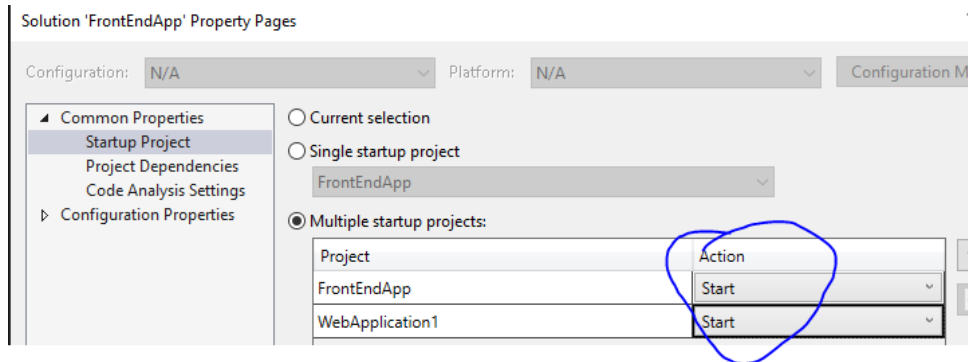


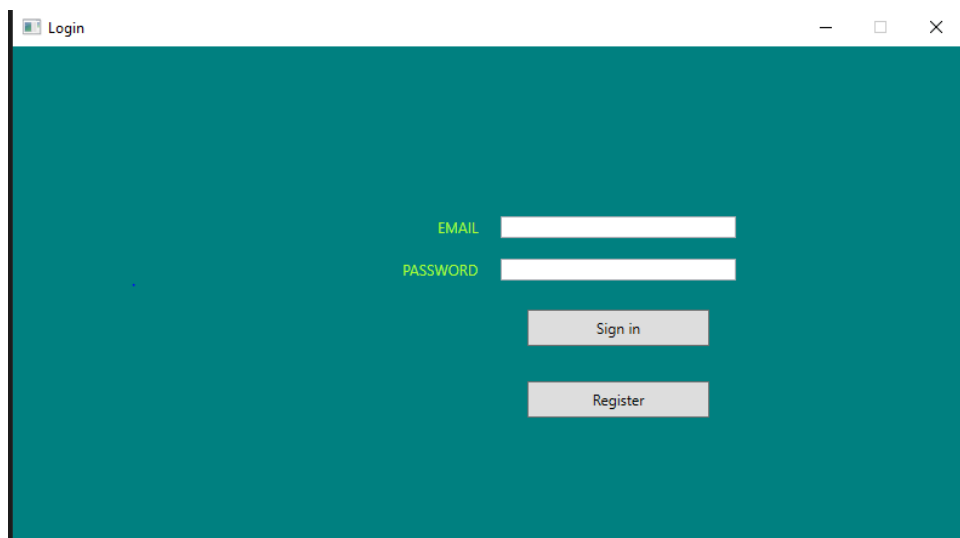
Aplikacja Frontendowa dla Web Api :

Wykorzystująca: WPF, Token JWT, Entity Framework, MS_SQL, Swagger, walidacje.

Pobieramy aplikacje z githuba.



Po uruchomieniu w Visual Studio 2022 Community , wchodzimy w Configure Startup Project , wybieramy Multiple startup projects i w Action zmieniamy na Start w obu aplikacjach. Potem uruchamiamy projekt. Uruchomi się aplikacja frontendowa i backendowa.



Po uruchomieniu, tworzy aplikacja za pomocą entity framework i migracji -> nową bazę danych ProjectWebApiDb i 4 tabele: Products, Prices, Inventories, Users. Dodatkowa tabela Users to dla użytkowników, którzy się zarejestrują w systemie i aby móc później się zalogować i korzystać z aplikacji.

The screenshot shows a web browser window titled "Register". The page has a teal background. In the top left corner, there is a "Return" button. The main form contains the following fields and values:

Field	Value
FIRST NAME	Patryk
LAST NAME	Prusko
EMAIL	patrykprusko@gmail.com
PASSWORD	*****
CONFIRM PASSWORD	*****
NATIONALITY	Polish
DATE OF BIRTH	13-01-1987
ROLE	

Below the form fields are two buttons: "Save" and "Clear".

Klikamy przycisk Register, zatwierdzamy przyciskiem Save.

This screenshot shows the same "Register" form as the previous one, but with a modal dialog box overlaid in the center. The dialog box has a white background and a red close button in the top right corner. It contains the following text:

Added new user
Status code: 200 -> OK

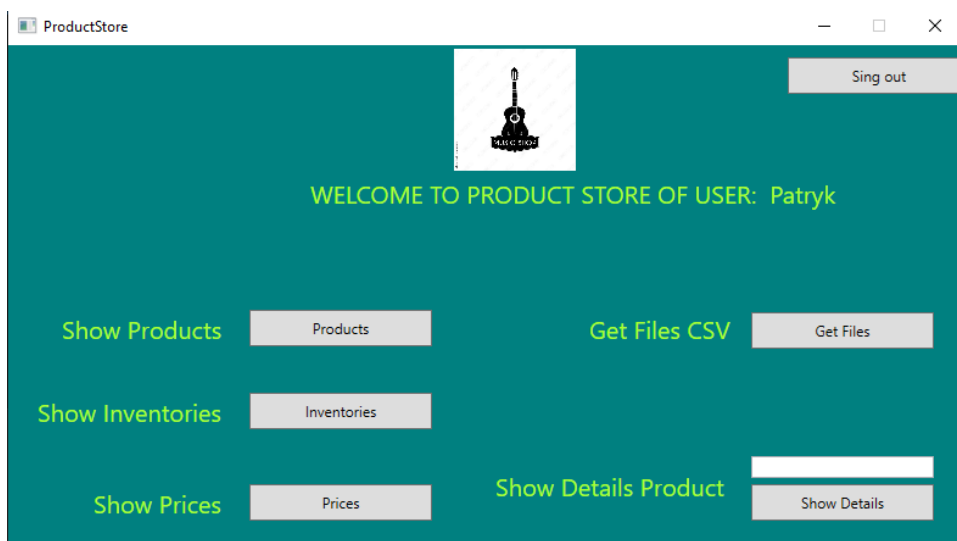
At the bottom of the dialog box is an "OK" button. A blue hand-drawn circle highlights the dialog box. The form fields and buttons are visible behind the dialog box.

Gdy prawidłowo dane się wprowadzi to dodało się nowego usera do bazy danych. Przechodzi automatycznie do okna logowania.

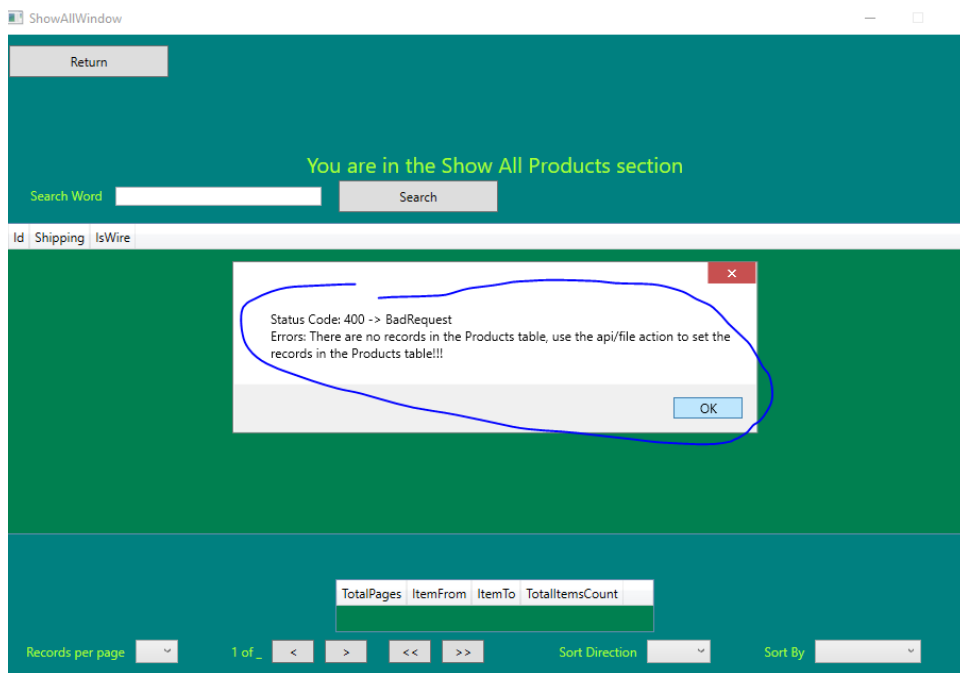
```
SELECT TOP (1000) [Id]
, [Email]
, [FirstName]
, [LastName]
, [DateOfBirth]
, [Nationality]
, [PasswordHash]
FROM [ProjectWebApiDb].[dbo].[Users]
```

Id	Email	FirstName	LastName	DateOfBirth	Nationality	PasswordHash
1	patrykprusko@gmail.com	Patryk	Prusko	1987-01-13 00:00:00.0000000	Polish	AQAAAAIAAYagAAAAElaEJDyuoyt

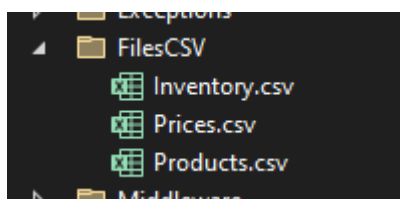
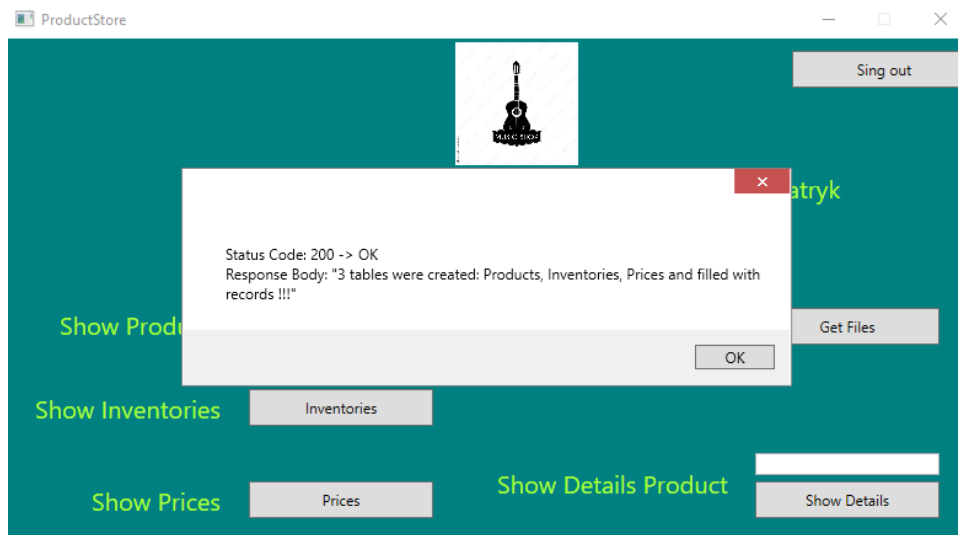
W bazie danych jest dodany nowy user.



Po zalogowaniu się i wygenerowaniu nowego Tokena JWT, można korzystać z przeglądania danych tabel.

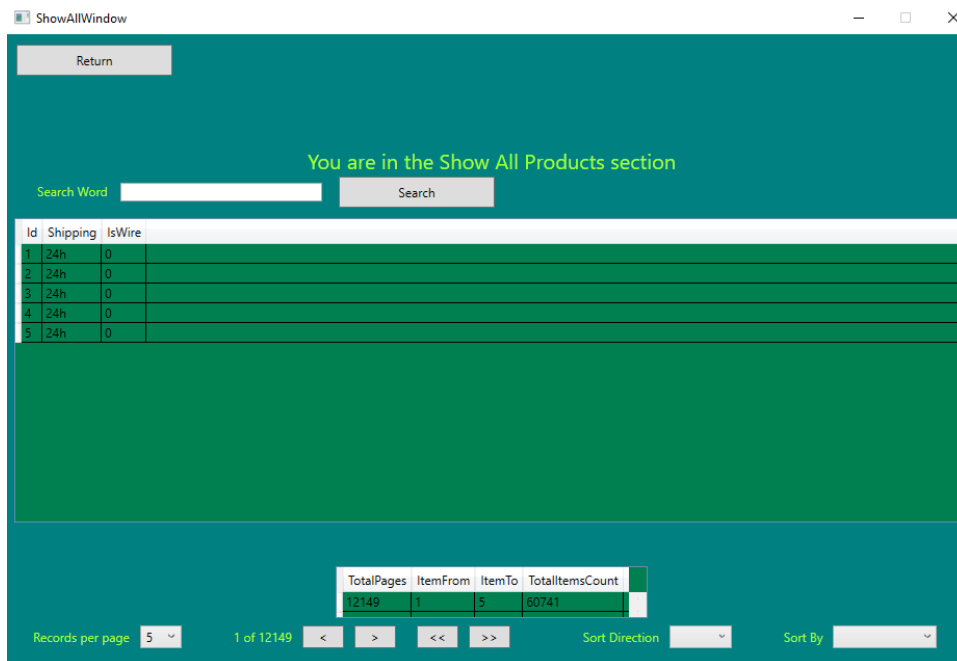


Gdy się sprawdzi czy są jakieś rekordy w bazie danych w Products (przycisk Products), to wyświetli się komunikat, że aby użyć akcji api/file, żeby załadować dane w tabeli, to jest przycisk Get Files, który ładuje wszystkie dane do tabel. Pojawi się odpowiedni komunikat.

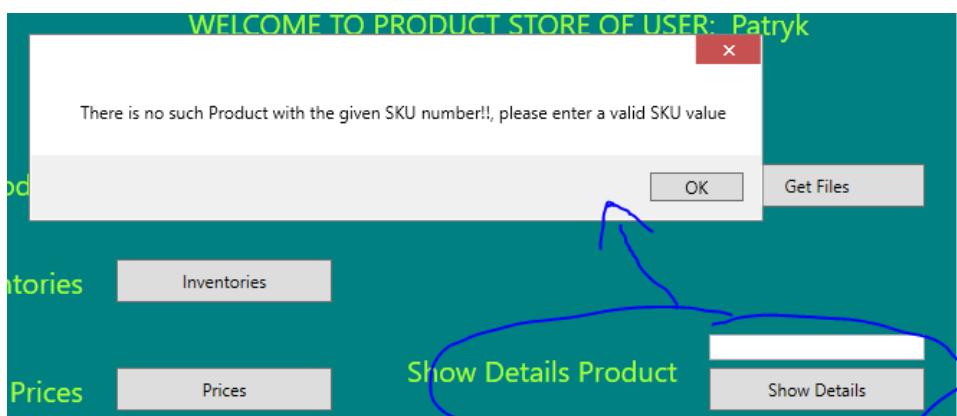


Po kliknięciu przycisku Get Files, gdy nie ma jeszcze żadnych plików CSV i danych w tabelach, to może potrwać kilka sekund, proszę poczekać chwilę. Pojawi się w projekcie w WebApplication1 -> FilesCSV katalog w nim 3 ściągnięte nowe z sieci pliki csv : Products, Prices, Inventory. Baza danych

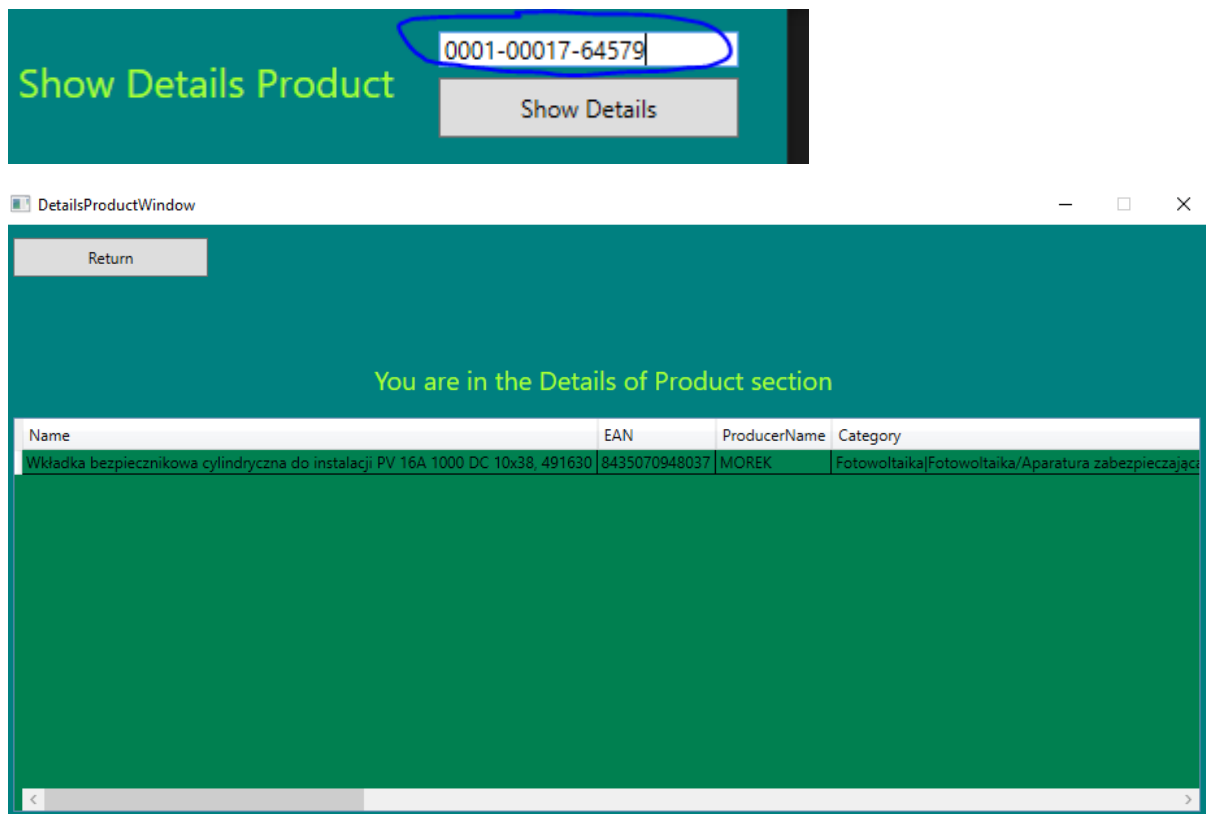
się wypełni nowymi rekordami w 3 tabelach : Products, Prices, Inventories. Potem ponownie jak wciśniemy przycisk get files to już potrwa chwilkę, sprawdzi czy są dane wszystkie i pliki i da informacje zwrotną. Teraz można przeglądać dane.



Wchodząc w Product przycisk pojawia się nowe okno do wyszukania. Można poruszać się po danej stronie, ustawić ilość stron na danej stronie. Można sortować rosnąco lub malejąco względem kolumny Id. Pokazuje wyniki TotalPages -> wszystkie rekordy znalezione w bazie danych w tabeli Products. Można wyszukać po szukanym słowie po kolumnie Id. Tak samo analogicznie działa przycisk Inventories i Prices i korzystają, wszystkie 3 opcje do przeglądania z jednego okna do wyświetlania.



Przycisk Show Details -> szuka danego produktu po numerze SKU podanym przez usera. Jeśli nie prawidłowy to komunikat.



Jeśli prawidłowy to znajduje go i wyświetla dane w tabeli szczegółów w nowym oknie

```
"ConnectionStrings": {  
  "ProductsDbContext": "Server=DESKTOP-89DCRPN;Database=ProjectWebApiDb;Trusted_Connection=True;TrustServerCertificate=True;"  
}  
"Logging": {  
  
}
```

```
private const string uri = @"https://localhost:7113/"; // local connection  
  
4 references  
private static string GetTokenJWT()
```

Trzeba ustawić prawidłowe połączenie z bazą danych w katalogu -> WebApplication1 -> appSettings.Development.json. Potem ustawić prawidłowy adres URL w katalogu FrontEndApp->Utilites-> HelperHttpClient i ustawić zmienna uri.

Jeśli jakiś problem będzie z stworzeniem automatycznym bazy danych, to można użyć w terminalu update-database -> aby uruchomić migracje manualnie (w Tools-> Nuget Package Manager -> Package Manage Console), albo usunąć aktualną migracje i stworzyć nową : add-migration Init i update database.

