Dokumentacja Projektu
Aplikacja Medium Społecznościowego

Arkadiusz Torba, 41286
Mateusz Bizoń, 41252
Urszula Gręzicka, 41260

Maj 2023

1 Opis funkcjonalny systemu

Aplikacja umożliwia użytkownikom rejestrację i logowanie, tworzenie postów z tekstem i zdjęciami, dodawanie znajomych, lajkowanie i komentowanie postów. Możliwe jest także wyszukiwanie innych użytkowników po nazwie użytkownika. Dodatkowo, użytkownicy mogą zmieniać swoje dane osobowe.

2 Opis technologii

- GitHub - narzędzie wykorzystywane do kontroli wersji kodu.

2.1 API

- Python - język programowania na którym opiera się cały system api.
- Django - framework umożliwiający w łatwy sposób obsługiwanie zapytań http.
- Django-Rest-Framework - framework rozszerzający Django ułatwiający przytwarzanie zapytań oraz umozliwiający obsługę tokenów jwt wykorzystanych w aplikacji do autentykacji.
- Sqlite3 - silnik bazy danych wykorzystanej w aplikacji.
- Heroku - aplikacja internetowa umożliwiająca hosting aplikacji do sieci www.

2.2 Web

- Javascript - język programowania używany we frameworku React.
- React - framework służący do renderowania w czasie rzeczywistym.

- Yarn - narzędzie służące do zarządzania zależnościami
- Netlify - aplikacja internetowa umożliwiająca hosting aplikacji do sieci www.

2.3 Mobile

- Kotlin - język programowania służący do implementacji aplikacji mobilnych, odpowiada za wysyłanie zapytań oraz renderowanie.

3 Wykorzystane wzorce projektowe

3.1 API

- MVC - Wzorzec Model-Widok-Kontroler to główny wzorzec wykorzystany przy tworzeniu aplikcaji w django.
- Serializer - Wykorzystywany do przetwarzania modeli django na obiekty mogące zostać wysłane w json responsie - słowniki lub listy.

3.2 Web

- Adapter - Wykorzystany do mapowania danych w celu przekazania do danego komponentu.

3.3 Mobile

- Singleton - Wykorzystany do tworzenia instancji komponentó wyświetlanych w aplikacji.
- Adapter - Wykorzystany do mapowania danych w celu przekazania do danego komponentu.

4 Instrukcje uruchomienia testów oraz systemów

4.1 API

4.1.1 Local API

- 1. Install python
- 2. Install pipenv:
  - python -m pip install pipenv
- 3. Install dependencies
  - git clone https://github.com/Suabyak/zmp-api

  - python -m pipenv install –dev

- 4. Run/make migrations
  - python -m pipenv shell
  - python manage.py makemigrations
  - python manage.py migrate

- 5. Run dev server
  - python manage.py runserver

  In case api doesn't work repeat from Step 3

4.1.2 Api tests

- Inside main project directory run:
  - pipenv shell
  - python manage.py test

4.2 Web

- First clone the repository
  - git clone https://github.com/Suabyak/zmp-web.git
- Enter the yarn command
  - yarn
- Enabling application on localhost
  - yarn start

5 Wnioski projektowe

5.1 API

Podejście programistyczne Test Driven Development pozwala na wykrycie błędów spowodowanych poprzez zmiany w innej części systemu(np. zmiana formatu w jakim są przesyłane są dane), dzięki czemu mamy łatwy dostęp do informacji gdzie wystąpił błąd. Testowanie endpointów aplikacji pozwala na zaoszczędzenie czasu po pierwsze na samym sprawdzaniu tych endpointów czy działają w odpowiedni sposób, ale przede wszystkim ułatwia nam rozwiązywanie problemów.

5.2 Web

Porównując tworzenie requestów w React i Android, React wydaje się być bardziej intuicyjny w tym zakresie. Nauczenie się tworzenia requestów w React nie powinno zająć sporo czasu osobom, które dopiero zaczynają pracować z Reactem.

5.3 Mobile

W przypadku testowania połączenia z API w aplikacji Android, konieczne jest wpisanie adresu IP komputera zamiast użycia "localhost"lub "127.0.0.1"podczas testowania w trybie lokalnym. Dodaje to niepotrzebnej komplikacji w procesie tworzenia.

6 Hosting

6.1 API

`https://zmp-social-app.herokuapp.com`

6.2 Web

`https://master--social-media-app-zmp.netlify.app/login`

6.3 Mobile

`https://drive.google.com/file/d/1rRrygQXK-JFlr6PYZL-OdO8ohpOADvS-/view?usp=share_link`