

**Dokumentacja Projektu PHP - BK Auto Serwis**

Aplikacja internetowa Auto Serwis jest aplikacją służącą do podstawowej obsługi klientów w warsztacie samochodowym.

Głównym zadaniem strony jest wykorzystanie portalu przez:

* **Serwisanta -**  możliwość logowania sie do serwisu (logowanie, rejestracja, odzyskiwanie hasła), profil administratora, dodawania klientów oraz ich pojazdów do bazy, historię serwisów.
* **Klienta:** reklama i informacja dla klienta na temat zakładu, serwisów i cen, oraz możliwości wysłania formularza z proźbą o serwis.

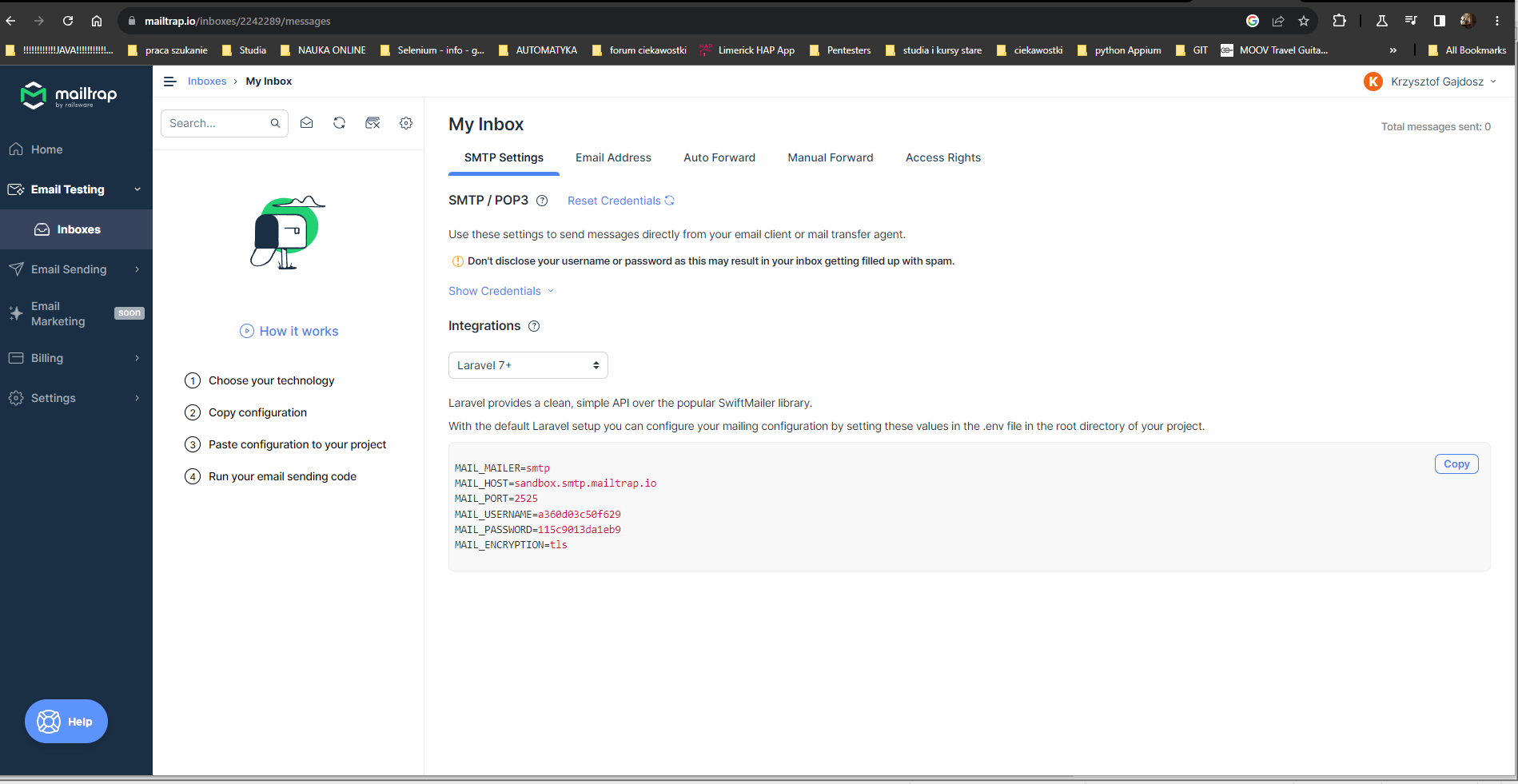
**Zawartość strony:**

Portal zawiera podstrony opisowe reklamujące AutoSerwis takie jak Strona główna, Historia oraz Galeria. Klient ma możliwość również skorzystanie z formularza kontaktowego w celu komunikacji z serwisem jak również ma możliwość przeglądniecia podstawowych stawek i cen serwisu pojadów

Z perspektywy serwisu administrator portalu ma możliwość::

* założenia konta w serwisie,
* zalogowania się do niego,
* odzyskanie hasła,
* sprawdzenie i modyfikację swoich danych,
* usunięcie konta,
* zmiana hasła
* dodanie klienta do wewnętrznej listy klientów
* dodania samochodu klienta
* przeglądnięcie historii serwisów
* przejrzenie oraz edycji cennika usług i napraw (dodanie nowych usług)
* zgłoszenie samochodu do naprawy razem z wyborem daty oraz usługami serwisowymi(wybieranymi z “select’a”)

**Do wykonania projektu wykorzystaliśmy:**

* HTML
* CSS
* Javascript
* PHP
* MySQL
* bootstrap
* Mailtrap (api) - testowy klient pocztowy wykorzystywany do odbierania poczty w razie resetowania hasła - klient pocztowy może być ustawiony w pliku .env
* 

Jako, że hasła są zahashowane, to tutaj podajemy loginy i hasła do logowania:

* **Admin** (dane mogą również być stworzone przez okno rejestracji):**login**: confesor123@gmail.com; **hasło**: aaddmmiinn
* **Admin** (dane mogą również być stworzone przez okno rejestracji):**login**: bwasko@gmail.com; **hasło**: aaddmmiinn

Projekt jest podzielony na sekcje:

-a) **frontendowa** [resources > views] - zawierająca wszystkie widoki (blady), pozwalająca określić jak dany widok ma wyglądać. Część ustawień jak ma wyglądać strona lub jaką ma mieć funkcjonalność można też znaleźć w znanych folderach css lub/i js (podfoldery resources)

b) **backendowa** [app > Http > Controllers] - pozwalająca zaprogramować funkcjonalność

c) **modelowa** - [app > Http > Models] pozwalająca stworzyć główne blueprinty / schematy jak bedą wyglądać wykorzystywane obiekty (jakie bedą miały pola w bazie, w jakiej tabeli będą sie znajdować, relacje między tabelami)

d) **routing** [routes > web.php] - warstwa łącząca nasz frontend z backendem

e) **bazy danych [database]**

**-**  migrations – zawiera wszystkie migracje (określa jak wygląda baza danych)

- seeders – zawiera przykladowe dane które mogą być wstawione do bazy

Pozostałe sekcje są odpowiedzialne za ustawienia Laravela (np. Ustawienie połączenia bazy danych, klientów pocztowych itp [można to zrobić w pliku: .env], oraz

**Opis wykorzystywanych folderów i klas**

1. **Frontendowa** [resources > views] wykorzystywane foldery i klasy
   1. **Auth** 
      1. **forgot-password.blade.php -** widok okna “Zapomniałeś hasła”
      2. **Login.blade.php -** widok okna “Logowania”
      3. **register.blade.php -** widok okna “Rejestracji”
      4. **reset-password.blade.php -** widok okna “Resetowania hasła”
   2. **Klienci**
      1. **Edit.blade.php -** widok okna “edycji klienta”
      2. **showAll.blade.php -** główny widok okna “klienta” - widoczne dodaj/edycja i usuń funkcjonalności
   3. **Kontakt**
      1. **contact.blade.php -** widok okna “sukcesu wysłania wiadomości”
      2. **kontakt.blade.php -** główny widok okna “formularza kontaktowego” - ***występuje walidacja Laravela***
   4. **Naprawy**
      1. **showAll.blade.php -** główny widok okna “napraw” - pozwala wybrać klienta w celu wylistowania jego napraw w zakładzie
   5. **Opisowe**
      1. **Galeria.blade.php -** główny widok okna “galeri”
      2. **Historia.blade.php -** główny widok okna “historii serwisu”
      3. **Start.blade.php -** główny widok okna “autoserwisu”  ***początkowe***
   6. **Profilowe**
      1. **Delete-user-form.blade.php -** widok okna “usunięcia użytkownika”
      2. **Update-password- form.blade.php -** widok okna “edycji hasła”
      3. **Update-profile-information.blade.php -** widok okna “edycji profilu”
   7. **Samochody**
      1. **Edit.blade.php -** widok okna “edycji pojazdu”
      2. **showAll.blade.php -** główny widok okna “pojazdu” - widoczne dodaj/edycja i usuń funkcjonalności
   8. **Serwisy**
      1. **newSerwise.php -** widok okna “dodania serwisu”
      2. **showAll.blade.php -** główny widok okna “Serwisów” - widoczne dodaj/edycja i usuń funkcjonalności
   9. **TypySerwisu**
      1. **Edit.blade.php -** widok okna “edycji Cennika”
      2. **showAll.blade.php -** główny widok okna “Cennika” - widoczne dodaj/edycja i usuń funkcjonalności
   10. **Main.blade.php -** widok wspolnych komponentów wszystkich widoków jak nagłówek i stopka
2. **backendowa** [app > Http > Controllers]
   1. **KlientController**.php - klasa zawierająca wszystkie metody pozwalające na dodanie, edycje, usunięcie, zapisanie klienta – zawiera walidacje
   2. **KontaktController** - klasa zawierająca metodę walidacji formularza oraz przekierowanie na widok sukcesu
   3. **NaprawyController**.php - klasa zawierająca metodę showAll - pozwalającą na wyświetlenien widoku napraw
   4. **OpisyController**.php - klasa zawierająca metodę pozwalającą na wyświetlenien widoku galerii oraz historii
   5. **ProfileController.php** - klasa zawierająca wszystkie metody pozwalające na dodanie, edycje, usunięcie, zapisanie profiluoraz wylogowanie
   6. **SamochodControlelr**.php - klasa zawierająca wszystkie metody pozwalające na dodanie, edycje, usunięcie oraz zapisanie samochodu
   7. **SerwisController**.php - klasa zawierająca wszystkie metody pozwalające na dodanie, edycje, oraz zapisanie samochodu
   8. **TypySerwisu**.php - - klasa zawierająca wszystkie metody pozwalające na dodanie, edycje, usunięcie, zapisanie cennika – zawiera walidacje
3. **Folder Auth** zawierający: Controllery stworzone przez Laravel, są przez nas wykorzystywane do autoryzacji użytkownika, odzyskania hasła logowania oraz rejestracji
4. **modelowa** - [app > Http > Models]
   1. Klient – model wyglądu Klienta w bazie
   2. Samochod – model wyglądu pojazd w bazie
   3. SerwisSamochodu – model wyglądu sarwis samochoduw bazie
   4. TySerwisu – model wyglądu cennik i typ serwisuw bazie
   5. User – model wyglądu Użytkowniaka w bazie
5. **routing** [routes > web.php] - warstwa łącząca nasz frontend z backendem

Plik **web.php** zawiera trasy po między Kontrolerami a Widokami.

Wszystkie z tras są przez nas tworzone poza trasami dla KlientKontroler, gdzie jest wykorzystywany kontroler zasobów. Ten sposób zapewnia wachlarz wszystkich metod takich jak tworzenie, usuwanie, edycja, zapis do bazy itp.