# Dokumentacja Systemu i Schematu Bazy Danych

# Spis Treści

## 1. Architektura Systemu

### 1.1 Komponenty Kontenerów Docker

Frontend Container:  
- Port: 3000:80  
- Odpowiedzialny za interfejs użytkownika.  
- Zbudowany w kontekście ./frontend.  
- Umożliwia interaktywną pracę (stdin\_open: true, tty: true).

Backend Container:  
- Port: 8000:8000  
- Obsługuje logikę biznesową i API.  
- Konfiguracja przez zmienne środowiskowe:  
 - POSTGRES\_USER  
 - POSTGRES\_PASSWORD  
 - POSTGRES\_DB  
 - POSTGRES\_HOST  
- Przechowuje media w persystentnym woluminie backend\_media.

PostgreSQL Container:  
- Port: 5432:5432  
- Baza danych PostgreSQL.  
- Dane przechowywane w woluminie postgres\_data.  
- Konfigurowana przez te same zmienne środowiskowe co backend.

Adminer Container:  
- Port: 8080:8080  
- Narzędzie do zarządzania bazą danych.  
- Zależny od kontenera PostgreSQL.

### 1.2 Konfiguracja Sieci

- Wszystkie kontenery (oprócz frontendu) połączone w sieci tagtron\_network.  
- Umożliwia bezpieczną komunikację między usługami.

## 2. Schema Bazy Danych

### 2.1 Zarządzanie Użytkownikami

Tabela Users:  
- id: Klucz główny.  
- email: Unikalny adres email.  
- username: Unikalna nazwa użytkownika.  
- first\_name: Imię.  
- last\_name: Nazwisko.  
- hashed\_password: Zaszyfrowane hasło.  
- is\_active: Status aktywności konta.  
- role\_id: Powiązanie z rolą systemową.

Tabela Roles:  
- id: Klucz główny.  
- name: Unikalna nazwa roli systemowej.

### 2.2 Zarządzanie Projektami

Tabela Projects:  
- id: Klucz główny.  
- title: Tytuł projektu.  
- description: Opis projektu.  
- owner\_id: ID właściciela (powiązanie z Users).

Tabela ProjectUsers:  
- id: Klucz główny.  
- project\_id: Powiązanie z projektem.  
- user\_id: Powiązanie z użytkownikiem.  
- role\_id: Powiązanie z rolą projektową.

Tabela ProjectRoles:  
- id: Klucz główny.  
- name: Unikalna nazwa roli projektowej.

### 2.3 Zarządzanie Obrazami

Tabela Images:  
- id: Klucz główny.  
- name: Nazwa obrazu.  
- path: Unikalna ścieżka do pliku.  
- upload\_date: Data przesłania.  
- size\_x: Szerokość obrazu.  
- size\_y: Wysokość obrazu.  
- project\_id: Powiązanie z projektem.  
- owner\_id: Powiązanie z właścicielem.

Tabela Labels:  
- id: Klucz główny.  
- image\_id: Powiązanie z obrazem.  
- label: Etykieta/adnotacja.  
- create\_time: Czas utworzenia.  
- position\_x1: Pozycja X początku bounding box.  
- position\_y1: Pozycja Y początku bounding box.  
- position\_x2: Pozycja X końca bounding box.  
- position\_y2: Pozycja Y końca bounding box.  
- owner\_id: Powiązanie z twórcą etykiety.

## 3. Wnioski i Rekomendacje

### 3.1 Mocne Strony Systemu

- Dobrze przemyślana architektura mikrousługowa.  
- Zaawansowany system uprawnień na dwóch poziomach.  
- Elastyczny system adnotacji obrazów.  
- Skalowalność dzięki konteneryzacji.  
- Bezpieczne przechowywanie danych w woluminach.

### 3.2 Potencjalne Obszary do Rozwoju

- Możliwość dodania cache'owania dla często używanych obrazów.  
- Implementacja systemu wersjonowania etykiet.  
- Rozważenie dodania systemu powiadomień.  
- Możliwość rozszerzenia o funkcje eksportu/importu projektów.  
- Dodanie mechanizmów automatycznego backupu.

### 3.3 Zalecenia Bezpieczeństwa

- Regularne aktualizacje kontenerów.  
- Monitoring dostępu do bazy danych.  
- Implementacja rate limitingu dla API.  
- Regularne audyty uprawnień użytkowników.  
- Szyfrowanie komunikacji między kontenerami.

### 3.4 Perspektywy Rozwoju

- Możliwość integracji z systemami CI/CD.  
- Dodanie wsparcia dla uczenia maszynowego.  
- Rozszerzenie funkcjonalności o pracę zespołową.  
- Implementacja API dla zewnętrznych integracji.  
- Rozwój funkcji raportowania i analityki.  
  
