

Oskar Kacała

Dokumentacja zadania rekrutacyjnego (BackEnd)

1. Opis architektury

Projekt oparty jest na architekturze warstwowej (Domain → Application → Infrastructure → API), z rozdzieleniem odpowiedzialności na:

- warstwę logiki domenowej (Domain),
- warstwę logiki biznesowej i DTO (Application),
- warstwę dostępu do danych (Infrastructure),
- warstwę interfejsu API (API).

2. Klasy i ich odpowiedzialności

Domain

User, Contact, Category, Subcategory (Entities)

- Reprezentują główne encje w systemie (użytkownik, kontakt, kategoria, podkategoria).
- Zawierają właściwości oraz relacje encji między sobą.

IUserRepository, IContactRepository, ICategoryRepository, ISubcategoryRepository (Interfaces)

- Abstrakcje repozytoriów do operacji CRUD na encjach.

Application

DTOs

- **UserLoginDto, UserRegisterDto, UserDto** – modele wymiany danych przy autoryzacji.
- **CreateContactDto, UpdateContactDto, ContactDto** – modele danych kontaktu.
- **CategoryDto, SubcategoryDto, AuthResultDto** – uproszczone modele odpowiedzi dla kategorii, subkategorii, autoryzacji.

Services

- **AuthService** – obsługuje logowanie i rejestrację użytkownika (BCrypt + JWT).

- **ContactService** – pełna logika zarządzania kontaktami (CRUD).
- **CategoryService, SubcategoryService** – logika pobierania kategorii i podkategorii.

Infrastructure

ApplicationDbContext

- Kontekst bazy danych (Entity Framework Core, PostgreSQL).

Configurations

- Klasy konfiguracyjne typu `IEntityTypeConfiguration<T>` dla EF Core (fluent API).
- `JwtConfiguration` – konfiguracja autoryzacji JWT w `IServiceCollection`.

Repositories

- Implementacje interfejsów repozytoriów (np. `UserRepository`, `ContactRepository`).

Shared

JwtGenerator

- Generuje token JWT z claimami dla zalogowanego użytkownika.

Mappings (AutoMapper)

- **AuthProfile, ContactProfile** – konfiguracje mapowania encji na DTO i odwrotnie.

API

AuthController

- `POST /login`, `POST /register`, `GET /me` – autoryzacja użytkowników (JWT).

ContactsController

- `GET`, `POST`, `PUT`, `DELETE` – operacje na kontaktach (część publiczna i zabezpieczona).

CategoriesController

- `GET`, `GET {id}/subcategories` – publiczny dostęp do kategorii i podkategorii.

Program.cs

- Konfiguracja aplikacji (`DbContext`, repozytoria, serwisy, JWT, CORS, Swagger, AutoMapper, middleware).

3. Wykorzystane biblioteki i ich zastosowanie

Biblioteka / Pakiet	Zastosowanie
Entity Framework Core	ORM – operacje na bazie danych (PostgreSQL).
AutoMapper	Mapowanie encji ↔ DTO.
BCrypt.Net-Next	Bezpieczne hashowanie i weryfikacja haseł.
Microsoft.AspNetCore.Authentication.JwtBearer	Obsługa tokenów JWT w autoryzacji użytkowników.
Microsoft.IdentityModel.Tokens	Walidacja i generowanie tokenów JWT.
Swashbuckle.AspNetCore (Swagger)	Dokumentacja i testowanie API.
ASP.NET Core (MVC, DI, Middleware)	Główna platforma backendowa – kontrolery, routing, DI, auth.
System.Security.Claims	Obsługa informacji o użytkowniku z tokena JWT (userId, email).
Npgsql.EntityFrameworkCore.PostgreSQL	Provider bazy danych PostgreSQL dla EF Core.