

**za<pro/>gramowani.com**

**</> webmaster**



**E-mail:** [zaprogramowani@filiposinski.com](mailto:zaprogramowani@filiposinski.com)

**Discord:** <https://tiny.pl/7k6jp>

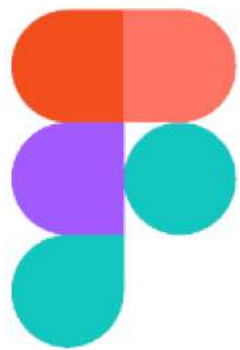
**GitHub:** [github.com/zaprogramowaniFO](https://github.com/zaprogramowaniFO)

# Zajęcia **nr 6**

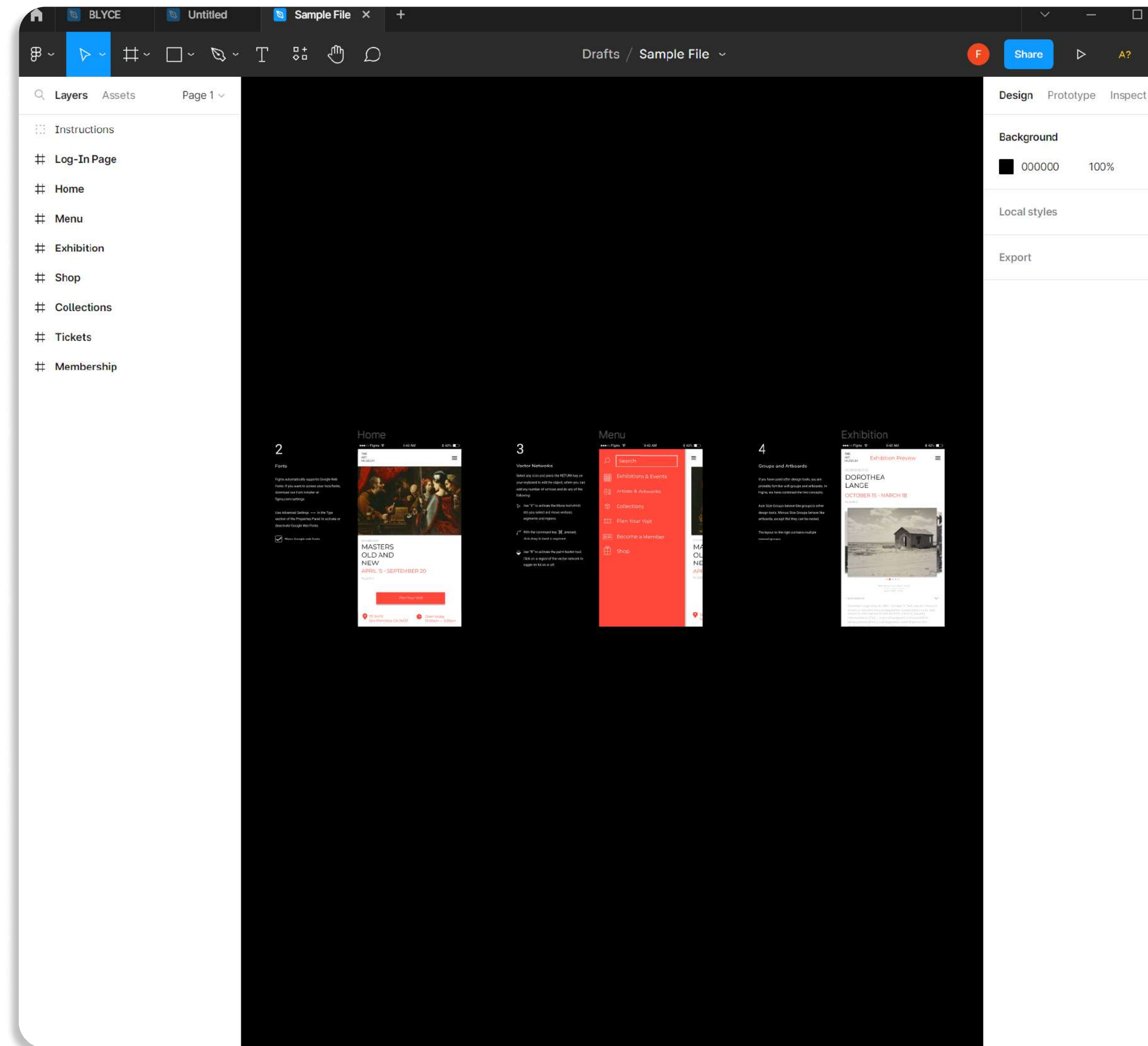
2023/04/20



# Figma

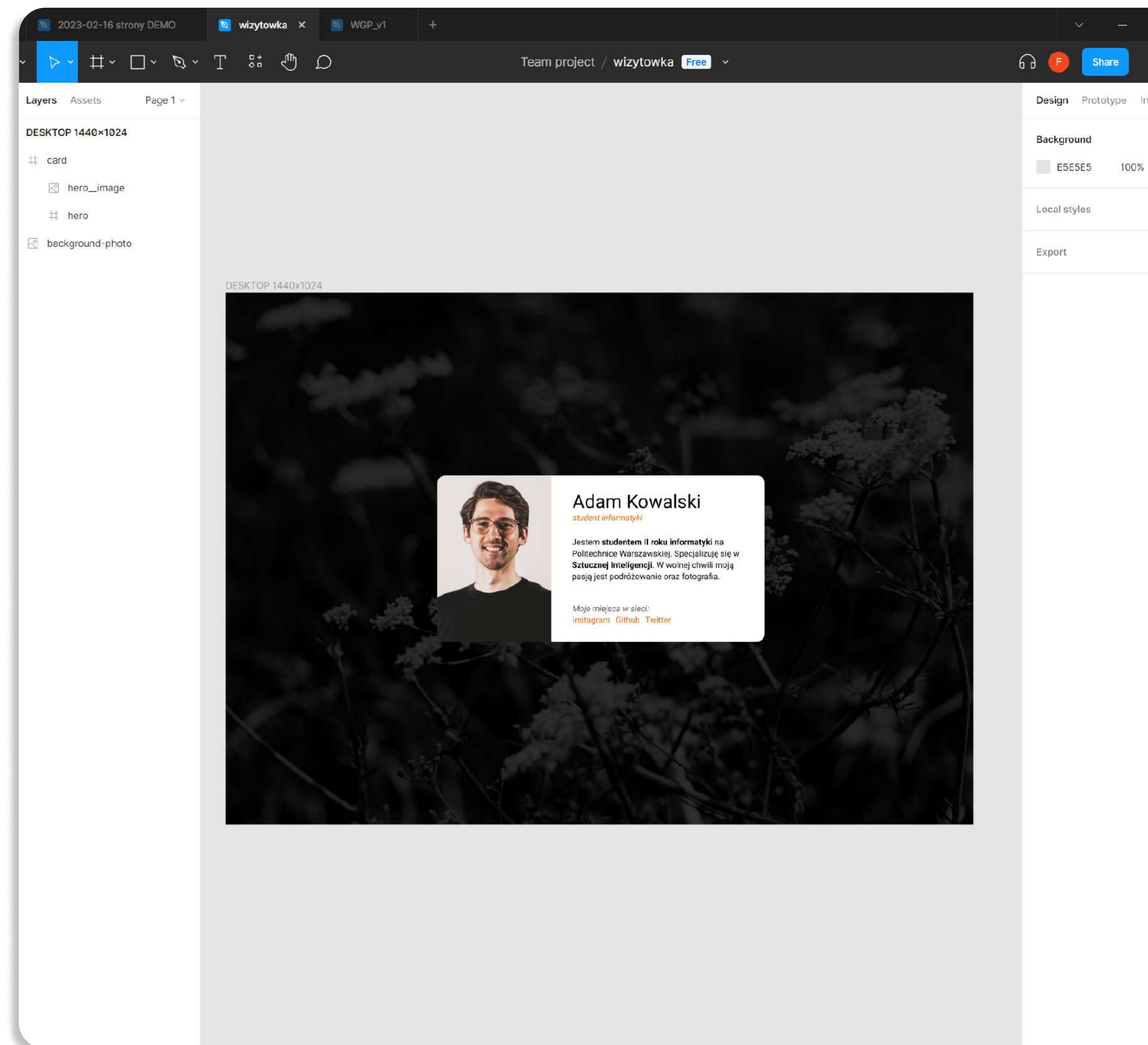


Edytor grafiki wektorowej służący jako narzędzie do projektowania i prototypowania stron internetowych i aplikacji mobilnych. Umożliwia tworzenie interaktywnych widoków oraz wspomaga 'cięcie' projektu pod kod.



# “Cięcie” projektu

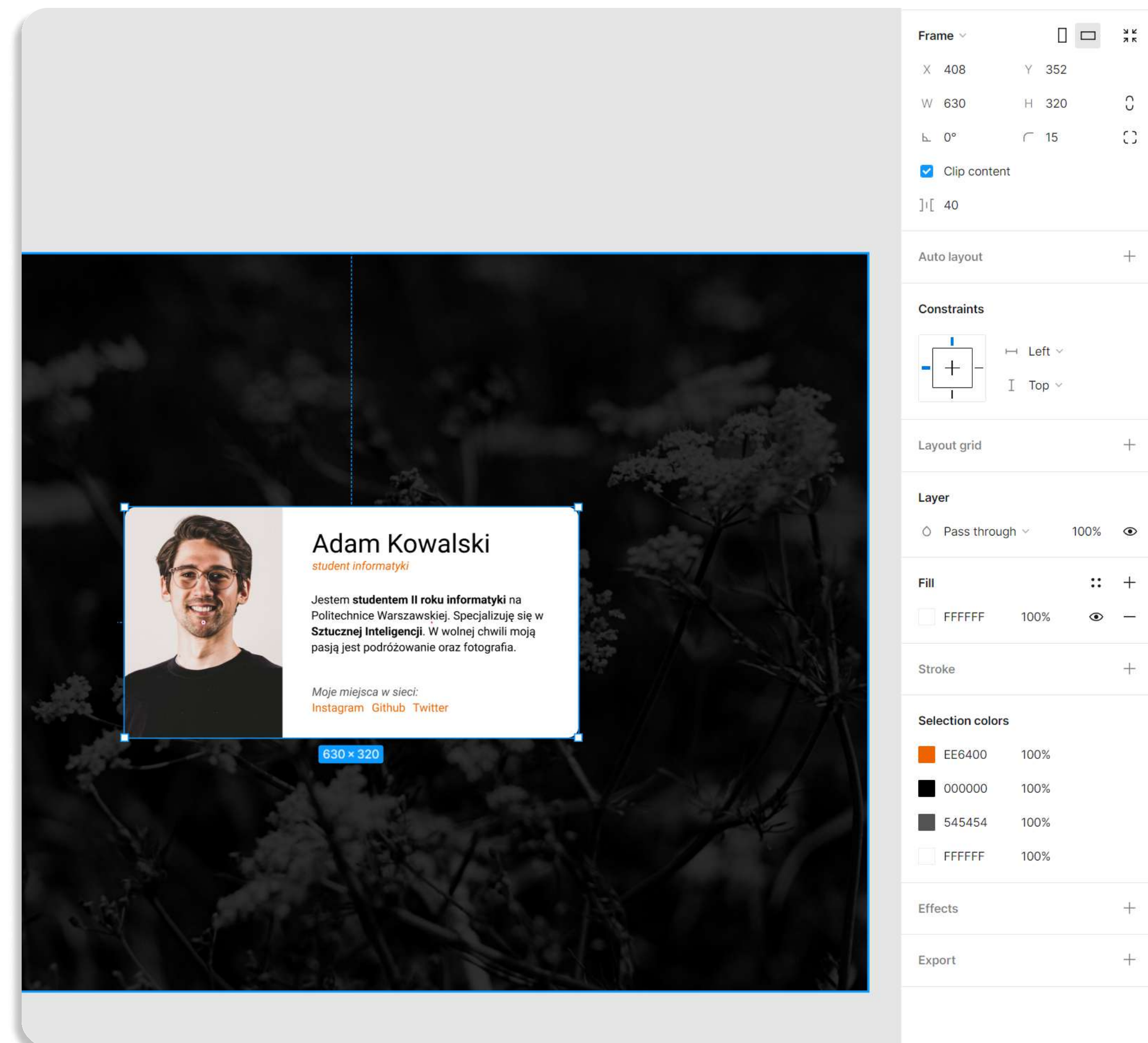
Gotowy projekt przygotowany przez grafika zostaje przekazany do **developer**a, który na podstawie projektu koduje stronę.





# Parametry elementów

**Każdy element** zaprojektowany w **Figmie** jest opisany poprzez różne parametry, takie jak wymiary, kolory, krój pisma i jego parametry, położenie względem innych elementów etc.

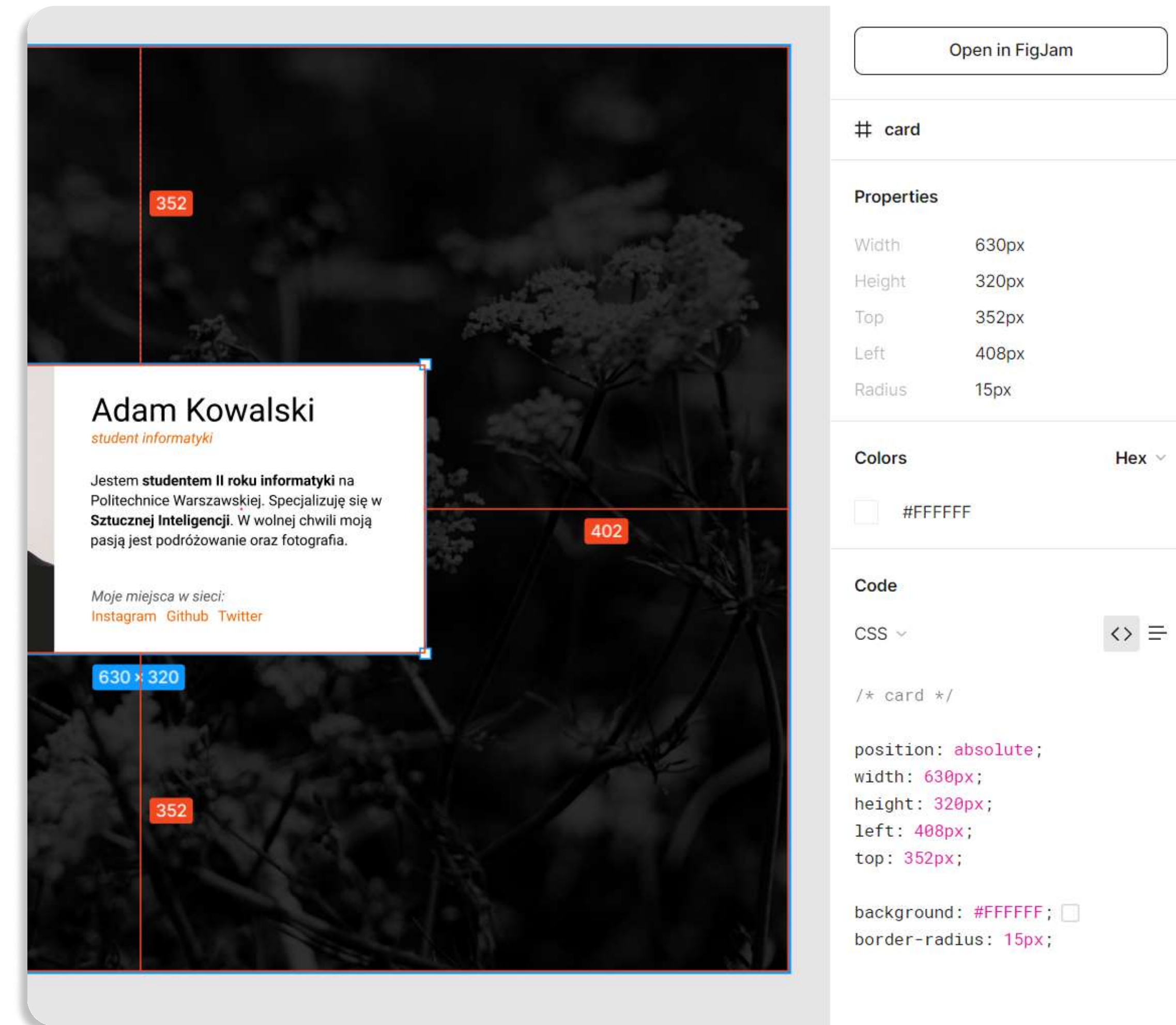


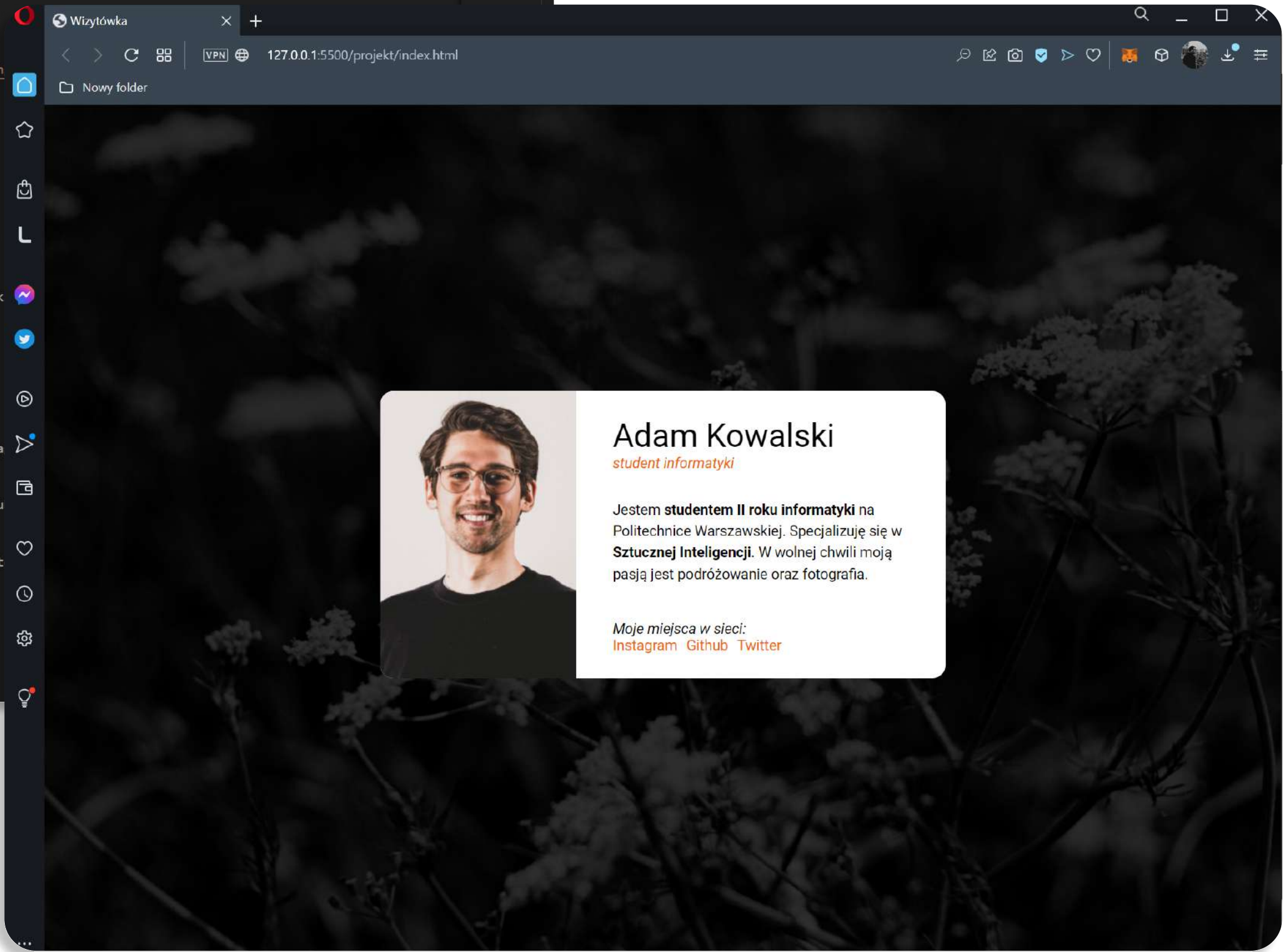
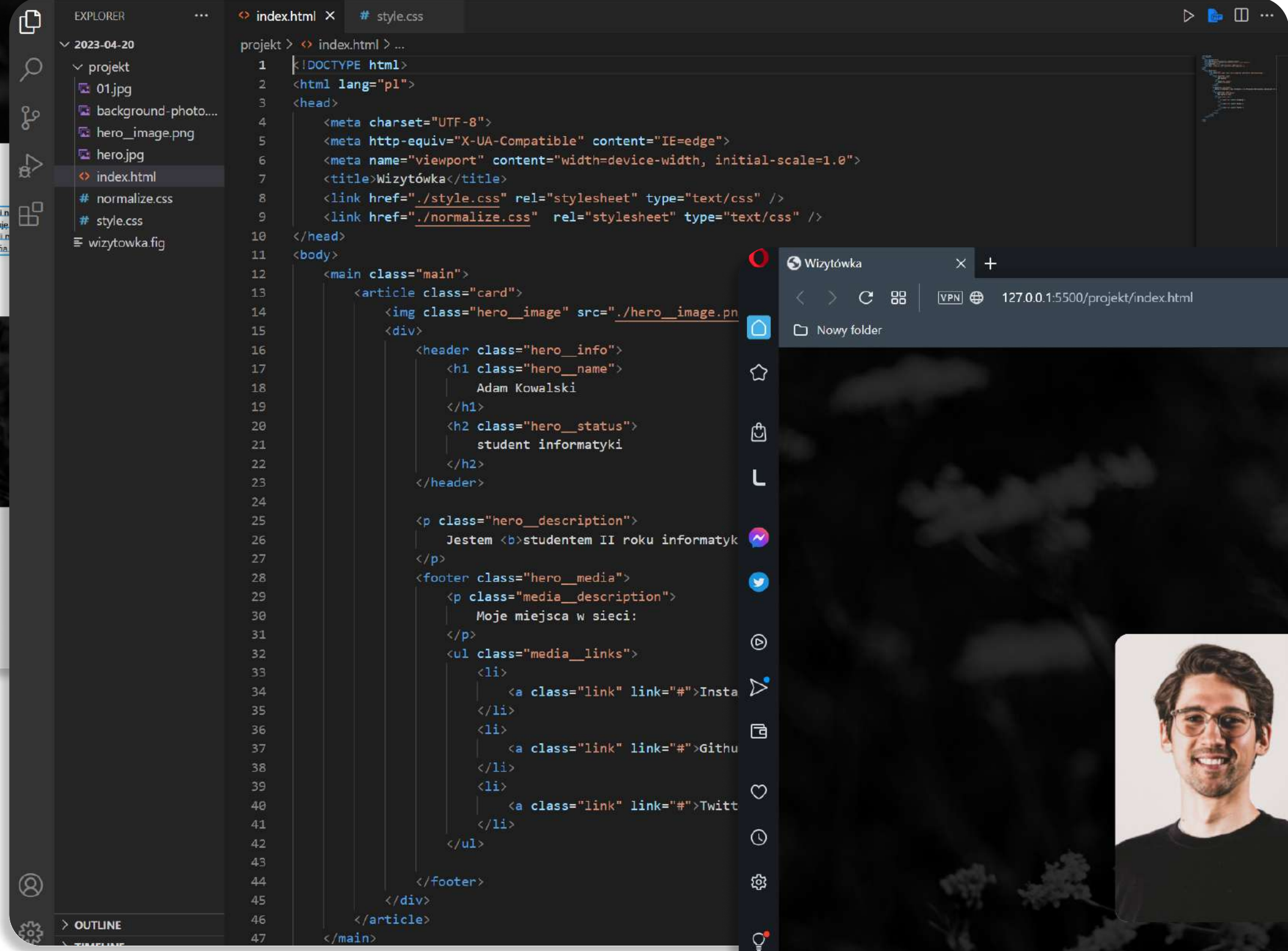
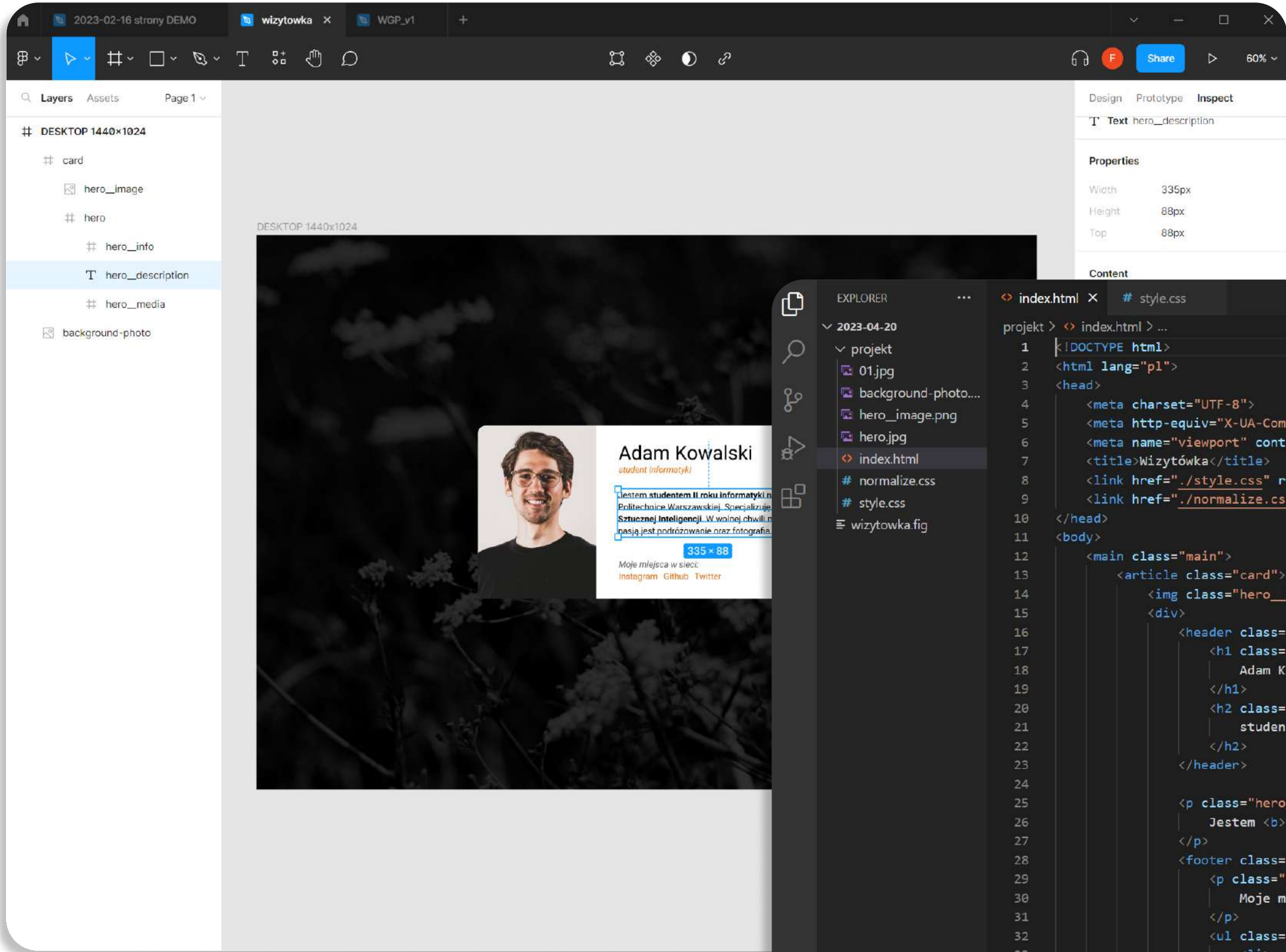


# Wsparcie pracy developera

**Figma** wspiera pracę developera, przedstawiając propozycję **stylu CSS**, dla wskazanego elementu, proponując właściwości oraz ich wartości.

Nie zawsze jednak są to najlepsze rozwiązania.









# Git, czyli rozproszony system kontroli wersji

<https://git-scm.com>

Stworzony przez **Linusa Torvaldsa**,  
twórcę **Linuxa**.

Pierwsza wersja Gita pojawiła się w  
kwietniu 2005 roku.

Git to typ oprogramowania, który  
służy przede wszystkim do  
śledzenia zmian w naszym kodzie.



# Git, czyli rozproszony system kontroli wersji

## Jeżeli:

- Wczoraj działało, a dzisiaj już nie,
- Chcemy wrócić do poprzedniej wersji kodu,
- Chcemy wprowadzić parę zmian, ale jednocześnie móc mieć obecną wersję programu,
- Chcemy pracować nad projektem w grupie, to...

...pomoże nam w tym Git ;)

# GitHub – czyli Twoje programistyczne portfolio

## Jeżeli:

- Wczoraj działało, a dzisiaj już nie,
- Chcemy wrócić do poprzedniej wersji kodu,
- Chcemy wprowadzić parę zmian, ale jednocześnie móc mieć obecną wersję programu,
- Chcemy pracować nad projektem w grupie, to...

...pomoże nam w tym Git ;)



# Git, czyli rozproszony system kontroli wersji

<https://github.com>

Serwis działa od kwietnia 2008 roku.

Jest to serwis, który pozwala na przechowywanie repozytoriów na swojej przestrzeni serwerowej.

Dzięki niemu wszystkie swoje projekty możesz opublikować oraz na bieżąco wprowadzać zmiany.

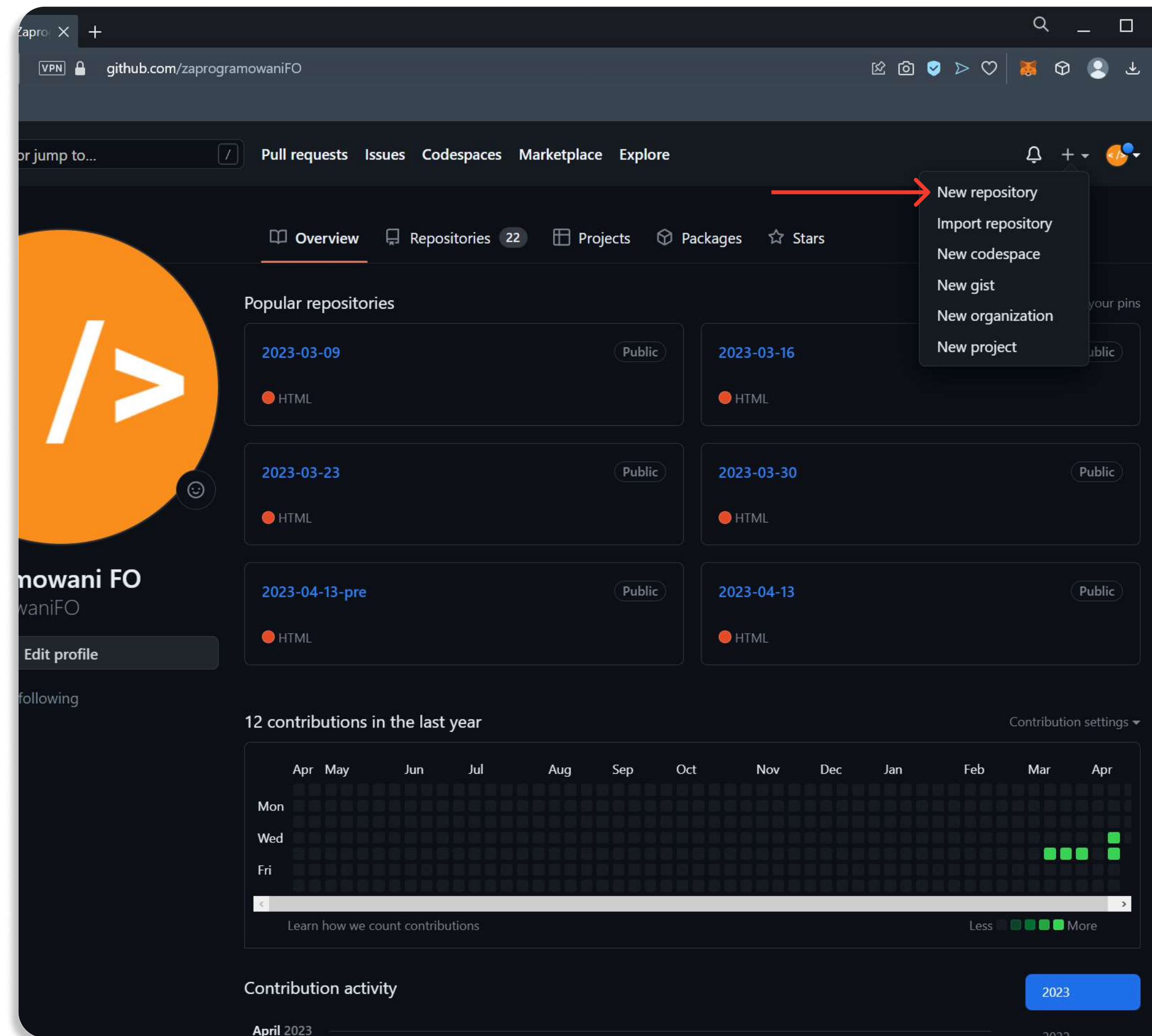


**</> GIT != Github**

**Tworzymy repozytorium na GITHUBIE**

# Tworzymy repozytorium

Będąc na naszym profilu na **Githubie**, klikamy w **plus** obok naszego logo w prawym górnym narożniku, a następnie z menu kontekstowego wybieramy **new repository**.



# Tworzymy repozytorium

Wpisujemy **nazwę naszego repozytorium (repository name)**, możemy również dodać od razu opcjonalny opis.

Jeżeli chcemy skorzystać z **hostingu** oferowanego przez **github**, nasze repozytorium musi być publiczne.

## Create a new repository

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

Owner \*



zaprogramowaniFO ▾

Repository name \*



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

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:

Skip this step if you're importing an existing repository.



Add a README file

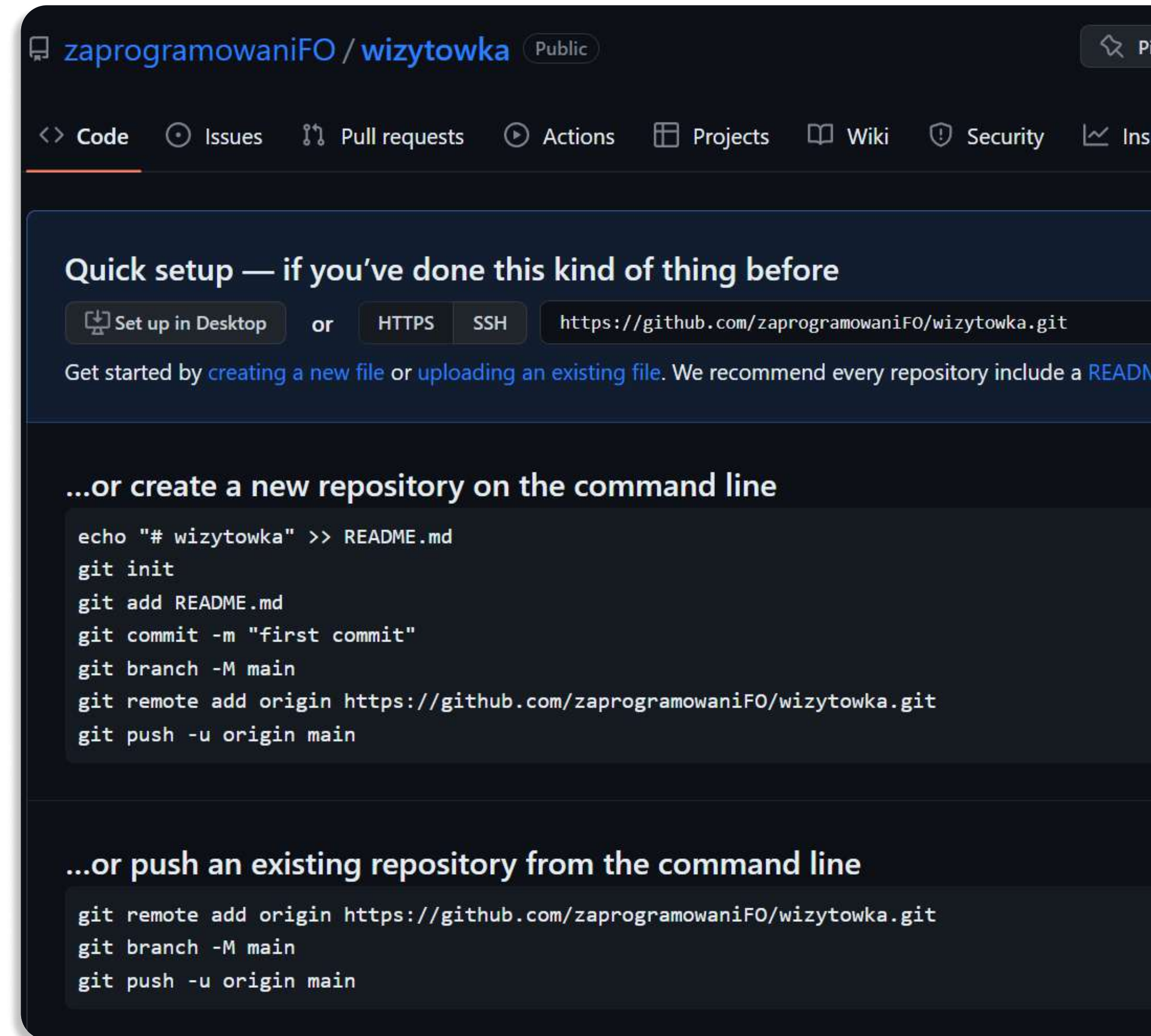
This is where you can write a long description for your project. [Learn more.](#)



# Tworzymy repozytorium

Następnie powinniśmy podpiąć repozytorium utworzone na naszym komputerze, do stworzonego repozytorium na platformie github, aby w prosty sposób synchronizować zmiany w projekcie.

Możemy również w prosty sposób umieszczać bezpośrednio pliki w repozytorium na stronie.  
Klikamy `upload an existing file`.





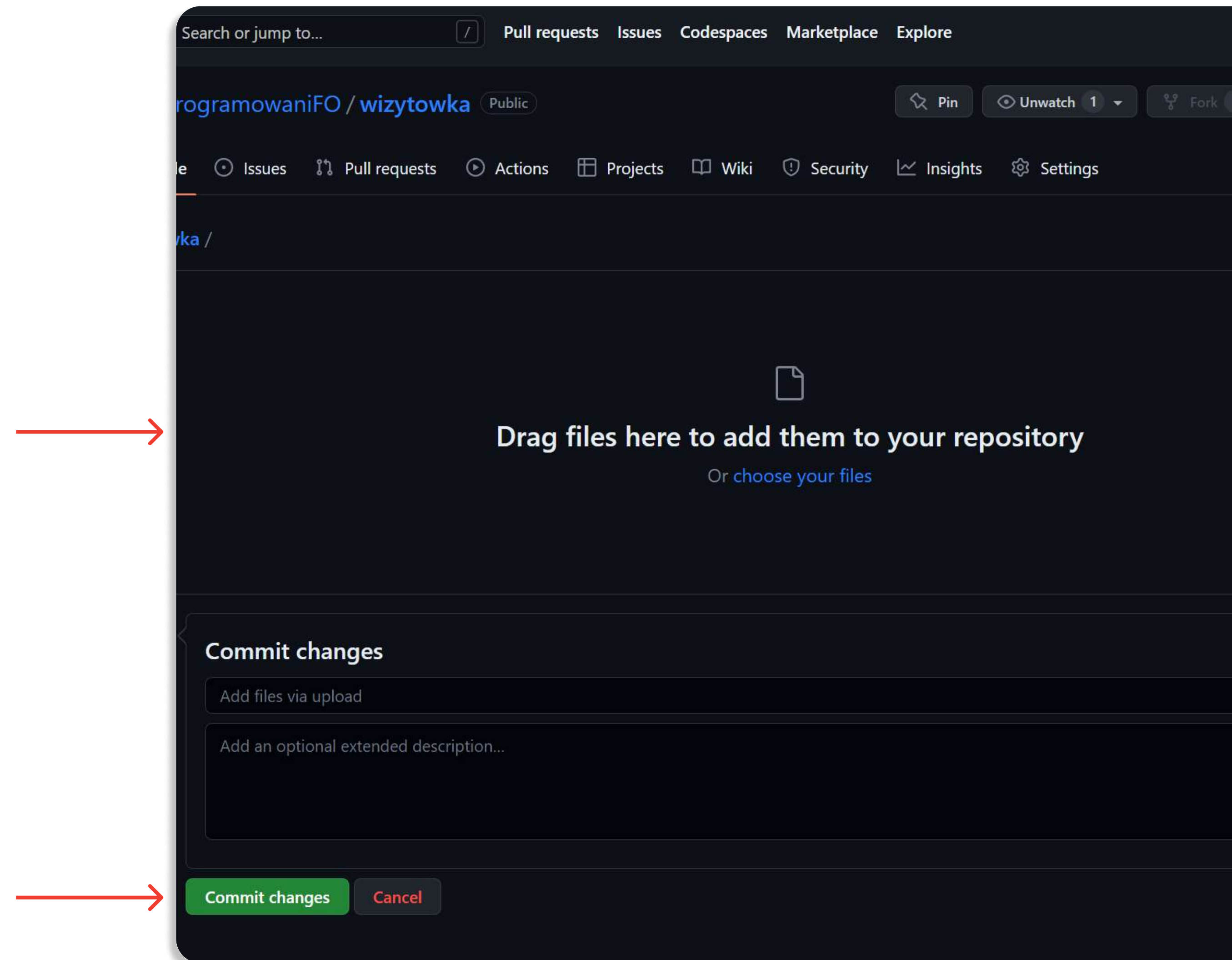
# Tworzymy repozytorium

W tym miejscu możemy bezpośrednio umieszczać pliki poprzez przeciąganie.

Możemy też napisać komentarz do naszego **commita**.

Po umieszczeniu plików należy kliknąć **commit changes**.

W ten sposób zrobiliśmy naszego pierwszego **commita** w projekcie! :)

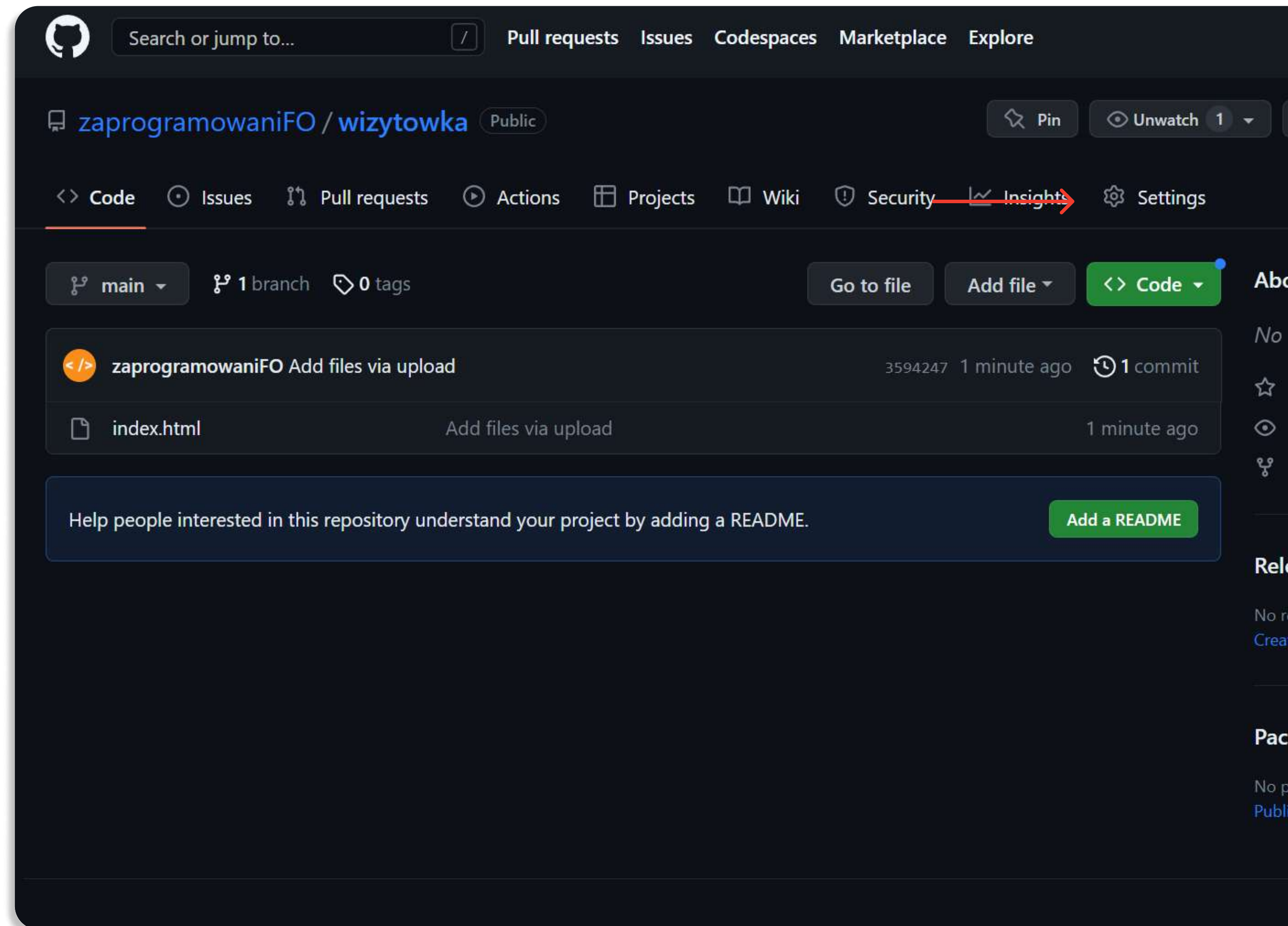


**Publikujemy stronę na GITHUBIE**

# Publikujemy stronę

Od publikacji strony dzieli nas jeszcze kilka kroków!

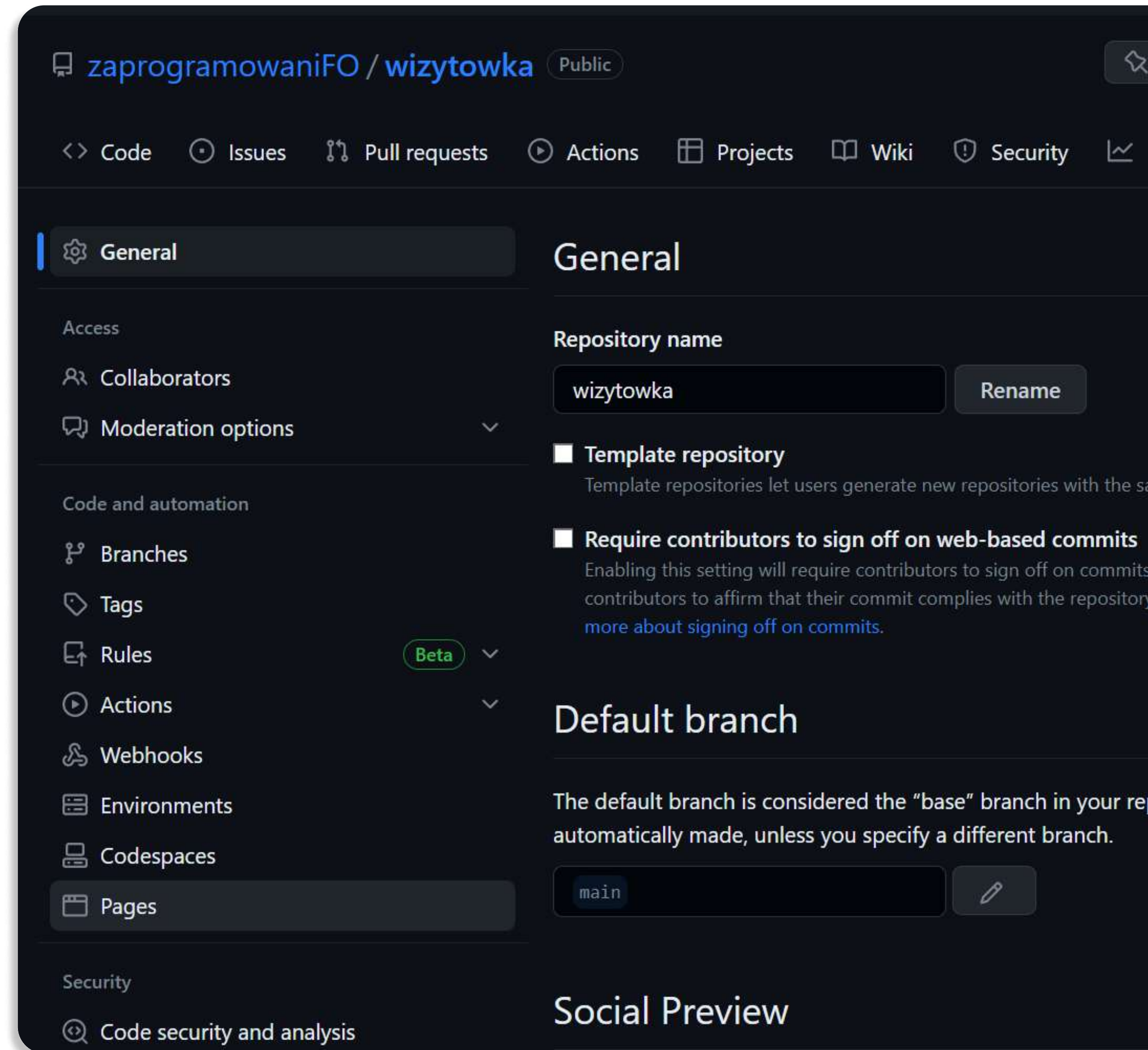
W gotowym repozytorium, które chcemy udostępnić jako stronę, klikamy w **ustawienia (settings)**.





# Publikujemy stronę

Następnie przechodzimy do sekcji **Pages**.

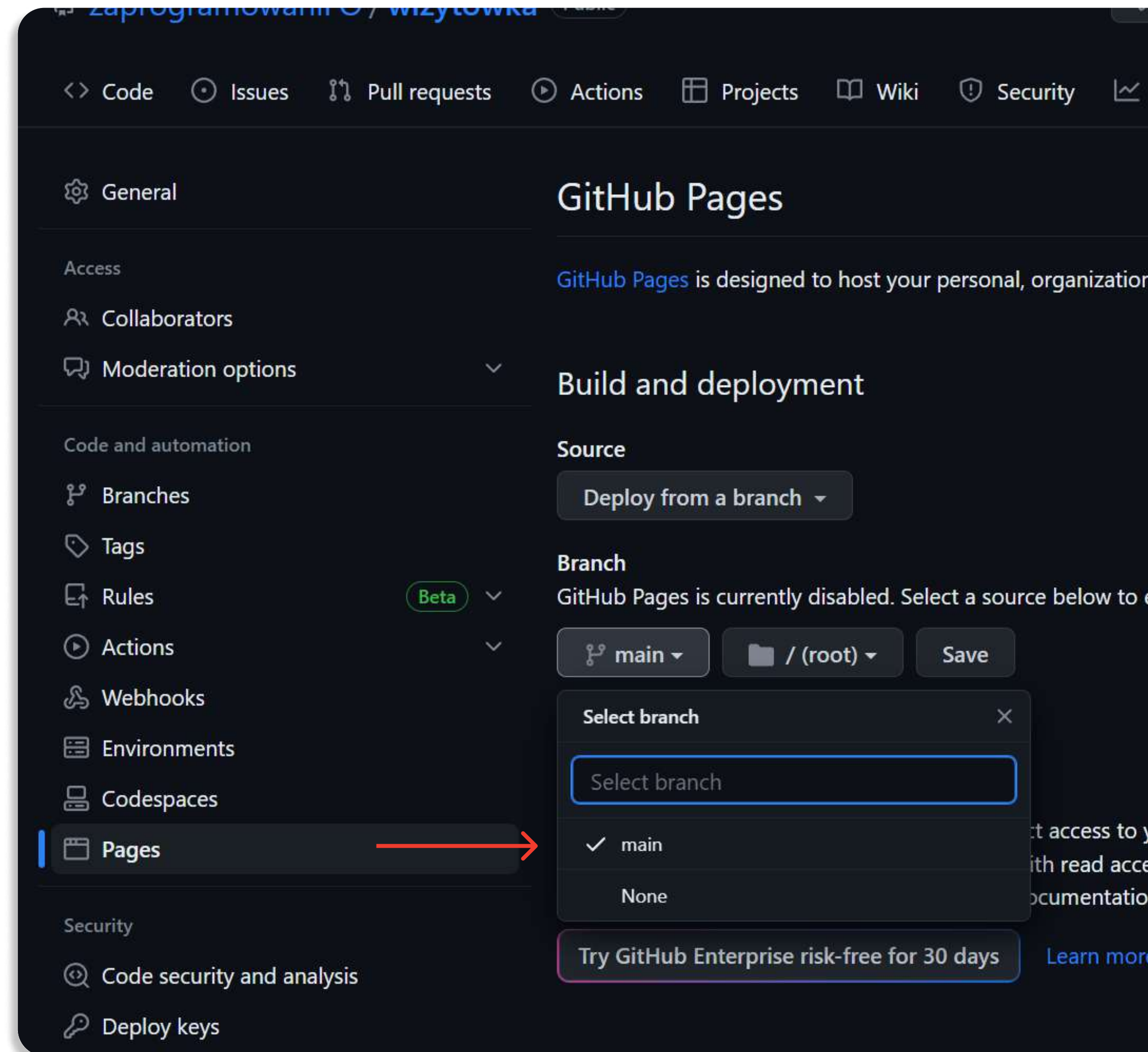


# Publikujemy stronę

W sekcji **Branch** wybieramy **main**, a następnie klikamy **save**.

Po tym procesie nasza treść jest przetwarzana i po jakimś czasie pojawi się dostępna pod adresem

**nazwa\_użytkownika.github.io/nazwa\_repozytorium /**





# GOTOWE!

**Strona opublikowana!**

Uwaga!  
Czasem przetworzenie danych, a  
zatem nim strona będzie widoczna w  
internecie, może zająć parę minut.

Cierpliwości! :)

