

# Instrukcja obsługi GitHub Desktop

Aleksandra Snopek

16 stycznia 2026

## Spis treści

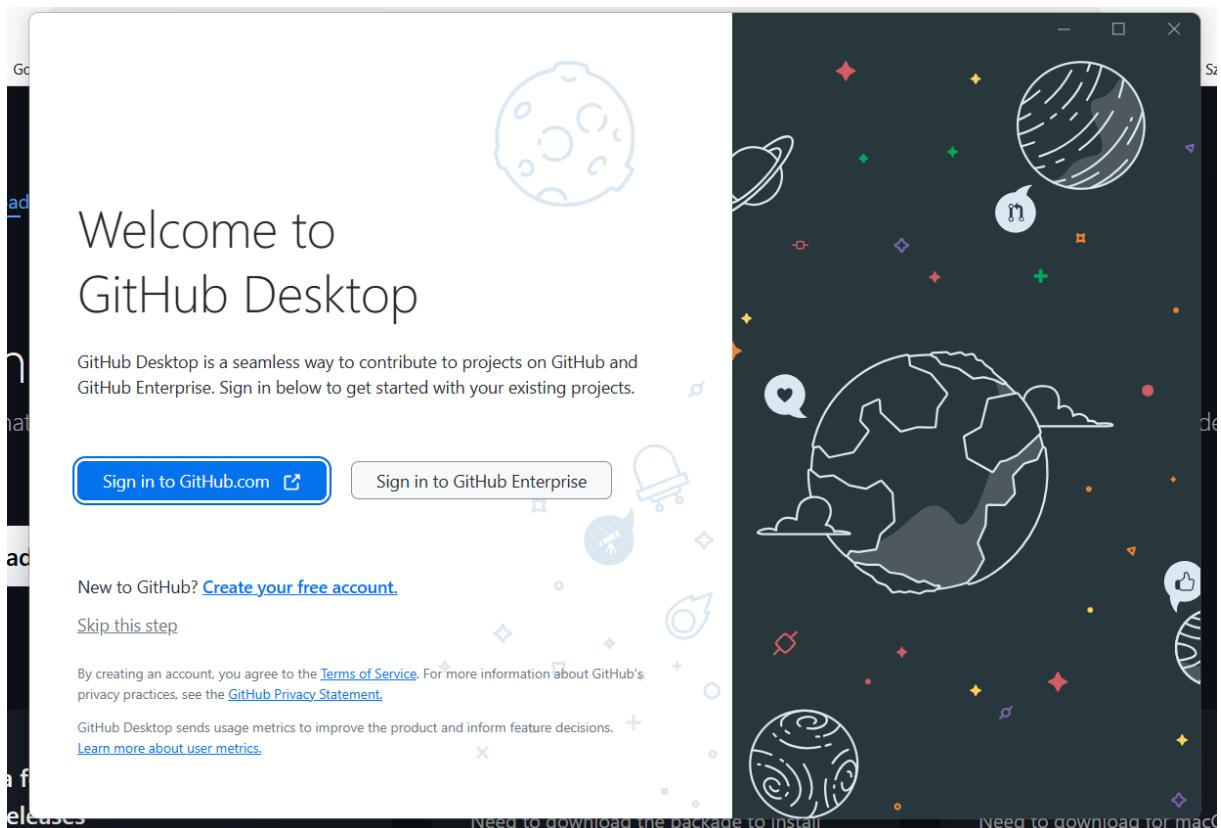
<b>1 Wprowadzenie</b>	<b>2</b>
<b>2 Logowanie do GitHub Desktop</b>	<b>2</b>
<b>3 Klonowanie repozytorium</b>	<b>2</b>
<b>4 Wprowadzanie zmian w projekcie</b>	<b>2</b>
<b>5 Tworzenie commitów</b>	<b>3</b>
<b>6 Wysyłanie zmian do repozytorium zdalnego</b>	<b>3</b>
<b>7 Tworzenie Pull Request</b>	<b>3</b>
<b>8 Podsumowanie</b>	<b>4</b>

# 1 Wprowadzenie

Niniejsza instrukcja opisuje szczegółowo obsługę aplikacji GitHub Desktop. Przedstawia proces logowania, klonowania repozytorium, wprowadzania zmian, tworzenia commitów, synchronizacji z repozytorium zdalnym oraz tworzenia Pull Request.

## 2 Logowanie do GitHub Desktop

Po uruchomieniu aplikacji GitHub Desktop użytkownik musi zalogować się na swoje konto GitHub.



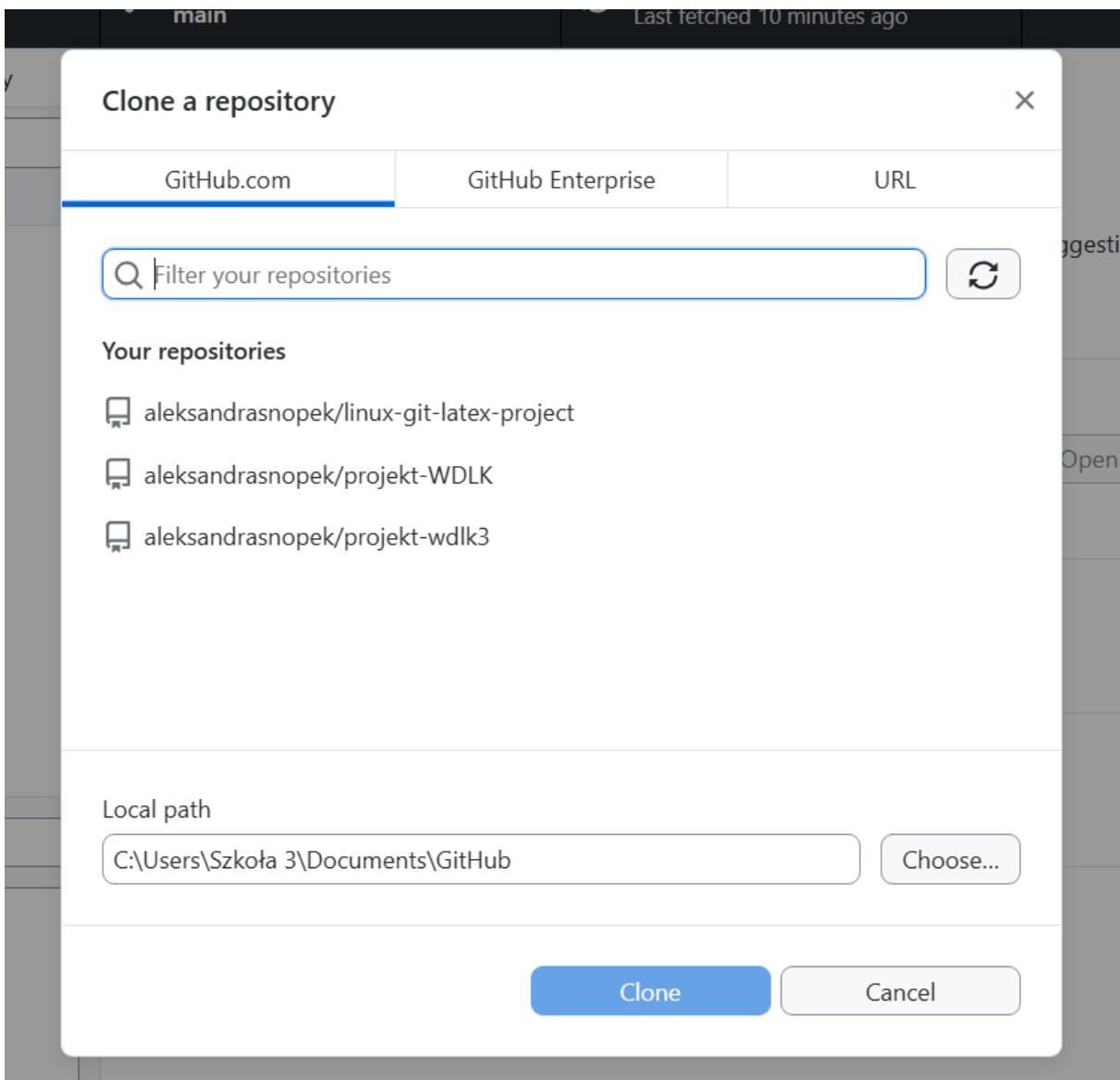
Rysunek 1: Logowanie do aplikacji GitHub Desktop

## 3 Klonowanie repozytorium

Po zalogowaniu należy wybrać opcję *Clone a repository*, a następnie wskazać repozytorium oraz lokalizację na dysku.

## 4 Wprowadzanie zmian w projekcie

Po otwarciu repozytorium użytkownik może edytować pliki projektu. GitHub Desktop automatycznie wykrywa wszystkie zmiany.



Rysunek 2: Wybór repozytorium do klonowania

## 5 Tworzenie commitów

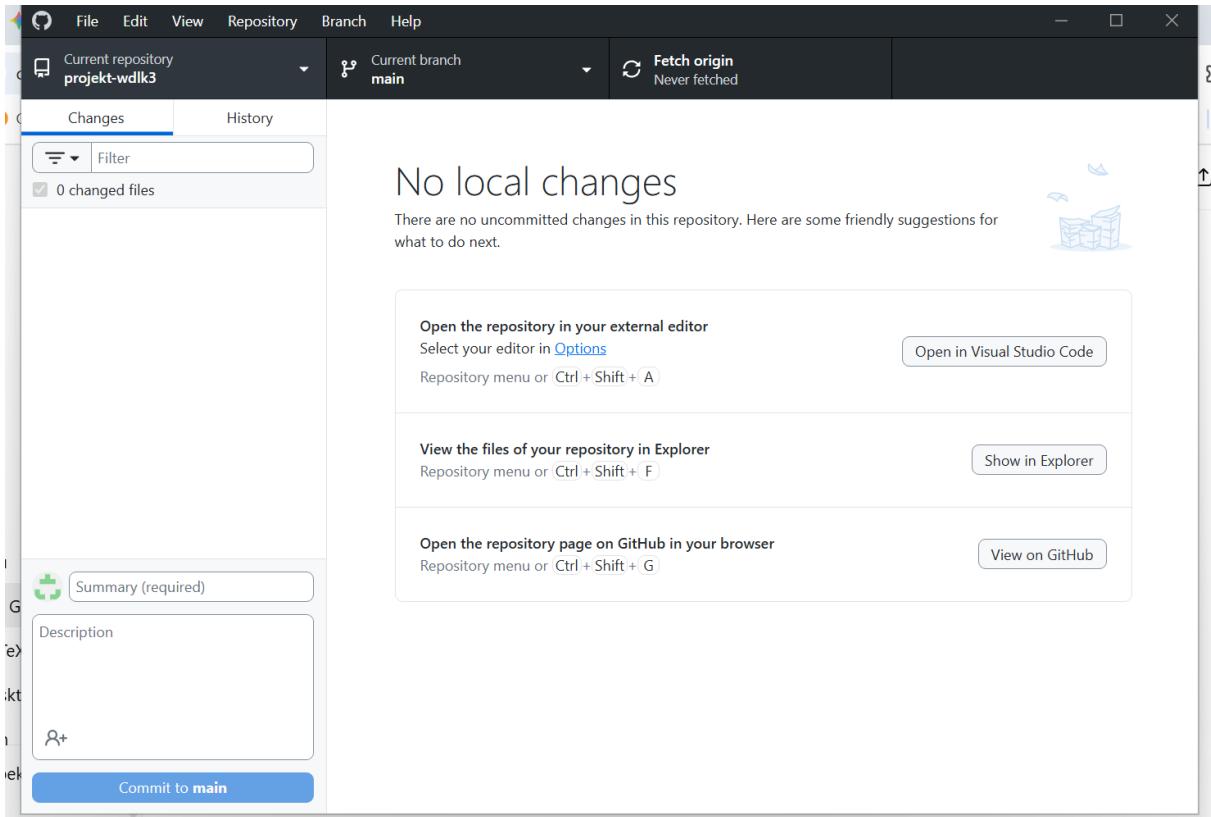
Commit zapisuje aktualny stan projektu wraz z opisem wprowadzonych zmian. Aby go utworzyć, należy wpisać krótki komunikat i zatwierdzić zmiany.

## 6 Wysyłanie zmian do repozytorium zdalnego

Po wykonaniu commitów zmiany znajdują się lokalnie. Opcja *Push origin* umożliwia ich wysłanie do repozytorium zdalnego.

## 7 Tworzenie Pull Request

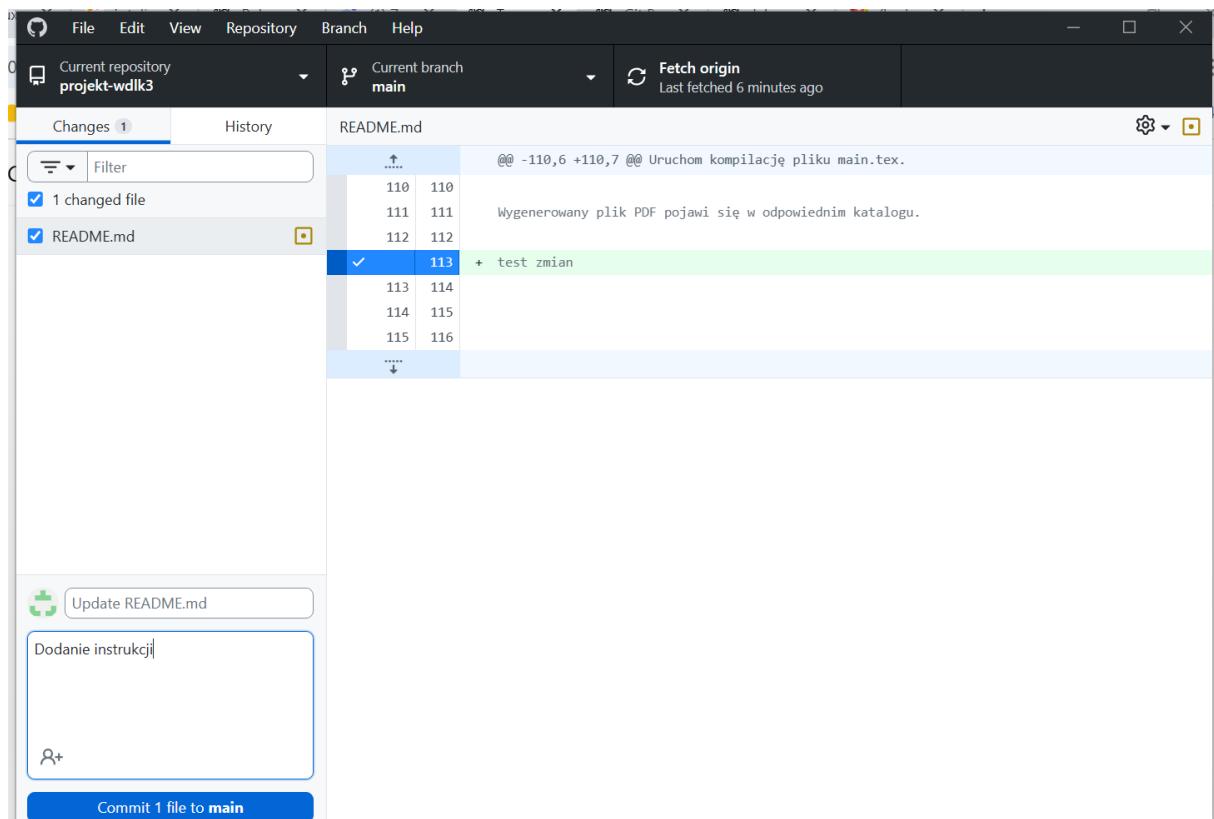
Pull Request służy do połączenia zmian z gałęzi roboczej z gałęzią główną repozytorium.



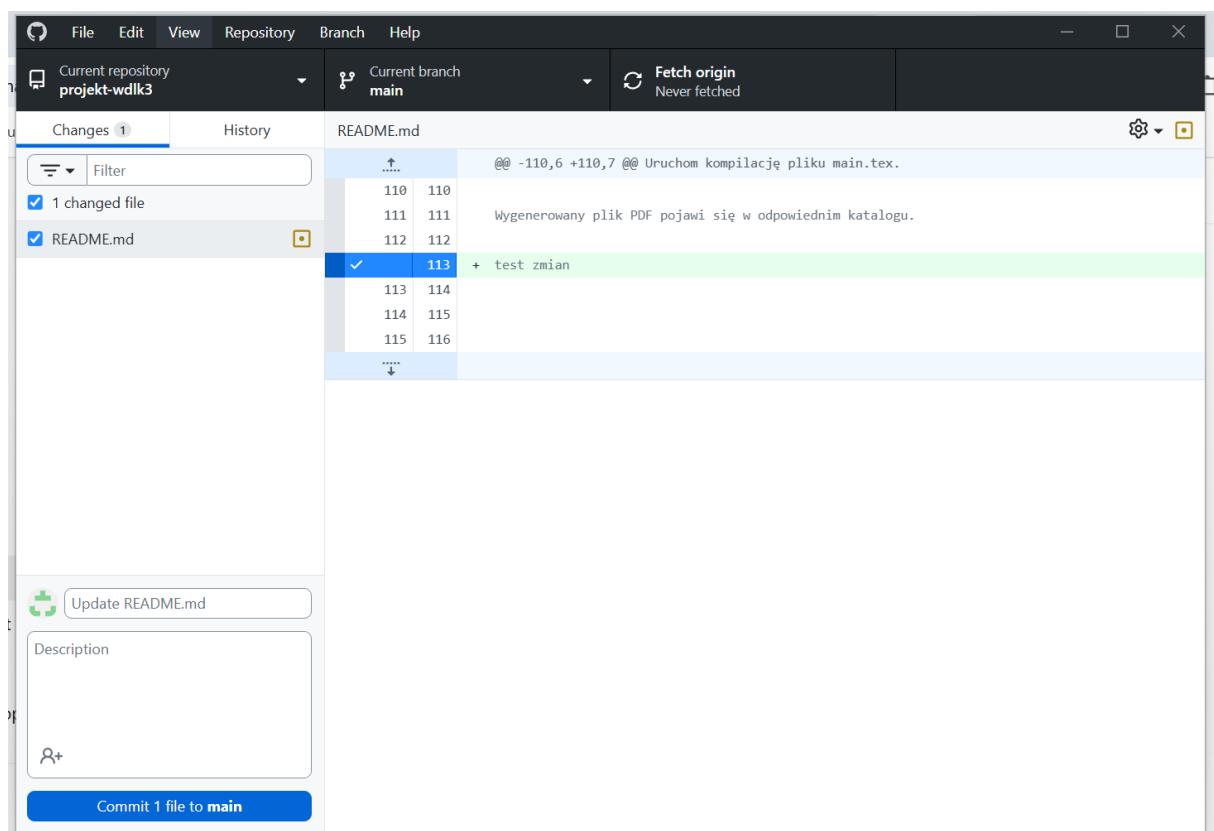
Rysunek 3: Widok repozytorium po klonowaniu

## 8 Podsumowanie

GitHub Desktop umożliwia intuicyjną pracę z systemem kontroli wersji Git. Dzięki przejrzystemu interfejsowi użytkownik może efektywnie zarządzać repozytoriami bez konieczności korzystania z linii poleceń.



Rysunek 4: Wykryte zmiany w plikach projektu



Rysunek 5: Tworzenie commita

Aktualizacja README.md

Aleksandra Snopek a1798a4

1 changed file

README.md

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	- # projekt-wdlk3	+ # \# Projekt WDLK 3 - Automatyzacja zadań w systemie Linu	x													
		+ #														
		+ # \## Autor														
		+ # Aleksandra Snopek														
		+ # ---														
		+ #														
		+ # \## Cel repozytorium														

Rysunek 6: Historia commitów w GitHub Desktop

Aktualizacja README.md

Aleksandra Snopek a1798a4

1 changed file

README.md

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	- # projekt-wdlk3	+ # \# Projekt WDLK 3 - Automatyzacja zadań w systemie Linu	x													
		+ #														
		+ # \## Autor														
		+ # Aleksandra Snopek														
		+ # ---														
		+ #														
		+ # \## Cel repozytorium														

Rysunek 7: Wysyłanie zmian do GitHuba

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

The screenshot shows a GitHub interface for creating a pull request. At the top, there are dropdown menus for 'base: main' and 'compare: develop'. A green checkmark indicates that the branches are 'Able to merge'. Below this, a message encourages users to 'Discuss and review the changes in this comparison with others' and provides a link to 'Learn about pull requests'. A large green button on the right says 'Create pull request'. In the center, it shows '2 commits' from '1 contributor', with '7 files changed'. Below this, a list of commits is shown:

- Dodanie skryptu klasyfikacji plików**  
aleksandrasnopek committed 8 hours ago
- Połączony skrypt LaTex z materiałów z Teamsa**  
aleksandrasnopek committed 2 hours ago

Rysunek 8: Tworzenie Pull Request