Aplikacja webowa - Car4Rent

Dokumentacja techniczna

Autor: Jakub Kurek

Nr albumu 120137

**Spis treści**

[**1.** **Opis projektu:** 3](#_Toc125558497)

[**2.** **Wykorzystane technologie w projekcie:** 3](#_Toc125558498)

[**3.** **Wymagania techniczne:** 3](#_Toc125558499)

[**4.** **Implementacja:** 4](#_Toc125558500)

[**5.** **Instalacja:** 5](#_Toc125558501)

[**6.** **Użytkowanie:** 5](#_Toc125558502)

[**7.** **Uwagi dotyczące bezpieczeństwa:** 5](#_Toc125558503)

[**8.** **Funkcjonalności programu:** 5](#_Toc125558504)

[**9.** **Serwis w tle** 6](#_Toc125558505)

[**10.** **Architektura aplikacji:** 6](#_Toc125558506)

[**11.** **Planowany rozwój** 7](#_Toc125558507)

# **Opis projektu:**

Projekt "**Car4Rent**" jest aplikacją webową do wypożyczania samochodów, napisanym w języku C# i wykorzystującym frameworki .Net i Auto mapper. Aplikacja łączy się z bazą danych, pobiera z niej dane i wstawia do niej dane. Projekt posiada możliwość rejestracji kont użytkowników, logowania i wylogowania. Aplikacja posiada podstrony dostępne publicznie oraz podstrony tylko dla zalogowanych użytkowników. Projekt uwzględnia indywidualną interakcję użytkownika, taką jak przesłanie formularza, z którego dane są przechowywane w bazie i przypisane do odpowiedniego użytkownika.

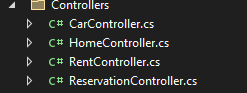
# **Wykorzystane technologie w projekcie:**

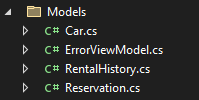
* **Microsoft.AspNetCore.Diagnostics.EntityFreamworkCore** - biblioteka umożliwiająca łatwe diagnozowanie i rozwiązywanie problemów związanych z bazą danych w projekcie opartym na ASP.NET Core i Entity Framework Core.
* **Microsoft.AspNetCore.Identity.EntityFreamworkCore** - biblioteka umożliwiająca implementację systemu zarządzania użytkownikami i uprawnieniami w projekcie opartym na ASP.NET Core i Entity Framework Core.
* **Microsoft.AspNetCore.Identity.UI** - biblioteka zawierająca gotowe komponenty interfejsu użytkownika do obsługi systemu zarządzania użytkownikami i uprawnieniami w projekcie opartym na ASP.NET Core.
* **Microsoft.VisualStudio.Web.CodeGeneration.Design** - narzędzie umożliwiające generowanie kodu w projekcie opartym na ASP.NET Core, na podstawie szablonów i modelu danych.
* **Microsoft.EntitFrameworkCore.Tools** - narzędzie umożliwiające wykonywanie różnych operacji na bazie danych, takich jak migracje czy tworzenie tabel na podstawie klas, w projekcie opartym na Entity Framework Core.
* **Microsoft.EntitFrameworkCore.SqlServer** - biblioteka umożliwiająca połączenie projektu opartego na Entity Framework Core z bazą danych MS SQL Server.
* **AutoMapper** to biblioteka umożliwiająca automatyczne mapowanie obiektów między różnymi typami danych, ułatwiając przetwarzanie i przesyłanie danych między warstwami aplikacji. Prosty w użyciu i możliwy do skonfigurowania za pomocą atrybutów lub metod konfiguracyjnych, dostępny zarówno z frameworkiem .NET jak i innymi platformami.

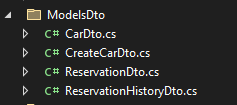
# **Wymagania techniczne:**

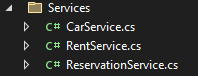
* .Net 6.0 lub nowszy
* Automapper 12.0.0 lub nowszy
* Baza danych MySQL lub Microsoft SQL Server
* Web server obsługujący aplikacje ASP.NET

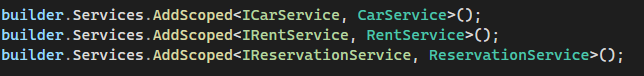
# **Implementacja:**



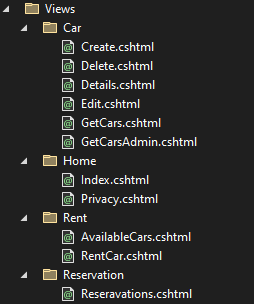


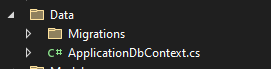






Brak opisu.





# **Instalacja:**

* Skonfiguruj bazę danych zgodnie z ustawieniami w pliku appsettings.json
* Uruchom aplikację za pomocą Visual Studio lub przez publikację na Azure obsługującym aplikacje ASP.NET

# **Użytkowanie:**

* Aby się zarejestrować, użytkownik musi podać swój adres e-mail oraz hasło dwukrotnie.
* Aby się zalogować, użytkownik musi podać swój adres e-mail i hasło.
* Po zalogowaniu, użytkownik ma dostęp do swojego profilu oraz do wypożyczenia samochodów.
* Użytkownik może wypełnić formularz z danymi dotyczącymi wypożyczenia samochodu, które zostaną przechowywane w bazie danych i przypisane do tego użytkownika.
* Aby się wylogować, użytkownik musi kliknąć przycisk "wyloguj" na swoim profilu.

# **Uwagi dotyczące bezpieczeństwa:**

* Hasła użytkowników są hashowane przed zapisaniem do bazy danych.
* Wymagane jest uwierzytelnianie przy wprowadzaniu danych do formularza wypożyczenia samochodu.
* Dostęp do podstron dostępnych tylko dla zalogowanych użytkowników jest chroniony przez mechanizmy autoryzacji. (Podstrona z wypożyczeniem auta)

# **Funkcjonalności programu:**

**System logowania:**

* Rejestracja
* Logowanie
* Przypisywanie roli

**Profil użytkownika:**

* Zmiana Hasła
* Sprawdzenie swoich rezerwacji
* Usuwanie oraz pobieranie danych użytkownika

**System rezerwacji:**

* Wybór daty oraz miasta
* Wyświetlenie dostępnych samochodów w tym terminie
* Rezerwacja auta
* Potwierdzenie rezerwacji
* Dodanie encji do bazy danych

**Zarządzanie samochodami:**

Admin:

* Wyświetlanie całej tabeli samochodów
* Dodawanie samochodów do tabeli
* Usuwanie samochodów do tabeli
* Edycja samochodów w tabeli
* Sprawdzenie szczegółów o danym samochodzie
  + **Zarządzanie plikami:**
  + Dodawanie zdjęcia oraz usuwanie zdjęcia

User/Osoba nie zalogowana

* Wyświetlanie tabeli samochodów które są w bazie
* Sprawdzenie szczegółów samochodów

**Seeder:**

* Wypełnianie pustej bazy danych przy uruchamianiu programu

# **Serwis w tle**

* Sprawdza on bazę danych w poszukiwaniu rezerwacji, które kończą się w danym dniu.
* Jeśli znajdzie takie rezerwacje, wysyła powiadomienie e-mailem do odpowiednich użytkowników informując ich o końcu rezerwacji.
* Następnie usuwa encje z bazy danych, które dotyczą rezerwacji, które kończą się w danym dniu.
* Serwis będzie działał w tle i będzie cyklicznie sprawdzał bazę danych w celu wykonania tej czynności.

# **Architektura aplikacji:**

* Projekt składa się z kilku warstw: warstwa prezentacji, warstwa logiki biznesowej i warstwa dostępu do danych oraz Automappera
* Warstwa prezentacji odpowiada za interfejs użytkownika oraz przetwarzanie danych przesłanych z formularza.
* Warstwa logiki biznesowej odpowiada za reguły dotyczące wypożyczania samochodów, takie jak sprawdzanie dostępności samochodów, ceny itp.
* Warstwa dostępu do danych odpowiada za komunikację z bazą danych, taką jak pobieranie i zapisywanie danych dotyczących wypożyczeń, użytkowników itp.
* Automapper jest używany do mapowania danych między warstwami oraz między modelem danych a widokami.

# **Planowany rozwój**

* Wdrożenie UI / Frontend: Aplikacja będzie miała nowy interfejs użytkownika, który będzie łatwiejszy w użyciu i bardziej przyjazny dla użytkownika.
* Zwiększenie bezpieczeństwa kodu: Kod aplikacji będzie przeglądany i ulepszany, aby zwiększyć jego bezpieczeństwo przed ewentualnymi atakami.
* Potwierdzenie rejestracji przez e-mail: Po rejestracji użytkownik otrzyma e-mail z potwierdzeniem, że jego konto zostało utworzone.
* Potwierdzenie rezerwacji przez e-mail: Po dokonaniu rezerwacji użytkownik otrzyma e-mail potwierdzający rezerwację.
* Metody płatności: Będzie możliwość dokonywania płatności za rezerwacje przez różne metody, takie jak karty kredytowe, PayPal lub przelewy bankowe.
* Logowanie przez zewnętrzne serwisy: Użytkownicy będą mogli zalogować się do aplikacji za pomocą swoich kont z różnych serwisów, takich jak Google, Facebook lub Twitter.
* Dodatkowe funkcje: Będą dodawane nowe funkcje, które poprawią funkcjonalność aplikacji i ułatwią korzystanie z niej.
* Wdrożenie do portfolio: Po zakończeniu prac nad aplikacją, będzie ona wdrożona do mojego portfolio w celu udowodnienia umiejętności oraz przyciągnięcia potencjalnych pracodawców.
* Pomoc w przyszłej karierze: Dzięki rozwojowi aplikacji, będę mógł posiłkować się nią w swojej przyszłej karierze, jako dowód na swoje umiejętności i doświadczenie w tworzeniu aplikacji.

Załączam również prototyp UI dla projektu

Graphical user interface, application

Description automatically generated A picture containing text, screenshot, road

Description automatically generated