

## 1. Cel stworzenia projektu

Głównym celem stworzenia projektu jest sterowanie komputerem za pomocą mowy. Program wykonuje odpowiednio przekazane mu głosowo polecenia. Słowo wywołujące po uruchomieniu programu to "Computer", po którym należy wypowiedzieć skonfigurowaną przez nas komendę. Program następnie wykona zadanie oraz do momentu wyłączenia będzie wyczekiwał następnej instancji słowa wywołującego.

## 2. Zakres pracy

11.03.2024 (pierwszy commit) - 19.06.2024

## 3. Wykorzystane technologie

- Python 3.12
- Django-ninja API - framework API dla Django, umożliwiający tworzenie API opartych na modelach Django
- Flet - framework do tworzenia aplikacji Flutter w Pythonie
- PyAudio - biblioteka umożliwiająca komunikację języka Python z API PortAudio19, wielo-platformowej biblioteki I/O audio

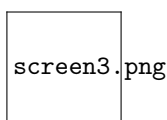
## 4. Struktura projektu

Cały program składa się z głównego folderu zawierającego plik `app_run.py`, służącego do uruchomienia całej aplikacji. Następnie podfolder `api` zawiera konfigurację Django API oraz model `documents` służący do zapisywania komend skonfigurowanych przez użytkownika w bazie danych. Podfolder `app_run_config` zawiera konfigurację oraz uruchamia potrzebne zależności, API oraz interfejs graficzny. Podfolder `gui` zawiera konfigurację oraz główny kod interfejsu graficznego. Wewnątrz podfolderu `gui_config` znajdują się `assets` - folder z ikonami oraz innymi użytymi assetami, `views` - folder zawierający skrypty z wyglądem odpowiednich podstron i sekcji aplikacji, `routing` - folder zawierający skrypt odpowiedzialny za zmianę wyglądu stron interfejsu `user_controls` - folder z kontrolkami dla użytkownika, w tym przypadku jest to jedynie kontrolny pasek zadań znajdujący się na samej górze aplikacji. Podfolder `my_bot` zawiera konfigurację oraz główny kod bota na platformie Discord komunikającego się z API.

`my-app/  
node_modules/public/favicon.icoindex.htmlrobots.txtsrc/index.cssindex.js.gitignorepackage.jsonREADME.md`

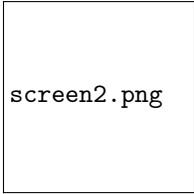
## 5. Dane wejściowe do działania projektu

Jako dane wejściowe projekt potrzebuje nazwy, bądź frazy którą będziemy używać do wywołania komendy, ścieżki lub komendy `shell/cmd` (w zależności od systemu użytkownika) oraz opcjonalnej dokumentacji na czym polega tworzona komenda.



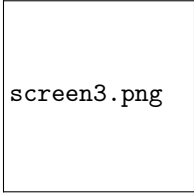
## 6. Przykłady działania

Na początku, zaraz po uruchomieniu programu nie będziemy w stanie od razu wywoływać głosem żadnej komendy, gdyż baza danych jest jeszcze pusta. Aby rozpocząć użytkowanie programu należy dodać komendę, do czego służy pierwszy główny przycisk "Dodaj komendę".



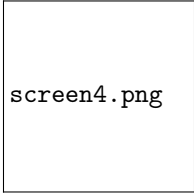
screen2.png

Następnie musimy wpisać odpowiednio opisane już wcześniej interesujące nas parametry oraz kliknąć przycisk “Dodaj”

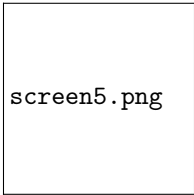


screen3.png

Jeśli jesteśmy tym zainteresowani, możemy ujrzeć listę wszystkich komend, które zostały dodane do bazy danych i możemy wykonywać. Jest to możliwe za pomocą drugiego głównego przycisku “Wyświetl komendy”.



screen4.png



screen5.png

Następnie wystarczy wypowiedzieć słowa wywołujące oraz komendę, którą wcześniej skonfigurowaliśmy.

## 7. Podsumowanie

### 7.1. Podział pracy w projekcie

- Rafał Bazan - implementacja Django-ninja API oraz działalności PyAudio
- Jakub Piasek - implementacja oraz design Graficznego interfejsu użytkownika Flet

Repozytorium na GitHub