

Dokumentacja techniczna

ZADANIE 1

MICHAŁ WITKOWSKI

Aplikacja składa się z dwóch głównych części: backendu oraz frontendowej aplikacji SPA. Backend został stworzony w języku C# z wykorzystaniem ASP.NET Core do implementacji REST API, natomiast frontend został zbudowany z użyciem frameworka Angular. Dane aplikacji przechowywane są w bazie danych SQL Server.

1. Opis poszczególnych klas i metod

Backend:

Zawiera klasy reprezentujące encje bazy danych o następującej strukturze:

a) Contact

```
public class Contact
{
    //Primary Key
    8 references
    public int Id { get; set; }
    //Properties
    4 references
    public string FirstName { get; set; } = null!;
    4 references
    public string LastName { get; set; } = null!;
    5 references
    public string NickName { get; set; } = null!;
    5 references
    public string PasswordHash { get; set; } = null!;
    3 references
    public DateOnly? BirthDate { get; set; }
    2 references
    public int RoleId { get; set; }
    //Navigation Properties
    7 references
    public virtual List<AboutCategory> AboutCategories { get; set; } = null!;
    1 reference
    public virtual Role Role { get; set; } = null!;
}
```

b) AboutCategory

```
public class AboutCategory
{
    //Primary Key
    public int Id { get; set; }
    //Properties
    public string? PhoneNumber { get; set; }
    public string Email { get; set; } = null!;
    //Navigation Properties
    public virtual Contact Contact { get; set; } = null!;
    public virtual Category Category { get; set; } = null!;
    9 references
    public virtual Subcategory? Subcategory { get; set; }

    //Foreign Keys
    1 reference
    public int CategoryId { get; set; }
    2 references
    public int? SubcategoryId { get; set; }
}
```

c) Category

```
public class Category
{
    //Primary Key
    1 reference
    public int Id { get; set; }
    //Properties
    10 references
    public string Name { get; set; } = null!;
}
```

d) Subcategory

```
public class Subcategory
{
    //Priamry Key
    2 references
    public int Id { get; set; }
    //Properties
    18 references
    public string Name { get; set; } = null!;
}
```

e) Role

```
public class Role
{
    //Primary Key
    public int Id { get; set; }
    //Properties
    3 references
    public string Name { get; set; } = null!;
}
```

Kontekst bazy danych oraz konfiguracja encji zawiera się w klasie ContactsDbContext.

Baza danych jest utworzona oraz uruchomiona w SQL Server 2022.

Dane słownikowe trzymane są w bazie danych.

Wszystkie nowo tworzone oraz aktualizowane obiekty są odpowiednio walidowane.

Kontrolery REST API obsługują zapytania HTTP. Odpowiadają one za komunikację API z klientem.

Logika aplikacji znajduje się wewnątrz serwisów. Do mapowania danych wykorzystywany jest AutoMapper.

Autoryzacja jest realizowana za pomocą przypisanych ról do kontaktów. Do przesyłania tej informacji wykorzystano tokeny JWT.

Frontend:

Aplikacja kliencka wykorzystuje bibliotekę React.js korzystającą z języka JavaScript. Zbudowana ona została z wykorzystaniem komponentów.

2. Wykorzystane biblioteki

Aplikacja kliencka korzysta z następujących pakietów:

- React-redux
- Redux
- React-bootstrap
- Bootstrap
- Jwt-decode
- Axios
- React

Aplikacja serwerowa korzysta z następujących pakietów nuget:

- Microsoft.AspNetCore.Authentication
- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools
- AutoMapper.Extensions.Microsoft.DependencyInjection
- FluentValidation.AspNetCore
- Microsoft.AspNetCore.Identity

3. Sposób kompilacji aplikacji

Backend:

Instalacja .NET Core SDK, otworenie solucji oraz uruchomienie aplikacji za pomocą przycisku F5.

Frontend:

Instalacja wymaganych pakietów „node_modules” za pomocą npm install w utworzonym folderze gdzie znajduje się aplikacja. Następnie należy użyć polecenia „npm run dev”.