Dokumentacja Projektu

LuxDrive - Wypożyczalnia Samochodów

Autor: Krystian Nowak
Nr indeksu: 177132
Data: Styczeń 2026

Spis treści

---

1. Wstęp

LuxDrive to nowoczesna aplikacja webowa przeznaczona do wypożyczania samochodów premium. System umożliwia przeglądanie dostępnych pojazdów, sprawdzanie ich dostępności w wybranych terminach oraz dokonywanie rezerwacji online. Aplikacja została zaprojektowana z myślą o intuicyjnej obsłudze i nowoczesnym wyglądzie.

Projekt wykorzystuje architekturę trzywarstwową, składającą się z warstwy prezentacji (frontend), warstwy logiki biznesowej (backend) oraz warstwy danych (baza danych). Takie podejście zapewnia separację odpowiedzialności i ułatwia rozwój oraz utrzymanie systemu.

---

2. Wykorzystane technologie

2.1. Frontend

- HTML5 - struktura dokumentu
- CSS3 - stylowanie i animacje
- JavaScript (ES6+) - logika aplikacji
- Font Awesome - ikony
- Google Fonts (Inter) - typografia

2.2. Backend

- .NET 8 - framework aplikacji
- Minimal API - budowa endpointów REST
- Npgsql - sterownik PostgreSQL dla .NET

2.3. Baza danych

- PostgreSQL - system zarządzania bazą danych
- Funkcje PL/pgSQL - logika biznesowa po stronie bazy

---

3. Struktura projektu

Plik/Katalog	Opis
index.html	Strona główna aplikacji
index.css	Arkusz stylów CSS
app.js	Logika JavaScript
server/Program.cs	Główny plik API .NET
server/CarRentApi.csproj	Plik projektu .NET
database/init_database.sql	Inicjalizacja bazy danych
database/function_*.sql	Funkcje SQL

---

4. Warstwa prezentacji (Frontend)

Warstwa prezentacji składa się z trzech głównych plików tworzących interfejs użytkownika.

4.1. index.html

Główny plik HTML definiujący strukturę strony. Zawiera sekcje: nagłówek z nawigacją, baner powitalny (hero), filtry wyszukiwania, siatkę samochodów oraz stopkę. Wykorzystuje semantyczne znaczniki HTML5 dla lepszej dostępności.

4.2. index.css

Arkusz stylów zawierający kompletny system designu. Wykorzystuje zmienne CSS do definiowania kolorów, animacji i efektów glassmorphism. Strona jest w pełni responsywna dzięki zastosowaniu CSS Grid i Flexbox.

4.3. app.js

Plik JavaScript obsługujący logikę aplikacji po stronie klienta. Główne funkcje to: pobieranie listy samochodów z API, filtrowanie i wyszukiwanie, sprawdzanie dostępności, obsługa formularza rezerwacji oraz wyświetlanie sugestii podobnych pojazdów.

---

5. Warstwa logiki biznesowej (Backend)

Backend został zbudowany przy użyciu .NET 8 z wykorzystaniem wzorca Minimal API, który umożliwia tworzenie lekkich i wydajnych endpointów REST bez nadmiarowego kodu.

5.1. Konfiguracja

Aplikacja konfiguruje CORS (Cross-Origin Resource Sharing) umożliwiając dostęp z dowolnego źródła. Połączenie z bazą danych jest realizowane za pomocą biblioteki Npgsql.

5.2. Modele danych

Aplikacja wykorzystuje dwa główne rekordy (records) do obsługi żądań:

- **CheckAvailabilityRequest** - sprawdzenie dostępności (CarId, PickupDate, ReturnDate)
- **ReservationRequest** - utworzenie rezerwacji (dane klienta, daty, cena)

---

6. Warstwa danych (Baza danych)

6.1. Tabela cars

Przechowuje dane samochodów:

Kolumna	Typ	Opis
id	SERIAL	Klucz główny
name	VARCHAR(100)	Nazwa samochodu
category	VARCHAR(50)	Kategoria (np. Luksusowe)
type	VARCHAR(30)	Typ techniczny
price	DECIMAL	Cena za dzień
rating	DECIMAL	Ocena (1-5)
seats	INT	Liczba miejsc
transmission	VARCHAR	Skrzynia biegów
fuel	VARCHAR	Rodzaj paliwa

6.2. Tabela reservations

Przechowuje rezerwacje klientów:

Kolumna	Typ	Opis
id	SERIAL	Klucz główny
car_id	INT	Klucz obcy do cars
customer_name	VARCHAR	Imię i nazwisko
customer_email	VARCHAR	Adres email
pickup_date	DATE	Data odbioru
return_date	DATE	Data zwrotu
total_price	DECIMAL	Całkowita cena
status	VARCHAR	Status rezerwacji

6.3. Funkcje SQL

check_car_availability(car_id, pickup_date, return_date) - funkcja sprawdzająca dostępność samochodu w podanym terminie. Zwraca TRUE jeśli samochód jest dostępny.

suggest_similar_car(car_id, pickup_date, return_date) - funkcja znajdująca podobny dostępny samochód gdy wybrany jest niedostępny. Priorytetyzuje: ten sam typ i podobną cenę, następnie ten sam typ, potem podobną cenę.

---

7. API REST

Backend udostępnia cztery endpointy REST obsługujące główne funkcje aplikacji:

Metoda	Endpoint	Opis
GET	/api/cars	Pobiera listę wszystkich samochodów
POST	/api/check-availability	Sprawdza dostępność samochodu
POST	/api/suggest-similar	Sugeruje podobny dostępny samochód
POST	/api/reservations	Tworzy nową rezerwację

---

8. Instrukcja uruchomienia

8.1. Wymagania systemowe

- PostgreSQL (wersja 12 lub nowsza)
- .NET SDK 8.0
- Przeglądarka internetowa (Chrome, Firefox, Edge)

8.2. Konfiguracja bazy danych

Wykonaj poniższe polecenia w terminalu:

```
psql -U postgres -c "CREATE DATABASE carrent;"
psql -U postgres -d carrent -f database/init_database.sql
psql -U postgres -d carrent -f database/function_check_availability.sql
psql -U postgres -d carrent -f database/function_suggest_similar_car.sql
```

8.3. Uruchomienie backendu

```
cd server
dotnet run
```

Serwer uruchomi się na porcie 3001 (http://localhost:3001).

8.4. Uruchomienie frontendu

Otwórz plik index.html w przeglądarce lub uruchom lokalny serwer HTTP:

```
npx serve
```

---

9. Podsumowanie

Projekt LuxDrive stanowi kompletne rozwiązanie do zarządzania wypożyczalnią samochodów. Aplikacja łączy nowoczesny frontend z wydajnym backendem .NET oraz relacyjną bazą danych PostgreSQL. System oferuje pełną funkcjonalność od przeglądania oferty przez sprawdzanie dostępności po finalizację rezerwacji.

Główne zalety rozwiązania:

- Nowoczesny, responsywny interfejs użytkownika
- Wydajne API REST z walidacją danych
- Inteligentny system sugestii podobnych pojazdów
- Bezpieczna obsługa rezerwacji z kontrolą dostępności
- Łatwa rozbudowa i utrzymanie dzięki modułowej architekturze