|  |  |
| --- | --- |
| **AI1** | Dokumentacja projektu |
| **Autor** | Konrad Pluta, 125152 |
| **Kierunek, rok** | Informatyka, II rok, st. stacjonarne (3,5-l) |
| **Temat  projektu** | *Ewidencja napraw gwarancyjnych i pogwarancyjnych* |

## Wstęp.

Celem projektu jest stworzenie aplikacji internetowej do ewidencji napraw gwarancyjnych i pogwarancyjnych przy użyciu frameworka Laravel wersja 11. Aplikacja jest przeznaczona do obsługi małego przedsiębiorstwa zajmującego się usługami serwisującymi urządzenia elektroniczne. Użytkownicy aplikacji mogą dodawać swoje urządzenia, zgłaszać naprawy, opłacić naprawy oraz wystawiać opinie odnośnie danej naprawy. Aplikacja zapewnia pełną automatyzację procesu usługi serwisowej, obliczania kosztów, oraz wystawianiu prostej płatności.

## Zakres działania aplikacji

Aplikacja obejmuje następujące funkcjonalności:

1. **Rejestracja i logowanie użytkowników:**
   1. Tylko zarejestrowani i zalogowani użytkownicy mogą korzystać z pełnych funkcji aplikacji, takich jak dodawanie urządzeń oraz napraw.
   2. Nowi użytkownicy mogą samodzielnie się zarejestrować, a po rejestracji uzyskują dostęp do swojego konta.
2. **Zarządzanie usługami, pracownikami, urządzeniami oraz naprawami przez administratora:**
   1. Administrator może dodawać pracowników, edytować ich pensje lub zwolnić.
   2. Administrator może edytować, dodawać oraz usuwać usługi.
   3. Administrator może przypisywać zgłoszenia do pracowników.
   4. Administrator może usuwać urządzenia z bazy danych.
3. **Automatyzacja procesów biznesowych:**
   1. Aplikacja automatycznie sprawdza czy urządzenia podlegają gwarancji lub zniżką za usługi.
   2. Docelowa cena jest zablokowana przed zmianami.
   3. Po zakończeniu usługi automatycznie generuje się oczekująca płatność dla użytkownika.
4. **Funkcjonalności dla użytkowników:**
   1. Użytkownicy mogą przypisywać urządzenia które są dostępne w serwisie poprzez podanie numeru seryjnego urządzenia oraz daty produkcji.
   2. Użytkownicy mogą dodawać urządzenia do swojego konta z poza serwisu.
   3. Użytkownicy mogą dodawać zgłoszenia napraw swoich urządzeń, śledzić stan naprawy oraz po zakończonej usłudze – opłacić ją.
   4. Użytkownicy do zakończonych napraw mogą wystawić opinie.
   5. Użytkownik ma dostęp do swojego profilu gdzie może edytować:

- hasło,

- adres e-mail,

- zdjęcie profilowe.

1. **Interfejs użytkownika:**
   1. Aplikacja posiada responsywny interfejs użytkownika, zaprojektowany przy użyciu silnika szablonów Blade oraz biblioteki CSS – Bootstrap.
   2. Interfejs zawiera różnorodne elementy, takie jak tabele, karty, formularze, przyciski i wiele więcej, tak aby poprawić doświadczenie użytkownika.
2. **Bezpieczeństwo:**
   1. Aplikacja posiada systemy uwierzytelniania i autoryzacji, zapewniając bezpieczeństwo danych użytkowników oraz samej aplikacji przed niepowołanym dostępem.
   2. Wrażliwe operacje są zabezpieczone, dostęp do funkcji administracyjnych jest ograniczony do uprawnionych użytkowników, momentami wymagana jest podwójna autoryzacja przed popełnieniem błędu.

### Główne elementy techniczne projektu

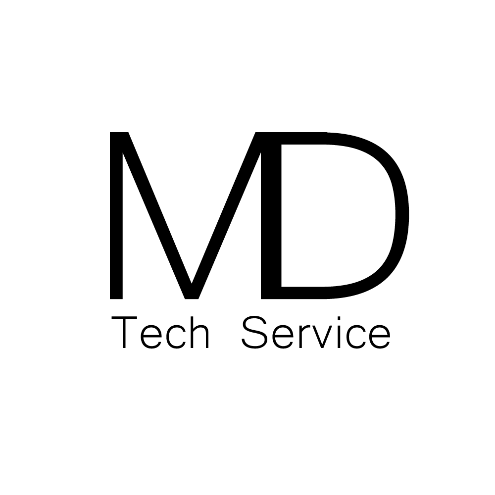
**- Framework:** Laravel 11.x

**- Baza danych:** MySQL

**- Font-end:** szablony Blade, CSS ( Bootstrap oraz własne style ), JavaScript

**- Backend:** PHP, Laravel

Aplikacja „MD Tech Service” została zaprojektowana w taki sposób, aby była wygodna i przyjemna w obsłudze zarówno dla zwykłego użytkownika, jak i dla administratorów oraz pracowników serwisu. Automatyzacja procesów biznesowych, interaktywne elementy podsumowywujące np. finanse z ostatniego miesiąca i wiele więcej pomaga w korzystaniu z aplikacji oraz upraszcza wykonywaną pracę. Przekłada się to również na efektywność działania systemu.



## Narzędzia i technologie

### Framework Laravel

Laravel to popularny framework do tworzenia aplikacji webowych, napisany w PHP przez Taylora Otwella. To zaawansowane narzędzie umożliwia szybkie i efektywne tworzenie skalowalnych aplikacji webowych, wykorzystując nowoczesne techniki programowania oraz wzorce projektowe. Dzięki temu kod jest czytelny i łatwy w utrzymaniu.

W naszym projekcie korzystamy z Laravel w wersji 11.x, która wprowadza liczne usprawnienia i nowe funkcjonalności w porównaniu do wcześniejszych wersji. Laravel jest udostępniany na licencji MIT, co pozwala na swobodne korzystanie, modyfikowanie i rozpowszechnianie frameworka.

Szczegółowe informacje na temat funkcji, klas, interfejsów i sposobu użycia różnych komponentów Laravel można znaleźć w oficjalnej dokumentacji na stronie Laravel Docs.

### Baza danych

Aplikacja wykorzystuje silnik bazodanowy MySQL, który jest oprogramowaniem typu open source, dostępnym na licencji GNU General Public License (GPL). Najnowszą wersję MySQL można pobrać ze strony MySQL Downloads.

##### Konfiguracja połączenia z serwerem bazodanowym:

Aby skonfigurować połączenie z serwerem bazodanowym MySQL, należy edytować plik konfiguracyjny config/database.php oraz plik .env. W tym pliku znajduje się sekcja connections, w której można zdefiniować ustawienia połączenia dla różnych środowisk (np. development, production).

Konfiguracja połączenia z bazą danych MySQL:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

W powyższym kodzie należy ustawić odpowiednie wartości dla parametrów host, port, database, username i password, zgodnie z konfiguracją Twojego serwera bazodanowego MySQL.

##### Frontend

Aplikacja wykorzystuje Blade jako silnik szablonów do generowania widoków HTML. Blade jest prosty w użyciu, a jednocześnie bardzo potężny, pozwala na wykorzystanie pełnych możliwości PHP bez komplikacji związanych z innymi silnikami szablonów. W połączeniu z CSS i JavaScript, zapewnia dynamiczny i responsywny interfejs użytkownika.

##### Middleware

Middleware w Laravelu służy do wykonywania różnych zadań pośrednich w trakcie przetwarzania żądań HTTP. W projekcie używamy middleware do sprawdzania uprawnień użytkowników, na przykład czy użytkownik jest administratorem:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

##### Kontrolery

Kontrolery w Laravelu organizują logikę aplikacji i odpowiadają za obsługę żądań HTTP. Przykładem może być kontroler AdminDashboardController, który zarządza m.in pracownikami, naprawami oraz usługami:

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

##### Modele

Modele w Laravelu odpowiadają za interakcje z bazą danych. Przykładowy model Device może wyglądać tak:

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

##### Walidacja danych

Laravel oferuje prosty sposób walidacji danych za pomocą reguł. Przykład walidacji dla emaila podczas dodawania pracownika lub rejestracji użytkownika:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

##### Seedowanie danych

W celu załadowania przykładowych danych do bazy, używamy seederów. Przykład seedera:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

## Baza danych

##### Diagram ERD

Obraz zawierający tekst, zrzut ekranu, numer, diagram

Opis wygenerowany automatycznie

##### Opis rozwiązań zastosowanych w bazie

MD Tech Service: Serwis napraw gwarancyjnych i pogwarancyjnych dla którego przygotowano bazę nazywa się: mdtechservice.

##### Opis tabel:

**USERS:** W tabeli users przechowywane są informacje o użytkownikach, którzy założyli konto na platformie. Tabela zawiera szczegółowe dane użytkowników, takie jak name, email, password, profile\_photo.

**ADMINS:** W tabeli admins przechowywane są informacje o administratorze, który posiada specjalne funkcje. Tabela zawiera takie dane jak name, email, password oraz profile\_photo.

**EMPLOYEES:** W tabeli employees przechowywane są informacje o pracowniku, który posiada specjalne funkcje. Tabela zawiera takie dane jak name, email, password oraz profile\_photo.

**DEVICES:** W tabeli devices przechowywane są informacje o urządzeniu użytkownika. Tabela zawiera: klucz główny id urządzenia, markę, model, numer seryjny, datę zakupu, datę końca gwarancji która jest wykorzystywana do obliczania adekwatnej ceny usługi serwisowej, kolumnę typu boolean która przechowuje informacje czy urządzenie było zarejestrowane wcześniej w serwisie, typ urządzenia oraz klucz obcy user\_id który służy do identyfikacji użytkownika do którego należy urządzenie.

**REPAIRS:** W tabeli są przechowywane informacje odnośnie napraw urządzeń. Tabela zawiera: klucz główny id naprawy, klucz obcy device\_id do określenia naprawianego urządzenia, klucz obcy employee\_id do określenia pracownika odpowiedzialnego za naprawę, klucz obcy user\_id do określenia użytkownika który zlecił naprawę, status do określenia na jakim etapie jest naprawa, report\_date do wyświetlenia informacji o tym, kiedy została złożona naprawa, completion\_date do wyświetlenia informacji kiedy naprawa została ukończona, user\_notes aby użytkownik mógł opisać dokładniej problem związany z urządzeniem, repairs\_title który przechowuje tytuł naprawy.

**PAYMENTS:** W tabeli są przechowywane płatności użytkowników za naprawy. Tabela zawiera takie informacje jak: id płatności, id użytkownika do którego należy płatność, id naprawy którą trzeba opłacić, kwota, status płatności oraz metoda płatności.

**RATINGS:** W tabeli są przechowywane opinie użytkowników na temat zleconych przez nich napraw. Tabela zawiera: id opinii, id użytkownika wystawiającego opinie, ocena od 1 do 5 oraz krótki komentarz.

**REPAIRNOTES:** W tabeli przechowywane są notatki dotyczące napraw wystawiane przez pracowników aby poinformować użytkownika co zostało zrobione lub co zostanie wykonane. Tabela zawiera takie kolumny jak: id notatki, id naprawy, wiadomość oraz data wysłania.

**TECHSERVICES:** W tabeli są przechowywane usługi które są dostępne na stronie. Przechowują one informacje np. ile orientacyjnie może kosztować wymiana ekranu oraz o ile zostanie odnowiona gwarancja urządzenia.

##### Migracje

Migracje w Laravelu służą do zarządzania strukturą bazy danych. Pozwalają na tworzenie, modyfikowanie i usuwanie tabel oraz innych elementów bazy danych przy użyciu kodu PHP, co ułatwia zarządzanie bazą danych w trakcie rozwoju projektu.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Każda migracja dziedziczy po klasie Migration i zawiera dwie metody: up() i down(). Metoda up() jest wywoływana podczas uruchamiania migracji i służy do utworzenia struktury bazy danych. Metoda down() jest wywoływana podczas wycofywania migracji i służy do usunięcia utworzonych wcześniej elementów.

##### Seedery

Seedery w Laravelu służą do wypełniania tabel w bazie danych przykładowymi danymi. Są one przydatne podczas testowania i rozwijania aplikacji, pozwalając na szybkie załadowanie danych początkowych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Seedery są przydatne w sytuacjach, gdy potrzebujemy przykładowych danych do testowania, inicjalizacji systemu lub tworzenia danych demonstracyjnych. Dzięki nim można łatwo wypełnić bazę danych wartościami początkowymi, co przyspiesza proces rozwoju aplikacji.

##### Opis przykładowej relacji w modelach

Pomiędzy tabelami users ( użytkownicy ) oraz devices ( urządzenia ) w projekcie występuje relacja jeden do wielu ( one-to-many ). Taka relacja umożliwia przypisanie wiele urządzeń do jednego użytkownika, za to jedno urządzenie nie może należeć do wielu użytkowników.

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

W modelu Devices zdefiniowano metodę user(), która ustanawia powiązanie jeden do jednego z użyciem modelu Users. Wykorzystuje ona metodę belongsTo().

W modelu Users zdefiniowano metodę devices(), która ustanawia powiązanie jeden do wielu z użyciem modelu Device. Ta metoda korzysta z metody hasMany().