



# DOKUMENTACJA PROJEKTU WEBLIFE

projekt programistyczny autorstwa CentreBit

## Streszczenie

W niniejszym dokumencie opisywany jest proces przygotowania projektu do działania oraz przykładowe użycie

Bruno Szewczyk  
Bruno.cobbler@gmail.com

# Spis treści

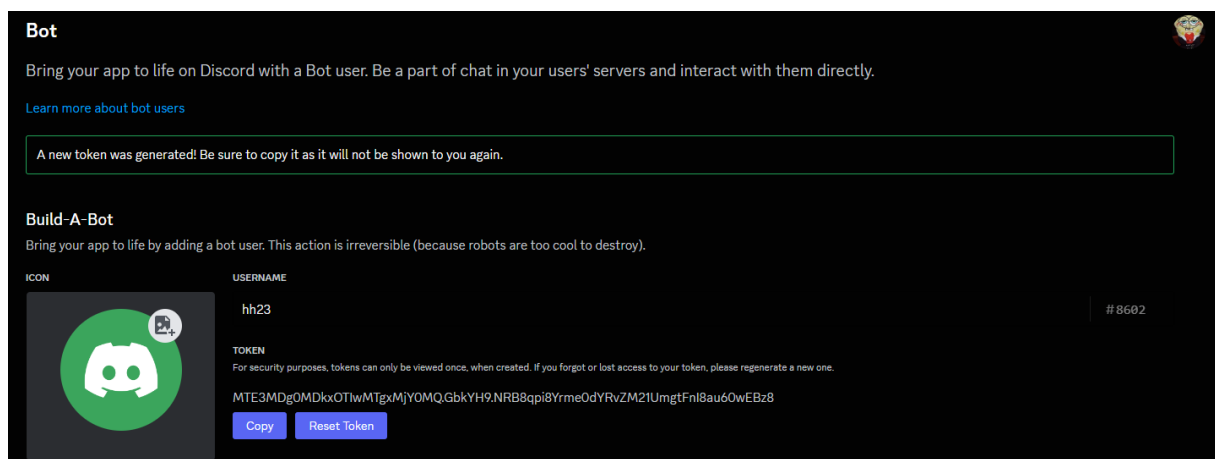
Spis treści.....	1
Przygotowanie projektu do działania .....	2
Boty Discord .....	2
Pliki .....	3
XAMPP .....	4
Baza danych.....	6
Skrypt python .....	6
Przygotowanie hostingu Github (nieobowiązkowe) .....	7
Testowanie projektu.....	9
Wejście na stronę .....	9
Moduł zadania .....	9
Powiadomienia o aktualizacji strony .....	10

# Przygotowanie projektu do działania

## Boty Discord

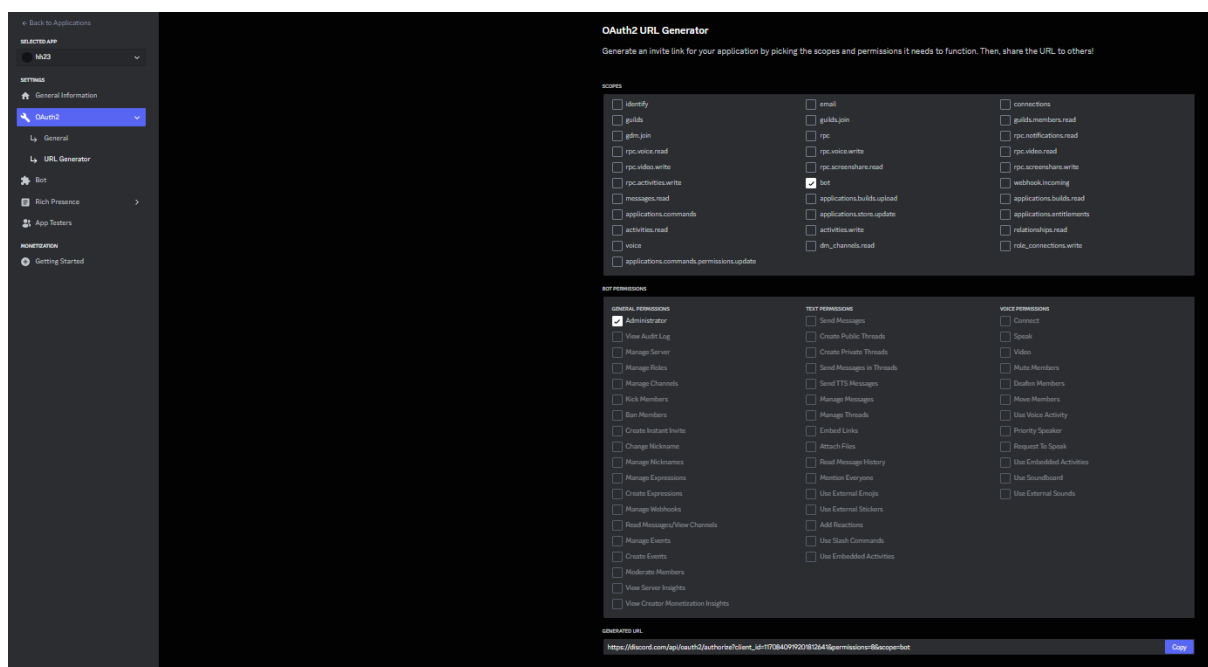
Do poprawnego działania projektu wymagane jest stworzenie dwóch botów discord. W tym celu muszą państwo odwiedzić [stronę discorda](#) i stworzyć na niej nową aplikację.

Żeby stworzyć nową aplikację, wymagane jest kliknięcie przycisku w prawym górnym rogu „New Application”



Zrzut ekranu nr 0: zakładka „Bot” i token

Następnie należy stworzyć URL w zakładce OAuth2 – URL generator.



Zrzut ekranu nr 1: zakładka OAuth2

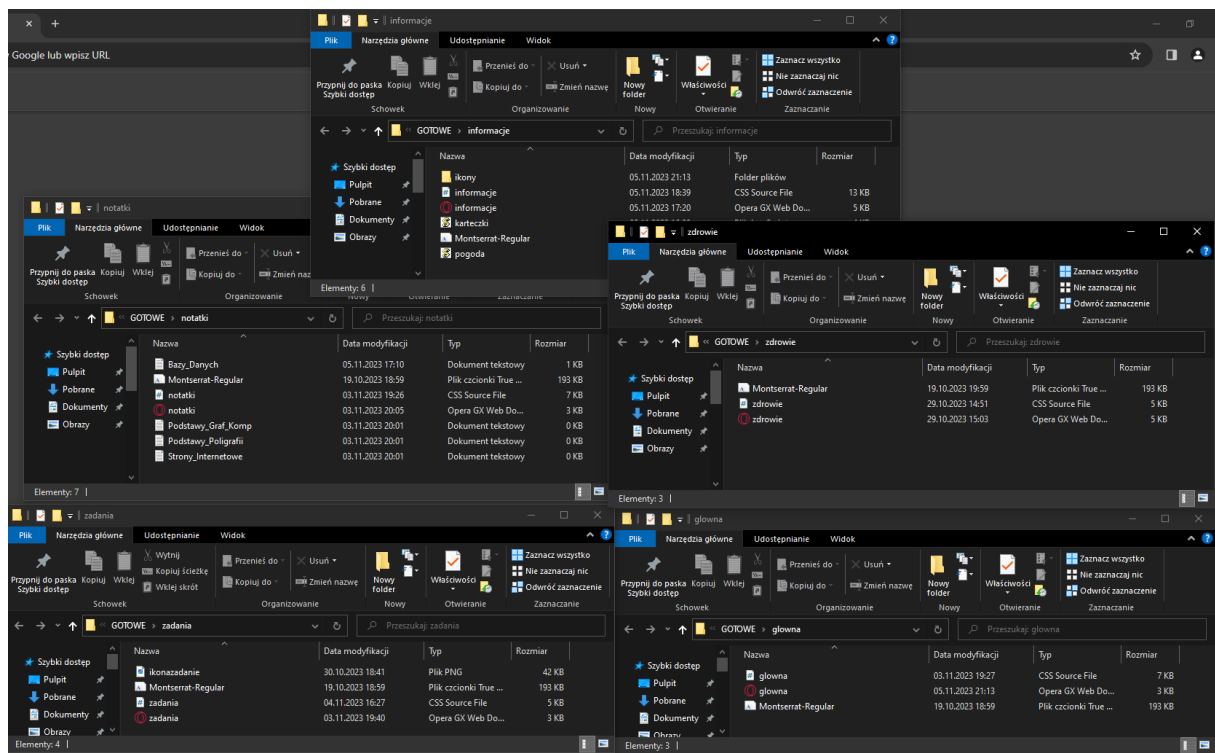
Później wymagane jest użycie linku i dodanie bota do serwera discord

## Pliki

Do uruchomienia projektu potrzebne są następujące pliki znajdujące się w następujących folderach

Nazwa	Data modyfikacji	Typ	Rozmiar
główna	03.11.2023 19:29	Folder plików	
informacje	05.11.2023 21:13	Folder plików	
notatki	03.11.2023 20:05	Folder plików	
zadania	30.10.2023 18:41	Folder plików	
zdrowie	05.11.2023 18:21	Folder plików	

Zrzut ekranu nr 2: główny folder projektu



Zrzut ekranu nr 3: zawartość wszystkich folderów w głównym folderze projektu

Nazwa	Data modyfikacji	Typ	Rozmiar
clear	04.11.2023 17:15	Microsoft Edge H...	7 KB
cloud	04.11.2023 17:15	Microsoft Edge H...	8 KB
haze	04.11.2023 17:15	Microsoft Edge H...	7 KB
rain	04.11.2023 17:15	Microsoft Edge H...	8 KB
snow	04.11.2023 17:15	Microsoft Edge H...	10 KB
storm	04.11.2023 17:15	Microsoft Edge H...	6 KB

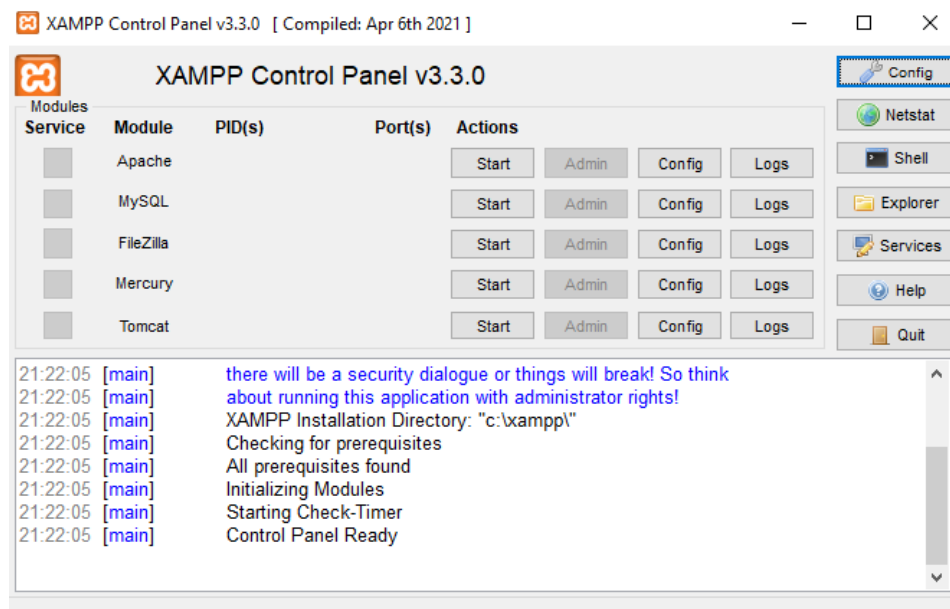
Zrzut ekranu nr 4: zawartość folderu „ikony” w folderze „informacje”

Potrzebne nam będą jeszcze pliki „Przypomnienia\_bot.py” oraz „submit.php”.

Wszystkie wymagane pliki znajdziecie Państwo na moim [repozytorium Github](#).

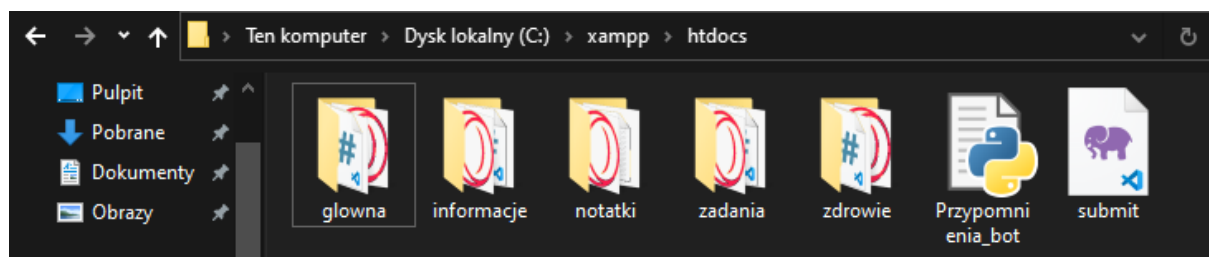
XAMPP

Do uruchomienia projektu potrzebny jest program [XAMPP](#). Po poprawnym zainstalowaniu programu XAMPP należy uruchomić XAMPP Control Panel.



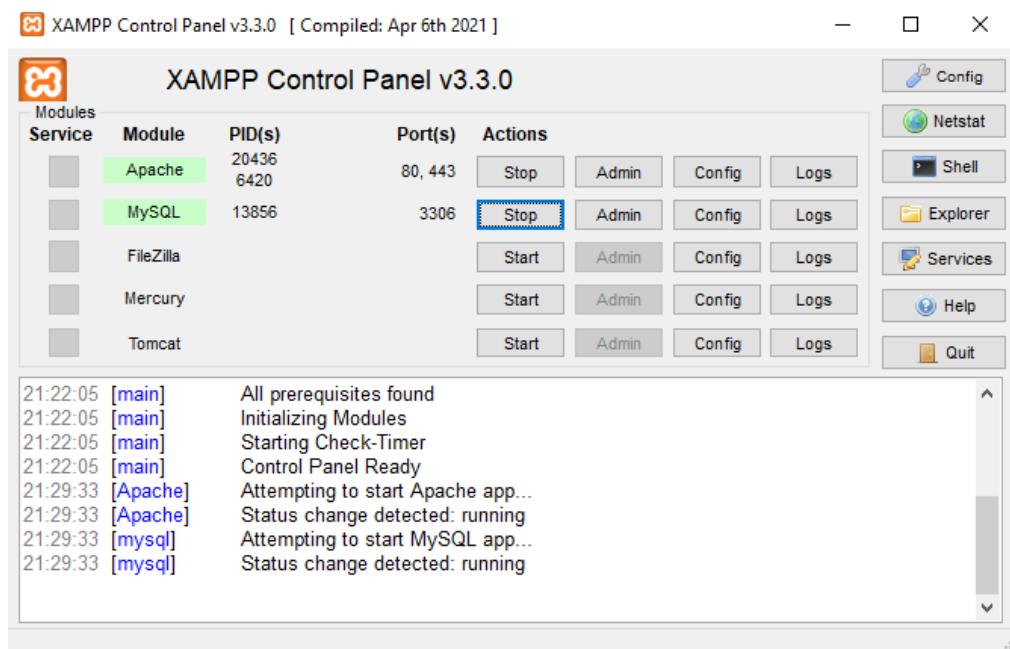
Zrzut ekranu nr 5: XAMPP Control Panel

Kolejnym krokiem będzie umieszczenie wszystkich plików do folderu „htdocs” w następujący sposób.



Zrzut ekranu nr 6: wszystkie potrzebne pliki umieszczone w folderze „htdocs” w folderze programu XAMPP

Następnym krokiem będzie uruchomienie funkcji Apache i MySQL w panelu XAMPP.



Zrzut ekranu nr 7: panel programu XAMPP po naciśnięciu przycisków „Start”

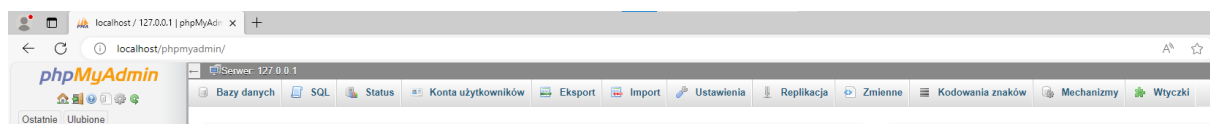
## Baza danych

Żeby projekt działał poprawnie, wymagane jest stworzenie nowej bazy danych. W tym celu w pasku przeglądarki należy wpisać *localhost/phpmyadmin*. Następnie trzeba skopiować poniższy kod do okna SQL

*CREATE DATABASE przypomnienia;*

*USE DATABASE przypomnienia;*

*CREATE TABLE wpisy ( id INT AUTO\_INCREMENT PRIMARY KEY, godzina VARCHAR(5), tekst TEXT);*



Zrzut ekranu nr 8: przykładowy panel phpMyAdmin

## Skrypt python

Do poprawnego działania projektu potrzebne są dwa boty napisane w języku python.

Pierwszy bot to „Przypomnienia\_bot”. Jego skrypt jest prawie gotowy do użycia od raz po pobraniu. Wystarczy zmienić następujące linijki według swoich celów.

```
if current_time == godzina:
    channel = bot.get_channel(1170709894924611596) #ID kanału discord na którym ma się pojawić wiadomość
    await channel.send(f"<@&1170711411383619698> {tekst}") #Treść wiadomości

    cursor.execute("DELETE FROM wpisy WHERE id = %s", (id_wiersza,))
    conn.commit()

    cursor.close()
    conn.close()

    await asyncio.sleep(60)

bot.run("MTE2Njc5MDE2ODEwODM0MzMDNg.GvCe-K.KcvwmYKAUy2zvvyVx6EtEr7lUrbE2wrg4PrH2E") #Token bota discord
```

Zrzut ekranu nr 9: wycinek skryptu „Przypomnienia\_bot” razem z komentarzami

Bot którego token jest zamieszczony w kodzie musi być na serwerze discord na którym znajduje się kanał którego ID jest zamieszczone w kodzie.

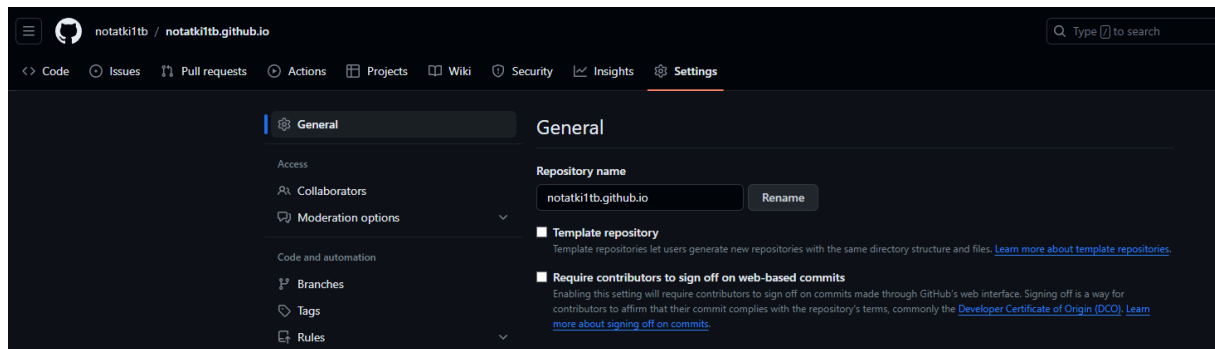
Drugi bot to „Github\_bot” który czytuje wiadomości z jednego kanału discord i wysyła dostosowaną wiadomość na drugi kanał. Uruchomienie tego bota nie jest niezbędne do działania całego projektu.

Na tym etapie, wymagane jest uruchomienie skryptu „Przypomnienia\_bot

## Przygotowanie hostingu Github (nieobowiązkowe)

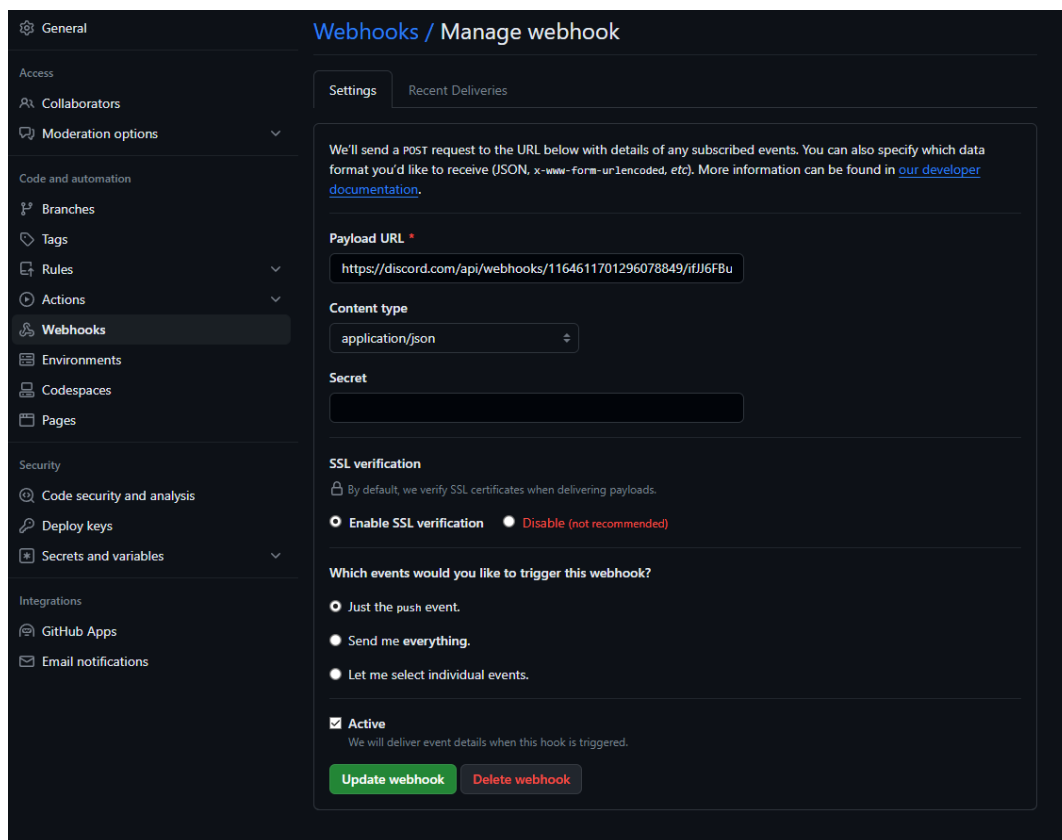
W celu hostingu strony na Github wymagane jest założenie darmowego konta na witrynie po czym stworzenie publicznego repozytorium o nazwie {nazwa konta}.github.io.





Zrzut ekranu nr n1: ustawienia repozytorium Github

Następnym krokiem będzie dodanie webhooka discord do repozytorium Github. Tym samym odsyłam państwa do [poradnika od Discord](#).



Zrzut ekranu nr n2: panel „Manage Webhook” na stronie github

Dalej wymagany jest wrzucenie wszystkich przygotowanych plików do repozytorium.

Później, muszą Państwo zmodyfikować kod bota „Github\_bot”.

```
@bot.event
async def on_message(message):
    if message.channel.id == 1170711543911038986: #ID monitorowanego kanału
        channel = bot.get_channel(1170709894924611596) #ID kanału na który ma zostać wysłana wiadomość
        if channel:
            await channel.send("<@&1170717399817928774> Stronka miała aktualizację, koniecznie sprawdź [tutaj](https://notatkiitb.github.io).") #Treść wiadomości
bot.run("MTE2NTMlOTIxNzIxNDYzMTcyMQ.GLQa8N.worHD5QDRHosxajk3PXF-qvrJwXJtguifpCg") #Token bota
```

Zrzut ekranu nr n3: wycinek skryptu „Github\_bot”

W celu poprawnego działania skryptu „Github\_bot” muszą Państwo podmienić ID monitorowanego kanału, ID kanału na który ma zostać wysłana wiadomość i token bota discord. Bot którego token jest umieszczony w kodzie musi być na serwerze discord na którym znajdują się odpowiednie kanały.”.

Na tym etapie, wymagane jest uruchomienie skryptu „Github\_bot

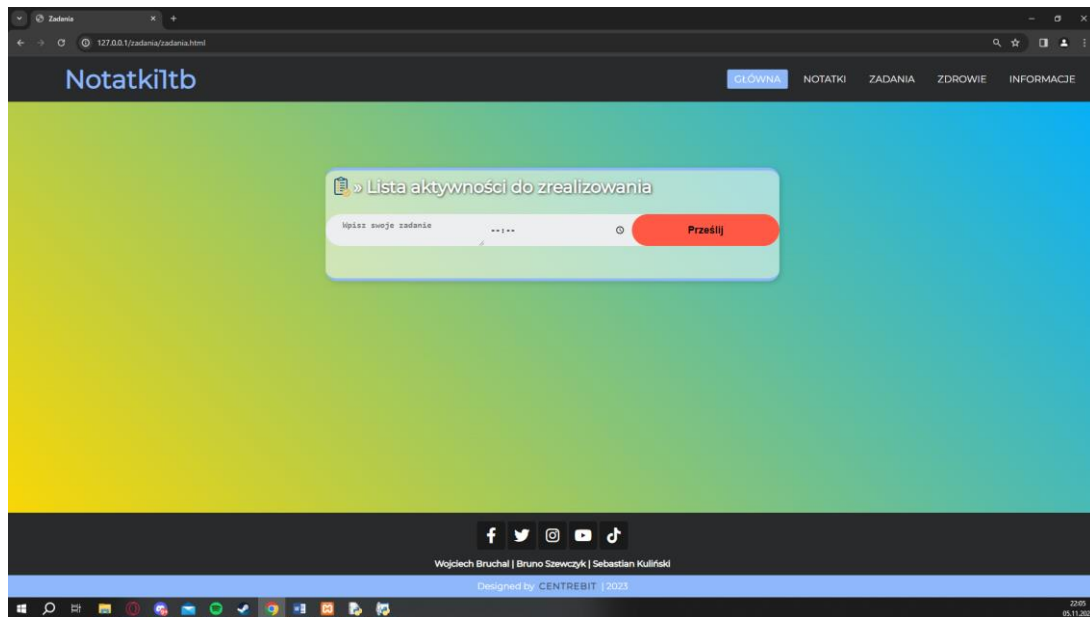
## Testowanie projektu

### Wejście na stronę

W przypadku XAMPP’a wystarczy wejść na adres  
*localhost/glowna/glowna.html*

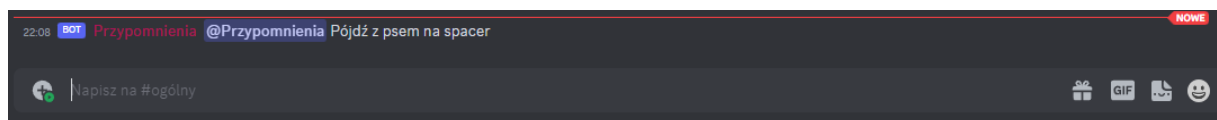
### Moduł zadania

Za pomocą modułu „Zadania” jesteśmy w stanie ustawić kiedy i jakie powiadomienie zostanie do nas wysłane na platformie discord.

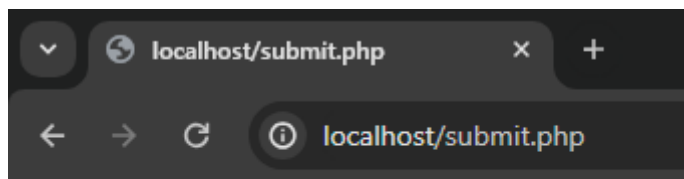


Zrzut ekranu 10: moduł zadania

Po wpisaniu swojego zadania i godziny, dostaniemy powiadomienie discord.



Zrzut ekranu 11: powiadomienie discord o podanej treści

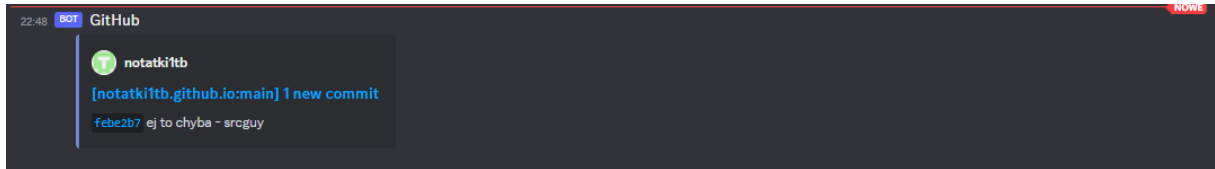


Nowy wpis dodany do bazy danych.

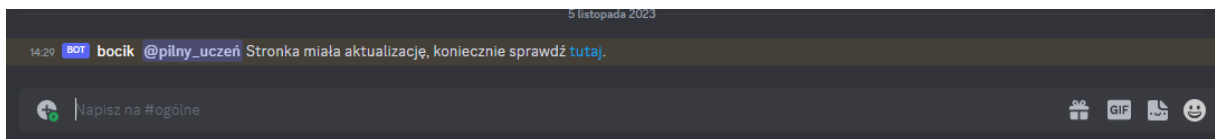
Zrzut ekranu 12: komunikat strony wyświetlany od razu po dodaniu przypomnienia

Powiadomienia o aktualizacji strony

W celu otrzymania powiadomienia o aktualizacji strony wymagane jest postawienie hostingu na platformie github. Powiadomienia prezentują się następująco:



Zrzut ekranu 13: powiadomienie wysłane przez Github o aktualizacji strony



Zrzut ekranu 14: powiadomienie discord o aktualizacji strony