

Dokumentacja techniczna aplikacji Jasne

1. Opis aplikacji

Aplikacja Jasne to aplikacja społeczna stworzona w celu fact checkingu realizowanego przez jej uczestników. Jej celem jest zapewnienie przestrzeni do weryfikacji informacji zdobywanych w Internecie oraz weryfikowanie ich dzięki ekspertom w danej dziedzinie. Aplikacja umożliwia dyskusję w danych tematach , które są wyznaczone przez kategorię. Każdy post oraz komentarz może zostać pozytywnie oceniony przez użytkowników, co powinno przekładać się na zainteresowanie postem lub rzetelność udzielonej odpowiedzi. Każdy użytkownik posiada możliwość edytowania swojego profilu, w tym ustawienia dziedziny w której się specjalizuje. Natywnym językiem aplikacji jest język angielski.

Aplikacja posiada panel administracyjny, który umożliwia:

- edytowanie postów
- usuwanie postów
- edytowanie kategorii
- blokowanie użytkowników

Aplikacja w zależności od potrzeb może zostać uruchomiona:

- W środowisku dockerowym rozłożona na kontenery: API, Nginx + Klient webowy, DB. Całość rozwiązana może być uruchamiana za pomocą rozwiązania docker-compose, które zapewnia połączenie między kontenerami oraz ustalanie zależności między nimi
- Klasycznie bez wirtualizacji na maszynie hosta jako API oraz klient webowy

2. Zastosowane technologie

Technologie wykorzystywane w projekcie to:

- Serwer aplikacji: Spring Boot w wersji 3.0.6
- Baza danych: (PostgreSQL) 15.3
- Klient Web: React 18.0.2
- Reverse proxy: Nginx
- Konteneryzacja: Docker

3. Wymagania techniczne

Wymagania techniczne sprecyzowane zostały głównie pod implementację za pomocą wirtualizacji.

- Ogólne wymagania sprzętowe dla serwera
 - System operacyjny: Ubuntu 20.04 LTS lub Windows z obsługą Docker.
 - CPU: Minimum 4 vCPU.
 - RAM: Minimum 8GB RAM.
 - Storage: Minimum 100GB SSD.
 - Network: Stabilne połączenie sieciowe z odpowiednimi przepustowościami dla komunikacji serwera aplikacji, bazy danych oraz klienta webowego.
- Specyfikacje Docker
 - Docker: Docker Engine w wersji 20.10 lub nowszy.
 - Docker Compose: Docker Compose w wersji 1.29 lub nowszy.
- Serwer aplikacji: Spring Boot 3.0.6
 - Obraz Docker: Oficjalny obraz JDK 17 lub nowszy, skonfigurowany do uruchamiania aplikacji Spring Boot.
 - Porty kontenera: Port 8080 (lub inny skonfigurowany w aplikacji) dla komunikacji HTTP/HTTPS.
- Baza danych: PostgreSQL 15.3
 - Obraz Docker: Oficjalny obraz PostgreSQL 15.3.
 - Porty kontenera: Port 5432 dla komunikacji z bazą danych.

- Persistent Storage: Volume Docker do przechowywania danych bazy danych poza kontenerem, aby zapewnić trwałość danych.
 - Konfiguracja:
 - Użytkownik i hasło dla bazy danych skonfigurowane przez zmienne środowiskowe.
- Klient Web: React 18.0.2
 - Obraz Docker: Node.js 16.0 lub nowszy do budowy aplikacji React.
 - Build: Skrypt builda do wygenerowania statycznych plików HTML, CSS, JS, które będą serwowane przez Nginx.
 - Persistent Storage: Volume Docker do przechowywania plików builda poza kontenerem.
- Reverse proxy: Nginx
 - Obraz Docker: Oficjalny obraz Nginx.
 - Porty kontenera: Port 80 dla HTTP oraz port 443 dla HTTPS.
 - Certyfikaty SSL/TLS: Volume Docker do przechowywania certyfikatów SSL/TLS poza kontenerem.
 - Konfiguracja:
 - Proxy pass do aplikacji Spring Boot na porcie 8080.

4. Konfiguracja oraz wdrożenie

- Kod serwera api znajduje się na branchu main: <https://github.com/optim77/jasne>
- Kod ze skryptami wdrażania na serwer oraz z klientem webowym (folder front) znajduje się na branchu deploy: <https://github.com/optim77/jasne/tree/deploy>

Poniższy opis dotyczy brancha deploy:

- Folder backend zawiera skompilowany serwer napisany Spring Boot
- Folder db zawiera skrypt inicjalizujący oraz wolium danych kontenera postgres
- Folder front zawiera pliki klienta webowego
- Folder nginx zawiera konfigurację reverse proxy
- Zmienne środowiskowe umieszczone są w pliku .env

W backend/Dockerfile w komendzie uruchomieniowej są kolejne zmienne, z którymi jest uruchamiany serwer

Aplikację można uruchomić za pomocą docker compose uruchamiając komendę docker compose up w głównym katalogu. Aplikacja zostanie uruchomiona pod adresem localhost.

Uruchomienie:

1. Zbudowana wersja aplikacji została udostępniona pod adresem: <http://20.123.57.10/>
2. Uruchomienie lokalnie za pomocą dockera: w głównym folderze aplikacji gdzie znajduje się plik docker-compose.yaml należy uruchomić polecenie docker-compose up -d
3. W celu lokalnego uruchomienia poszczególnych serwisów aplikacji należy:

Uruchomić serwer backendowy za pomocą polecenia `java -jar dstay-0.0.6-SNAPSHOT.jar`

Serwer wymaga połączenia do bazy danych postgres pod adresem `postgresql://db:5432/jasne` username: postgres hasło: root

Serwer zostanie uruchomiony na standardowym porcie: 8080

Uruchomienie deweloperskiego serwera klienckiego należy wykonać w folderze /front za pomocą polecenia `npm start`

W pliku /front/.env znajduje się zmienna środowiskowa, która odpowiada za adresację backendu, zakomentowana wartość to adres zdalnego serwera API z punktu 1

Poświadczenia dla panelu administratora, który znajduje się pod adresem /admin:

Mail: admin@jasne.com hasło: 123123

Przykładowe dane do serwera spring wgrywane są bezpośrednio w aplikacji, znajdują się one w ścieżce `dstay\src\main\resources\data.sql`

5. Instrukcje użytkownika

Podstawowe funkcje: Opis głównych funkcji systemu i jak z nich korzystać.

Przykłady scenariuszy użycia:

Podstawowe funkcje

1. Rejestracja

Opis funkcji:

Rejestracja umożliwia nowym użytkownikom założenie konta w systemie.

Kroki:

- Na stronie głównej, kliknij przycisk "Sign up".
- Wypełnij formularz rejestracyjny, podając:
 - Email
 - Hasło
 - Potwierdzenie hasła
 - Zaakceptuj zasady
- Kliknij przycisk "Sign-up".

2. Logowanie

Opis funkcji:

Logowanie umożliwia zarejestrowanym użytkownikom dostęp do ich konta.

Kroki:

- Na stronie głównej, kliknij przycisk "Sign in".
- Wprowadź swój adres e-mail i hasło.
- Kliknij przycisk "Login".
- Jeśli dane są poprawne, zostaniesz przekierowany na stronę główną aplikacji.

3. Tworzenie postów

Opis funkcji:

Użytkownicy mogą tworzyć nowe posty, aby dzielić się treściami z innymi użytkownikami.

Kroki:

- Po zalogowaniu, przejdź do sekcji "Ask question".
- Wypełnij formularz tworzenia posta, podając:
 - Tytuł
 - Treść posta
 - Dodaj linki oraz kategorie.
 - Kliknij przycisk "Send".

4. Edytowanie postów

Opis funkcji:

Użytkownicy mogą edytować swoje posty, aby wprowadzać zmiany lub poprawki.

Kroki:

- Przejdź do swoich aktywności klikając w navbarze przycisk „Profile” a następnie „Activity”
- Kliknij przycisk "Edit" przy wybranym poście.
- Wprowadź zmiany w treści posta.
- Kliknij przycisk "Update".

5. Dodawanie komentarzy

Opis funkcji:

Użytkownicy mogą dodawać komentarze do postów innych użytkowników, aby wyrażać swoje opinie lub zadawać pytania.

Kroki:

- Przejdź do posta, który chcesz skomentować.
- Wprowadź treść komentarza w polu komentarza.
- Kliknij przycisk "Send".

6. Edytowanie komentarzy

Opis funkcji:

Użytkownicy mogą edytować swoje komentarze, aby wprowadzać zmiany lub poprawki.

Kroki:

- Przejdź do swoich aktywności klikając w navbarze przycisk „Profile” a następnie „Activity”
- Kliknij przycisk "Edit" przy wybranym komentarzu.
- Wprowadź zmiany w treści komentarza.
- Kliknij przycisk "Update".

7. Ocenianie postów

Opis funkcji:

Użytkownicy mogą oceniać posty innych użytkowników, wyrażając swoje poparcie lub niechęć.

Kroki:

- Przejdź do posta, który chcesz ocenić.
- Kliknij przycisk "Up".

8. Ocenianie komentarzy

Opis funkcji:

Użytkownicy mogą oceniać komentarze innych użytkowników, wyrażając swoje poparcie lub niechęć.

Kroki:

- Przejdź do komentarza, który chcesz ocenić.
- Kliknij przycisk "Up".

Uwagi do aplikacji:

Aplikacja nie posiada pełnej responsywności na urządzeniach mobilnych

Niektóre funkcje nie zostały w pełni dopracowane