Specyfikacja Techniczna

Klasy i metody (Aplikacja ASP.NET):

- **AutoMapperProfiles -** Klasa zawierająca automapowanie z obiektów klasy Contact na obiekty klas DTO służących do komunikacji. Korzysta z biblioteki AutoMapper.
- **ContactController –** Klasa implementująca kontrolery dla endpointu /contacts. Zawiera metody służące do zwrócenia z bazy danych: wszystkich kontaktów, pojedynczego kontaktu, dodania, usunięcia lub edycji kontaktu oraz endpoint testowy sprawdzający połączenie z API za pomocą metody patch.
- **ApplicationDbContext –** Klasa implementująca DbContext z biblioteki EntityFrameworkCore. Posiada metode OnModelCreating wykorzystującą ModelBuilder do stworzenia kontekstu dla bazy danych, tj. utworzenia odpowiednich tabel ze schematów „Entity" i inicjalnego zaludnienia ich z przykładowymi danymi.
- **DTOs –** Klasy służące do zdefiniowania schematu wymiany informacji z API. Inaczej mówiąc w jakiej formie dane z naszej bazy danych będą wysyłane po zmapowaniu z oryginalnego obiektu.
- **Entities –** Klasy będące modelami reprezentującymi tabele w bazie danych.
- **ClaimsPrincipalExtensions** – Klasa implementująca metodę która zwraca role klienta API.
- **ResourceAccessClaimValue –** Klasy reprezentujące role, potrzebne dla zewnętrznego Identity Providera
- **ContactRepository i IContactRepository –** Interfejs I klassa implementująca ten interfejs mająca na celu stworzenie metod do komunikacji z bazą danych, które następnie wykorzystywane są przez kontrolery. Metody takie jak Add, Delete, Edit, GetById, GetAll, SaveChangesAsync
- **Program –** klasa główna z asynchroniczną metodą Main, tworząca serwer dla API.

Biblioteki:

- Automapper – mapowanie object-object
- Keycloak.Authservice Authorization/Authentication – Biblioteka służąca do komunikacji z identity providerem jakim jest keycloak w celach autentykacji i autoryzacji api.
- EntitiyFrameworkCore – Biblioteka będąca lżejszą wersją EntityFramework będąca technologią dostępu do danych, w tym pracy z bazami danych.
- Azure Containers Tools Targets – Służąca do odblokowania Visual Studio Tools dla kontenerów.
- Npgsql EntityFramework – Bibliteka obsługująca komunikację z bazą danych PostgreSQL w ramach frameworku Entity.
- **FRONTEND: Next.js i React**

Kompilacja:

- Aplikacja jest skonteneryzowana I działa w środowisku docker. Aby odpalić aplikacje należy znajdować się w folderze głównym aplikacji i odpalić plik docker-compose komendą docker-compose up --build z CLI. Do odpalenia wymagany oczywiście jest zainstalowany docker. Aplikacja poza środowiskiem dockera będzie miała problemy z komunikacją pomiędzy mikroserwisami, które skonfigurowane są pod wewnętrzną sieć dockera. Szczególnie jeżeli chodzi o bazy danych i identity providera. **Gdyby kontener backend-1 (api) scrashowal się przy pierwszym odpaleniu, wystarczy odpalić go jeszcze raz i cała apka będzie wtedy działała.**