**1. Opis projektu**

Aplikacja internetowa do zarządzania wypożyczalnią samochodów. Umożliwia użytkownikom dokonywanie rezerwacji pojazdów oraz administrowanie rezerwacjami i danymi systemu. Projekt opiera się na technologii ASP.NET Core z wykorzystaniem Entity Framework Core oraz Identity do zarządzania użytkownikami.

**Dokumentacja**

**1. Wymagania systemowe**

* **System operacyjny:** Windows 10/11, Linux, macOS
* **Środowisko uruchomieniowe:** .NET 7.0 SDK
* **Baza danych:** SQL Server (np. lokalny LocalDB lub SQL Server Express)
* **Przeglądarka internetowa:** Chrome, Firefox, Edge
* **Biblioteki i zależności:**
  + Microsoft.AspNetCore.Identity.EntityFrameworkCore
  + Microsoft.EntityFrameworkCore
  + Microsoft.EntityFrameworkCore.SqlServer

**2. Instalacja**

1. **Klonowanie repozytorium:**

git clone https://github.com/TwojeRepozytorium/CarReservation.git

cd CarReservation  
  
**Przygotowanie środowiska:**

* Zainstaluj .NET SDK 7.0 lub nowszy.
* Upewnij się, że SQL Server jest uruchomiony.

**3. Konfiguracja**

* **Testowi użytkownicy i hasła:**

| **Rola** | **E-mail** | **Hasło** |
| --- | --- | --- |
| Administrator | admin@demo.pl | Admin123! |
| Użytkownik | user@demo.pl | User123! |

* **Rola administratora:**
  + Zarządzanie samochodami.
  + Akceptowanie lub odrzucanie rezerwacji.
  + Przegląd wszystkich rezerwacji.
* **Rola użytkownika:**
  + Przegląd dostępnych samochodów.
  + Tworzenie rezerwacji na swoje konto.

**4. Opis działania aplikacji z punktu widzenia użytkownika**

1. **Logowanie i rejestracja:**
   * Użytkownik rejestruje się, podając imię, nazwisko, e-mail oraz hasło.
   * Po zalogowaniu użytkownik może przeglądać dostępne samochody i dokonywać rezerwacji.
2. **Rezerwacja samochodu:**
   * Użytkownik wybiera samochód z listy dostępnych pojazdów.
   * Wypełnia daty rezerwacji (początek i koniec).
   * Kliknięcie przycisku **Zarezerwuj** tworzy rezerwację z przypisanym statusem "Pending".
3. **Zarządzanie rezerwacjami:**
   * Administrator widzi wszystkie rezerwacje z możliwością ich akceptacji lub odrzucenia.
   * Rezerwacje są oznaczane statusami:
     + **Pending** (oczekujące)
     + **Approved** (zaakceptowane)
     + **Rejected** (odrzucone)
4. **Zarządzanie samochodami:**
   * Administrator może dodawać, edytować i usuwać pojazdy w systemie.

**5. Najważniejsze pliki i funkcjonalności**

* **Kontrolery:**
  + AccountController: obsługuje rejestrację, logowanie i wylogowanie użytkowników.
  + ReservationsController: zarządza rezerwacjami.
  + CarsController: zarządza samochodami.
* **Modele:**
  + ApplicationUser: przechowuje dane użytkownika (rozszerzony IdentityUser).
  + Car: opisuje pojazdy dostępne do wypożyczenia.
  + Reservation: zarządza danymi dotyczącymi rezerwacji.
* **Widoki:**
  + Views/Reservations/Index.cshtml: lista rezerwacji.
  + Views/Reservations/Create.cshtml: formularz tworzenia rezerwacji.
  + Views/Cars/Index.cshtml: lista samochodów.
  + Views/Account/Login.cshtml: formularz logowania.