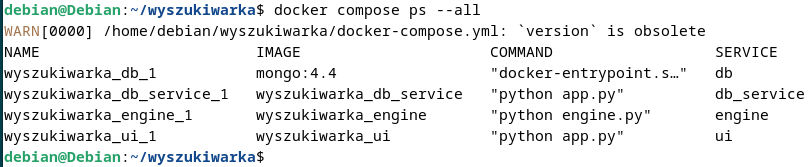
|  |  |
| --- | --- |
| **AKADEMIA NAUK STOSOWANYCH**  **INSTYTUT INFORMATYKI STOSOWANEJ IM. KRZYSZTOFA BRZESKIEGO**  **Przetwarzanie równoległe i rozproszone** | |
| Sprawozdanie wykonał:  Dominik Świerczyński Andrzej Posim | Tytuł ćwiczenia:  „Web scraper” |
| Nr indeksu:  20486 20475 | Data wykonania: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

1. Zakres prac nad aplikacją:
   1. Aplikacja pobiera, selekcjonuje i składuje  wybrane dane o narzuconym profilu z witryn internetowych.
   2. Profil danych jest ustalony przez realizującego projekt. Profil danych powinien obejmować min. 4 grupy, np. adresy email, adresy korespondencyjne, schemat organizacyjny itp.
   3. 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.
   4. Do parsowania kontentu należy użyć beautifulsoup.
   5. Dane mają być zapisywane w BD, np. MongoDB
   6. Program ma posiadać interfejs graficzny zrealizowany w Python (Flask lub Django)
   7. 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:
   1. Nazwa produktu
   2. Cena produktu
   3. Zdjęcie produktu
   4. 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:
   1. Wyszukiwarka\_db\_1:
      1. Odpowiada za poprawne działanie bazy danych.
      2. Obsługiwana przez MongoDB, która zapewnia przechowywanie   
         i zarządzanie danymi produktów.
   2. Wyszukiwarka\_db\_service\_1:
      1. Jego zadanie polega na zapisywaniu rekordów do bazy danych.
      2. Zapewnia integralność i spójność danych przechowywanych   
         w MongoDB.
   3. Wyszukiwarka\_engine\_1:
      1. Na tym kontenerze uruchomiony jest silnik web scraper’a.
      2. Odpowiada za przeszukiwanie stron internetowych CENEO i OLX   
         oraz pobieranie danych o produktach.
   4. Wyszukiwarka\_ui\_1:
      1. Odpowiada za wygląd i działanie interfejsu użytkownika.
      2. Zapewnia użytkownikom dostęp do funkcji aplikacji poprzez przejrzysty i intuicyjny interfejs webowy.

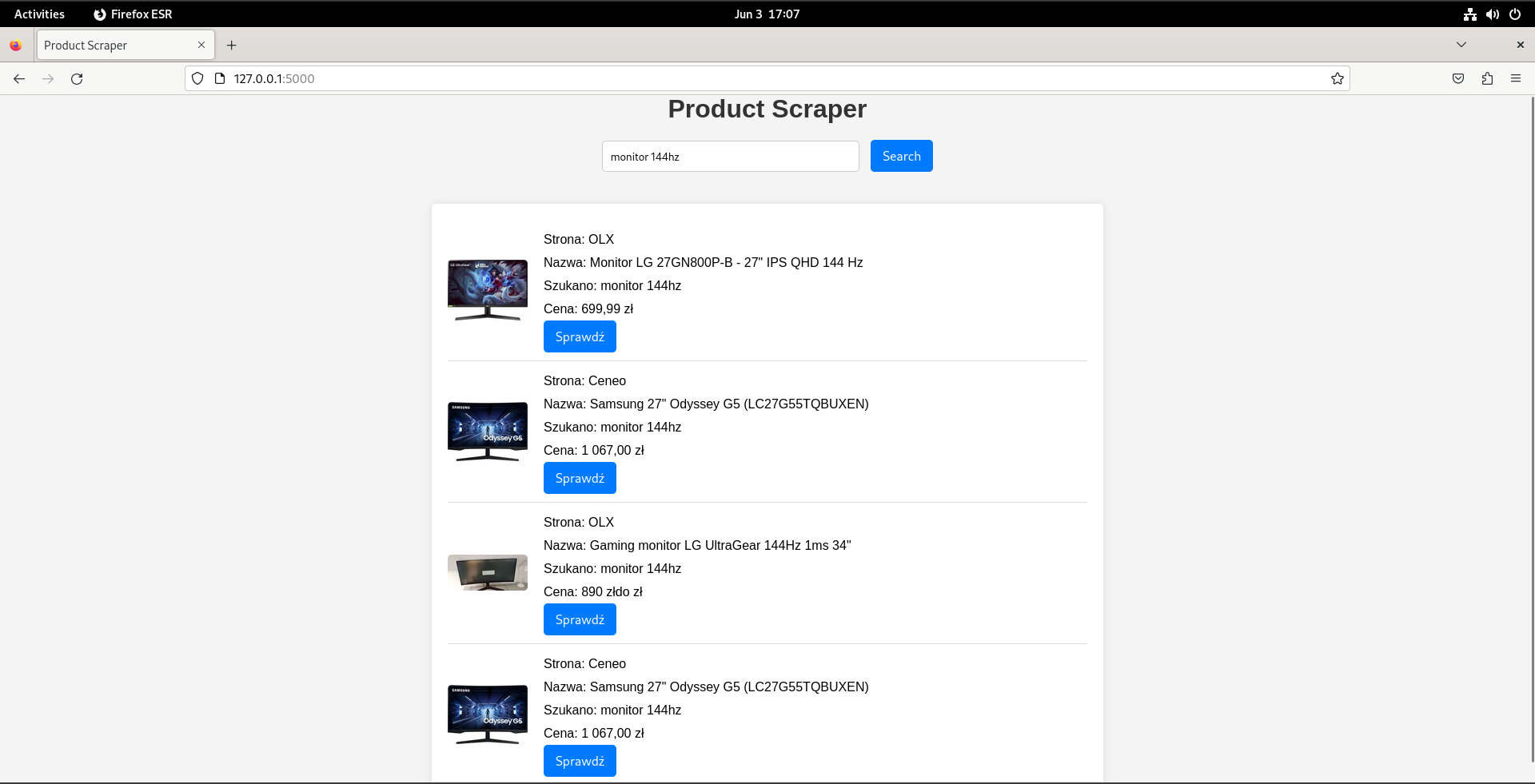


Fot. nr 1

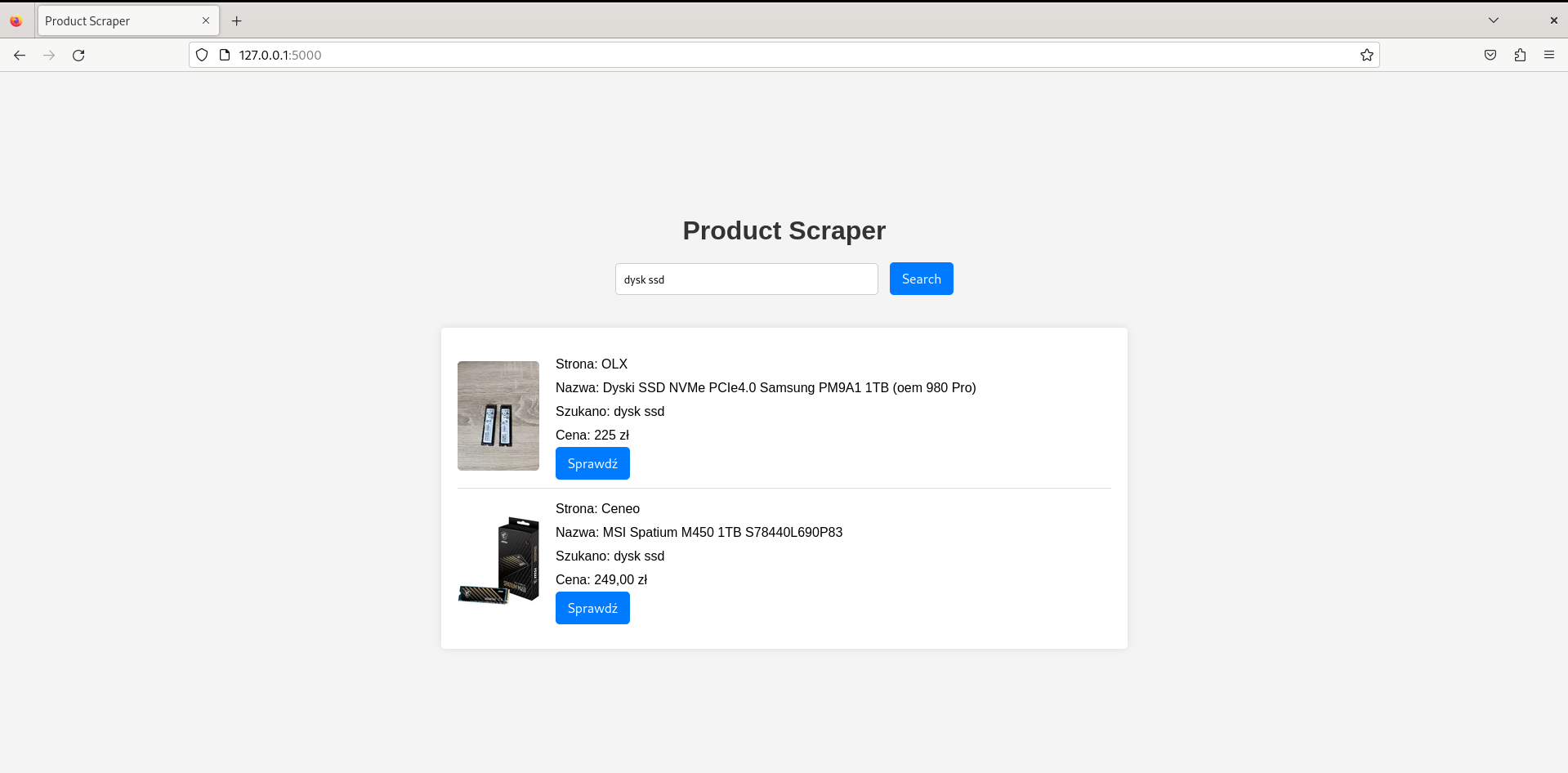
Fotografia nr 1 pokazuje, że kontenery zostały utworzone i działają poprawnie.

1. Zdjęcia z działania programu

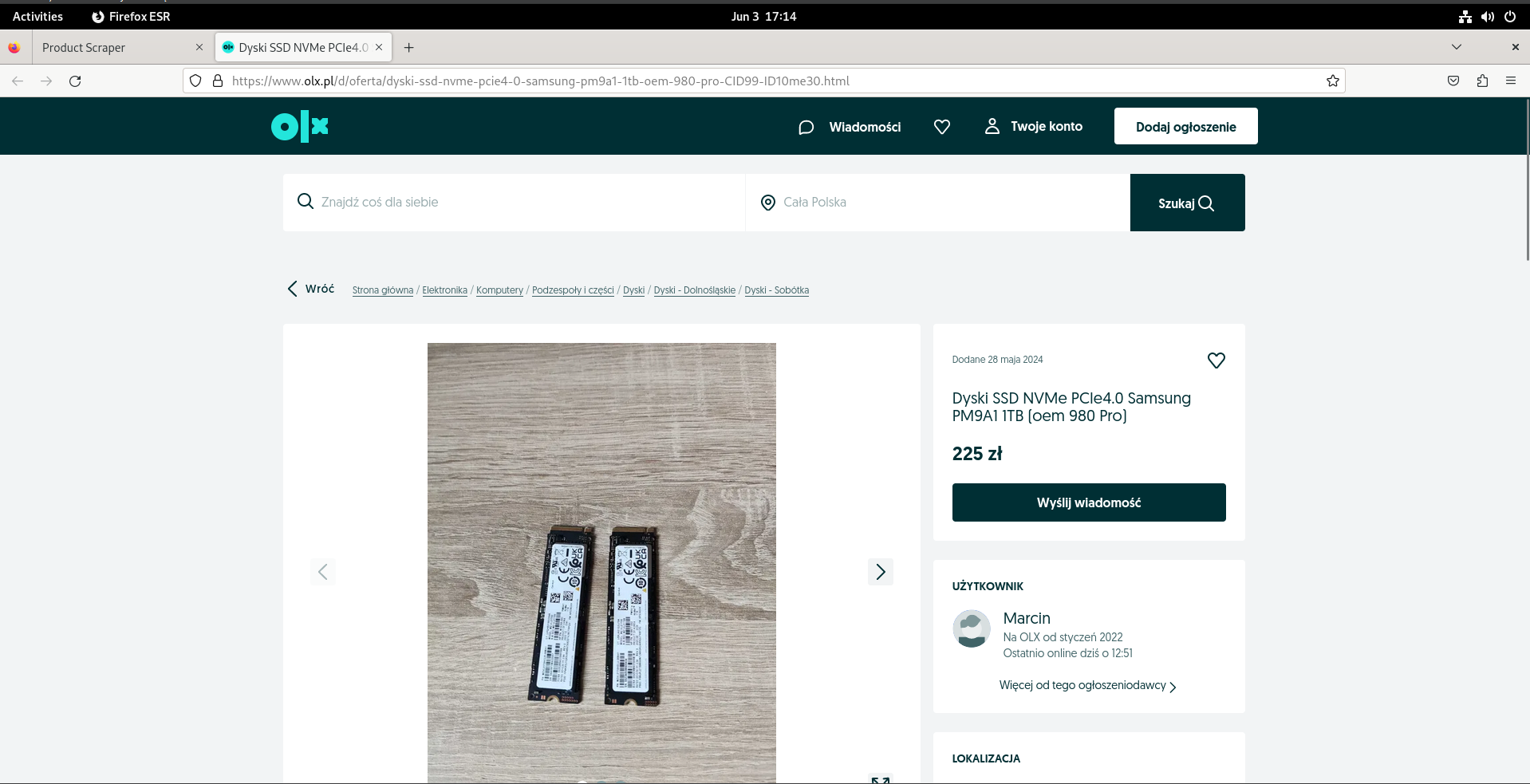
Fotografie nr 2 i 3 pokazują przykładowe działanie naszego web scraper’a. Aplikacja pobiera nazwę, cenę oraz zdjęcie produktu. Po kliknięciu przycisku „Sprawdź” automatycznie przechodzimy na stronę z danym produktem(fot. nr 4).



Fot. nr 2



Fot. nr 3



Fot. nr 4

1. Link do kodu github