

Błażej Belowski

Wojciech Borzęcki

## 1. Opis Ogólny

Projekt "SpyShark" to aplikacja dostępna dla każdego użytkownika umożliwiająca analizowanie ruchu na sklepach internetowych postawionych na platformie Shopify. Aplikacja back-endowa zawiera wymianę informacji między użytkownikiem a bazą danych. Obsługuje ona endpointy, do których można się odwołać z poziomu front-endu. Projekt jest napisany w technologii ASP.NET z użyciem architektury MVC i Swaggera. Autentykacja endpointów odbywa się przy użyciu Azure AD B2C i protokołu OAuth 2.0. Obsługa błędów jest zaimplementowana.

## 2. Struktura Projektu

Projekt składa się z następujących głównych elementów:

- Controllers:
  - StoreController.cs
- Data:
  - MyDatabaseContext.cs
- Models:
  - MainObject.cs
  - ProductData.cs
  - Store.cs
  - Todo.cs
  - UserRegisterModel.cs
  - UserStoreassociations.cs
- Inne:
  - Program.cs (plik domyślny)

## 3. Technologie

Projekt korzysta z wersji .NET 7, wykorzystuje Entity Framework oraz kontrolery MVC oraz TypeScript + React dla front-endu.

## 4. Uruchamianie Projektu

Aplikacja back-endowa jest hostowana w chmurze Azure. Można dostać się do jej UI poprzez link <https://shopifystatsapi.azurewebsites.net/swagger> . Wymagana jest autoryzacja tokenem który można uzyskać poprzez odpowiedni strzał do API przez Postman lub wyciągnięcie go z sesji zalogowanego użytkownika. Jeśli chcemy uruchomić aplikację lokalnie wymagana jest instalacja .NET 7.

## 5. Konfiguracja

Konfiguracja projektu obejmuje ustawienia konieczne do autentykacji endpointów za pomocą Azure AD B2C oraz konfiguracji bazy danych. Wymagało to utworzenie specjalnego tenanta B2C na platformie Azure oraz zarejestrowanie w tym naszej aplikacji. Następnie ustawienie odpowiednich przekierowań oraz połączyć aplikację back-endową z bazą danych używając client secret. Ze strony front-endu także musimy skonfigurować autoryzację.

## 6. Punkt Końcowy API

Projekt udostępnia następujące punkty końcowe API:

- POST AddStoresUserDataToDb
- GET GetUserStores
- POST RemoveUserStoreAssociation
- GET GetStoreUpdatedAts
- POST UpdatedAtDatabaseLogic
- GET GetTopStores

Powyższe endpointy umożliwiają wykonywanie operacji na bazie danych oraz wykonywanie odpowiedniej logiki do zbierania danych na temat sklepów.

## 6. Modele Danych

Projekt korzysta z następujących modeli danych:

- MainObject reprezentuje główny obiekt danych pobranych z zewnętrznego źródła, zawierającego listę produktów, nazwę sklepu oraz datę ostatniej zmiany na sklepie
- StoreControllerMainRequest reprezentuje domyślny obiekt dla niektórych zapytań do API, parametry podajemy w body zapytania POST a mianowicie Guid użytkownika oraz adres do sklepu
- UpdatedAt reprezentuje obiekt służący do zbierania głównej logiki naszej aplikacji, a mianowicie naliczania ruchu na stronie, zawiera on Id jako klucz, StoreName jako nazwę sklepu i Date ostatniej aktualizacji.
- UserStoreAssociation reprezentuje obiekt który odczytuje z bazy danych powiązania użytkownika z sklepami które obserwuje. Zawiera on Guid użytkownika, nazwę sklepu, flagę IsTracked do określenia czy dany sklep jest sledzony oraz link do sklepu

## 8. Zabezpieczenia

Autentykacja endpointów odbywa się przy użyciu Azure AD B2C i protokołu OAuth 2.0. Każdy strzał do API musi być autoryzowany tokenem generowanym przez Azure podczas logowania lub odpowiednim strzałem do API Microsoftu z użyciem client secret.

## 9. Swagger

Projekt integruje Swaggera, który dostarcza obsługę danych endpointów oraz ich graficzną implementację na czas developmentu lub testów

## 10. Testowanie

Zalecane metody testowania obejmują testy end-to-end z powodu licznych integracji z różnymi systemami czy technologiami.

## 11. Rozszerzalność

Projekt jest zorganizowany w sposób umożliwiający łatwe rozszerzanie funkcjonalności. W każdej chwili można dodać nowy endpoint umożliwiający nowe operacje na bazach według własnego upodobania.

## 12. Błędy i Obsługa Wyjątków

Projekt posiada obsługę błędów, a błędy są odpowiednio logowane.

## 13. Przykłady Użycia

Poniżej przedstawiono kilka przykładowych scenariuszy użycia, ilustrujących funkcjonalność aplikacji.

Dodawanie danych sklepu do bazy danych. POST /AddStoresUserDataToDb

Ciało żądania:

```
json

{
  "UserId": "exampleUserId",
  "shopUrl": "exampleShopUrl"
}
```

## 14. Wdrożenie

Przed wdrożeniem projektu zaleca się przeczytanie instrukcji dotyczących konfiguracji i bezpieczeństwa. Projekt można wdrożyć na dowolny serwer, spełniający wymagania systemowe. Nasz projekt jest hostowany na platformie Azure.

## 15. Monitorowanie i Analiza Danych

Projekt może być monitorowany przy użyciu dostępnych narzędzi monitorowania. Analiza danych może być przeprowadzana w oparciu o dane zgromadzone w bazie danych, lub specjalnych narzędzi na platformie Azure typu Application Insights

## 16. Front-end

### 16.1. Technologie

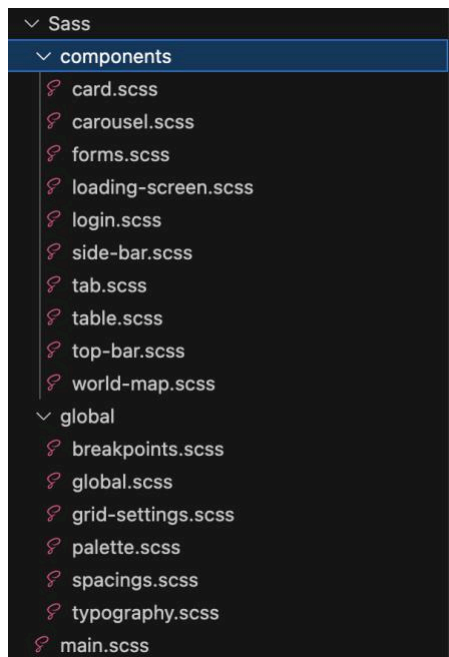
Aplikacja front-end została utworzona na frameworku React wraz z następującymi paczkami:

```
"@azure/msal-browser": "^2.38.0",
"@azure/msal-react": "^1.5.7",
"@fontawesome/fontawesome-free": "^6.4.0",
"@types/chart.js": "^2.9.37",
"@types/react-dom": "^18.2.4",
```

"axios": "^1.4.0",

"react-bootstrap": "^2.8.0",

Layout aplikacji został oparty na 12kolumnowym bootstrapowym gridzie dzięki czemu aplikacja jest w pełni responsywna. Aplikacja korzysta z preprocesora css (scss) w którym struktura stylów jest podzielona hierarchicznie



## 16.2. Struktura projektu

Projekt składa się z pliku app.tsx który jest głównym corem projektu. są w nim renderowane komponenty stron na podstawie nawigacji.

- App.tsx
  - Główny plik w którym zawarte jest sprawdzanie logowania użytkownika oraz cały routing strony
- dashboard.tsx
  - Główny plik który jest wyświetlany po zalogowaniu, są na nim wyświetlane statystyki najlepiej sprzedających sklepów w danym okresie czasu
- sales-tracker.tsx
  - Dodawanie sklepów do obserwowania oraz wyświetlanie obserwowanych sklepów na podstawie danych z endpointu asp.net
- store.tsx
  - Strona na której są wyświetlane informacje o konkretnym sklepie wybranym na stronie sales-tracker. Dynamiczne pobieranie danych o marketingu na platformie tiktok oraz informacje seo/traffic z utworzonych endpointów node.js