

Projektowanie i programowanie systemów internetowych II

PROJEKT ZESPOŁOWY

TELETEXT



Mikołaj Śniechowski
Krzysztof Balicki
Adrian Kurlej
Patryk Biernacki
Przemysław Baszak

1 Opis funkcjonalny systemu

Nasza aplikacja ma na celu odwzorowanie rzeczywiście istniejącej telegazety znanej z telewizji do formy aplikacji internetowej. Tak jak w „tradycyjnej” formie aplikacja Teletext przedstawia szereg najważniejszych informacji takich jak skrót bieżących wydarzeń z różnych kategorii, kursy walut i kryptowalut, program telewizyjny, czy też aktualną pogodę. Ponadto w aplikacji istnieje też możliwość głosowania w ankietach, oraz wystawiania krótkich ogłoszeń. W celu skorzystania z elementów interaktywnych telegazety konieczne jest założenie konta przez użytkownika w naszej aplikacji. Kontrolę nad ogłoszeniami sprawuje moderacja stworzona ze wybranych przez administratora kont.

2 Streszczenie opisu technologicznego

Nasza aplikacja została stworzona w oparciu o Vue.js (jako frontend) oraz Django i PostgreSQL (jako backend). Z pomocą frameworka Vue.js została stworzona aplikacja kliencka Teletext, którą której korzysta użytkownik. Aplikacja kliencka zawiera Telegazetę wraz wszystkimi treściami oraz część dostępną uwierzytelnieniu czyli możliwość dodania ogłoszenia i zagłosowania w ankiecie. Aplikacja komunikuje się z backendem stworzonym w Django. W naszym projekcie skorzystaliśmy z panelu administracyjnego Django aby zarządzać zawartością. Część backendowa przechowuje wszystkie dane na osobnej bazie danych PostgreSQL.

3 Instrukcja lokalnego i zdalnego uruchomienia systemu

3.1 Instrukcja uruchomienia lokalnego

Frontend:

1. Pobieramy repozytorium
2. Instalujemy środowisko uruchomieniowe Node.js
3. Otwierając repozytorium za pomocą Visual Studio Code otwórz terminal i wpisujemy polecenie „npm install”, które zainstaluje wszystkie niezbędne pakiety
4. Wypełniamy plik `.env` wszystkimi niezbędnymi danymi
5. W terminalu wpisujemy „npm run serve”
6. Aplikacja powinna być uruchomiona pod adresem: `localhost:8080`

Backend:

1. Pobieramy repozytorium
2. Instalujemy PostgreSQL 15.1 oraz Python w wersji 3.9
3. W Visual Studio Code zainstalujemy dodatek „Python” i wybieramy interpreter Pythona
4. W folderze `teletext` stwórz plik `.env` w którym zawarte będą wszystkie niezbędne informacje takie jak dane do bazy danych czy `secret key`
5. Instalujemy wszystkie wymagane pakiety z pomocą „pip install -r requirements.txt”
6. Tworzymy migracje za pomocą „python manage.py makemigrations”
7. Wyprowadzamy migracje do serwera bazodanowego za pomocą „python manage.py migrate”
8. Tworzymy superusera stosując „python manage.py createsuperuser” (niezbędny do rejestracji do panelu administracyjnego)
9. Uruchamiamy serwer z pomocą „python manage.py runserver”
10. Aplikacja powinna być dostępna pod adresem `http://127.0.0.1:8000/`

3.2 Instrukcja zdalnego uruchomienia systemu

Frontend: Naszą aplikację Frontendową upubliczniliśmy z wykorzystaniem platformy Azure a dokładnie z użyciem Azure Static Web Apps. Komponent ten został podłączony do repozytorium na GitHub. Z użyciem GitHub Actions zawartość repozytorium jest deployowana przy każdej aktualizacji głównego branch'a. Backend: Część backendową zdeployowaliśmy z pomocą komponentów Azure App Service (skonfigurowany dla Pythona 3.9) oraz Baza danych PostgreSQL. Zawartość repozytorium backendowego jest wdrażana na App Service za każdym commitem lub pull requestem. W konfiguracji App Service zdefiniowane są wszystkie zmienne niejawne niezbędne do działania aplikacji w tym połączenie do bazy danych. Z pomocą wewnętrznego SSH zostały też przeprowadzone wszystkie migracje oraz został stworzony superuser.

4 Instrukcja uruchamiania testów oraz opis funkcjonalności

Testy backendu

Instrukcja:

Aby przeprowadzić testy jednostkowe backendu należy w terminalu wpisać "python manage.py test".

Opis funkcjonalności:

Testowane są moduły "accounts" oraz "annoucement". Testy "accounts" sprawdzają funkcjonalności jak logowanie, rejestracja oraz endpoint z profile detail. Testy "announcements" testują wszystkie usługi (dodawanie ogłoszenia, aktualizacja ogłoszenia, usuwanie ogłoszenia), zarówno z perspektywy użytkownika zalogowanego jak i niezalogowanego. Ponadto sprawdzane jest wyświetlanie list kategorii ogłoszeń oraz samych ogłoszeń.

5 Link do dokumentacji projektu

<https://github.com/mikolajsniechowski/Teletext-frontend/tree/master/docs>

6 Wnioski projektowe

Proces tworzenia aplikacji był procesem wymagającym dużej ilości uwagi. Liczne problemy napotkane po drodze, przyczyniły się skomplikowania naszej pracy, ale i do nabrania większego doświadczenia. W szczególności problemów przysporzyła publikacja aplikacji na zewnętrznym serwisie. Dzięki korzystaniu w projekcie z rozmaitych technologii mogliśmy rozwinać swoje umiejętności w tworzeniu aplikacji internetowych, które to w pewnością przydadzą się przy tworzeniu kolejnych projektów. Ponadto poprawiliśmy swoje kompetencje w organizacji pracy.