Dokumentacja
Techniki Internetowe

System Zarządzania Projektami

Wydział Fizyki i Informatyki Stosowanej

Krystsina Mironenka

29.01.2026

Spis treści

1 Opis Projektu

TaskFlow to autorska platforma do zarządzania zadaniami, inspirowana metodologią Agile i narzędziami typu Jira. Aplikacja pozwala na pełną kontrolę nad cyklem życia zadania w środowisku wieloprojektowym.

2 Szczegółowe Funkcjonalności

System oferuje szeroki zakres funkcji wspierających współpracę zespołową:

2.1 Zarządzanie Projektami

- **Tworzenie i dołączanie:** Użytkownik może tworzyć nowe projekty lub dołączać do istniejących przy użyciu unikalnego kodu zaproszenia (np. 275358).
- **Kategoryzacja:** Projekty dzielą się na *Aktywne* oraz *Archiwalne*. Archiwizacja projektu blokuje edycję zadań (ikona kłódki).

2.2 Tablica Kanban i Zadania

- **Workflow:** Zadania przechodzą przez statusy: `TO DO`, `IN PROGRESS`, `IN REVIEW` oraz `DONE`.
- **Typy zadań:** System rozróżnia zadania typu *Bug* , *Feature* oraz standardowe *Task*.
- **Atrybuty:** Każde zadanie posiada unikalny identyfikator (np. KAN-690), tytuł oraz przypisanego wykonawcę z widocznym awatarem.

2.3 Zarządzanie Zespołem i Profilem

- **Role użytkowników:** System obsługuje role takie jak *Owner*, *Admin* oraz *User*, co pozwala na granulację uprawnień (np. usuwanie członków).
- **Edycja profilu:** Możliwość zmiany imienia, nazwiska oraz adresu e-mail.
- **System Awatarów:** Integracja z backendem umożliwia przesyłanie plików graficznych, które są dynamicznie wyświetlane w interfejsie.

3 Bezpieczeństwo i Autoryzacja JWT

Kluczowym elementem systemu jest bezpieczny mechanizm autoryzacji oparty na standardzie **JSON Web Token (JWT)**.

3.1 Mechanizm działania

1. Podczas logowania serwer weryfikuje dane i generuje zaszyfrowany token JWT.
2. Token jest przesyłany do klienta i zapisywany w `localStorage`.
3. Każde kolejne żądanie do API zawiera token w nagłówku *Authorization: Bearer <token>*.

3.2 Protected Routes

W aplikacji zaimplementowano komponent `ProtectedRoute`, który sprawdza obecność tokena przed załadowaniem komponentów prywatnych. W przypadku braku autoryzacji użytkownik widzi powiadomienia o błędzie generowane przez **React-Toastify** i jest przekierowywany do strony logowania.

4 Wykorzystane Technologie

- **Frontend:** React.js z wykorzystaniem hooków (*useState*, *useEffect*) oraz *React Router* do nawigacji.
- **Baza Danych:** MongoDB Atlas – dane są przechowywane w dokumentach, co pozwala na łatwe mapowanie relacji użytkownik-projekt.
- **Stylizacja:** Bootstrap 5 – wykorzystany do budowy responsywnego Sidebaru oraz kart zadań.
- **Komunikacja:** Axios z obsługą błędów sieciowych i przechwytywaniem statusów 401.

5 Rozwiązywanie Problemów Wdrożeniowych

Dokumentacja uwzględnia proces wdrożenia na **AWS EC2**. Kluczowe aspekty to:

- **Docker Networking:** Separacja usług i wystawienie portów 80 (web) oraz 5000 (API).
- **CORS:** Konfiguracja backendu umożliwiająca komunikację między różnymi adresami IP.
- **Persistent Storage:** Użycie wolumenów Dockera, aby zdjęcia profilowe nie znikały po restarcie kontenera.