



```
MINGW64:/c/Programy/Git/ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 ~
$ cd /c/Programy/Git/

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ git clone https://github.com/InnyInaczej69/ASDT_4
Cloning into 'ASDT_4'...
warning: You appear to have cloned an empty repository.

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ cd ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (main)
$ git checkout -b feature/gitignore
Switched to a new branch 'feature/gitignore'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ touch out.log

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ touch .gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ |
```

5. W repozytorium utworzyłem plik .gitignore za pomocą komendy touch .gitignore, a następnie edytowałem go, dodając regułę ignorującą wszystkie pliki z rozszerzeniem .log. Treść pliku .gitignore to *.log.

```
MINGW64:/c/Programy/Git/ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ git clone https://github.com/InnyInaczej69/ASDT_4
Cloning into 'ASDT_4'...
warning: You appear to have cloned an empty repository.

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git
$ cd ASDT_4

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (main)
$ git checkout -b feature/gitignore
Switched to a new branch 'feature/gitignore'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ touch out.log

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ touch .gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ echo "*.log" >> .gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$
```

6. Zmiany zostały dodane do staging area komendą git add ., następnie skomitowane poleceniem git commit -m "Dodano .gitignore i regułę ignorowania plików .log", a na końcu wypchnięte na zdalne repozytorium na gałąź feature/gitignore za pomocą git push.

```
MINGW64:/c/Programy/Git/ASDT_4

$ touch .gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ echo "*.log" >> .gitignore

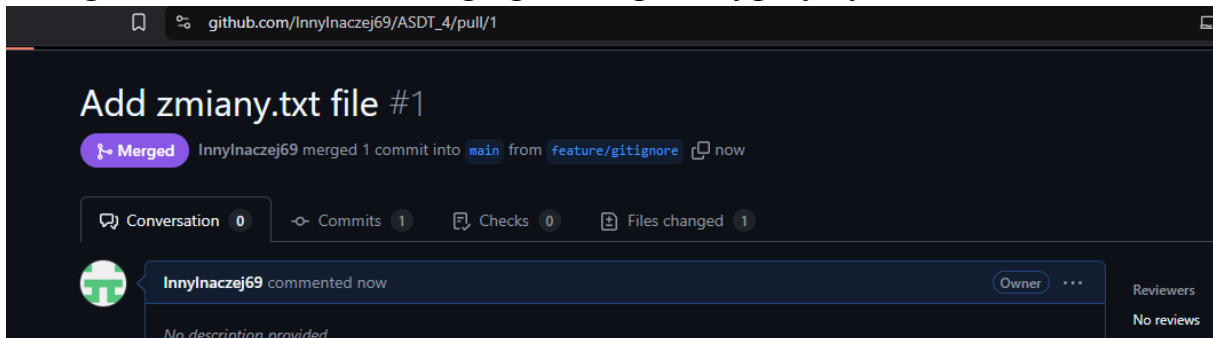
InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git commit -m "Add .gitignore and out.log"
[feature/gitignore (root-commit) 5ae1502] Add .gitignore and out.log
1 file changed, 1 insertion(+)
create mode 100644 .gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git push origin feature/gitignore
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 236 bytes | 236.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/InnyInaczej69/ASDT_4
 * [new branch]      feature/gitignore -> feature/gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$
```

7 Utworzyłem pull request w serwisie GitHub i po jego zatwierdzeniu zmergowałem branch feature/gitignore z główną gałęzią main.



8. Wyświetliłem bieżący wykres historii repozytorium za pomocą polecenia `git log`, które pokazuje szczegółowe informacje o commitach, takie jak identyfikator commitów (hash), autora, datę oraz wiadomości dołączone do commitów. Dodatkowo, używając opcji takich jak `--oneline`, `--graph` lub `--all`, mogłem uzyskać skrócony widok, zobaczyć strukturę gałęzi oraz pełną historię repozytorium w bardziej czytelnej formie.

```
MINGW64:/c/Programy/Git/ASDT_4
1 file changed, 1 insertion(+)
create mode 100644 zmiany.txt

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 373 bytes | 373.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/InnyInaczej69/ASDT_4
   583dcbd..f201cfd feature/gitignore -> feature/gitignore

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git log --oneline --graph --all
* f201cfd (HEAD -> feature/gitignore, origin/feature/gitignore) Add zmiany.txt f
file
* 583dcbd (origin/main, main) Update .gitignore to ignore .log files
* 4305c51 Add .gitignore file
* 1315c6c Add out.log file

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ |
```

9. Stworzyłem nowy tag dla *main* brancha z plikiem wyświetlającym „Hello word”. Nazywając go „*Hello world TAG*”.

```
InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ echo "Hello world" > hello.txt

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git add hello.txt
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git commit -m "Add Hello world file"
[feature/gitignore b1e14d7] Add Hello world file
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello world TAG"
fatal: 'Hello world TAG' is not a valid tag name.

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ ^C

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello world TAG"
fatal: 'Hello world TAG' is not a valid tag name.

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag Hello\ world\ TAG
fatal: 'Hello world TAG' is not a valid tag name.

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-world-TAG"

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ |
```

10. Utworzyłem kolejny tag Hello ASDT world TAG.

```
MINGW64;c/Programy/Git/ASDT_4
fatal: 'Hello world TAG' is not a valid tag name.

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-world-TAG"

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ echo "Hello ASDT world" > hello_asdt.txt

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git add hello_asdt.txt
warning: in the working copy of 'hello_asdt.txt', LF will be replaced by CRLF the next time Git touches it

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git commit -m "Add Hello ASDT world file"
[feature/gitignore 33b465f] Add Hello ASDT world file
1 file changed, 1 insertion(+)
create mode 100644 hello_asdt.txt

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-ASDT-world-TAG"

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ |
```

11. Stworzyłem nowy tag dla *main* brancha z plikiem wyświetlającym „Hello ASDT world with stashing”. Nazwałem go „*Hello ASDT world with stashing TAG*”.

```
MINGW64:/c/Programy/Git/ASDT_4
create mode 100644 hello_asdt.txt

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-ASDT-world-TAG"

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ echo "Hello ASDT world with stashing" > hello_asdt_stash.txt

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git add hello_asdt_stash.txt
warning: in the working copy of 'hello_asdt_stash.txt', LF will be replaced by C
RLF the next time Git touches it

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git commit -m "Add Hello ASDT world with stashing file"
[feature/gitignore c4deb53] Add Hello ASDT world with stashing file
1 file changed, 1 insertion(+)
create mode 100644 hello_asdt_stash.txt

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-ASDT-world-with-stashing-TAG"

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$
```

12. Spushowałem wszystkie tagi do zdalnego repozytorium za pomocą polecenia `git push origin --tags`. Polecenie to przesyła wszystkie lokalne tagi, zarówno lekkie, jak i adnotowane, do wskazanego zdalnego repozytorium (w tym przypadku origin). Dzięki temu tagi są dostępne dla innych użytkowników pracujących na tym repozytorium.

```
MINGW64:/c/Programy/Git/ASDT_4
[feature/gitignore c4deb53] Add Hello ASDT world with stashing file
1 file changed, 1 insertion(+)
create mode 100644 hello_asdt_stash.txt

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag "Hello-ASDT-world-with-stashing-TAG"

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git push origin --tags
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 905 bytes | 905.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/InnyInaczrej69/ASDT_4
 * [new tag]      Hello-ASDT-world-TAG -> Hello-ASDT-world-TAG
 * [new tag]      Hello-ASDT-world-with-stashing-TAG -> Hello-ASDT-world-with
-stashing-TAG
 * [new tag]      Hello-world-TAG -> Hello-world-TAG

InnyInaczrej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$
```


13. Wyświetliłem wszystkie tagi w repozytorium za pomocą polecenia `git tag`, które listuje wszystkie utworzone tagi w porządku alfabetycznym.

```
MINGW64:/c/Programy/Git/ASDT_4
InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git push origin --tags
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 905 bytes | 905.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/InnyInaczzej69/ASDT_4
 * [new tag]          Hello-ASDT-world-TAG -> Hello-ASDT-world-TAG
 * [new tag]          Hello-ASDT-world-with-stashing-TAG -> Hello-ASDT-world-with-stashing-TAG
 * [new tag]          Hello-world-TAG -> Hello-world-TAG

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag
Hello-ASDT-world-TAG
Hello-ASDT-world-with-stashing-TAG
Hello-world-TAG

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$
```

14. Utworzyłem nową gałąź `feature/hello_world_asdt_from_tag` z istniejącego tagu `Hello ASDT world TAG` za pomocą polecenia `git checkout -b feature/hello_world_asdt_from_tag Hello ASDT world TAG`. Dzięki temu gałąź zaczyna się dokładnie w punkcie, który został oznaczony przez ten tag.

```
MINGW64:/c/Programy/Git/ASDT_4
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/InnyInaczzej69/ASDT_4
 * [new tag]          Hello-ASDT-world-TAG -> Hello-ASDT-world-TAG
 * [new tag]          Hello-ASDT-world-with-stashing-TAG -> Hello-ASDT-world-with-stashing-TAG
 * [new tag]          Hello-world-TAG -> Hello-world-TAG

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git tag
Hello-ASDT-world-TAG
Hello-ASDT-world-with-stashing-TAG
Hello-world-TAG

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git checkout -b feature/hello_world_asdt_from_tag "Hello ASDT world TAG"
fatal: 'Hello ASDT world TAG' is not a commit and a branch 'feature/hello_world_asdt_from_tag' cannot be created from it

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git checkout -b feature/hello_world_asdt_from_tag "Hello-ASDT-world-TAG"
Switched to a new branch 'feature/hello_world_asdt_from_tag'

InnyInaczzej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$
```


15. Stworzyłem plik programu w Pythonie, a następnie zmodyfikowałem go, aby wyświetlał na terminalu komunikat Hello ASDT world with tagging

```
MINGW64:/c/Programy/Git/ASDT_4
$ git checkout -b feature/hello_world_asdt_from_tag "Hello ASDT world TAG"
fatal: 'Hello ASDT world TAG' is not a commit and a branch 'feature/hello_world_
asdt_from_tag' cannot be created from it

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/gitignore)
$ git checkout -b feature/hello_world_asdt_from_tag "Hello-ASDT-world-TAG"
Switched to a new branch 'feature/hello_world_asdt_from_tag'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ print("Hello ASDT world with tagging")
bash: syntax error near unexpected token `"Hello ASDT world with tagging"'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ print("Hello-ASDT-world-with-tagging")
bash: syntax error near unexpected token `"Hello-ASDT-world-with-tagging"'

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ touch hello_asdt.py

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ nano hello_asdt.py

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$
```

```
MINGW64:/c/Programy/Git/ASDT_4
GNU nano 8.2 hello_asdt.py
print("Hello ASDT world with tagging")

[ Read 1 line ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

16. Zmiany zostały skomitowane i spushowane do zdalnego repozytorium

```
MINGW64:/c/Programy/Git/ASDT_4
InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ git commit -m "Update file to display Hello ASDT world with tagging"
[feature/hello_world_asdt_from_tag c852ebb] Update file to display Hello ASDT world with tagging
1 file changed, 1 insertion(+)
create mode 100644 hello_asdt.py

InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$ git push origin feature/hello_world_asdt_from_tag
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/hello_world_asdt_from_tag' on GitHub by visiting:
remote:   https://github.com/InnyInaczej69/ASDT_4/pull/new/feature/hello_world_asdt_from_tag
remote:
To https://github.com/InnyInaczej69/ASDT_4
 * [new branch]      feature/hello_world_asdt_from_tag -> feature/hello_world_asdt_from_tag


InnyInaczej@DESKTOP-A7D7T5M MINGW64 /c/Programy/Git/ASDT_4 (feature/hello_world_asdt_from_tag)
$
```

17. Utworzyłem pull request, a następnie zmergowałem branch feature/hello_world_asdt_from_tag z główną gałęzią main.

Feature/hello world asdt from tag #2


Merged InnyInaczej69 merged 3 commits into `main` from `feature/hello_world_asdt_from_tag` [now](#)


Conversation 0 Commits 3 Checks 0 Files changed 3


**InnyInaczej69** commented now


No description provided.

**InnyInaczej69** added 3 commits 15 minutes ago


 Add Hello world file

 Add Hello ASDT world file

 Update file to display Hello ASDT world with tagging

**InnyInaczej69** merged commit `fcd877d` into `main` now

[Revert](#)

**Pull request successfully merged and closed**

You're all set—the `feature/hello_world_...` branch can be safely deleted.

[Delete branch](#)

Review

No re

Assign

No on

Labels

b1e14d7

None

33b465f

Proje

c852ebb

None

Milest

No mi

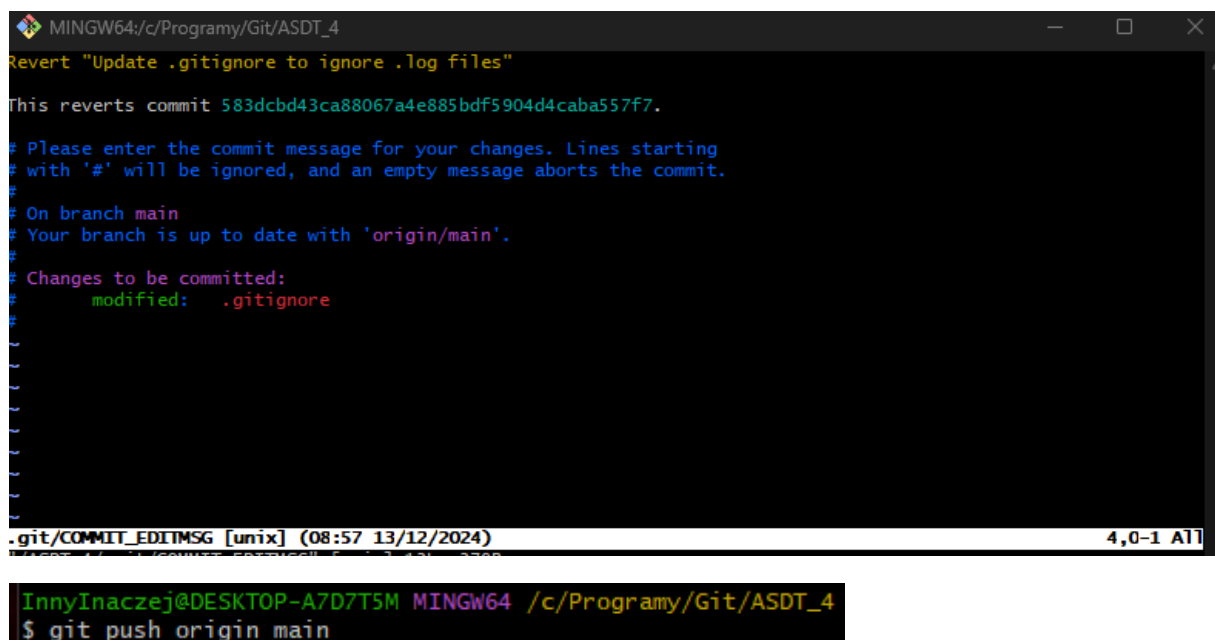
Devel

Succes

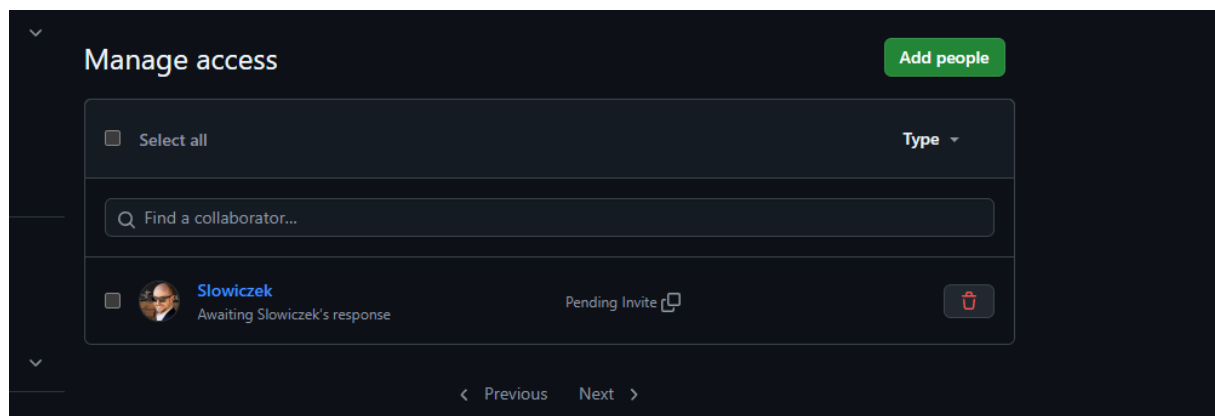
issues

None

18. Wykonałem revert ostatniego commitu z brancha main.



19. Zaprosiłem użytkownika Slowiczek do repozytorium, korzystając z funkcji zarządzania zespołem na platformie GitHub



Podsumowanie

W trakcie pracy nad repozytorium ASDT_4, wykonałem szereg działań związanych z zarządzaniem gałęziami, plikami oraz tagami w systemie kontroli wersji Git. Zaczynając od utworzenia nowej gałęzi, przez dodanie i edytowanie plików (.gitignore, out.log), aż po ich skomitowanie i spushowanie na zdalne repozytorium. Następnie, za pomocą tagów, oznaczyłem ważne wersje projektu, a także stworzyłem nowe gałęzie oparte na tagach. Po dokonaniu zmian w kodzie, wykonałem kolejne commitowanie oraz łączenie gałęzi (merge) z główną gałęzią main. Cały proces zakończyłem utworzeniem raportu PDF, który dokumentuje wykonane kroki, oraz zaproszeniem użytkownika do repozytorium.