Specyfikacja techniczna aplikacji

Opis projektu

Projekt to aplikacja webowa napisana w języku C# przy użyciu frameworka ASP.NET Core, która umożliwia zarządzanie kontaktami. Użytkownicy mogą przeglądać, dodawać, edytować i usuwać kontakty z bazy danych, a także zarządzać kategoriami i podkategoriami, do których przypisane są kontakty.

Wykorzystane biblioteki

- 1. **Microsoft.AspNetCore.Mvc**: Framework do tworzenia interfejsu API dla aplikacji ASP.NET Core.
- 2. **Microsoft.EntityFrameworkCore:** Biblioteka umożliwiająca mapowanie obiektoworelacyjne (ORM) w celu interakcji z bazą danych.
- 3. **Microsoft.AspNetCore.Authorization:** Biblioteka do obsługi autoryzacji i uwierzytelniania użytkowników.
- 4. WebAppReact.Data: Biblioteka zawierająca kontekst bazy danych i modele encji.
- 5. **WebAppReact.Models:** Biblioteka zawierająca modele danych aplikacji.

Struktura klas i metod

1. KontaktController:

- Klasa kontrolera obsługująca żądania dotyczące kontaktów.
- Metody:
 - ☐ **GetContacts():** Pobiera skrócone informacje o wszystkich kontaktach z bazy danych.
 - DeleteContact(int id): Usuwa kontakt o określonym identyfikatorze z bazy danych.
 - GetContact(int id): Pobiera szczegółowe informacje o kontakcie o określonym identyfikatorze.
 - ☐ AddContact(Kontakt kontakt): Dodaje nowy kontakt do bazy danych.
 - EditContact(int id, Kontakt updatedContact): Edytuje istniejący kontakt w bazie danych.
 - ☐ **IsEmailUnique(string email):** Sprawdza, czy adres e-mail jest unikalny w bazie danych.

2. Kategoria:

 ○ Klasa modelu reprezentująca kategorię kontaktu. ○ Zawiera pola identyfikatora (Id) i nazwy (Nazwa).

3. Podkategoria:

Klasa modelu reprezentująca podkategorię kontaktu.
 Zawiera pola identyfikatora (Id), nazwy (Nazwa) oraz identyfikatora powiązanej kategorii (Kategoriald).

4. ApplicationDbContext:

 Klasa kontekstu bazy danych, która dziedziczy po DbContext. Zawiera właściwości DbSet reprezentujące tabele w bazie danych.

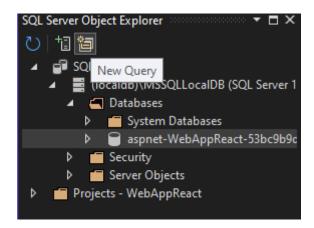
(

Kompilacja aplikacji

Aplikację należy uruchomić w środowisku Visual Studio. Następnie należy otworzyć okno "Konsola menadżera pakietów". I wpisać po kolei:

- 1. enable-migrations
- 2. add-migration "Initial setup"

Z racji, że baza danych będzie pusta to, żeby dodać Kategorie i Podkategorie musimy otworzyć okno "SQL Server Object Explorer"



Znajdujemy naszą bazę i tworzymy nowe Query.

```
INSERT INTO Kategorie (Nazwa)
VALUES ('Służbowy'),
VALUES ('Prywatny'),
VALUES ('Inny');

INSERT INTO Podkategorie (Nazwa, KategoriaId)
VALUES ('Szef'),
VALUES ('Klient');
```

Po tych krokach aplikację można uruchomić w programi Visual Studio