

Aplikacja webowa - **Car4Rent**

Dokumentacja techniczna

Autor: Jakub Kurek

Nr albumu 120137

Spis treści

| | |
|--|---|
| 1. Opis projektu: | 3 |
| 2. Wymagania techniczne: | 3 |
| 3. Wykorzystane technologie w projekcie: | 3 |
| 4. Implementacja: | 4 |
| 5. Instalacja: | 5 |
| 6. Użytkowanie: | 5 |
| 7. Uwagi dotyczące bezpieczeństwa: | 5 |
| 8. Funkcjonalności programu: | 5 |
| 9. Serwis w tle | 6 |
| 10. Architektura aplikacji: | 7 |
| 11. Planowany rozwój | 7 |

1. 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.

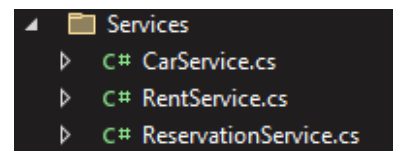
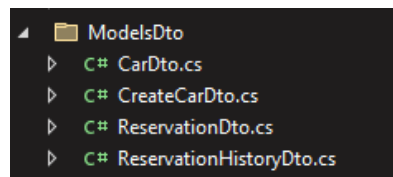
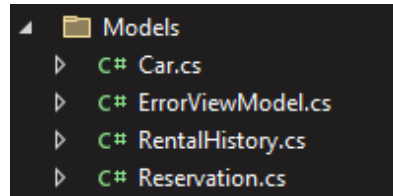
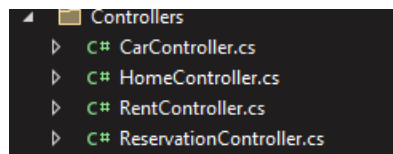
2. Wymagania techniczne:

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

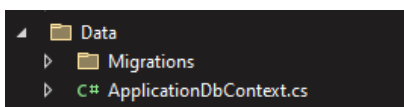
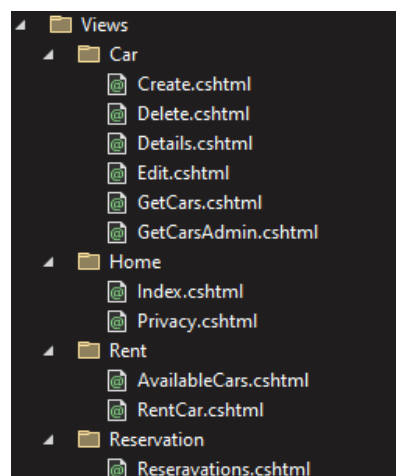
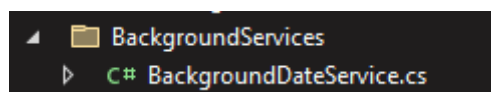
3. Wykorzystane technologie w projekcie:

- **Microsoft.AspNetCore.Identity.UI:**
 - Biblioteka z gotowymi komponentami interfejsu użytkownika do obsługi systemu zarządzania użytkownikami i uprawnieniami w projektach opartych na ASP.NET Core.
- **Microsoft.VisualStudio.Web.CodeGeneration.Design:**
 - Narzędzie do generowania kodu na podstawie szablonów i modelu danych w projektach opartych na ASP.NET Core.
- **Microsoft.EntityFrameworkCore.Tools:**
 - Narzędzie do wykonywania różnych operacji na bazie danych, takich jak migracje czy tworzenie tabel na podstawie klas, w projektach opartych na Entity Framework Core.
- **Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore:**
 - Biblioteka do diagnozowania i rozwiązywania problemów z bazą danych w projektach opartych na ASP.NET Core i Entity Framework Core.
- **Microsoft.AspNetCore.Identity.EntityFrameworkCore:**
 - Biblioteka do implementacji systemu zarządzania użytkownikami i uprawnieniami w projektach opartych na ASP.NET Core i Entity Framework Core.
- **Microsoft.EntityFrameworkCore.SqlServer:**
 - Biblioteka do połączenia projektu opartego na Entity Framework Core z bazą danych MS SQL Server.
- **AutoMapper**
 - Biblioteka do automatycznego mapowania obiektów między różnymi typami danych, ułatwiająca przetwarzanie i przesyłanie danych między warstwami aplikacji.
 - Dostępna zarówno z frameworkiem .NET jak i innymi platformami.

4. Implementacja:



```
builder.Services.AddScoped<ICarService, CarService>();  
builder.Services.AddScoped<IRentService, RentService>();  
builder.Services.AddScoped<IReservationService, ReservationService>();
```



5. Instalacja:

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

6. 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.

7. 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)

8. 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

9. 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.

10. 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.

11. 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

