

<b>AI1</b>	<b>Dokumentacja Projektu</b>
<b>Autor</b>	Mateusz Kiszka
<b>Kierunek, rok</b>	Informatyka i ekonometria, II rok, st. Stacjonarne (3.5-I)
<b>Temat</b>	Wypożyczalnia samochodów elektrycznych

<b>Tematyka projektu</b>	<b>1</b>
<b>Realizacja projektu</b>	<b>1</b>
Etap 1: Utworzenie zestawu bazy danych	2
Etap 2: Tworzenie widoków oraz controllerów	3
Etap 3: Tworzenie share/form oraz share/index	4
Etap 4: Zmiana działania zasobów	5
Etap 5: Wytworzenie finansów oraz własnych zasobów	5
Etap 6: Zmiana wyglądu oraz strony głównej	6
Etap 7: Przetestowanie działania całej aplikacji	7
<b>Testowanie aplikacji</b>	<b>7</b>
Testowanie standardowego użytkownika	7
Testowanie z użyciem Lighthouse	8
<b>Implementacja</b>	<b>8</b>

## Tematyka projektu

Celem projektu jest stworzenie aplikacji internetowej dla wypożyczalni samochodów elektrycznych. Aplikacja ma być w pełni funkcjonalna z bazą danych PostgreSQL, zawierać logowanie oraz rejestrację, podział na role (user i admin), zawierać dostęp do wszystkich tabel z opcją CRUD dla widoku administratora, zarządzanie własnymi zasobami dla użytkownika, pełną obsługę profilu oraz logikę biznesową dla zasobów. Aplikacja zawiera 5 tabel powiązanych relacyjnie Users, Cars, Reservations, Rentals, Payments. Każda z nich posiada własną walidację oraz jest responsywna z panelu administratora czy widoku użytkownika względem operacji wykonywanych w pozostałych tabelach.

## Realizacja projektu

Aplikacja została zbudowana w oparciu o nowoczesne technologie:

- Backend: Laravel (PHP), Eloquent ORM, PostgreSQL
- Frontend: HTML5, CSS3 (Tailwind, Bootstrap), JavaScript, SB Admin
- Autoryzacja: Laravel Breeze

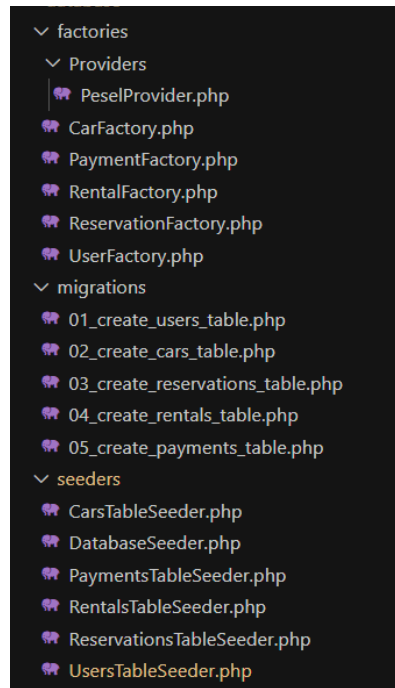
#### Etap 1:

```
class DatabaseSeeder extends Seeder
{
    0 references | 0 overrides
    public function run(): void {
        $this->call(class: [
            UsersTableSeeder::class,
            CarsTableSeeder::class,
            ReservationsTableSeeder::class,
            RentalsTableSeeder::class,
            PaymentsTableSeeder::class,
        ]);
    }
}
```

#### Utworzenie zestawu bazy danych

Pierwszym etapem projektu było utworzenie bazy danych oraz przygotowanie systemu jak i aplikacji do obsługi bazy w PostgreSQL 17. Zostały utworzone modele, migracje, factories, seedery oraz providery dla obsługi tworzenia bazy danych.

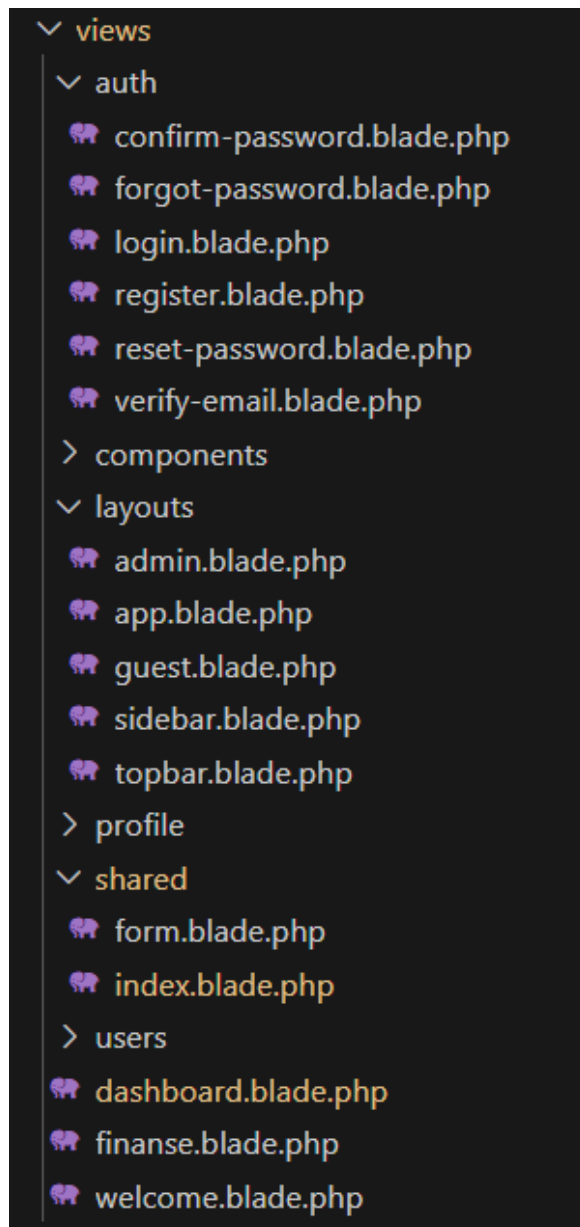
*Rysunek 1. Wywołanie seederów*



Rysunek 2. Pliki związane z bazą danych

## Etap 2: Tworzenie widoków oraz controllerów

Następnie utworzyłem widoki oraz kontrolery dla obsługi widoku tabel oraz rejestracji z logowaniem. Widoki zostały początkowo utworzone jako widoki pakietu blade jednak z czasem pozamieniałem większość widoków oraz ich obsługę dla utworzenia własnych metod oraz stylów. Posłużyłem się również wtyczką SB Admin aby uczynić stronę bardziej wizualną.



Rysunek 3. Pliki widoków

### Etap 3: Tworzenie share/form oraz share/index

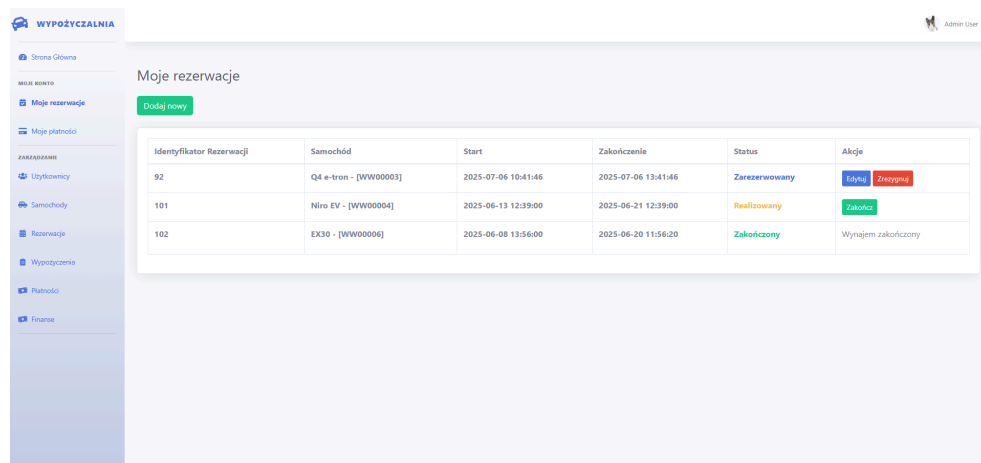
Stworzyłem dwa główne widoki które są stosowane do wyświetlania oraz edycji lub dodawania danych w aplikacji. Działają one na zasadzie pobrania kluczy głównych i nagłówków oraz ścieżek aby skutecznie wyświetlać pożądane zasoby.

## Etap 4: Zmiana działania zasobów

Postanowiłem zacząć pracę nad działaniem logiki dla rekordów w bazie danych. Zostały one ze sobą w ten sposób połączone logicznie iż oddziałują bezpośrednio na zmiany w pozostałych zasobach i aktualizują swoje statusy.

## Etap 5: Wytworzenie finansów oraz własnych zasobów

Utworzyłem widok finansów dla administratora gdzie mógł weryfikować własne zasoby oraz mieć podgląd do efektywności swojej działalności. Dodatkowo stworzyłem dla zarówno użytkownika jak i administratora widok dla zarządzania własnymi zasobami (rezerwacje oraz płatności).



Wypożyczalnia Admin User

Strona Główna

MOJE KONTO

Moje rezerwacje

Moje płatności

ZARZĄDZANIE

Użytkownicy

Samochody

Rezerwacje

Wypożyczenia

Płatności

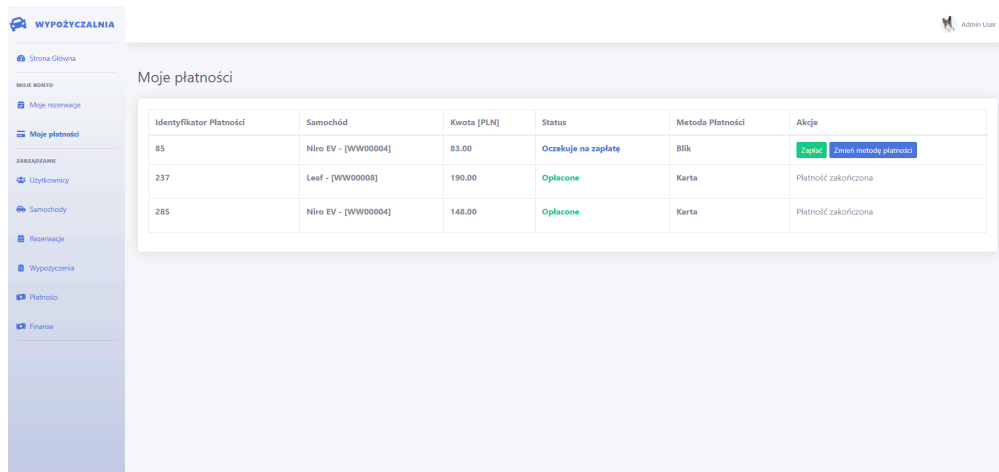
Finanse

Moje rezerwacje

Dodaj nowy

Identyfikator Rezerwacji	Samochód	Start	Zakończenie	Status	Akcje
92	Q4 e-tren - [WW00003]	2025-07-06 10:41:46	2025-07-06 13:41:46	Zarezerwowany	<a href="#">Edytuj</a> <a href="#">Zmierzaj</a>
101	Niro EV - [WW00004]	2025-06-13 12:39:00	2025-06-21 12:39:00	Realizowany	<a href="#">Zakończ</a>
102	EX30 - [WW00006]	2025-06-08 13:56:00	2025-06-20 11:56:20	Zakończony	Wynajem zakończony

Rysunek 4. Podgląd rezerwacji użytkownika



Wypożyczalnia Admin User

Strona Główna

MOJE KONTO

Moje rezerwacje

Moje płatności

ZARZĄDZANIE

Użytkownicy

Samochody

Rezerwacje

Wypożyczenia

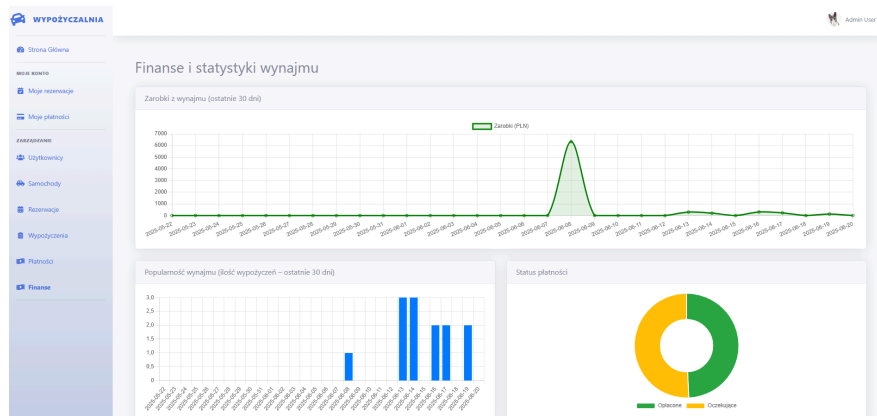
Płatności

Finanse

Moje płatności

Identyfikator Płatności	Samochód	Kwota [PLN]	Status	Metoda Płatności	Akcje
85	Niro EV - [WW00004]	83.00	Oczekuje na zapłatę	Blisk	<a href="#">Zapłać</a> <a href="#">Zmień metodę płatności</a>
237	Leaf - [WW00008]	190.00	Oplacone	Karta	Płatność zakończona
285	Niro EV - [WW00004]	148.00	Oplacone	Karta	Płatność zakończona

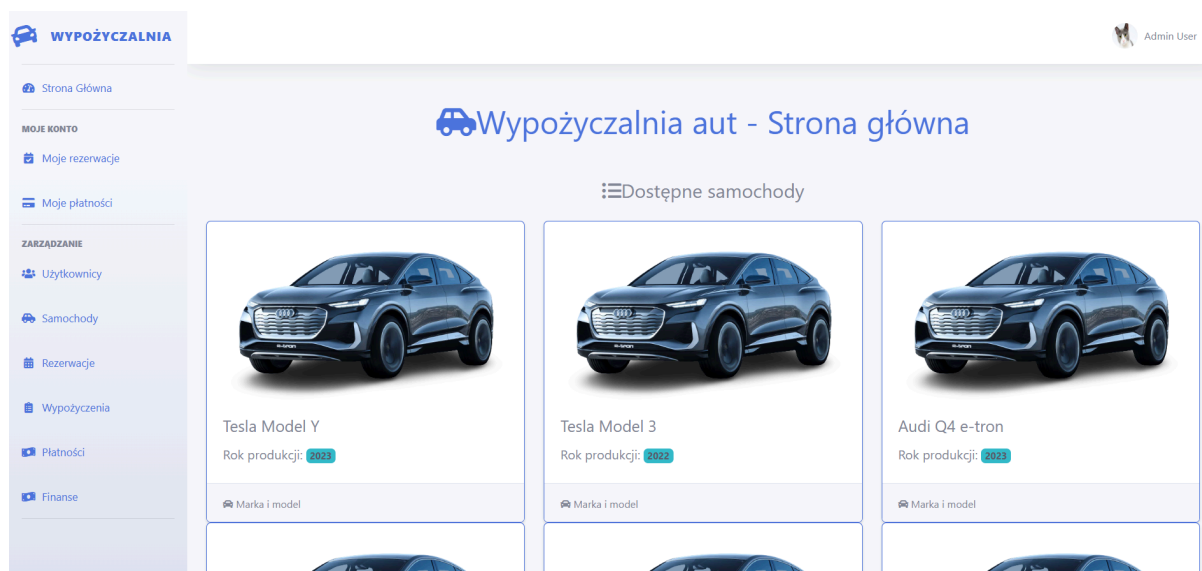
Rysunek 5. Podgląd płatności użytkownika



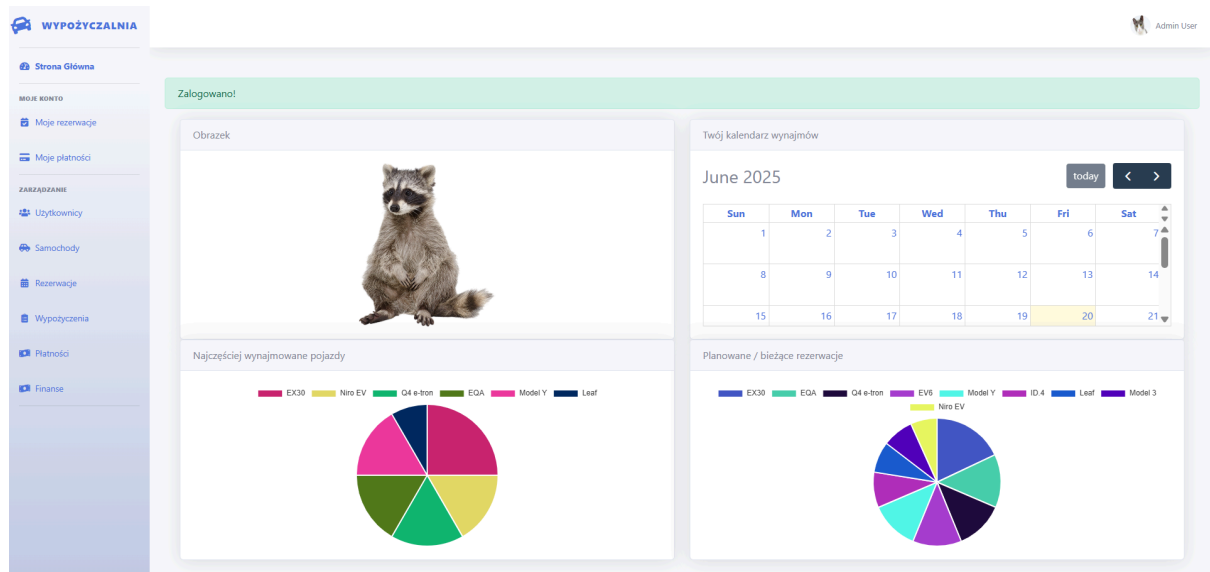
Rysunek 6. Podgląd panelu finansów

## Etap 6: Zmiana wyglądu oraz strony głównej

Ostatecznie przeprowadziłem zmianę dla wyglądu i obsługi strony welcome oraz dashboard aby były one bardziej przyjazne w obsłudze dla użytkownika



Rysunek 7. Podgląd strony startowej



Rysunek 8. Podgląd strony głównej

## Etap 7: Przetestowanie działania całej aplikacji

Ostatecznie przeprowadziłem test Lighthouse dla strony oraz przeprowadziłem własne testy walidacji dla aplikacji aby upewnić się że spełnia wszystkie normy walidacji czy funkcjonalności.

## Testowanie aplikacji

Aplikację postanowiłem przetestować na dwa sposoby.

- Pierwszy sposób to standardowe użycie strony, używanie jej funkcji, przycisków, selectorów oraz ogólnego CRUD'a.
- Drugi to przetestowanie za pomocą wtyczki webowej Lighthouse.

## Testowanie standardowego użytkownika

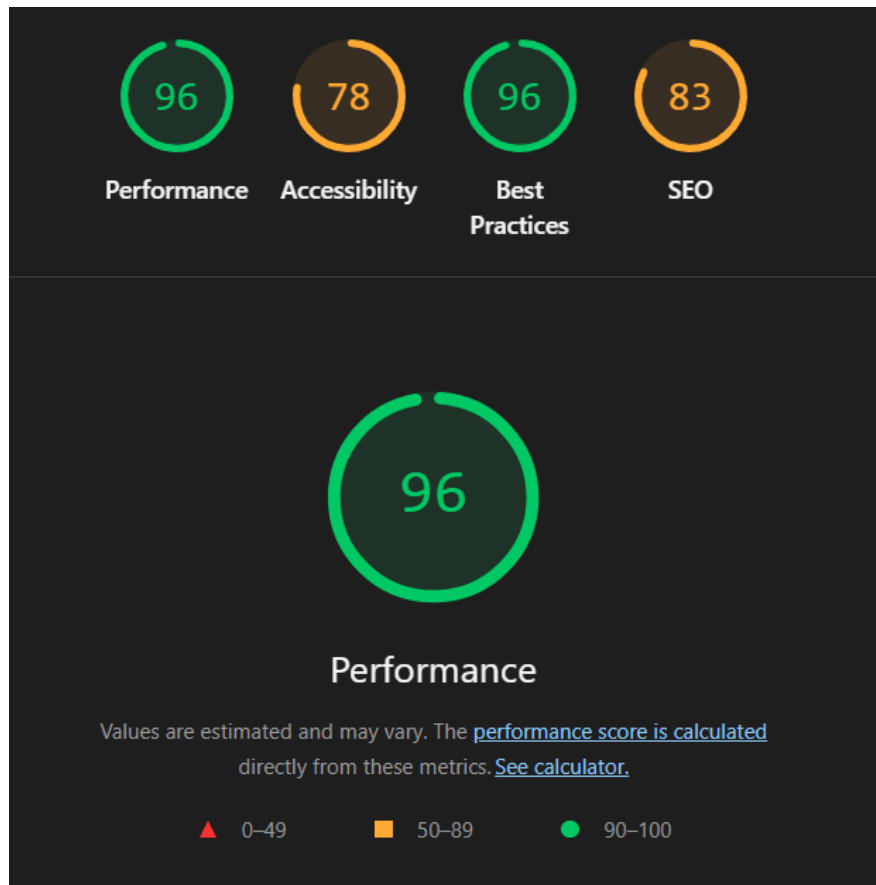
**Założenie testu:** Poprawnie działające wszystkie funkcje strony.

**Rezultat testu:** Po spędzeniu dwóch godzin używając stronę i wyciskając z niej wszystkie możliwości, doszedłem do rezultatu w którym okazało się strona działa poprawnie jednak występowały błędy przy zbyt szybkim odświeżeniu strony czy nagłym dodaniu dwóch rekordów, gdzie inne metody nie zdążyły sprawdzić zmienionego stanu zmiennej.

## Testowanie z użyciem Lighthouse

**Założenie testu:** Wysoki wynik testu Lighthouse

**Rezultat testu:** Pozytywny wynik testu lighthouse jednak pozostawia on miejsce do poprawy.



Rysunek 9. Wynik testu Lighthouse

## Implementacja

W ramach projektu dla stylizacji interfejsu użytkownika zastosowano Tailwind CSS oraz Bootstrap. Wykorzystano mechanizmy grid/flexbox, a także komponenty Laravel Blade. Dla użytkowników stworzone zostały widoki strony wejściowej, strony głównej panel logowania oraz rejestracji, widok ich rezerwacji, płatności oraz ich profilu. Dla administratora dodatkowo dodano pełną obsługę CRUD dla wszystkich tabel oraz finanse. Zastosowane zostało kilka modułów funkcyjnych takich jak kalendarz czy wykresy.