

# Tytuł projektu

**HomeBudget**

## Krótki opis działania projektu

Celem projektu "Budżet domowy" jest stworzenie interaktywnej strony internetowej w ASP.NET, która umożliwi użytkownikom zarządzanie ich finansami osobistymi. Strona będzie oferować funkcje logowania, przeglądania historii transakcji oraz zarządzania transakcjami, dodawanie, modyfikowanie, usuwanie.

## Autorzy projektu

Wiktor Luck

Bartosz Kołacz

## Specyfikacja wykorzystanych technologii

- Backend - ASP.NET Core 8.0
- Frontend - Razor Pages, HTML5, CSS3, JavaScript
- Baza danych - Microsoft SQL Server
- Inne - Entity Framework Core, Identity Framework dla autentykacji i autoryzacji, toastr

## Instrukcje pierwszego uruchomienia projektu

1. Skonfiguruj środowisko programistyczne (*Visual Studio*)
2. Pobierz repozytorium projektu z GitHub
3. Za pomocą NuGet Package Manager zainstaluj:
  - a. `AspNetCore.Identity.UI`; `EntityFrameworkCore.SqlServer`; `EntityFrameworkCore.Tools`; `Web.CodeGeneration.Design` – dla *HomeBudget.MVC*, *HomeBudget.Infrastructure*
  - b. `Extensions.Identity.Stores` – dla *HomeBudget.Infrastructure*
  - c. `AutoMapper.Extension.Microsoft.DependencyInjection`; `FluentValidation.AspNetCore`; `MetiaTR.Extensions.Microsoft.DependencyInjection`; `AspNetCore.Identity.UI`; `AspNetCore.Identity.EntityFrameworkCore`; `Extensions.DependencyInjection.Abstractions` – dla *HomeBudget.Application*
4. Wykonaj migracje bazy danych przy użyciu Entity Framework
5. Uruchom projekt w środowisku programistycznym

## Opis struktury projektu

Projekt oparty jest o architekturę trójwarstwową:

- Warstwa prezentacji - front-end napisany z wykorzystaniem Razor
- Warstwa logiki biznesowej - C# i .NET 8 zarządzanie logiką aplikacji
- Warstwa dostępu do danych - Entity Framework do zarządzania relacjami i operacjami na bazie danych

## Wylistowane wszystkie modele

- CurrentUser
- UserContext
- Transaction

## Wylistowane kontrolery wraz z metodami

### HomeController

- **Index()** - asynchronicznie pobiera wszystkie transakcje i wyświetla je
- **NoAccess()** - wyświetla widok informujący o braku dostępu
- **Privacy()** - wyświetla widok polityki prywatności
- **Budget()** - wyświetla widok budżetu (wymagana rola "Parent")
- **Reports()** - wyświetla widok raportów (wymagana rola "Parent")
- **Family()** - wyświetla widok zarządzania rodziną (wymagana rola "Parent")
- **Transactions()** - wyświetla widok transakcji
- **Error()** - wyświetla widok błędu z identyfikatorem żądania

### TransactionController

- **Index()** - asynchronicznie pobiera wszystkie transakcje i wyświetla je (wymagana rola "Parent")
- **Details(string encodedName)** - asynchronicznie pobiera szczegóły transakcji według zakodowanej nazwy
- **Create()** - wyświetla formularz tworzenia nowej transakcji
- **Create(CreateTransactionCommand command)** - asynchronicznie tworzy nową transakcję po weryfikacji modelu
- **Edit(string encodedName)** - asynchronicznie pobiera transakcję do edycji według zakodowanej nazwy, sprawdza dostępność i wyświetla formularz edycji
- **Edit(string encodedName, EditTransactionCommand command)** - asynchronicznie zapisuje zmiany w transakcji po weryfikacji modelu
- **Delete(string encodedName)** - asynchronicznie pobiera transakcję do usunięcia według zakodowanej nazwy, sprawdza dostępność i wyświetla formularz potwierdzenia usunięcia (wymagana rola "Parent")

- **Delete(string encodedName, DeleteTransactionCommand command)** - asynchronicznie usuwa transakcję po weryfikacji modelu (wymagana rola "Parent")

## Opis systemu użytkowników

System użytkowników korzysta z ASP.NET Identity Framework do zarządzania autentykacją i autoryzacją użytkowników. Funkcje obejmują hashowanie haseł, resetowanie haseł oraz zarządzanie sesjami użytkowników.

## Krótką charakterystyka najciekawszych funkcjonalności

Logowanie i rejestracja użytkownika - umożliwia tworzenie konta oraz logowanie się za pomocą nazwy użytkownika i hasła

Przeglądanie transakcji - użytkownicy mogą przeglądać swoje dotychczasowe transakcje, filtrować je według daty oraz kategorii

Dokonywanie płatności - integracja z zewnętrznymi systemami płatności do zarządzania płatnościami online

Zarządzanie transakcjami – użytkownicy mogą edytować, usuwać swoje transakcje, jeśli użytkownik posiada rolę *Parent*, może edytować wszystkie transakcje

## Schemat bazy danych

