

Dokumentacja techniczna aplikacji "Wypożyczalnia Samochodów"

Tytuł projektu

Wypożyczalnia Samochodów

Krótki opis działania projektu

Aplikacja "Wypożyczalnia Samochodów" umożliwia użytkownikom wynajem samochodów na określony czas. Zapewnia funkcjonalności takie jak rejestracja i logowanie użytkowników, przeglądanie listy dostępnych samochodów, filtrowanie i sortowanie pojazdów, wypożyczanie oraz zarządzanie historią wypożyczeń. Administratorzy mają dostęp do zarządzania użytkownikami i pojazdami.

Autorzy projektu

- Mateusz Ratajczak

Specyfikacja wykorzystanych technologii

- .NET 8.0
- Entity Framework Core 8.0.11
- Microsoft SQL Server (Express) jako baza danych
- Blazor Server jako framework frontendowy
- Microsoft.AspNetCore.Authentication.JwtBearer - wersja 8.0.11
- Microsoft.AspNetCore.Components.Authorization - wersja 8.0.11
- Microsoft.EntityFrameworkCore - wersja 8.0.11
- Microsoft.EntityFrameworkCore.SqlServer - wersja 8.0.11
- Microsoft.EntityFrameworkCore.Tools - wersja 8.0.11 (z ustawionym elementem `<PrivateAssets>all</PrivateAssets>`, co oznacza, że pakiet nie jest ujawniany jako zależność w projektach, które używają tej biblioteki).
- Model MVC (elementarnie)

Instrukcje pierwszego uruchomienia projektu

1. Upewnij się, że masz zainstalowane .NET 8 SDK oraz SQL Server.
2. Skonfiguruj bazę danych:
 - a. Dodaj bazę danych do swojego serwera wraz z danymi.
 - b. Ustaw odpowiedni ConnectionString w appsettings.json
3. Uruchom aplikację

Opis struktury projektu

Blazor:

Blazor Web App w .NET

Blazor to framework od Microsoftu wchodzący w skład platformy .NET, który umożliwia tworzenie interaktywnych aplikacji webowych z wykorzystaniem języka C# oraz Razor (HTML z kodem serwerowym). Dzięki Blazorowi, programiści mogą korzystać z pełnych możliwości platformy .NET w przeglądarce, eliminując konieczność użycia JavaScriptu w dużej części aplikacji.

Główne cechy Blazora

1. **Wykorzystanie C# zamiast JavaScriptu:** Kod frontendu pisany jest w C#, co pozwala na pełną integrację z backendem .NET oraz współdzielenie logiki między warstwami.
2. **Komponentowa architektura:** Aplikacje w Blazorze składają się z komponentów, które są samodzielnymi jednostkami zawierającymi zarówno logikę, jak i widok (HTML + C#).
3. **Model Razor:** Blazor wykorzystuje Razor, który umożliwia mieszanie kodu C# z HTML, co znacząco przyspiesza tworzenie interfejsu użytkownika.
4. **Cross-platformowość:** Aplikacje Blazora można uruchamiać na różnych platformach i przeglądarkach.

Projekt składa się z następujących elementów:

- **Models:** Klasy reprezentujące dane (np. Samochod, Uzytkownik, Wypozyczenie).
- **Pages:** Widoki aplikacji w technologii Blazor.
- **AppDbContext:** Klasa kontekstowa do komunikacji z bazą danych.

Modele

1. Samochod

Opis: Reprezentuje dane samochodu w systemie.

- **Id** (int) - Klucz główny.
- **Marka** (string) - Marka samochodu. [Maks. 50 znaków]
- **Model** (string) - Model samochodu. [Maks. 50 znaków]
- **RokProdukcji** (int) - Rok produkcji. [Wartość od 1900]
- **CenaZaDzien** (decimal) - Cena wynajmu za dzień. [Wartość > 0]

- **Dostepnosc** (bool) - Status dostępności pojazdu.

2. Uzytkownik

Opis: Reprezentuje użytkownika systemu.

- **Id** (int) - Klucz główny.
- **Imie, Nazwisko** (string) - Imię i nazwisko użytkownika. [Maks. 50 znaków]
- **Email** (string) - Adres e-mail. [Wymagane, maks. 100 znaków]
- **Haslo** (string) - Hasło użytkownika. [Wymagane]
- **Pesel** (string) - PESEL użytkownika. [Długość: 11 znaków]
- **Plec** (string) - Płeć użytkownika (np. "Mężczyzna").
- **DataUrodzenia** (DateTime) - Data urodzenia.
- **Rola** (string) - Rola użytkownika (np. "Administrator", "Uzytkownik"). [Domyślna: "Uzytkownik"]

3. Wypozyczenie

Opis: Reprezentuje historię wypożyczeń samochodów.

- **Id** (int) - Klucz główny.
- **IdUzytkownika** (int) - Powiązany użytkownik (FK).
- **IdSamochodu** (int) - Powiązany samochód (FK).
- **DataWypozyczenia** (DateTime) - Data rozpoczęcia wynajmu.
- **DataZwrotu** (DateTime?) - Data zakończenia wynajmu.
- **Koszt** (decimal) - Całkowity koszt wypożyczenia. [> 0]

Kontrolery i metody

Projekt nie posiada Kontrolerów. Logika jest w odpowiednich stronach.

Pages

Znajdują się w folderze Components.

Zaczynając od layout

MainLayout i NavMenu. Są to paski górne i boczne.

Następnie Strony odpowiadające za konto w folderze "Konto"

Error – strona do której zostaniemy przeniesieni przy błędzie logowania.

Login – Formularz Logowania Użytkownika

Logout – Wylogowanie Użytkownika

Register – Formularz rejestracji dla użytkownika.

Następnie Reszta Stron: Error, Home, ListaSamochod,
MojeWypozyczenia,Users,WypożyczanieSamochodu.

Error – odpowiada za komunikat błędu gdy coś złego stanie się na stronie (taki ogólny komunikat)

Home – Jest to strona główna

ListaSamochod - Wyświetla listę wszystkich samochodów w wypożyczalni z funkcjami filtrowania i sortowania.

MojeWypozyczenia- Wypozyczenia dla danego użytkownika

Users – Panel użytkowników Widoczny tylko dla administratora

WypożyczanieSamochodu – Zawiera formularz wypożyczenia Samochodu

Opis systemu użytkowników

- **Role:**
 - Administrator: Dostęp do zarządzania użytkownikami i samochodami.
 - Użytkownik: Dostęp do wynajmu samochodów.
 - Gość: Może przeglądać listę samochodów.
- **Rola domyślna:** "Użytkownik".
- **Zarządzanie rolami:** Rola może być przypisana podczas tworzenia użytkownika w bazie danych.

Najciekawsze funkcjonalności

1. **Filtrowanie i sortowanie samochodów:** Możliwość wyszukiwania pojazdów według marki, rocznika lub ceny za dzień.
2. **Panel administracyjny:** Intuicyjne zarządzanie użytkownikami i zasobami systemu.
3. **Dynamiczne obliczanie kosztów wynajmu:** Koszt wynajmu jest automatycznie obliczany na podstawie liczby dni i ceny za dzień.

Kilka Screenów z aplikacji z perspektywy gościa (jak widać nie widać niektórych paneli dostępne dla użytkowników i administratora)



