

# Sprawozdanie z Pracy Projektowej

Dominik Gwóźdź

8.06.2024

## Spis treści

<b>1</b>	<b>Opis funkcjonalny systemu</b>	<b>2</b>
1.1	Główne funkcjonalności . . . . .	2
<b>2</b>	<b>Opis technologiczny</b>	<b>2</b>
<b>3</b>	<b>Wyszczególnione wdrożone zagadnienia kwalifikacyjne</b>	<b>2</b>
<b>4</b>	<b>Instrukcja lokalnego i zdalnego uruchomienia systemu</b>	<b>3</b>
4.1	Lokalna instalacja . . . . .	3
<b>5</b>	<b>Wnioski projektowe</b>	<b>4</b>

# 1 Opis funkcjonalny systemu

System to forum internetowe poświęcone tematyce anime. Użytkownicy mogą zakładać nowe wątki, odpowiadać na istniejące posty, rejestrować się, logować, zmieniać język interfejsu oraz przeglądać najnowsze i najpopularniejsze wątki. Administratorzy mają możliwość zarządzania wątkami oraz postami.

## 1.1 Główne funkcjonalności

- Rejestracja i logowanie użytkowników
- Zakładanie nowych wątków i dodawanie postów
- Przeglądanie kategorii i wątków
- Zmiana języka interfejsu
- Wyświetlanie najnowszych i najpopularniejszych wątków
- Edycja i usuwanie postów (dla autora i administratora)
- System cache'owania dla poprawy wydajności
- Logowanie akcji użytkowników (logowania, wylogowania, tworzenie wątków i postów)

## 2 Opis technologiczny

Projekt został zrealizowany przy użyciu następujących technologii:

- **Python 3.12:** Język programowania wykorzystywany do tworzenia logiki aplikacji.
- **Django 5.0.6:** Framework webowy użyty do budowy aplikacji.
- **Django REST Framework:** Rozszerzenie Django do tworzenia API.
- **MySQL:** Relacyjna baza danych do przechowywania informacji o użytkownikach, wątkach i postach.
- **Bootstrap 4:** Framework CSS użyty do stylizacji interfejsu użytkownika.
- **JavaScript:** Używany do dynamicznych interakcji w przeglądarce.
- **Git:** System kontroli wersji używany do zarządzania kodem źródłowym.

## 3 Wyszczególnione wdrożone zagadnienia kwalifikacyjne

- **Uwierzytelnianie użytkowników:** Implementacja rejestracji, logowania i wylogowania.
- **CRUD dla wątków i postów:** Tworzenie, odczyt, aktualizacja i usuwanie wątków oraz postów.

- **Cache’owanie:** Wykorzystanie Redis do cache’owania danych dla poprawy wydajności.
- **Miedzynarodowość:** Możliwość zmiany języka interfejsu przez użytkownika.
- **Logowanie akcji:** Rejestrowanie akcji użytkowników takich jak logowanie, wylogowanie, tworzenie wątków i postów.
- **API:** Implementacja API do zarządzania wątkami przy użyciu Django REST Framework.
- **Responsywny design:** Zastosowanie Bootstrap do zapewnienia responsywności interfejsu.

## 4 Instrukcja lokalnego i zdalnego uruchomienia systemu

### 4.1 Lokalna instalacja

1. Sklonuj repozytorium projektu:

```
git clone https://github.com/dominog125/PPSI-Project.git
```

2. Utwórz i aktywuj wirtualne środowisko:

```
python -m venv venv
source venv/bin/activate # Na Windows: venv\Scripts\activate
```

3. Zainstaluj zależności:

```
pip install -r requirements.txt
```

4. Skonfiguruj baze danych:

- Upewnij się, że MySQL jest zainstalowany i uruchomiony.
- Utwórz baze danych:

```
CREATE DATABASE anime_forum CHARACTER SET UTF8;
```

- Skonfiguruj połączenie do bazy danych w `settings.py`.

5. Wykonaj migracje bazy danych:

```
python manage.py makemigrations
python manage.py migrate
```

6. Uruchom serwer:

```
python manage.py runserver
```

## 5 Wnioski projektowe

Projekt został zrealizowany zgodnie z wymaganiami. Implementacja funkcji takich jak cache'owanie, API, międzynarodowość oraz logowanie akcji użytkowników znacznie poprawiła wydajność i funkcjonalność systemu.

Projekt dostarczył cenne doświadczenie w pracy z Django, REST Framework, MySQL i innymi technologiami, które będą przydatne w przyszłych projektach.