

# Projekt zaliczeniowy

Proces ETL

Grupa projektowa:...

Imię	Nazwisko	Numer albumu	Grupa dziekańska	Wkład w prace nad projektem <sup>1</sup>	Udział procentowy

\_\_\_/70  
pkt

<sup>1</sup> proszę wymienić konkretne zadania

## Table of Contents

1. <i>DOKUMENTACJA TECHNICZNA</i> .....	3
1.1. INTERFEJS UŻYTKOWNIKA.....	3
2. DOKUMENTACJA INTERFEJSU UŻYTKOWNIKA.....	5
2.1. <i>INTERFEJS UŻYTKOWNIKA</i> .....	5
2.2 DZIAŁANIE APLIKACJI.....	6

## 1. DOKUMENTACJA TECHNICZNA

### 1.1. INTERFEJS UŻYTKOWNIKA

Interfejs użytkownika został napisany w języku python. Dodatkowo wykorzystana została technologia Tkinter. Tkinter jest standardem pozwalającym utworzyć interfejs graficzny oraz jest oparta na bibliotece Tk. Interfejs użytkownika składa się z jednego pliku python o nazwie *main.py*, który składa się z jednej klasy GUI oraz funkcji -

- *do\_nothing()* - funkcja wykorzystana w klasie GUI, nie wykonująca żadnej czynności
- *extract()* - funkcja wykorzystana w klasie GUI do wyodrębniania danych ze strony '<https://www.x-kom.pl/g-4/c/1590-smartfony-i-telefony.html?page={ }>' za pomocą API
- *transform()* - funkcja wykorzystana w klasie GUI do przeobrażenia wyodrębnionych danych za pomocą API
- *load()* - funkcja wykorzystana w klasie GUI do wczytania wyodrębnionych i przeobrażonych danych do bazy danych za pomocą API
- *saveToFile()* - funkcja wykorzystana w klasie GUI do zapisania danych z bazy danych do pliku 'output.json' za pomocą API
- *data()* - funkcja wykorzystana w klasie GUI do wyświetlenia danych znajdujących się w bazie danych za pomocą API
- *etl()* - funkcja wykorzystana w klasie GUI do przeprowadzenia całego procesu etl za pomocą API
- *delete()* - funkcja wykorzystana w klasie GUI do usunięcia danych z bazy danych za pomocą API

Klasa GUI składa się z dwóch przycisków 'Select' i 'Quit oraz siedmiu przycisków opcji -

- *EXTRACT*
- *TRANSFORM*
- *LOAD*
- *ETL*
- *DELETE*
- *DATA*

- *SAVE TO FILE*

Klasa GUI posiada również funkcję *clicked()*, która odpowiada za menu wyboru w interfejsie użytkownika i wywołuje wybrane przez użytkownika funkcje.

## 1.2. API

Czołowymi wykorzystywanymi technologiami w tworzeniu funkcjonalności API, były 3 technologie:

- \* Flask – Mikroframework zapewniający przejrzyste mechanizmy do tworzenia API
- \* BeautifulSoup – Biblioteka języka Python służąca do wyciągania danych z wnętrza języków znacznikowych.
- \* Docker – Kontenery jako środowisko uruchomieniowe dla bazy danych oraz API.
- \* MongoDB jako nierelacyjna baza danych.

Struktura projektu API składa się z następujących plików:

app.py – Jako główny plik uruchomieniowy a zarazem inicjujący Flaska.

db\_connection.py – Plik inicjujący połączenie z bazą danych.

Extract.py, transform.py, load.py, database\_data.py jako pliki zawierające poszczególne funkcjonalności aplikacji z odwołaniem do konkretnego zasobu.

Katalog „cache” przechowuje aktualny stan danych całego procesu ETL, zgodnie z wytycznymi narzuconymi na projekt, to znaczy że:

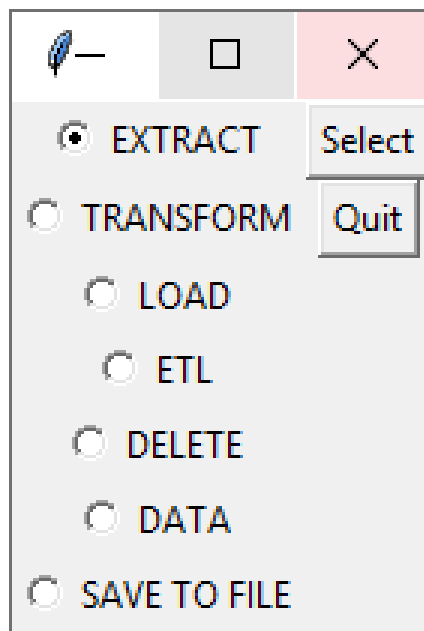
- W przypadku wykonania zapytania *extract*, zostaną *ściągnięte wszystkie HTMLe zawierające poszukiwane przez nas informacje oraz zostaną one umieszczone w katalogu „cache”*

- W przypadku zapytania *transform*, do katalogu *cache* trafi nowy plik *data.json*, zawierający wyszczególnione dane z plików *HTMLi* za to pobrane wcześniej pliki zostaną usunięte.

- W przypadku zapytania *load*, dane trafią do bazy danych, a plik *data.json* również zostanie usunięty.

## 2. DOKUMENTACJA INTERFEJSU UŻYTKOWNIKA

### 2.1. INTERFEJS UŻYTKOWNIKA



Rysunek 1. Interfejs

Interfejs użytkownika składa się z 7 przycisków opcji -

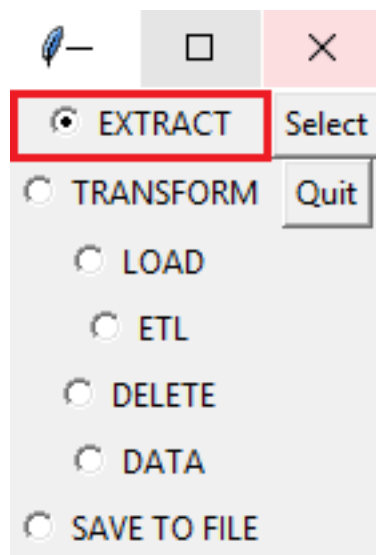
- EXTRACT
- TRANSFORM
- LOAD
- ETL
- DELETE
- DATA
- SAVE TO FILE

oraz 2 przycisków -

- SELECT
- QUIT

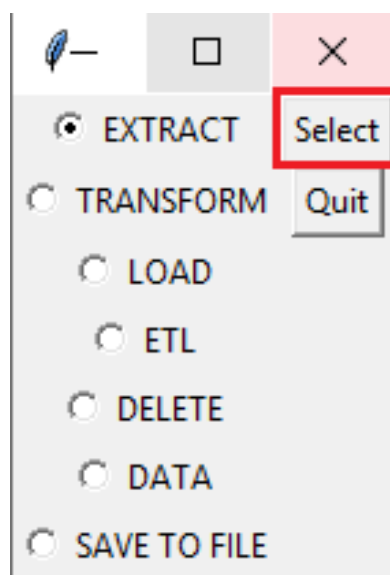
## 2.2 DZIAŁANIE APLIKACJI

1. Wybierz jeden z siedmiu przycisków opcji (przykład. EXTRACT)



Rysunek 2.  
Zaznaczony przycisk

2. Wciśnij przycisk SELECT



Rysunek 3. Przycisk