1. **Opis tematyki**: Aplikacja Komisu samochodowego z podziałem na użytkowników (klientów) oraz adminów (admin/super admin). Aplikacja wykonuje podstawowe akcje CRUD. Zawiera prostą logikę biznesową manipulując statusami samochodów, historią zamówień użytkownika itp.
2. **Opis wykorzystanych technologii:**
3. Język programowania PHP 8.1 (<https://www.php.net/releases/8.1/en.php>)
4. Framework Laravel 9.17.0 (<https://laravel.com/docs/9.x/releases>)
5. Baza danych PostgreSQL 13 (<https://www.postgresql.org/download/>)
6. Biblioteka Spatie do zarządzania rolami użytkowników (<https://spatie.be/docs/laravel-permission/v5/introduction>)
7. Biblioteka Bootstrap 5 (<https://getbootstrap.com/docs/5.0/getting-started/introduction/)>
8. **Opis wykorzystanych narzędzi**
9. Środowisko programistyczne PHPStorm (<https://www.jetbrains.com/phpstorm/documentation/?source=google&medium=cpc&campaign=14335686150&term=phpstorm&gclid=Cj0KCQjwwJuVBhCAARIsAOPwGASHIvS-g9nCdJltnWleFVyT4pDWXDpwd5dhMVWevogYbhJ68CK0PL0aAmdHEALw_wcB)>

Darmowa licencja dla studentów

1. pgAdmin 4 (<https://www.pgadmin.org/download/>)

Narzędzie, pozwalające pracować z bazą PostgreSQL korzystając z GUI

1. Composer CLI (<https://getcomposer.org/)>
2. Node.js (<https://nodejs.org/en/)>
3. **Instrukcja uruchomienia aplikacji**
4. Zainstalować PHP w wersji min 8.1
5. Zainstalować serwis bazy danych PostgreSQL 13
6. Zainstalować „composer”
7. Zainstalować „Node.js”
8. Uzupełnić dane do połączenia z bazą danych w pliku .env

**Polecenia do uruchomienia w głównym folderze projektu:**

1. composer update
2. composer install
3. npm install
4. php artisan db:create (polecenie utworzy bazę danych. UWAGA! Jeśli baza o nazwie podanej w pliku konfiguracyjnym już istnieje zostanie ona usunięta)
5. php artisan migrate
6. php artisan db:seed
7. php artisan serve

Aplikacja będzie dostępna pod adresem: <http://localhost:8000>

1. **Omówienie kodu**
2. Parametry połączenia z bazą danych ustawiają się w pliku .env

Obraz zawierający tekst

Opis wygenerowany automatycznie

Połączenie z bazą następuje w momencie wykonywania zapytań i jest obsługiwane poprzez framework.

1. Seed przykładowych danych do bazy następuje w momencie wykonywania polecenia „php artisan db:seed”.

Przykład:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Wszystkie polecenia są opisane w pliku /database/seeders/DatabaseSeeder.php

1. Przykładowa funkcjonalność aplikacji:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. Realizacja uwierzytelniania i autoryzacji:

Przykład autoryzacji:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Przykład uwierzytelniania:

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. Przykład walidacji danych przekazywanych od użytkownika:

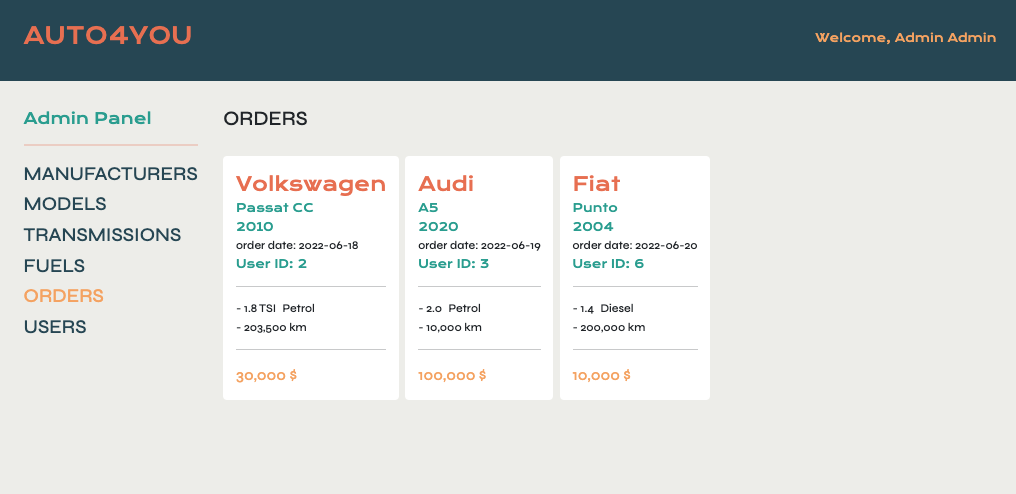
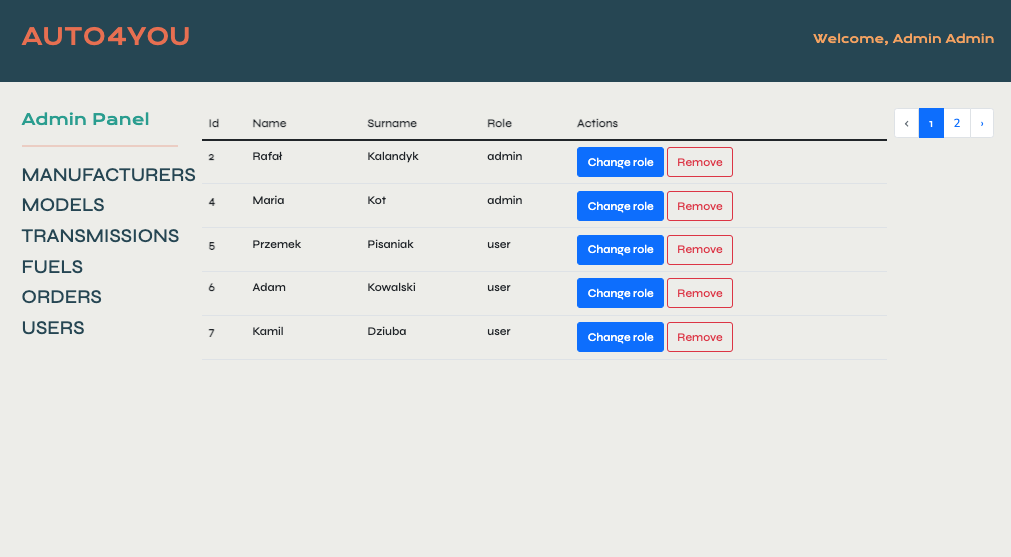
Obraz zawierający tekst

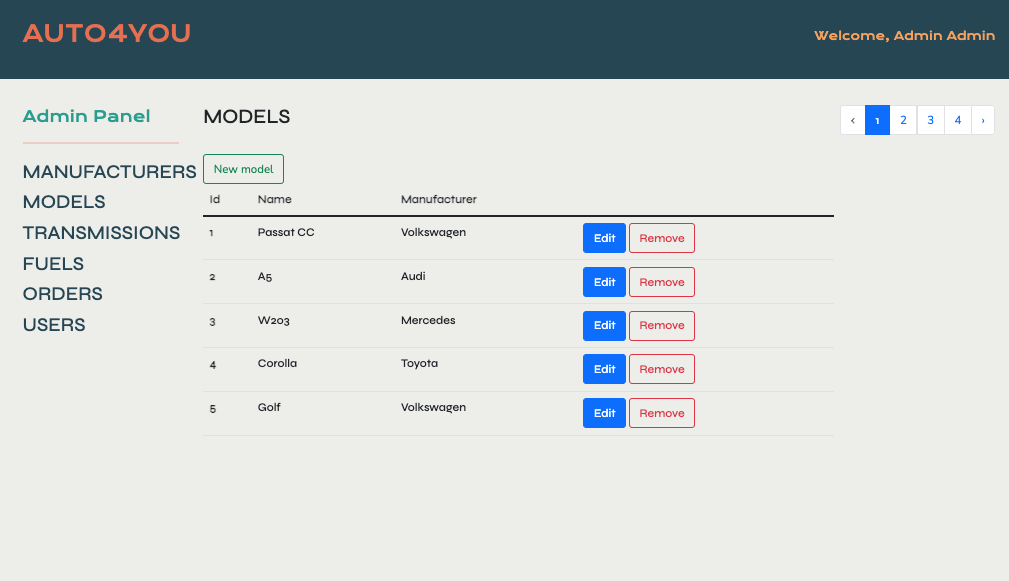
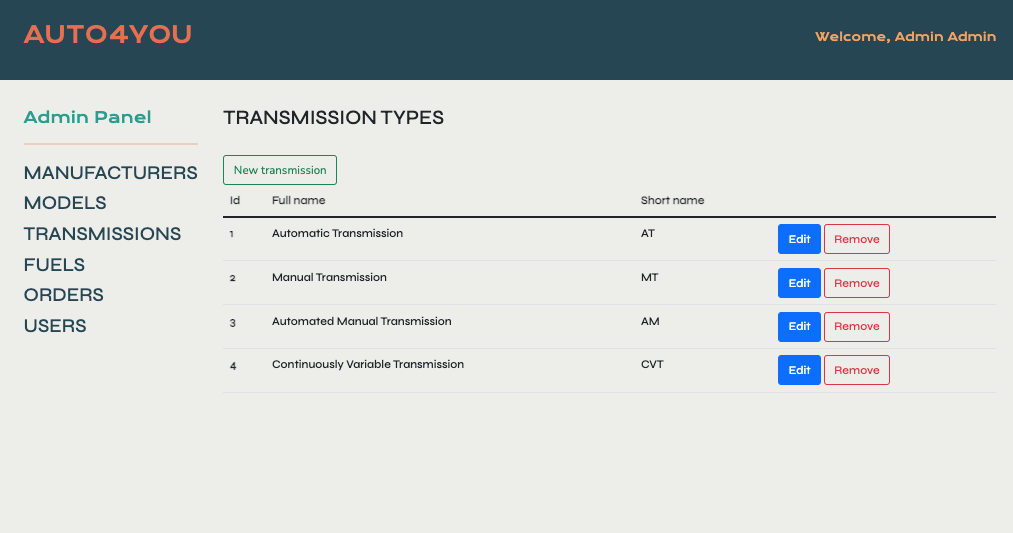
Opis wygenerowany automatycznie

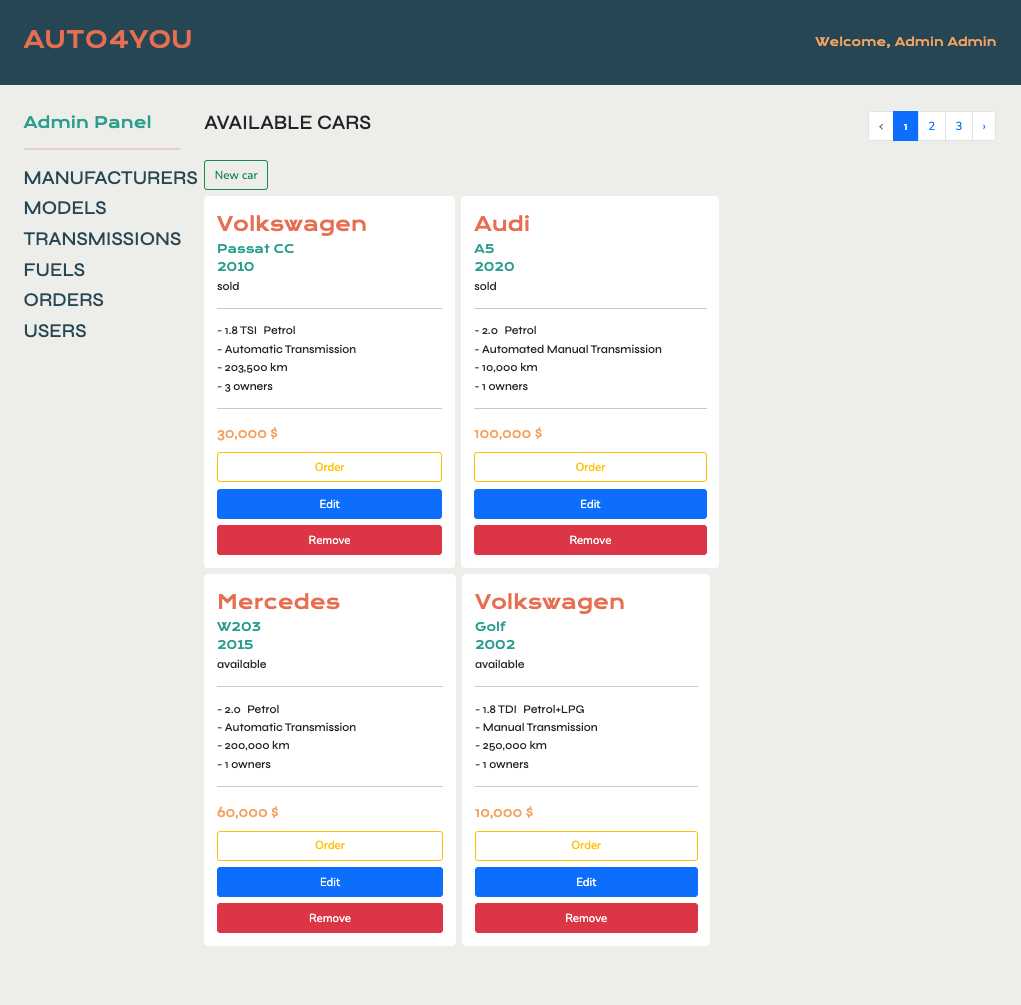
Obraz zawierający tekst

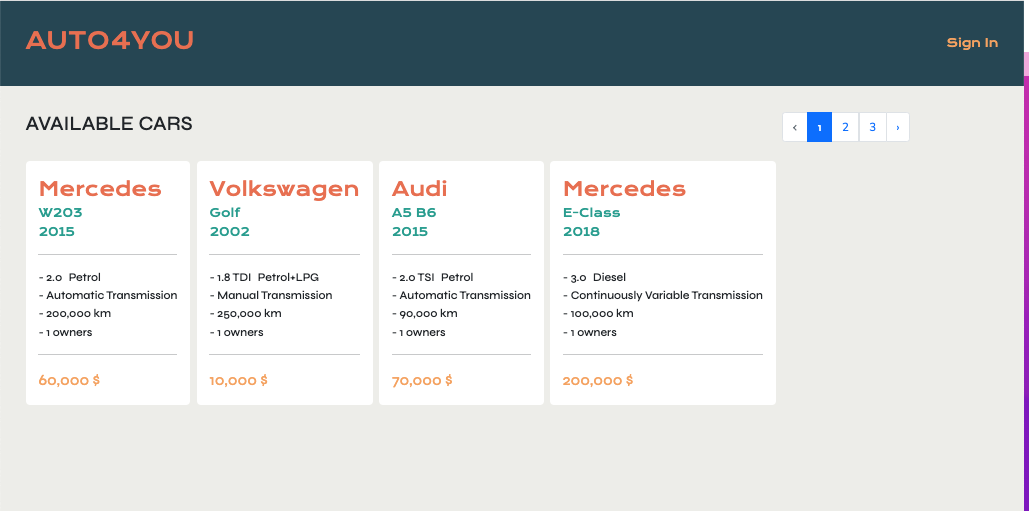
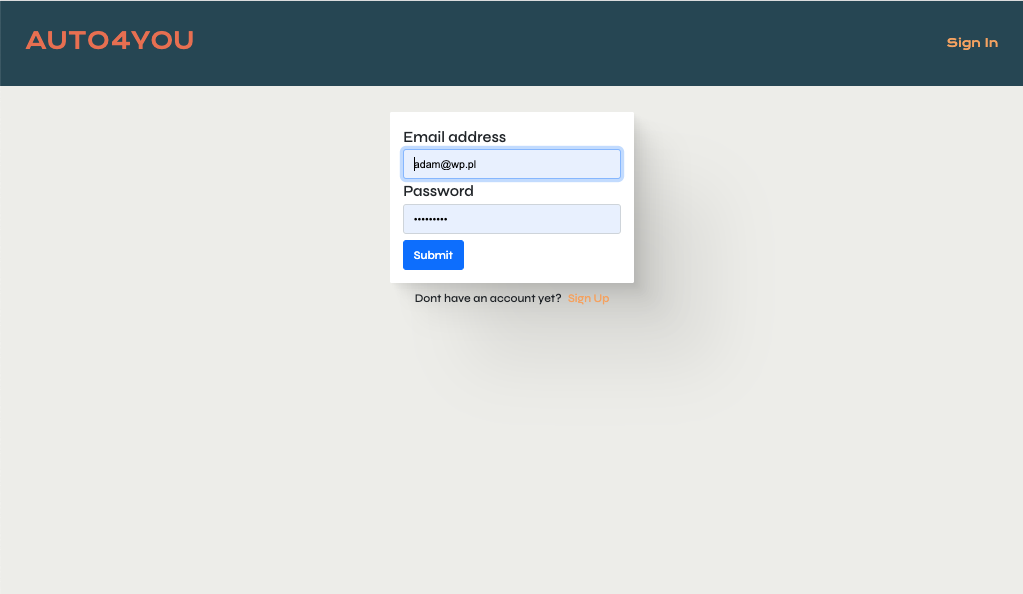
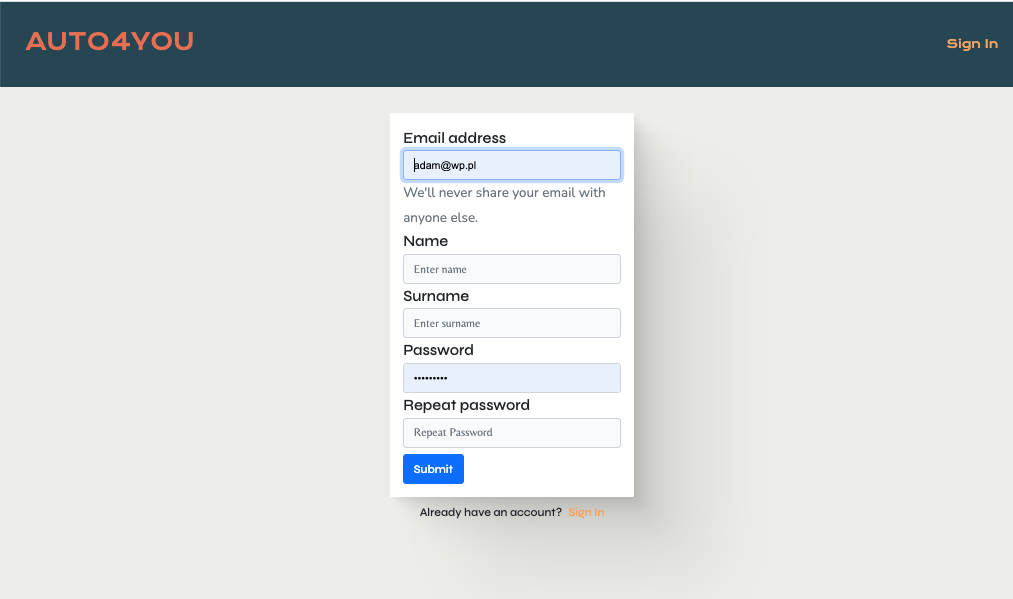
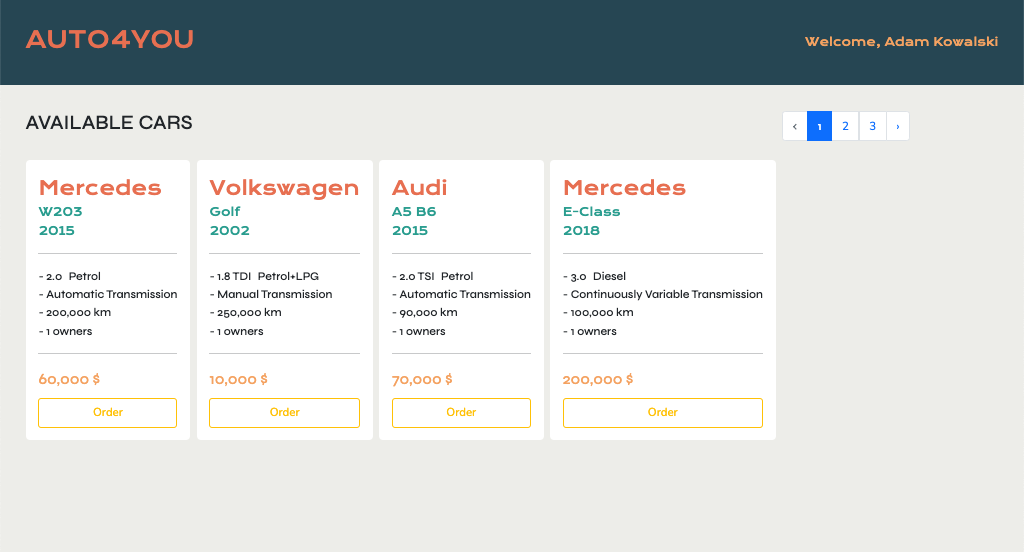
Opis wygenerowany automatycznie

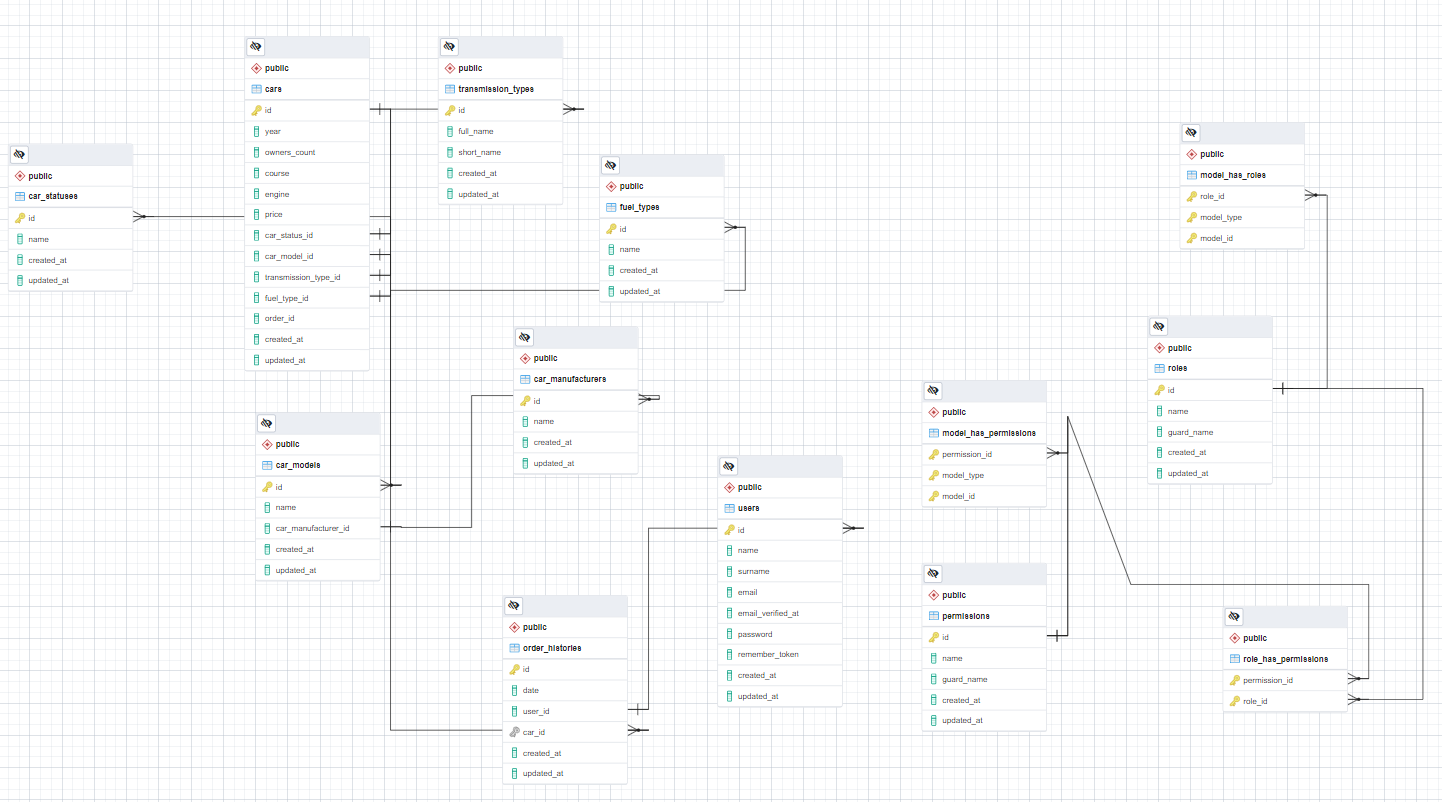


1. Zrzuty ekranu wybranych podstron aplikacjiObraz zawierający stół

   Opis wygenerowany automatycznieObraz zawierający stół

   Opis wygenerowany automatycznieObraz zawierający tekst

   Opis wygenerowany automatycznie
2. **Omówienie zawartości bazy**
3. Diagram ERD



1. Wymienienie i opis tabel:

**- Users**

Przechowuje dane użytkownika

**- Cars**

Przechowuje rekordy wszystkich samochodów

**- Car\_Statuses**

Przechowuje możliwe statusy samochodów

**- Transmission\_Types**

Przechowuje rekordy dostępnych rodzajów przekładni samochodowych

**- Fuel\_Types**

Przechowuje rekordy dostępnych rodzajów paliw

**- Car\_Manufacturers**

Przechowuje rekordy producentów samochodów

**- Car\_Models**

Przechowuje rekordy modeli samochodów

**- Orders\_Histories**

Przechowuje referencje pomiędzy użytkownikiem a zakupionym przez niego samochodem

**- Roles**

Przechowuje dostępne roli

**- Permissions**

Przechowuje dostępne pozwolenia

- **Role\_Has\_Permissions**

Powiązuje rolę a dostęp

- Model\_Has\_Permissions

Powiązuje użytkownika a dostęp

- **Model\_Has\_Roles**

Powiązuje użytkownika a rolę