AKADEMIA NAUK STOSOWANYCH INSTYTUT INFORMATYKI STOSOWANEJ IM. KRZYSZTOFA BRZESKIEGO Przetwarzanie równoległe i rozproszone	
Sprawozdanie wykonał: Dominik Świerczyński Andrzej Posim	Tytuł ćwiczenia: „Web scraper”
Nr indeksu: 20486 20475	Data wykonania: 02.06.2024

1. Zakres prac nad aplikacją:
    a. Aplikacja pobiera, selekcjonuje i składuje wybrane dane o narzuconym profilu z witryn internetowych.
    b. Profil danych jest ustalony przez realizującego projekt. Profil danych powinien obejmować min. 4 grupy, np. adresy email, adresy korespondencyjne, schemat organizacyjny itp.
    c. Program wykorzystuje wielowątkowość/wieloprocesowość. Silnik należy zrealizować we własnym zakresie wykorzystując: multiprocessing i asyncio. Przetwarzanie ma być wieloprocesowe, najlepiej z możliwością skalowania na rdzenie procesora, dalej na komputery, dalej na klastry itp.
    d. Do parsowania kontentu należy użyć beautifulsoup.
    e. Dane mają być zapisywane w BD, np. MongoDB
    f. Program ma posiadać interfejs graficzny zrealizowany w Python (Flask lub Django)
    g. Docelowo aplikacja ma być rozproszona na min 3 moduły: interfejs (1 lub więcej kontenerów), silnik (1 kontener), BD (1 kontener). Sposób ulokowania należy opracować we własnym zakresie i potrafić uzasadnić wybory.

2. Aplikacja przeszukuje dwie popularne strony internetowe, CENEO i OLX, aby znaleźć i wyświetlić wybrane przez użytkownika produkty. Automatycznie przegląda te strony i zwraca poszukiwane kryteria takie jak:
    a. Nazwa produktu
    b. Cena produktu
    c. Zdjęcie produktu
    d. Bezpośredni link do produktu

3. Dodatkowo, do zrealizowania interfejsu graficznego został użyty Flask a do parsowania danych używamy BeautifulSoup. Flask zapewnia nam prostą i elastyczną platformę do stworzenia naszej aplikacji. Umożliwiło to na łatwe i przejrzyste prezentowanie wyników wyszukiwania naszego web scraper’a. BeautifulSoup pozwala na efektywne przetwarzanie i analizowanie zawartości stron internetowych, co pozwala na pobierania informacji z nich.

4. Aplikacja działa na czterech kontenerach, z których każdy pełni określoną rolę:
    a. Wyszukiwarka_db_1:
        i. Odpowiada za poprawne działanie bazy danych.
        ii. Obsługiwana przez MongoDB, która zapewnia przechowywanie i zarządzanie danymi produktów.

    b. Wyszukiwarka_db_service_1:
        i. Jego zadanie polega na zapisywaniu rekordów do bazy danych.
        ii. Zapewnia integralność i spójność danych przechowywanych w MongoDB.

    c. Wyszukiwarka_engine_1:
        i. Na tym kontenerze uruchomiony jest silnik web scraper'a.
        ii. Odpowiada za przeszukiwanie stron internetowych CENEO i OLX oraz pobieranie danych o produktach.

    d. Wyszukiwarka_ui_1:
        i. Odpowiada za wygląd i działanie interfejsu użytkownika.
        ii. Zapewnia użytkownikom dostęp do funkcji aplikacji poprzez przejrzysty i intuicyjny interfejs webowy.

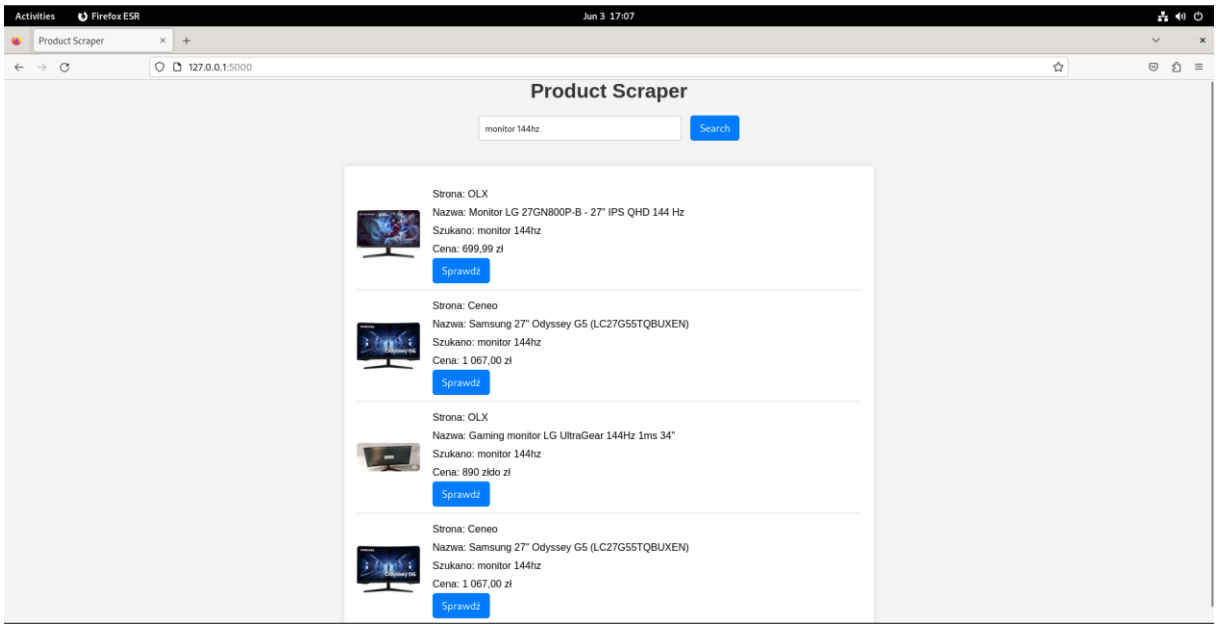
```
debian@Debian:~/wyszukiwarka$ docker compose ps --all
WARN[0000] /home/debian/wyszukiwarka/docker-compose.yml: `version` is obsolete
NAME                        IMAGE                       COMMAND                  SERVICE
wyszukiwarka_db_1           mongo:4.4                   "docker-entrypoint.s…"   db
wyszukiwarka_db_service_1   wyszukiwarka_db_service     "python app.py"          db_service
wyszukiwarka_engine_1       wyszukiwarka_engine         "python engine.py"       engine
wyszukiwarka_ui_1           wyszukiwarka_ui             "python app.py"          ui
debian@Debian:~/wyszukiwarka$
```

Fot. nr 1

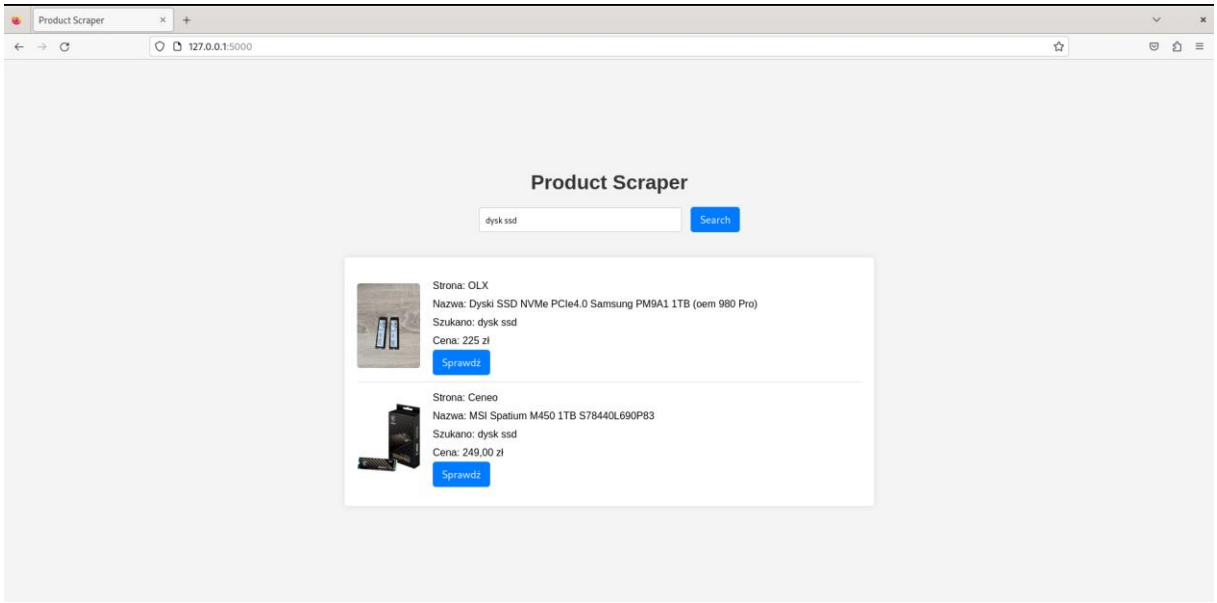
Fotografia nr 1 pokazuje, że kontenery zostały utworzone i działają poprawnie.

5. Zdjęcia z działania programu

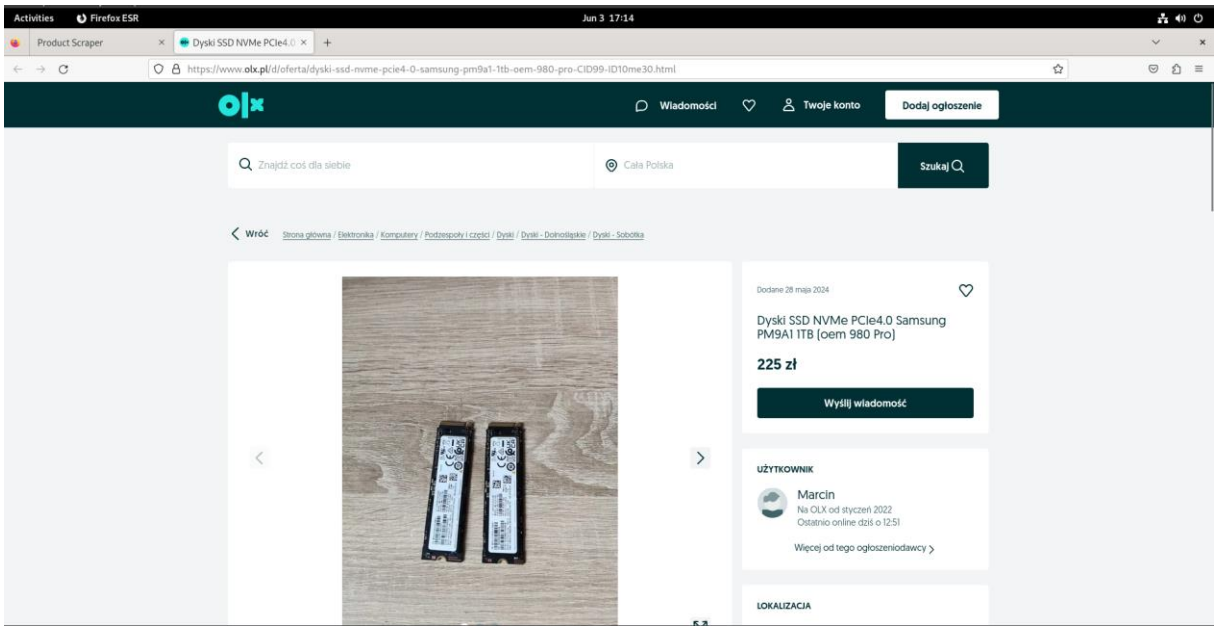
    Fotografie nr 2 i 3 pokazują przykładowe działanie naszego web scraper'a. Aplikacja pobiera nazwę, cenę oraz zdjęcie produktu. Po kliknięciu przycisku „Sprawdź" automatycznie przechodzimy na stronę z danym produktem(fot. nr 4).



Fot. nr 2



Fot. nr 3



Fot. nr 4

6. Link do kodu Git Hub
   https://github.com/swierczynski02/projekt_prir