Platformy Programistyczne	
Kierunek *Informatyczne Systemy Automatyki - IZT*	**Termin** *wtorek TN* 17^{05}
Imię, nazwisko, numer albumu - osoby zarządzającej zespołem *Jędrzej Boruczkowski 272521, Krzysztof Brylski 272490*	**Data** *10.06.2025*

Raport – nr 2 (końcowy)

1 Wstęp

1.1 Cel dokumentu

Celem dokumentu jest podsumowanie realizacji projektu z przedmiotu Platformy Programistyczne. W ramach dokumentu przedsatwiono opis API oraz widoków obsługoiwanych przez naszą aplikację oraz pokazano screenshoty działającej aplikacji z perpsketywy użytkownika.

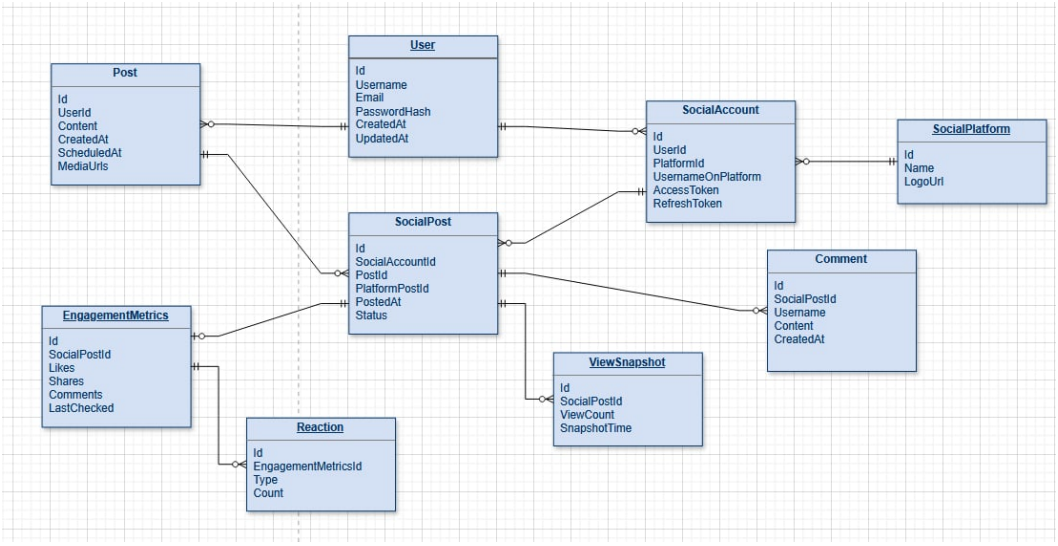
1.2 Zakres projektu

Nasz projekt polega na implementacji prostej aplikacji webowej w .NET z użyciem Entity Framework umożliwiającej planowanie wstawiania postów na media społecznościowe w danym terminie. Zrealizowanie projektu wymagało od nas dodatkowego przestudiowania API wybranych platform (X/Twitter, Facebook). Ostateczna wersja naszej aplikacji zawiera wsparcie dla API Facebooka oraz interfejs umożliwiający łatwe dodanie nowych platform społecznościowych w przyszłości. Niestety połączenie z API X/Twittera nie udało się ze względu na skomplikowane warunki użytkowania i wymóg akceptacji naszej aplikacji ze strony firmy X.

2 Baza danych

2.1 Diagram encji

Jako pierwszy krok pracy zaplanowaliśmy wszystkie funkcjonalności, a następnie przygotowaliśmy diagram encji, aby ułatwić sobie wdrożenie modeli w Entity Framework. Ostateczna wersja relacji w bazie danych nieznacznie różni się od tej zaplanowanej przez nas na początku prac. Wdrożyliśmy głównie uproszczenia takie jak zamiana encji SocialPlatform na prosty ENUM.



Rysunek 1: Diagram encji - pierwsza wersja z kwietnia

2.2 Wdrożenie

Do wdrożenia bazy danych wykorzystaliśmy Entity Framework połączony z bazą MySQL oraz systemem migracji. W pełni zaimplementowaliśmy wszystkie encje przedstawione na opracowanym przez nas diagramie encji, który powstał na początku projektu. Dzięki temu, mimo napotkanych ograniczeń API w dalszych etapach, nasza aplikacja jest gotowa na przyszły rozwój — zawiera wszystkie relacje i tabele niezbędne do wdrożenia kolejnych funkcji, takich jak obsługa większej liczby portali społecznościowych.

3 Backend

3.1 Dokumentacja

Do naszej aplikacji wygenerowaliśmy automatycznie dokumentacje z opisem najważniejszych klas i metod - głównie kontrolerów. Pliki HTML dokumentacji można znaleźć w katalogu "docfx_project/_site".

Poniżej zamieściliśmy przykłady danych wstawionych do naszej bazy danych:

Id	Content	ScheduledAt	MediaUrls	UserId	CreatedAt	UpdatedAt
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
868c93d5-0b3e-44cc-a55f-04498efd2e7d	d	NULL	/uploads/im1dwdp4.jpg	c93c4656-d25b-41ab-8ad0-b4cac575c5b6	2025-06-14 17:13:49.366844	2025-06-14 17:13:49.3€

Rysunek 2: Post utworzony w aplikacji

Id	Status	SocialAccountId	PostId	CreatedAt	UpdatedAt
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1a9d0963-f076-4f3e-9bc0-000f59805eff	1	9082a03a-bce4-4646-8971-6362499fe37f	868c93d5-0b3e-44cc-a55f-04498efd2e7d	2025-06-14 17:14:57.371261	2025-06-14 17:14:57.371261

Rysunek 3: Post po wstawieniu do platform społecznościowych

3.2 Możliwość wdrożenia w chmurze

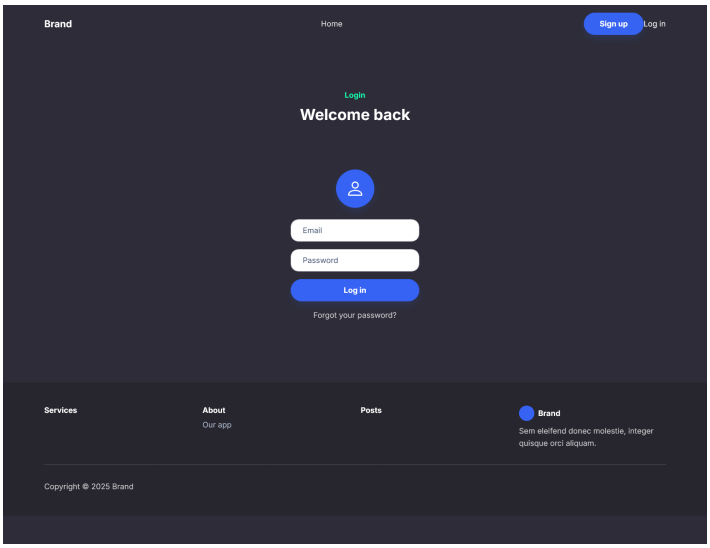
W naszym repozytorium znajdują się pliki Dockerfile oraz docker-compose.yml, które zawierają konfigurację umożliwiającą uruchomienie całej aplikacji w kontenerze za pomocą jednej komendy. Przed uruchomieniem należy jedynie ustawić odpowiednie zmienne środowiskowe zawierające klucze API do Facebook Graph API. Takie rozwiązanie umożliwia łatwe wdrożenie aplikacji w dowolnym środowisku chmurowym obsługującym kontenery.

4 Frontend

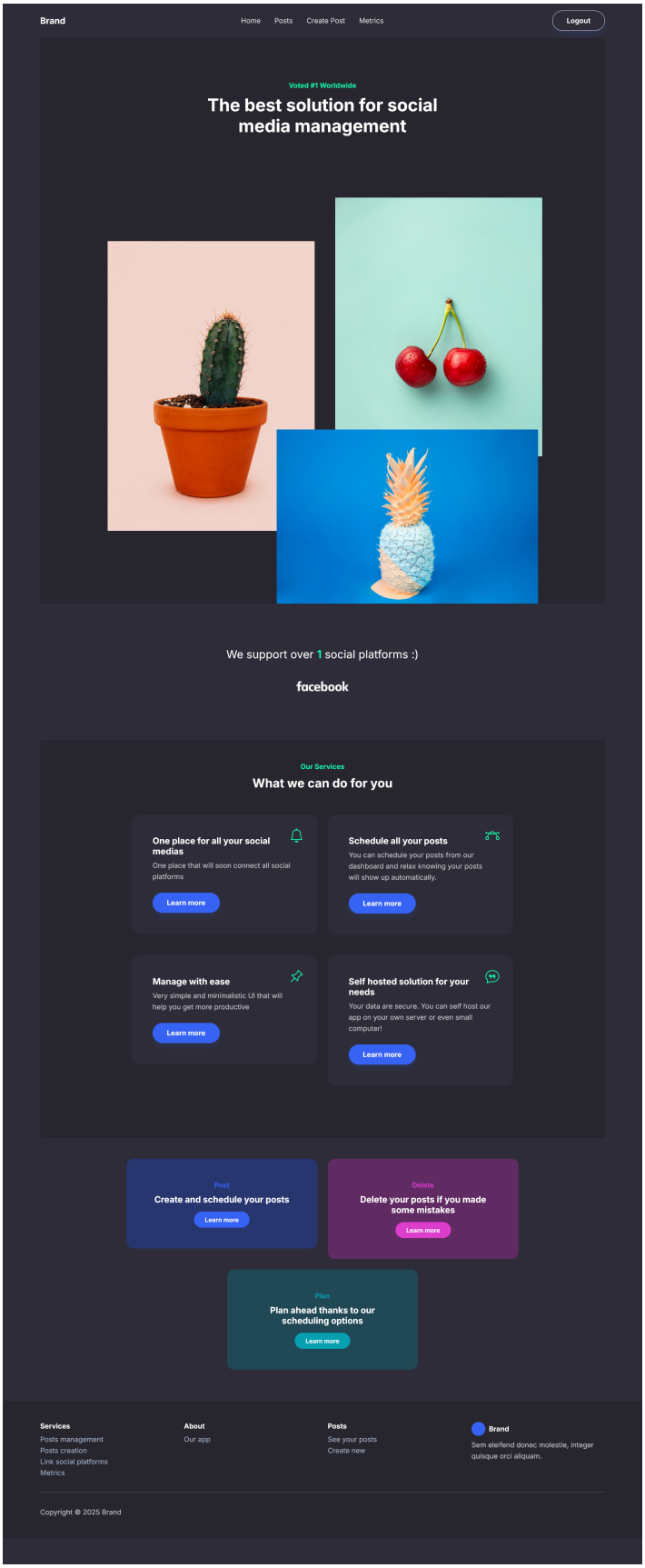
4.1 Użycie Bootstrap - przejrzystość i responsywność

Do stworzenia frontendu naszej aplikacji wykorzystaliśmy Bootstrap – zestaw stylów CSS, który umożliwia szybkie tworzenie nowocześnie wyglądających stron internetowych. Framework ten ułatwia dostosowanie interfejsu do różnych rozmiarów ekranów, co zapewnia responsywność aplikacji webowych.

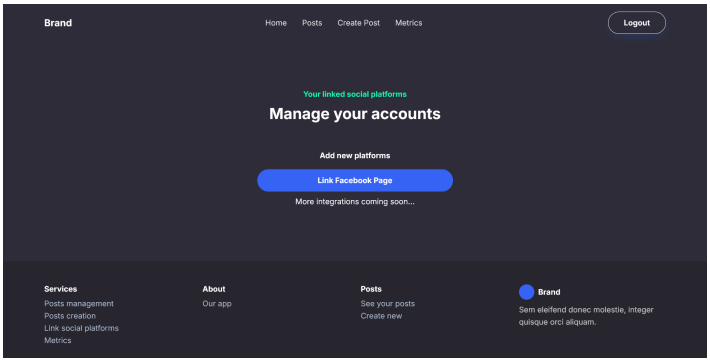
4.2 Prezentacja aplikacji z perspektywy użytkownika



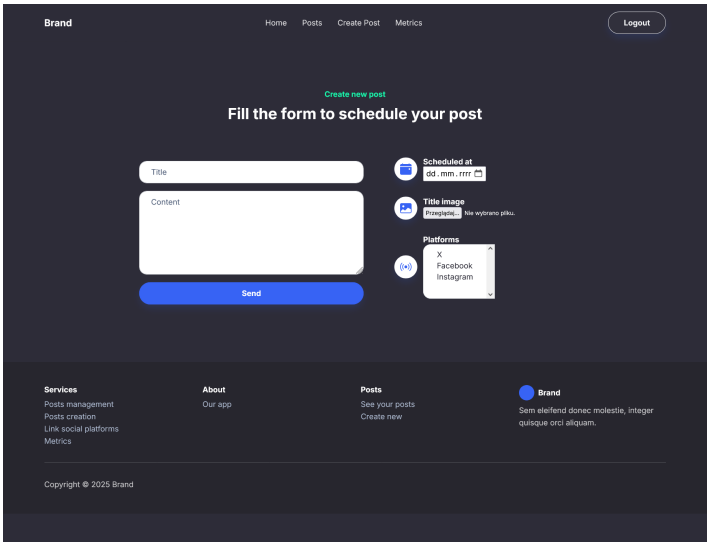
Rysunek 4: Strona logowania



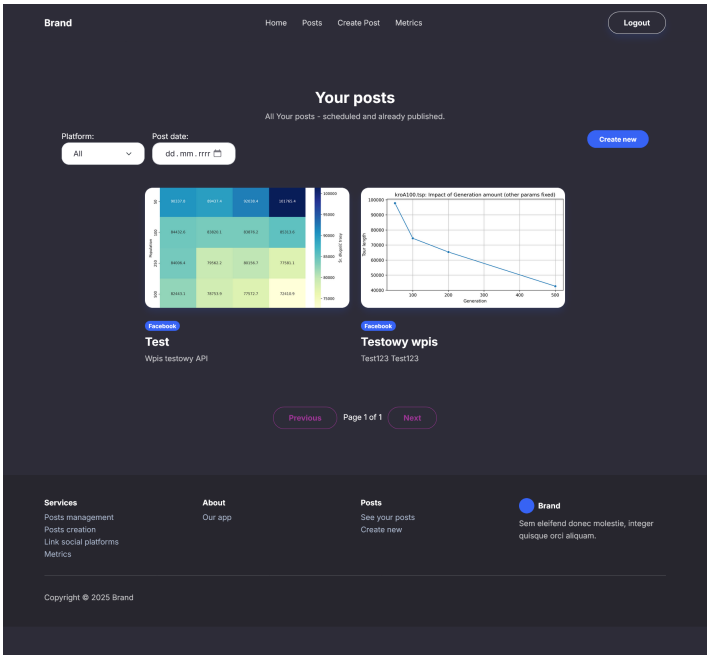
Rysunek 5: Strona główna



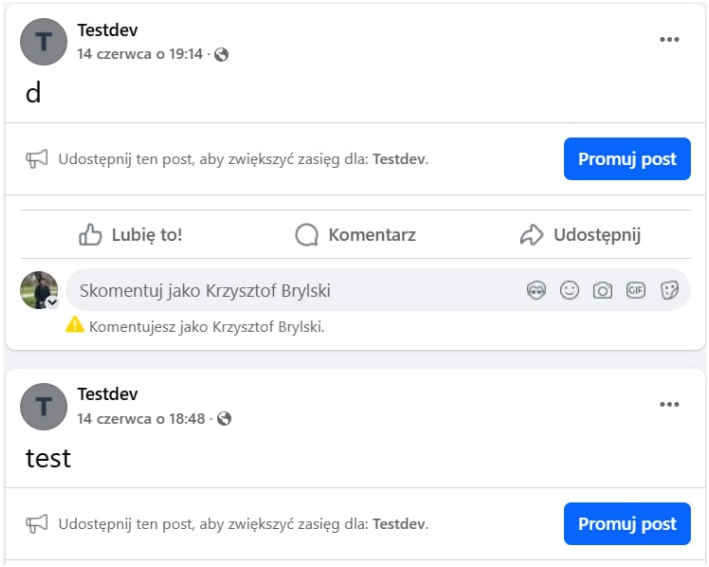
Rysunek 6: Strona łączenia konta z Facebookiem (OAuth)



Rysunek 7: Tworzenie posta



Rysunek 8: Lista postów do wrzucenia na media społecznościowe



Rysunek 9: Posty wstawione automatycznie na Facebooka

5 Problemy napotkane podczas realizacji projektu

Największym problemem okazało się połączenie z zewnętrznymi API platform społecznościowych. Główną trudność stanowiło uzyskanie odpowiednich uprawnień do wykonywania zapytań na poszczególnych portalach.

W przypadku np. Twittera/X integracja się nie powiodła — mimo informacji o dostępności darmowego planu testowego, po utworzeniu konta deweloperskiego konieczne było osobne zgłoszenie w celu uzyskania dostępu do API. Ostatecznie, pomimo złożenia wniosku, nie otrzymaliśmy odpowiedzi ani dostępu do interesujących nas funkcji.

Proces integracji z Facebookiem był prostszy, ponieważ nasze konto deweloperskie zostało automatycznie zatwierdzone po krótkim czasie oczekiwania. Największe trudności napotkaliśmy jednak w ustawieniach aplikacji w panelu deweloperskim. Panel ten wydaje się przestarzały w porównaniu do reszty ekosystemu Facebooka i usług firmy Meta. Ostatecznie udało nam się utworzyć aplikację, korzystając ze starszego sposobu jej rejestracji, i uzyskać dostęp do API umożliwiającego zarządzanie stronami na Facebooku. Dzięki temu w pełni zintegrowaliśmy logowanie do Facebooka z naszą aplikacją przy użyciu protokołu OAuth, a następnie zaimplementowaliśmy funkcję publikowania postów na stronie.

Nasza aplikacja posiada klasę interfejsu umożliwiającą szybkie wdrażanie obsługi kolejnych platform społecznościowych po uzyskaniu dostępu do ich API. Obecna wersja wspiera jedynie podstawową integrację z Facebookiem. Każda nowa platforma powinna dziedziczyć po wspólnym interfejsie i działać jako osobna usługa umożliwiająca dostęp do API danej platformy.