|  |  |
| --- | --- |
| **AI1** | Dokumentacja projektu |
| **Autor** | Dawid Olko, 125148 |
| **Kierunek, rok** | Informatyka, II rok, st. stacjonarne (3,5-l) |
| **Temat  projektu** | *Platforma wypożyczania filmów VOD* |

**Spis treści**

[**1.** **Wstęp** 4](#_Toc167053109)

[**Zakres Działania Aplikacji** 4](#_Toc167053110)

[**Główne Elementy Techniczne** 5](#_Toc167053111)

[**2.** **Narzędzia i technologie** 6](#_Toc167053112)

[**Framework Laravel** 6](#_Toc167053113)

[**Baza danych** 6](#_Toc167053114)

[**Konfiguracja połączenia z serwerem bazodanowym:** 6](#_Toc167053115)

[**Frontend** 7](#_Toc167053116)

[**Middleware** 7](#_Toc167053117)

[**Kontrolery** 7](#_Toc167053118)

[**Modele** 8](#_Toc167053119)

[**Usługi** 8](#_Toc167053120)

[**Walidacja Danych** 9](#_Toc167053121)

[**Seedowanie Danych** 9](#_Toc167053122)

[**3.** **Baza danych** 10](#_Toc167053123)

[**Diagram ERD** 10](#_Toc167053124)

[**Opis rozwiązań zastosowanych w bazie** 10](#_Toc167053125)

[**MOVIES** 10](#_Toc167053126)

[**USERS** 10](#_Toc167053127)

[**LOANS** 10](#_Toc167053128)

[**LOAN\_MOVIE** 10](#_Toc167053129)

[**OPINIONS** 10](#_Toc167053130)

[**CATEGORIES** 11](#_Toc167053131)

[**Migracje** 11](#_Toc167053132)

[**Seedery** 11](#_Toc167053133)

[**Opis Przykładowej Relacji w Modelach** 12](#_Toc167053134)

[**4.** **GUI** 13](#_Toc167053135)

[**Opis Interfejsu Użytkownika** 13](#_Toc167053136)

[**Główne Widoki Aplikacji** 13](#_Toc167053137)

[**5.** **Uruchomienie aplikacji** 20](#_Toc167053138)

[**Wymagania** 20](#_Toc167053139)

[**Kroki przy pierwszym uruchomieniu** 21](#_Toc167053140)

[**Kroki przy kolejnych uruchomieniach** 22](#_Toc167053141)

[**6.** **Funkcjonalności Aplikacji** 22](#_Toc167053142)

[**Logowanie i Rejestracja** 22](#_Toc167053143)

[**Logowanie:** 22](#_Toc167053144)

[**Rejestracja:** 23](#_Toc167053145)

[**Przykładowi użytkownicy:** 24](#_Toc167053146)

[**Przeglądanie Zasobów** 24](#_Toc167053147)

[**Strona Główna:** 24](#_Toc167053148)

[**Katalog Filmów:** 25](#_Toc167053149)

[**Szczegóły Filmu:** 26](#_Toc167053150)

[**Regulamin Strony:** 27](#_Toc167053151)

[**Zarządzanie przez Administratora** 28](#_Toc167053152)

[**CRUD Filmów:** 28](#_Toc167053153)

[**Zarządzanie Użytkownikami:** 28](#_Toc167053154)

[**Zarządzanie Wypożyczeniami:** 30](#_Toc167053155)

[**Funkcjonalności dla Zalogowanych Użytkowników** 30](#_Toc167053156)

[**Profil Użytkownika:** 30](#_Toc167053157)

[**Koszyk:** 31](#_Toc167053158)

[**Opinie:** 31](#_Toc167053159)

[**Proces Wypożyczania Filmów** 32](#_Toc167053160)

[**Dodawanie do Koszyka:** 32](#_Toc167053161)

[**Finalizacja Zamówienia:** 32](#_Toc167053162)

[**Historia Wypożyczeń:** 33](#_Toc167053163)

[**7.** **Walidacja danych** 33](#_Toc167053164)

[**Walidacja dla różnych typów żądań** 33](#_Toc167053165)

[**8.** **Realizacja wymagań** 43](#_Toc167053166)

[**Wymagania Techniczne** 43](#_Toc167053167)

[**9.** **Podsumowanie** 46](#_Toc167053168)

[**Założenia Projektu** 46](#_Toc167053169)

[**Narzędzia i Technologie** 46](#_Toc167053170)

[**Baza Danych** 46](#_Toc167053171)

[**Interfejs Użytkownika (GUI)** 46](#_Toc167053172)

[**Uruchomienie Aplikacji** 46](#_Toc167053173)

[**Funkcjonalności Aplikacji** 47](#_Toc167053174)

[**Użytkownicy mogą:** 47](#_Toc167053175)

[**Administratorzy mają możliwość:** 47](#_Toc167053176)

[**Walidacja Danych** 47](#_Toc167053177)

[**Wnioski** 47](#_Toc167053178)

# **Wstęp**

Projekt dotyczy stworzenia aplikacji internetowej do wypożyczania filmów VOD (Video on Demand) przy użyciu frameworka Laravel. Aplikacja jest przeznaczona do obsługi małego przedsiębiorstwa zajmującego się wypożyczaniem filmów online. Użytkownicy aplikacji mogą przeglądać katalog dostępnych filmów, wypożyczać je na określony czas oraz zarządzać swoimi zamówieniami i danymi. Aplikacja zapewnia pełną automatyzację procesu wypożyczania filmów, obliczania kosztów, obsługi promocji oraz przyznawania punktów lojalnościowych.

## **Zakres Działania Aplikacji**

Aplikacja obejmuje następujące funkcjonalności:

1. **Rejestracja i logowanie użytkowników**:
   * Tylko zarejestrowani i zalogowani użytkownicy mogą korzystać z pełnych funkcji aplikacji, takich jak wypożyczanie filmów.
   * Nowi użytkownicy mogą samodzielnie się zarejestrować, a po rejestracji uzyskują dostęp do swojego konta.
2. **Zarządzanie filmami i zamówieniami przez administratora**:
   * Administrator ma możliwość dodawania, edytowania oraz usuwania filmów w bazie danych.
   * Administrator zarządza również zamówieniami oraz zmianą danych użytkowników czy ich usunięciem.
3. **Automatyzacja procesów biznesowych**:
   * Aplikacja automatycznie oblicza koszt wypożyczenia filmów, uwzględniając dynamiczne ceny promocyjne.
   * Przyznawanie punktów lojalnościowych za wypożyczenia oraz zarządzanie ich stanem jest również zautomatyzowane.
   * Obsługa zmiany dostępności filmów oraz inne operacje związane z zarządzaniem katalogiem filmów są wykonywane automatycznie.
4. **Funkcjonalności dla użytkowników**:
   * Użytkownicy mogą przeglądać dostępne filmy, korzystając z różnych filtrów i sortowań.
   * Mogą dodawać filmy do koszyka, wypożyczać je oraz zarządzać swoimi zamówieniami.
   * Użytkownicy mają dostęp do swojego profilu, gdzie mogą edytować swoje dane oraz przeglądać historię wypożyczeń.
5. **Interfejs użytkownika**:
   * Aplikacja posiada responsywny interfejs użytkownika, zaprojektowany z użyciem silnika szablonów Blade.
   * Interfejs zawiera różnorodne elementy, takie jak formularze, przyciski, listy, oraz dynamiczne elementy, które poprawiają doświadczenie użytkownika.
6. **Bezpieczeństwo**:
   * Aplikacja implementuje mechanizmy uwierzytelniania i autoryzacji, zapewniając bezpieczeństwo danych użytkowników.
   * Wszystkie wrażliwe operacje są zabezpieczone, a dostęp do funkcji administracyjnych jest ograniczony do uprawnionych użytkowników.

## **Główne Elementy Techniczne**

* **Framework**: Laravel 11.x
* **Baza danych**: MySQL
* **Front-end**: Blade (szablony), CSS, JavaScript
* **Backend**: PHP, Laravel
* **Inne technologie**: Faker (do seedowania danych czy migracji tabel do bazy), Bootstrap (do stylizacji interfejsu)

Aplikacja "Wypożyczalnia Filmów VOD" jest zaprojektowana tak, aby zapewnić wygodę użytkowania zarówno dla klientów, jak i administratorów. Automatyzacja wielu procesów biznesowych znacznie upraszcza zarządzanie zamówieniami i katalogiem filmów, co przekłada się na efektywność działania całego systemu.



**RentalVOD**

# **Narzędzia i technologie**

## **Framework Laravel**

Laravel to popularny framework aplikacji webowych oparty na języku PHP, stworzony przez Taylora Otwella. Jest to zaawansowane narzędzie, które umożliwia szybkie i efektywne tworzenie skalowalnych aplikacji webowych, wykorzystując nowoczesne techniki programowania oraz wzorce projektowe, co ułatwia tworzenie czytelnego i łatwego w utrzymaniu kodu.

W projekcie wykorzystujemy Laravel w wersji 11.x, który oferuje liczne funkcjonalności i usprawnienia w porównaniu do poprzednich wersji. Laravel jest udostępniany na podstawie licencji MIT, co oznacza, że możemy swobodnie korzystać, modyfikować i rozpowszechniać framework. Oficjalna dokumentacja Laravel jest dostępna na stronie Laravel Docs, gdzie można znaleźć szczegółowe informacje na temat funkcji, klas, interfejsów i sposobu użycia różnych komponentów frameworka.

## **Baza danych**

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

### **Konfiguracja połączenia z serwerem bazodanowym:**

Aby skonfigurować połączenie z serwerem bazodanowym MySQL, należy edytować plik konfiguracyjny config/database.php. 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:



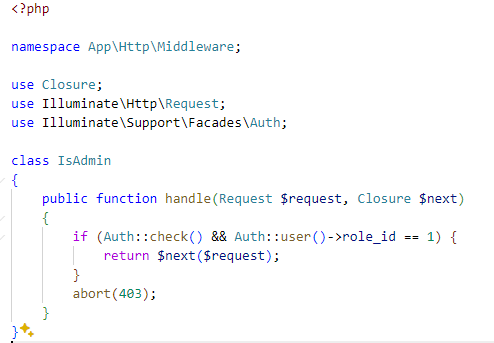
W powyższym kodzie należy ustawić odpowiednie wartości dla parametrów host, port, database, username i password, zgodnie z konfiguracją Twojego serwera bazodanowego MySQL.

## **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:



## **Kontrolery**

Kontrolery w Laravelu organizują logikę aplikacji i odpowiadają za obsługę żądań HTTP. Przykładem może być kontroler AdminController, który zarządza użytkownikami i filmami:



## **Modele**

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



## **Usługi**

Aplikacja korzysta z usług (services), które zawierają logikę biznesową, taką jak obliczanie dynamicznych cen filmów:



## **Walidacja Danych**

Laravel oferuje prosty sposób walidacji danych za pomocą form requestów. Przykład walidacji dla dodawania nowego filmu:



## **Seedowanie Danych**

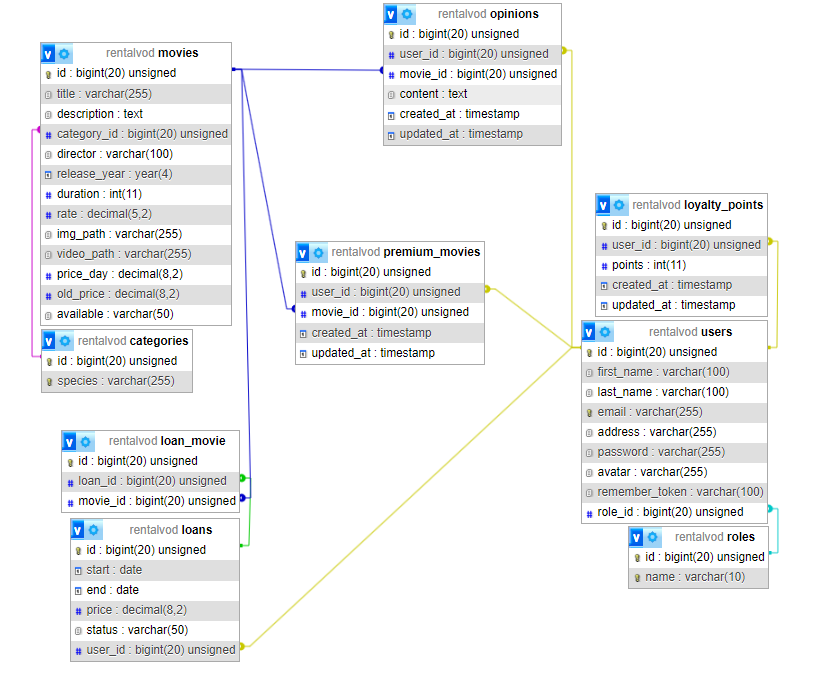
W celu załadowania przykładowych danych do bazy, używamy seederów. Przykład seedera:



Każdy z powyższych elementów technologicznych i narzędzi jest starannie zintegrowany w aplikacji, co zapewnia jej sprawne działanie, łatwość utrzymania oraz możliwość dalszego rozwoju.

# **Baza danych**

## **Diagram ERD**



## **Opis rozwiązań zastosowanych w bazie**

WYPOŻYCZALNIA FILMÓW VOD: Wypożyczalnia, dla której przygotowano bazę, nosi nazwę RentalVOD.

**MOVIES**: W tabeli movies przechowujemy wszystkie nasze filmy z najważniejszymi danymi o nich. Kolumna available wskazuje, czy film jest dostępny do wypożyczenia ("dostępny") czy jest obecnie wypożyczony ("niedostępny"). Dodatkowo tabela zawiera kolumny takie jak title, description, director, release\_year, price\_day, img\_path, category\_id.

**USERS**: W tabeli users przechowujemy informacje o użytkownikach, którzy założyli konto na naszej platformie. Tabela zawiera szczegółowe dane użytkowników, takie jak first\_name, last\_name, email, address, city, password, role\_id (odpowiedzialna za uprawnienia użytkowników), avatar, loyalty\_points.

**LOANS**: W tabeli loans przechowujemy informacje o wypożyczeniach, w tym datę rozpoczęcia (start\_date), datę zakończenia (end\_date), status wypożyczenia (status) oraz cenę wypożyczenia (price). Kolumna user\_id przechowuje identyfikator użytkownika dokonującego wypożyczenia.

**LOAN\_MOVIE**: Jest to tabela pośrednicząca łącząca tabele loans i movies, co umożliwia przechowywanie informacji o tym, które filmy zostały wypożyczone w ramach danego wypożyczenia. Tabela zawiera kolumny loan\_id i movie\_id, które tworzą powiązania między tymi tabelami.

**OPINIONS**: Tabela opinions przechowuje opinie użytkowników o filmach, które wypożyczyli. Tabela zawiera kolumny takie jak content (treść opinii), movie\_id i user\_id, co pozwala na powiązanie opinii z odpowiednim filmem i użytkownikiem.

**CATEGORIES**: Tabela categories przechowuje informacje o kategoriach filmów. Kolumna species zawiera nazwę kategorii.

## **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.



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**

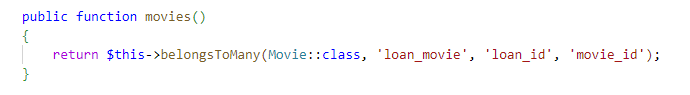
Seedery w Laravelu służą do wypełniania tabel w bazie danych przykładowymi danymi. Są one przydatne podczas testowania i rozwijania aplikacji, pozwalając na szybkie załadowanie danych początkowych.



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 movies (filmy) i loans (wypożyczenia) w projekcie występuje relacja wiele do wielu (many-to-many) za pośrednictwem tabeli loan\_movie. Taka relacja umożliwia, że jeden film może być wypożyczany w ramach wielu wypożyczeń, a jedno wypożyczenie może obejmować wiele filmów.



W modelu Movie zdefiniowano metodę loans(), która ustanawia powiązanie wiele do wielu z modelem Loan. Wykorzystuje ona metodę belongsToMany() i określa nazwę tabeli pośredniczącej loan\_movie. Dodatkowo, ustala nazwy kluczy obcych dla modelu Movie i Loan. Obraz zawierający tekst, linia, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

W modelu Loan również zdefiniowano metodę movies(), która ustanawia powiązanie wiele do wielu z modelem Movie. Ta metoda także korzysta z metody belongsToMany() i określa nazwę tabeli pośredniczącej loan\_movie. Podobnie jak w przypadku modelu Movie, określone są nazwy kluczy obcych dla modelu Loan i Movie.

Dzięki zdefiniowaniu tych relacji, można łatwo uzyskać powiązane filmy dla danego wypożyczenia poprzez dostęp do atrybutu movies na obiekcie Loan. Podobnie, można uzyskać wypożyczenia powiązane z danym filmem poprzez dostęp do atrybutu loans na obiekcie Movie.

W tabeli pośredniczącej loan\_movie przechowywane są klucze obce dla wypożyczeń (loan\_id) i filmów (movie\_id), które tworzą powiązania między nimi.



Relacja wiele do wielu umożliwia elastyczne odwzorowanie przypadków, w których jeden obiekt może być powiązany z wieloma innymi obiektami, a także odwrotnie. Ta relacja jest szczególnie przydatna w przypadku modelowania związku między filmami a wypożyczeniami, gdzie film może być wypożyczany w ramach różnych wypożyczeń, a jedno wypożyczenie może dotyczyć wielu filmów.

# **GUI**

## **Opis Interfejsu Użytkownika**

Interfejs użytkownika (GUI) aplikacji "Wypożyczalnia Filmów VOD" został zaprojektowany z myślą o łatwości użytkowania i estetyce. Aplikacja wykorzystuje **Blade** jako silnik szablonów, co umożliwia dynamiczne generowanie widoków HTML i integrację z backendem Laravel. Całość interfejsu została zbudowana z użyciem nowoczesnych technologii front-endowych, takich jak Bootstrap do stylizacji oraz JavaScript do obsługi dynamicznych elementów interfejsu.

## **Główne Widoki Aplikacji**

**1. Strona Główna (home.blade.php)**

Strona główna jest pierwszym widokiem, który użytkownik widzi po wejściu na stronę aplikacji. Zawiera ona:

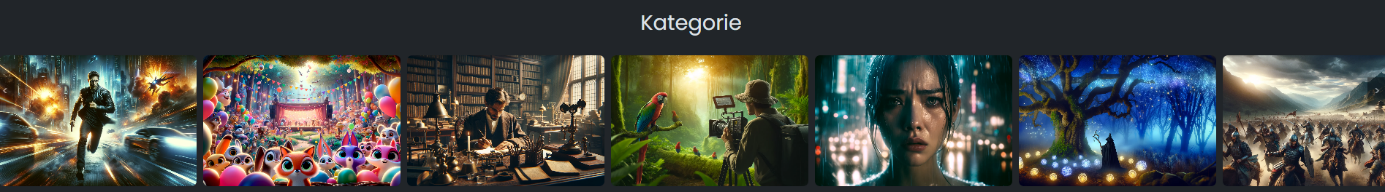
* **Nagłówek**: Zawiera logo aplikacji oraz nawigację umożliwiającą przejście do różnych sekcji strony.



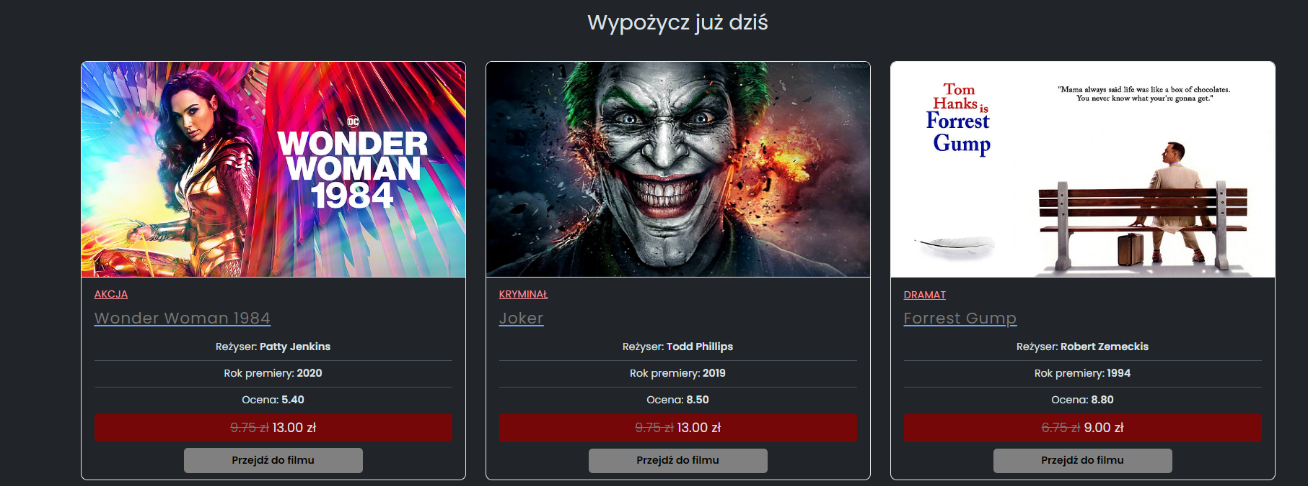
* **Karuzela**: Dynamiczny slider wyświetlający promowane filmy.



* **Kategorie**: Sekcja wyświetlająca różne kategorie filmów dostępnych w wypożyczalni, z możliwością filtrowania filmów według kategorii.



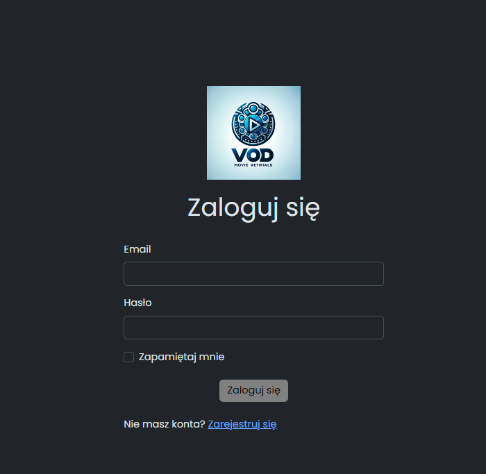
* **Lista filmów**: Prezentacja wybranych filmów, które są aktualnie dostępne do wypożyczenia. Filmy są wyświetlane w postaci kart z obrazkiem, tytułem, kategorią, reżyserem, rokiem premiery i oceną.



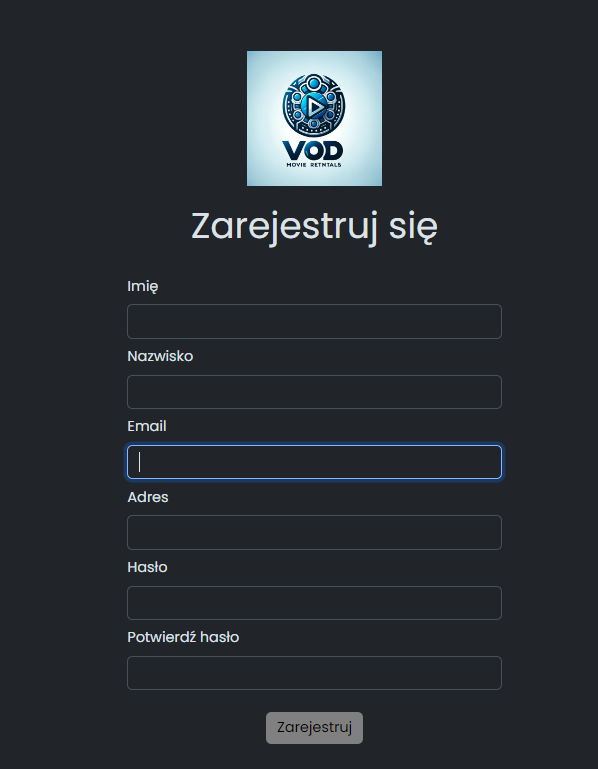
**2. Strona Logowania (login.blade.php) i Rejestracji (register.blade.php)**

Te widoki umożliwiają użytkownikom logowanie się do aplikacji oraz rejestrację nowych kont:

* **Formularz Logowania**: Umożliwia użytkownikom wprowadzenie adresu e-mail i hasła w celu zalogowania się do swojego konta.



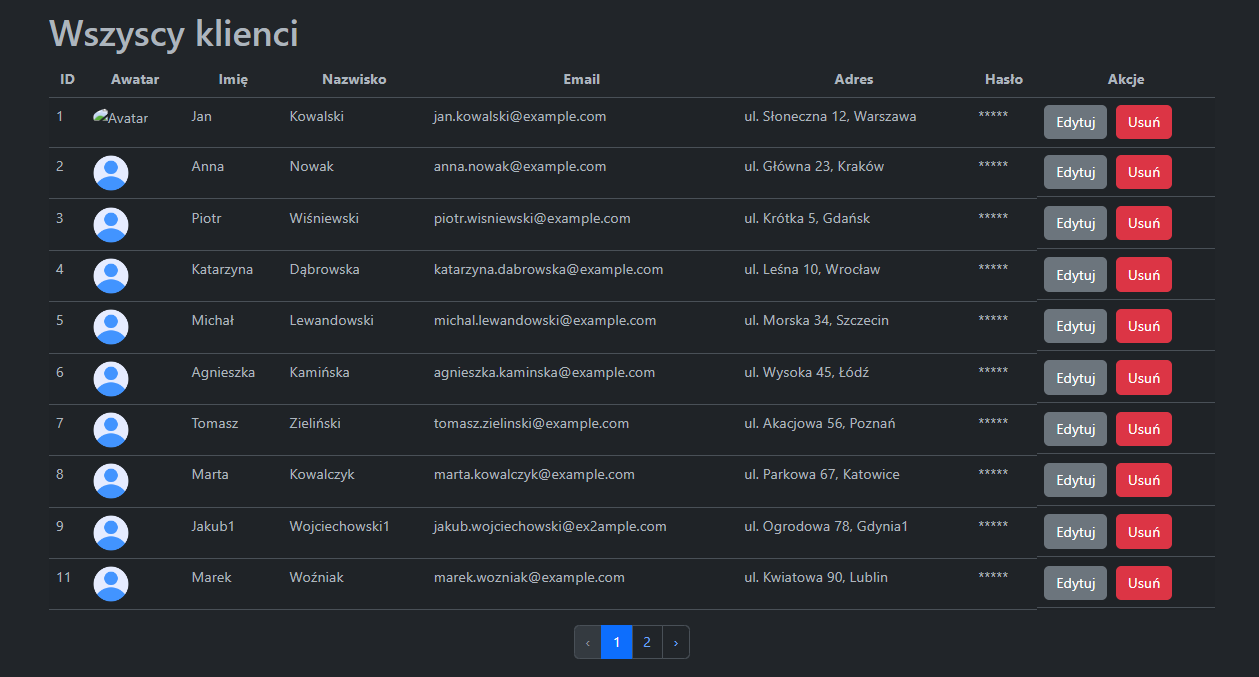
* **Formularz Rejestracji**: Umożliwia nowym użytkownikom utworzenie konta poprzez podanie takich informacji jak imię, nazwisko, adres e-mail, adres, miasto i hasło.



**3. Panel Administratora (admin.blade.php)**

Panel administratora jest dostępny tylko dla użytkowników z rolą administratora i zawiera narzędzia do zarządzania aplikacją:

* **Zarządzanie użytkownikami**: Widok umożliwiający administratorowi przeglądanie, edytowanie i usuwanie kont użytkowników.



* **Zarządzanie filmami**: Widok umożliwiający dodawanie nowych filmów, edytowanie istniejących oraz usuwanie filmów z bazy danych.



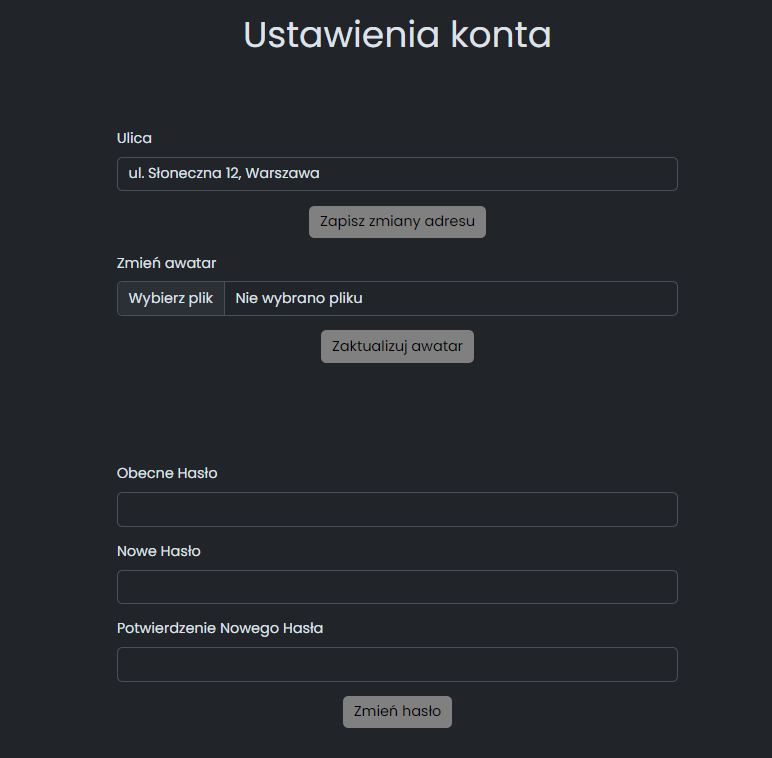
* **Zarządzanie zamówieniami**: Widok pozwalający na przeglądanie i zarządzanie zamówieniami złożonymi przez użytkowników.



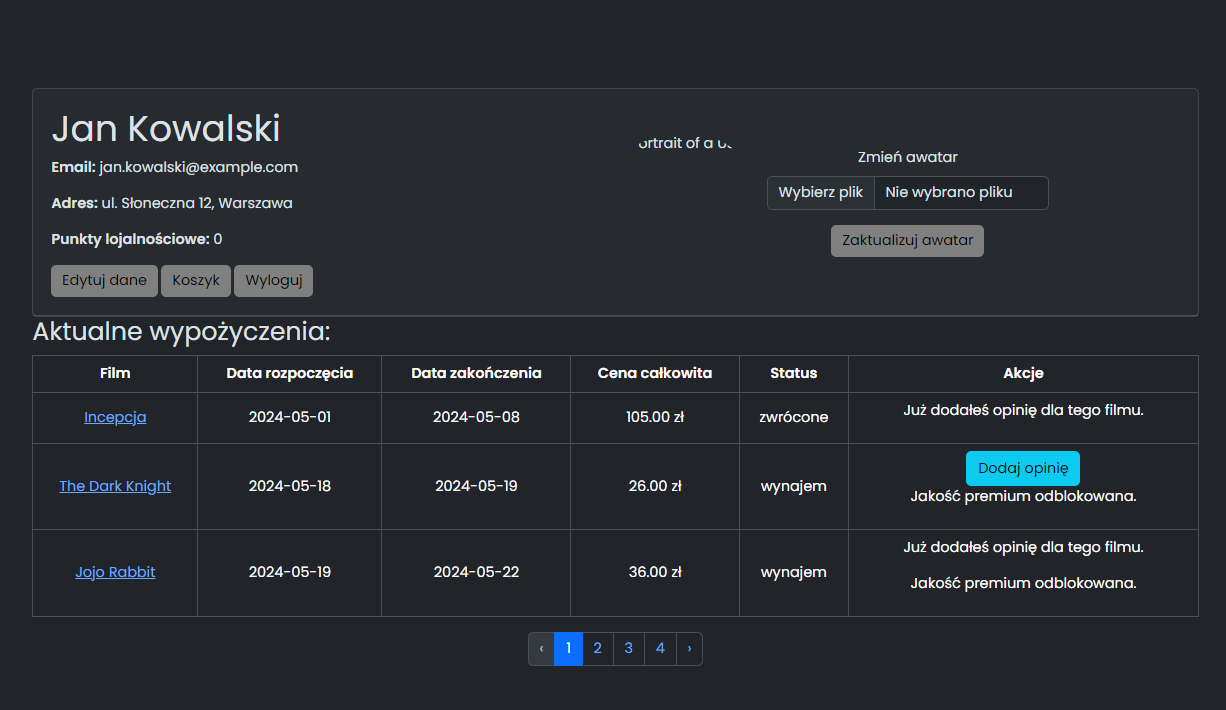
**4. Strona Profilu Użytkownika (profile.blade.php)**

Strona profilu użytkownika pozwala użytkownikom na przeglądanie i edytowanie swoich danych oraz historii wypożyczeń:

* **Dane użytkownika**: Sekcja wyświetlająca podstawowe informacje o użytkowniku oraz umożliwiająca ich edytowanie.



* **Historia wypożyczeń**: Lista filmów wypożyczonych przez użytkownika, z informacjami o dacie wypożyczenia, dacie zwrotu oraz statusie wypożyczenia.



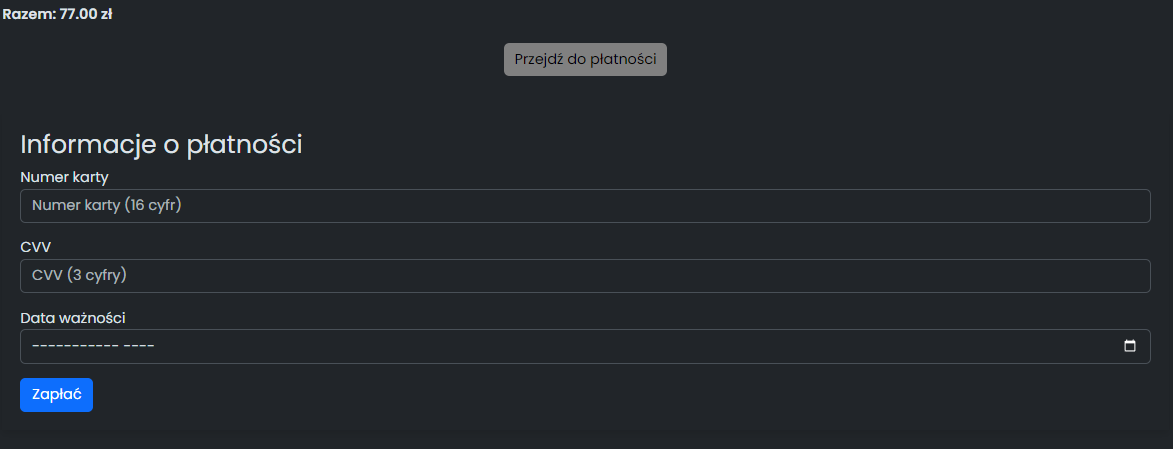
**5. Koszyk (cart.blade.php)**

Widok koszyka umożliwia użytkownikom przeglądanie filmów dodanych do koszyka, zarządzanie datami wypożyczeń oraz finalizowanie zamówienia:

* **Lista filmów w koszyku**: Wyświetla wszystkie filmy dodane do koszyka wraz z ich ceną i wybranymi datami wypożyczenia.

****

* **Przycisk finalizacji zamówienia**: Umożliwia przejście do procesu zamawiania i opłacenia wypożyczenia.

****

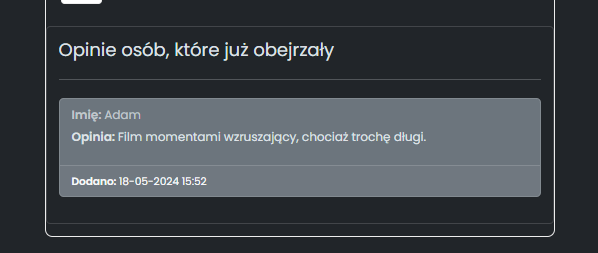
**6. Strona Filmu (show.blade.php)**

Widok szczegółowy filmu wyświetla pełne informacje o wybranym filmie:

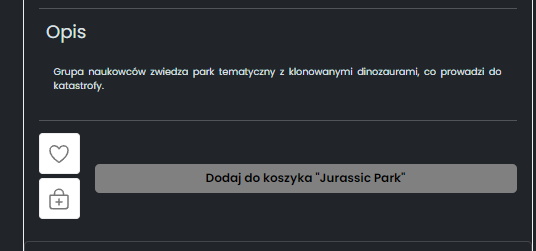
* **Informacje o filmie**: Zawiera tytuł, opis, reżysera, rok premiery, ocenę, kategorię oraz cenę wypożyczenia.

****

* **Opinie użytkowników**: Sekcja wyświetlająca opinie innych użytkowników, którzy wypożyczyli film, wraz z możliwością dodania własnej opinii.

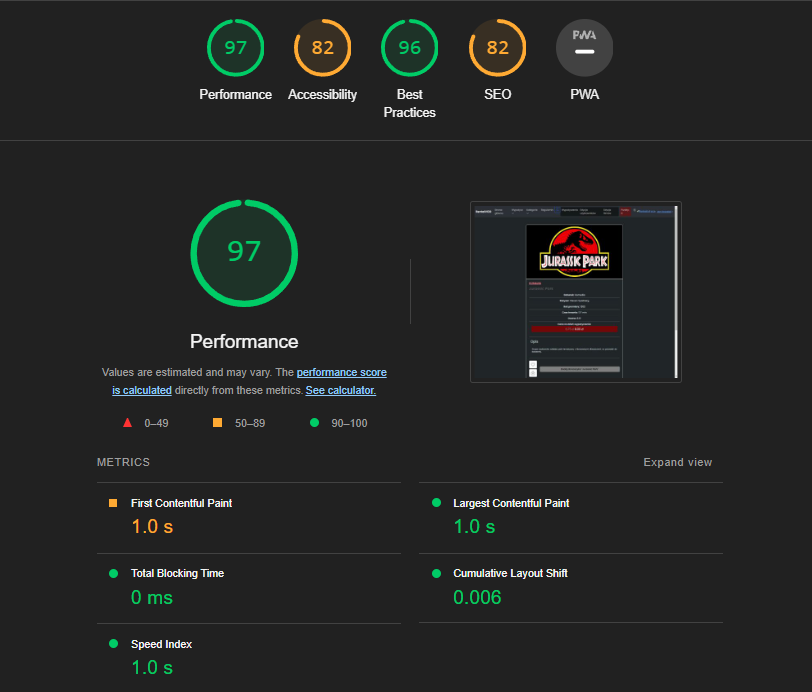
****

* **Przycisk dodania do koszyka**: Umożliwia dodanie filmu do koszyka w celu wypożyczenia.

****

**Stylizacja i Responsywność**

Aplikacja jest w pełni responsywna, co oznacza, że interfejs dostosowuje się do różnych rozmiarów ekranów, od komputerów stacjonarnych po urządzenia mobilne. Stylizacja została wykonana z użyciem **Bootstrap**, co zapewnia spójny i estetyczny wygląd wszystkich elementów interfejsu.



**Dynamiczne Elementy**

* **JavaScript**: Wykorzystany do obsługi dynamicznych elementów, takich jak karuzele, formularze, przyciski akcji i inne interaktywne komponenty.
* **Animacje**: Subtelne animacje dodają dynamiki do interfejsu, poprawiając doświadczenie użytkownika.

# **Uruchomienie aplikacji**

## **Wymagania**

1. **Pakiet XAMPP**: Pakiet XAMPP zawiera serwer Apache, bazę danych MySQL oraz interpreter PHP, które są niezbędne do uruchomienia aplikacji Laravel. Możesz pobrać XAMPP ze strony: [XAMPP Download](https://www.apachefriends.org/download.html).
2. **Composer**: Composer to narzędzie do zarządzania zależnościami w projekcie Laravel. Używane jest do instalacji wszystkich niezbędnych bibliotek i zależności frameworka. Możesz pobrać Composer ze strony: [Composer Download](https://getcomposer.org/Composer-Setup.exe).
3. **Visual Studio Code**: Visual Studio Code to popularne środowisko programistyczne, które ułatwia edycję kodu. Możesz pobrać Visual Studio Code ze strony: [VS Code Download](https://code.visualstudio.com/).

## **Kroki przy pierwszym uruchomieniu**

1. **Uruchom XAMPP**:
   * Po zainstalowaniu XAMPP, uruchom go i włącz moduły MySQL oraz Apache.
   * XAMPP zapewnia graficzny interfejs, który pozwala na łatwe zarządzanie usługami serwera.
2. **Przejdź do folderu projektu i uruchom plik start.bat**:
   * W folderze projektu znajduje się plik **start.bat**, który automatyzuje proces konfiguracji i uruchomienia aplikacji.
   * Otwórz folder projektu i uruchom plik **start.bat** jako administrator, jeśli używasz innego systemu nie szkodzi jest plik start.sh dla macOs i linux(jeśli używasz tych systemów wejdź w drzewie katalogu na poprawną ścieżkę i wpisz w konsoli „bash start.sh”). Skrypt ten wykonuje następujące kroki:

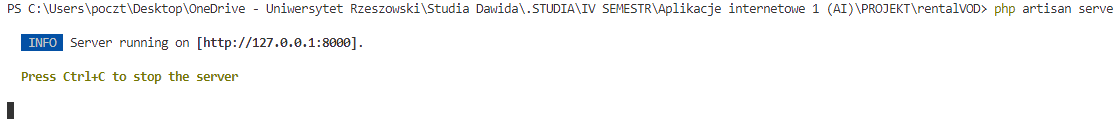


* + Skrypt ten wykonuje następujące operacje:
    - Tworzy bazę danych **rentalVOD**.
    - Kopiuje plik konfiguracyjny **.env.example** do **.env**.
    - Instaluje wszystkie zależności za pomocą **composer install**.
    - Wymaga dodatkowe pakiety takie jak **stripe/stripe-php** oraz **bensampo/laravel-enum**.
    - Generuje klucz aplikacji za pomocą **php artisan key:generate**.
    - Tworzy symbolic link dla przechowywania plików za pomocą **php artisan storage:link**.
    - Migruje bazę danych oraz seedy początkowe za pomocą **php artisan migrate:fresh --seed**.
    - Instaluje pakiet **barryvdh/laravel-ide-helper** dla ulepszenia doświadczenia programistycznego.
    - Uruchamia lokalny serwer deweloperski za pomocą **php artisan serve**.
    - Otwiera przeglądarkę na adresie [http://127.0.0.1:8000](http://127.0.0.1:8000/).
    - Otwiera projekt w Visual Studio Code.

1. **Uruchom wbudowany serwer deweloperski**:
   * Po uruchomieniu pliku **start.bat**, aplikacja powinna być dostępna pod adresem [http://127.0.0.1:8000](http://127.0.0.1:8000/).

## **Kroki przy kolejnych uruchomieniach**

1. **Przejdź do katalogu projektu**:
   * Otwórz terminal lub wiersz poleceń.
   * Przejdź do katalogu, w którym znajduje się projekt Laravel.
2. **Uruchom wbudowany serwer deweloperski**:
   * Uruchom serwer deweloperski za pomocą komendy:



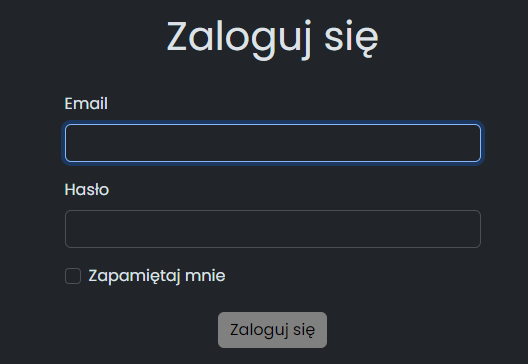
* + Aplikacja powinna być dostępna pod adresem [http://127.0.0.1:8000](http://127.0.0.1:8000/).

# **Funkcjonalności Aplikacji**

## **Logowanie i Rejestracja**

### **Logowanie:**

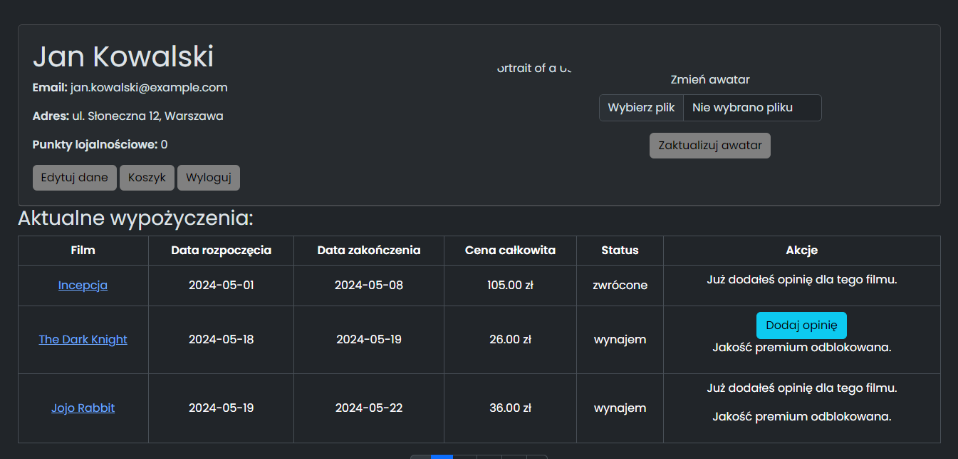
* Na stronie logowania użytkownik zostaje poproszony o wprowadzenie swojego adresu e-mail oraz hasła.



* Wprowadzone dane są walidowane pod kątem poprawności formatu i weryfikowane w bazie danych, aby sprawdzić, czy istnieje konto użytkownika o podanych danych uwierzytelniających.



* Jeśli podane dane są poprawne, użytkownik jest przekierowany do chronionej części aplikacji, gdzie ma dostęp do swojego konta.



* Jeśli podane dane są nieprawidłowe, użytkownik otrzymuje komunikat o błędzie i proszony jest o ponowne wprowadzenie danych logowania.

### **Rejestracja:**

* Na stronie rejestracji użytkownik jest proszony o podanie swojego adresu e-mail, imienia, nazwiska, poprawnego adresu i hasła.



* Podane dane są walidowane pod kątem poprawności formatu, a także sprawdzane pod kątem unikalności, np. czy wybrany adres e-mail jest już zajęty przez innego użytkownika.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

* Podane informacje są zapisywane w bazie danych wraz z danymi logowania użytkownika.
* Po pomyślnej rejestracji, użytkownik jest automatycznie logowany do aplikacji i przekierowany na stronę główną.

## **Przykładowi użytkownicy:**

* **Administrator**:
  + E-mail: jan.kowalski@example.com
  + Hasło: Admin12345&
* **Zwykły użytkownik**:
  + E-mail: anna.nowak@example.com
  + Hasło: password12345&A



## **Przeglądanie Zasobów**

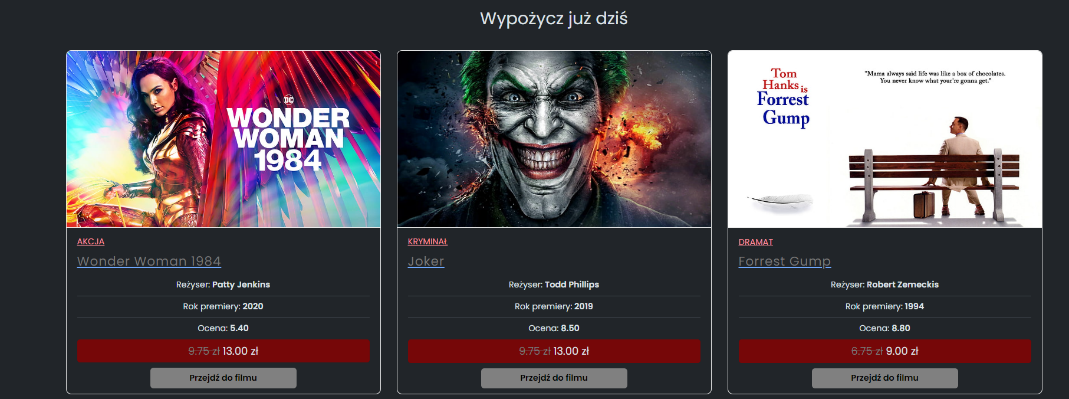
### **Strona Główna:**

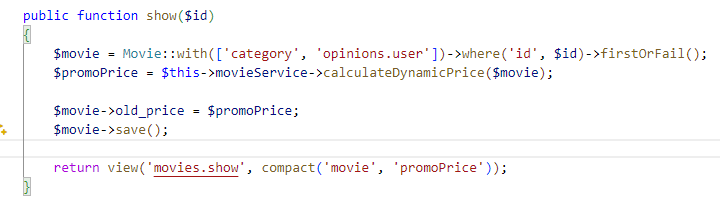
* Strona główna wyświetla promowane filmy w formie karuzeli.

Obraz zawierający tekst, zrzut ekranu, projekt graficzny, Grafika

Opis wygenerowany automatycznie

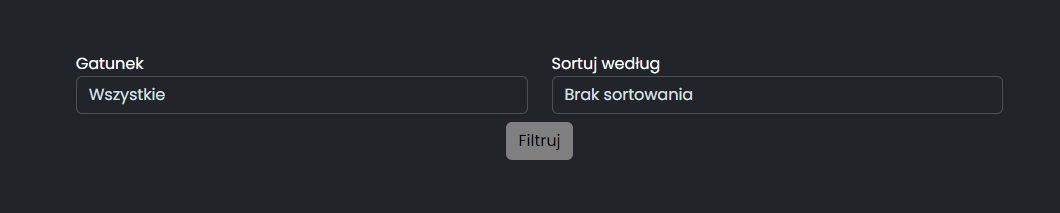
* Zawiera sekcje z popularnymi kategoriami filmów oraz losowo wybranymi filmami dostępnymi do wypożyczenia.





### **Katalog Filmów:**

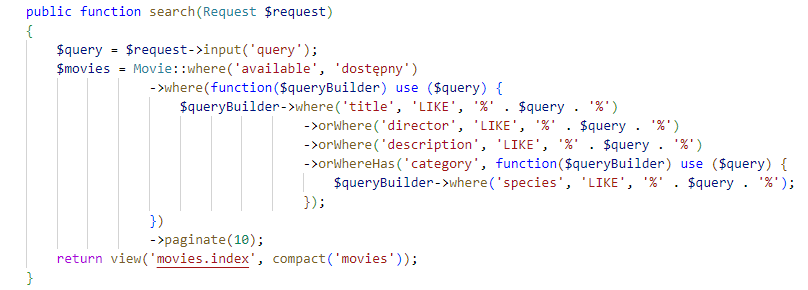
* Użytkownicy mogą przeglądać pełny katalog filmów, korzystając z filtrów i opcji sortowania według kategorii, roku premiery, oceny itp.





* Możliwość przeszukiwania filmów na podstawie słów kluczowych.





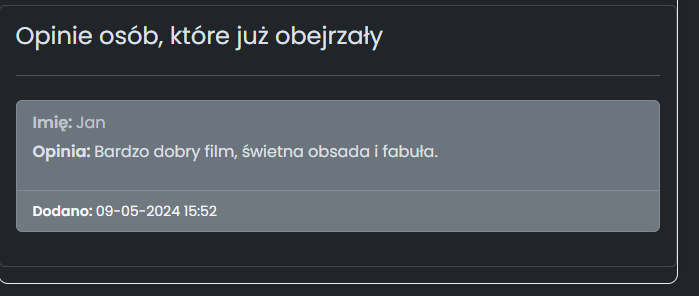
### **Szczegóły Filmu:**

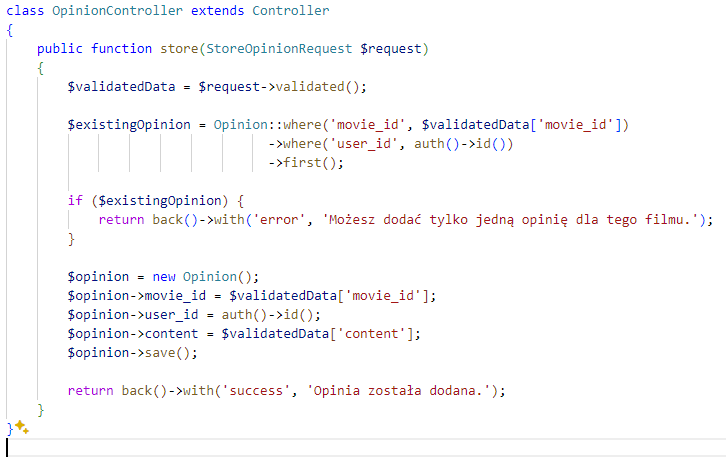
* Każdy film ma stronę szczegółową, która wyświetla pełne informacje o filmie, takie jak opis, reżyser, rok premiery, ocena oraz dostępność.





* Użytkownicy mogą dodawać opinie oraz przeglądać opinie innych użytkowników.





### **Regulamin Strony:**

* Strona z regulaminem, który określa zasady korzystania z serwisu oraz warunki wypożyczeń.



## **Zarządzanie przez Administratora**

### **CRUD Filmów:**

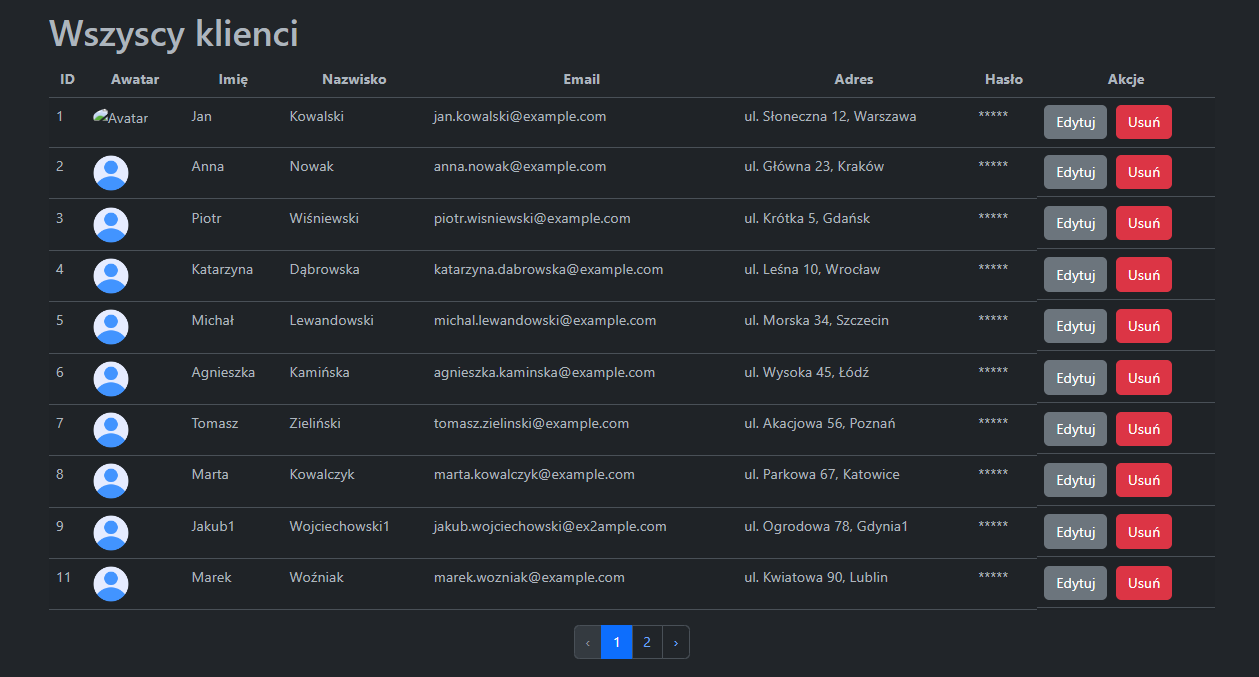
* **Dodawanie filmów**: Administrator może dodawać nowe filmy do bazy danych, wypełniając formularz z informacjami o filmie.
* **Edycja filmów**: Administrator może edytować istniejące filmy, aktualizując ich dane.
* **Usuwanie filmów**: Administrator może usuwać filmy z bazy danych.

****



### **Zarządzanie Użytkownikami:**

* **Dodawanie i edycja użytkowników**: Administrator może dodawać nowych użytkowników oraz edytować dane istniejących użytkowników.
* **Usuwanie użytkowników**: Administrator może usuwać konta użytkowników.

****

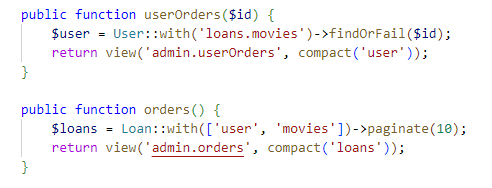




### **Zarządzanie Wypożyczeniami:**

* **Przeglądanie zamówień**: Administrator może przeglądać wszystkie zamówienia złożone przez użytkowników, zarządzać ich statusem oraz szczegółami wypożyczeń.
* **Obsługa zgłoszeń problemów**: Administrator ma dostęp do panelu zgłoszeń, gdzie może przeglądać i rozwiązywać problemy zgłoszone przez użytkowników.

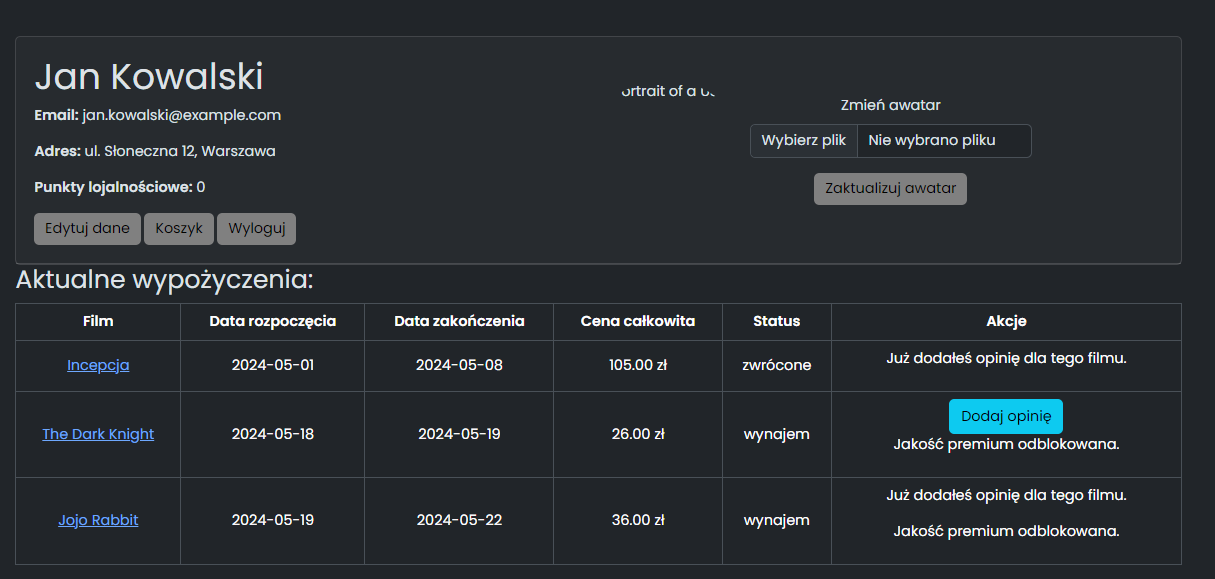
****



## **Funkcjonalności dla Zalogowanych Użytkowników**

### **Profil Użytkownika:**

* Zalogowani użytkownicy mają dostęp do swojego profilu, gdzie mogą przeglądać swoje dane, historię wypożyczeń oraz zarządzać swoimi zamówieniami.
* Możliwość edycji danych osobowych i zmiany hasła.



### **Koszyk:**

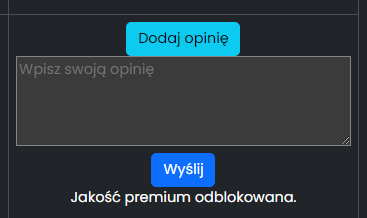
* Użytkownicy mogą dodawać filmy do koszyka, zarządzać datami wypożyczeń oraz finalizować zamówienia.
* Koszyk wyświetla listę filmów, łączną cenę oraz daty wypożyczenia.





### **Opinie:**

* Użytkownicy mogą dodawać opinie do wypożyczonych filmów, które będą widoczne dla innych użytkowników.
* Możliwość przeglądania i zarządzania własnymi opiniami.

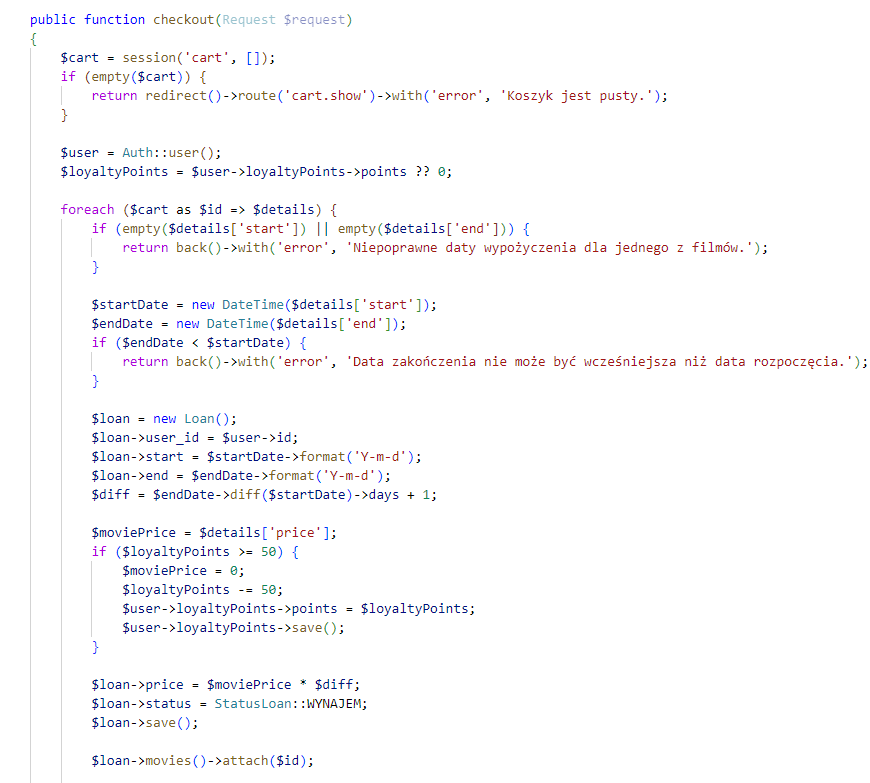


## **Proces Wypożyczania Filmów**

### **Dodawanie do Koszyka:**

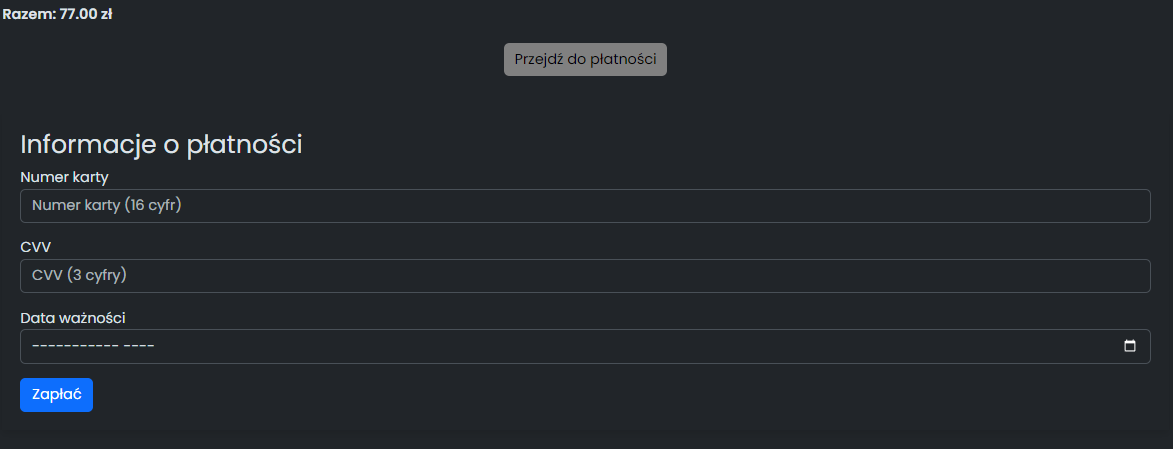
* Użytkownik dodaje filmy do koszyka, wybierając daty wypożyczenia.
* Koszyk wyświetla łączną cenę wypożyczenia na podstawie liczby dni i wybranych filmów.





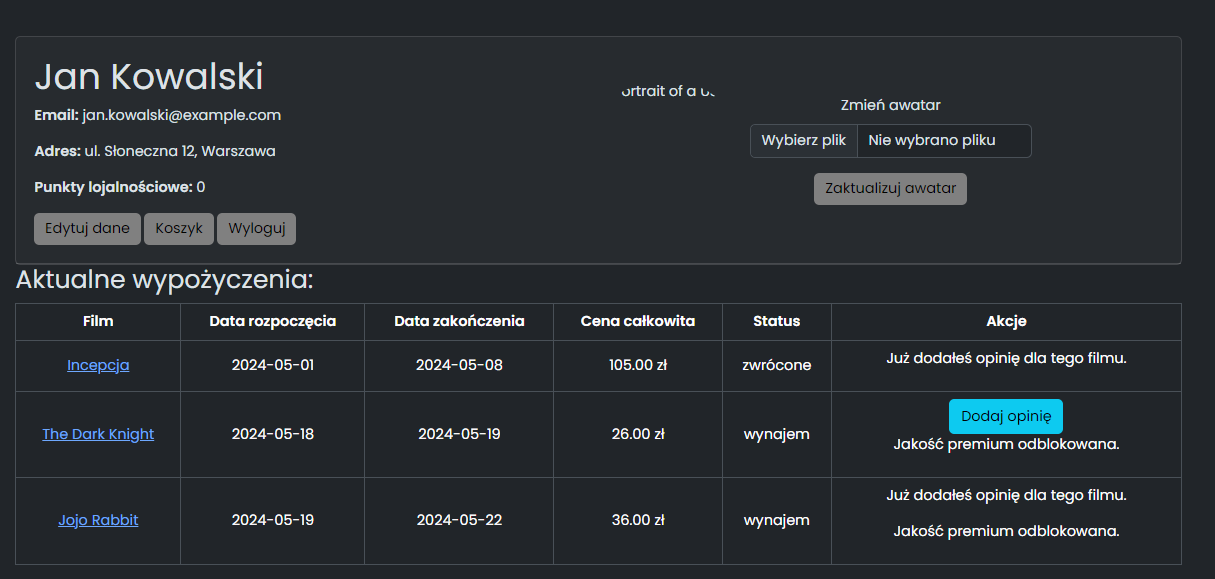
### **Finalizacja Zamówienia:**

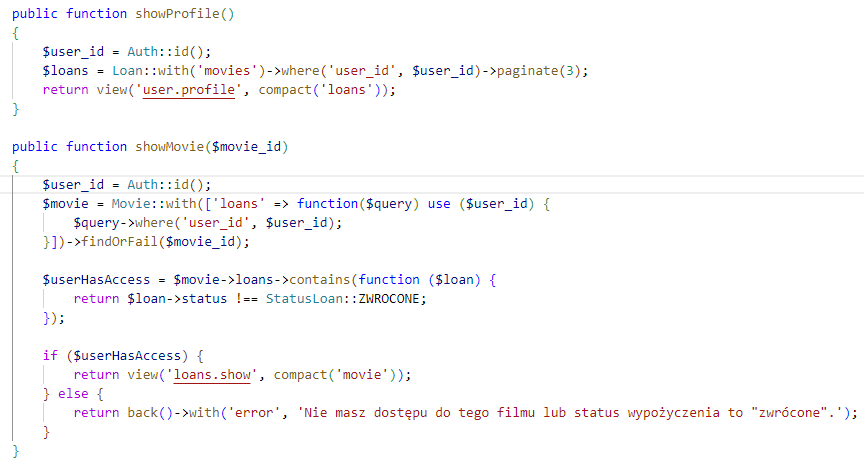
* Po zatwierdzeniu koszyka użytkownik przechodzi do płatności.
* Płatność może być realizowana za pomocą punktów lojalnościowych lub kartą kredytową.



### **Historia Wypożyczeń:**

* Użytkownik może przeglądać historię swoich wypożyczeń, sprawdzać status zamówień oraz szczegóły każdego wypożyczenia.





# **Walidacja danych**

W aplikacji "Wypożyczalnia Filmów VOD" walidacja danych jest kluczowym elementem zapewniającym poprawność i bezpieczeństwo operacji przeprowadzanych przez użytkowników. Laravel oferuje mechanizm walidacji, który pozwala na łatwe definiowanie reguł walidacji dla różnych żądań.

## **Walidacja dla różnych typów żądań**

**1. Dodawanie Kategorii (AddCategoryRequest)**

* **Opis**: Żądanie dodania nowej kategorii filmowej.
* **Reguły walidacji**:
  + **genre**: wymagane, tekst, maksymalnie 255 znaków, unikalne w tabeli **categories** (kolumna **species**).
* **Kod**:



**2. Dodawanie Filmu (AddMovieRequest)**

* **Opis**: Żądanie dodania nowego filmu do bazy danych.
* **Reguły walidacji**:
  + **title**: wymagane, tekst, maksymalnie 255 znaków.
  + **description**: wymagane, tekst.
  + **category\_id**: wymagane, liczba całkowita.
  + **director**: wymagane, tekst, maksymalnie 255 znaków.
  + **release\_year**: wymagane, liczba całkowita, minimalnie 0.
  + **duration**: wymagane, tekst, minimalnie 0, format liczbowy z opcjonalną częścią dziesiętną.
  + **rate**: wymagane, liczba, minimalnie 0.
  + **video\_path**: wymagane, tekst.
  + **price\_day**: wymagane, liczba, minimalnie 0.
  + **available**: wymagane, jeden z wartości: **dostępny**, **niedostępny**.
  + **img\_path**: opcjonalne, obrazek, formaty: jpeg, png, jpg, gif, maksymalnie 2048 kB.
* **Kod**:



**3. Autoryzacja (AuthenticateRequest)**

* **Opis**: Żądanie logowania użytkownika.
* **Reguły walidacji**:
  + **email**: wymagane, format email, maksymalnie 255 znaków.
  + **password**: wymagane, tekst, minimalnie 8 znaków, zawiera małe i duże litery, cyfry oraz znak specjalny.
* **Kod**:



**4. Zmiana Hasła (ChangePasswordRequest)**

* **Opis**: Żądanie zmiany hasła użytkownika.
* **Reguły walidacji**:
  + **current\_password**: wymagane.
  + **new\_password**: wymagane, tekst, minimalnie 8 znaków, zawiera małe i duże litery, cyfry oraz znak specjalny, potwierdzenie hasła.
* **Komunikaty błędów**:
  + **new\_password.min**: "Hasło musi zawierać co najmniej 8 znaków."
  + **new\_password.confirmed**: "Potwierdzenie hasła nie zgadza się."
  + **new\_password.regex**: "Hasło musi zawierać co najmniej jedną wielką literę, jedną małą literę, jedną cyfrę i jeden znak specjalny."
* **Kod**:



**5. Rejestracja (RegisterRequest)**

* **Opis**: Żądanie rejestracji nowego użytkownika.
* **Reguły walidacji**:
  + **first\_name**: wymagane, tekst, maksymalnie 100 znaków, tylko litery.
  + **last\_name**: wymagane, tekst, maksymalnie 100 znaków, tylko litery.
  + **email**: wymagane, format email, maksymalnie 255 znaków, unikalne w tabeli **users**.
  + **password**: wymagane, potwierdzenie hasła, minimalnie 8 znaków, zawiera małe i duże litery, cyfry oraz znak specjalny.
  + **address**: wymagane, tekst, maksymalnie 255 znaków.
* **Komunikaty błędów**:
  + **first\_name.required**: "Imię jest wymagane."
  + **last\_name.required**: "Nazwisko jest wymagane."
  + **email.required**: "Adres email jest wymagany."
  + **email.email**: "Proszę podać prawidłowy adres email."
  + **email.unique**: "Podany adres email jest już używany."
  + **password.required**: "Hasło jest wymagane."
  + **password.confirmed**: "Hasła nie są identyczne."
  + **password.min**: "Hasło musi zawierać co najmniej 8 znaków."
  + **password.regex**: "Hasło musi zawierać małe i duże litery, cyfry oraz znak specjalny."
  + **address.required**: "Adres jest wymagany."
* **Kod**:



**6. Dodawanie Opinii (StoreOpinionRequest)**

* **Opis**: Żądanie dodania nowej opinii o filmie.
* **Reguły walidacji**:
  + **content**: wymagane, minimalnie 1 znak.
  + **movie\_id**: wymagane, istnieje w tabeli **movies**.
* **Komunikaty błędów**:
  + **content.required**: "Treść opinii jest wymagana."
  + **content.min**: "Treść opinii musi mieć co najmniej 1 znak."
  + **movie\_id.required**: "ID filmu jest wymagane."
  + **movie\_id.exists**: "Podany film nie istnieje."
* **Kod**:



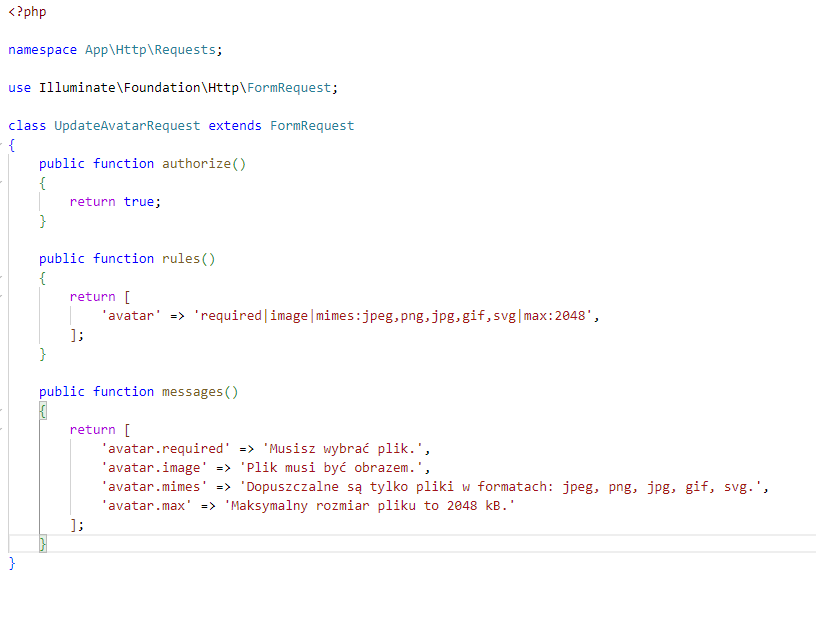
**7. Aktualizacja Adresu (UpdateAddressRequest)**

* **Opis**: Żądanie aktualizacji adresu użytkownika.
* **Reguły walidacji**:
  + **address**: wymagane, tekst, maksymalnie 255 znaków, może zawierać tylko litery, cyfry, spacje oraz znaki ,.-.
* **Komunikaty błędów**:
  + **address.required**: "Adres jest wymagany."
  + **address.regex**: "Adres może zawierać tylko litery, cyfry, spacje oraz znaki ,.-."
* **Kod**:



**8. Aktualizacja Awatara (UpdateAvatarRequest)**

* **Opis**: Żądanie aktualizacji awatara użytkownika.
* **Reguły walidacji**:
  + **avatar**: wymagane, obrazek, formaty: jpeg, png, jpg, gif, svg, maksymalnie 2048 kB.
* **Komunikaty błędów**:
  + **avatar.required**: "Musisz wybrać plik."
  + **avatar.image**: "Plik musi być obrazem."
  + **avatar.mimes**: "Dopuszczalne są tylko pliki w formatach: jpeg, png, jpg, gif, svg."
  + **avatar.max**: "Maksymalny rozmiar pliku to 2048 kB."
* **Kod**:



**9. Aktualizacja Koszyka (UpdateCartRequest)**

* **Opis**: Żądanie aktualizacji dat wypożyczenia filmów w koszyku.
* **Reguły walidacji**:
  + **start**: wymagane, data.
  + **end**: wymagane, data, musi być późniejsza lub równa dacie **start**.
* **Komunikaty błędów**:
  + **start.required**: "Data rozpoczęcia jest wymagana."
  + **start.date**: "Data rozpoczęcia musi być prawidłową datą."
  + **end.required**: "Data zakończenia jest wymagana."
  + **end.date**: "Data zakończenia musi być prawidłową datą."
  + **end.after\_or\_equal**: "Data zakończenia nie może być wcześniejsza niż data rozpoczęcia."
* **Kod**:



**10. Aktualizacja Filmu (UpdateMovieRequest)**

* **Opis**: Żądanie aktualizacji danych filmu.
* **Reguły walidacji**:
  + **title**: wymagane, tekst, maksymalnie 255 znaków.
  + **description**: wymagane, tekst.
  + **director**: wymagane, tekst, maksymalnie 255 znaków.
  + **release\_year**: wymagane, liczba całkowita, minimalnie 0.
  + **duration**: wymagane, tekst, minimalnie 0, format liczbowy z opcjonalną częścią dziesiętną.
  + **rate**: wymagane, liczba, minimalnie 0.
  + **video\_path**: wymagane, tekst.
  + **price\_day**: wymagane, liczba, minimalnie 0.
  + **available**: wymagane, jeden z wartości: **dostępny**, **niedostępny**.
  + **img\_path**: opcjonalne, obrazek, formaty: jpeg, png, jpg, gif, maksymalnie 2048 kB.
* **Kod**:



**11. Aktualizacja Użytkownika (UpdateUserRequest)**

* **Opis**: Żądanie aktualizacji danych użytkownika.
* **Reguły walidacji**:
  + **first\_name**: wymagane, tekst, maksymalnie 100 znaków.
  + **last\_name**: wymagane, tekst, maksymalnie 100 znaków.
  + **email**: wymagane, format email, maksymalnie 255 znaków, unikalne w tabeli **users** z wyjątkiem aktualizowanego użytkownika.
  + **address**: wymagane, tekst, maksymalnie 255 znaków.
* **Kod**:



**Podsumowanie**

Mechanizmy walidacji w aplikacji "Wypożyczalnia Filmów VOD" zapewniają poprawność danych wprowadzanych przez użytkowników na różnych etapach korzystania z aplikacji. Dzięki temu aplikacja działa sprawnie i bezpiecznie, minimalizując ryzyko błędów oraz zapewniając pozytywne doświadczenie użytkownikom.

# **Realizacja wymagań**

Aplikacja "Wypożyczalnia Filmów VOD" spełnia szereg wymagań dotyczących funkcjonalności i jakości, które są niezbędne do osiągnięcia określonej oceny. Poniżej znajduje się szczegółowy opis każdego z wymagań oraz sposób, w jaki zostały zrealizowane w projekcie.

## **Wymagania Techniczne**

1. **Laravel >= 11.x**
   * Aplikacja została zbudowana w oparciu o najnowszą wersję frameworka Laravel 11.x, co zapewnia dostęp do najnowszych funkcji i zabezpieczeń.
2. **Minimum dwie, powiązane ze sobą tabele**
   * Aplikacja posiada wiele tabel w bazie danych, które są ze sobą powiązane. Na przykład, tabele **users** i **loans** są powiązane relacją jeden-do-wielu.
3. **Minimum trzy, powiązane ze sobą tabele**
   * Oprócz powyższego, aplikacja zawiera również powiązania między tabelami **users**, **loans**, **movies**, które tworzą skomplikowane relacje.
4. **Minimum cztery, powiązane ze sobą tabele**
   * Aplikacja wykorzystuje jeszcze więcej tabel, takich jak **users**, **loans**, **movies**, **categories**, co pokazuje elastyczność i złożoność relacji.
5. **Jedna z tabel jest tabelą transakcji/rezerwacji/wypożyczeń itp.**
   * Tabela **loans** służy do przechowywania informacji o wypożyczeniach filmów, w tym daty rozpoczęcia i zakończenia wypożyczenia, a także powiązane filmy i użytkowników.
6. **Występują więzy integralności encji**
   * Integralność danych jest zapewniona przez klucze obce w tabelach, takie jak **user\_id** w tabeli **loans** czy **movie\_id** w tabeli **loan\_movie**.
7. **Seed bazy przykładowymi danymi**
   * Baza danych jest wypełniana przykładowymi danymi przy użyciu seederów, co pozwala na szybkie uruchomienie aplikacji z gotowymi danymi testowymi.
8. **Utworzenie użytkownika/ów pełniących rolę administratora, zarządzającego zasobami aplikacji**
   * Aplikacja pozwala na tworzenie użytkowników z rolą administratora, którzy mogą zarządzać filmami, użytkownikami oraz zamówieniami.
9. **Administratorzy zarządzają użytkownikami**
   * Administratorzy mają dostęp do funkcji zarządzania użytkownikami, w tym dodawania, edycji oraz usuwania kont użytkowników.
10. **Możliwość przeglądania ogólnie dostępnych zasobów**
    * Użytkownicy mogą przeglądać dostępne filmy bez konieczności logowania się do aplikacji.
11. **Paginacja najliczniej występujących zasobów**
    * Aplikacja wykorzystuje mechanizm paginacji do wyświetlania listy filmów oraz użytkowników w widokach administracyjnych.
12. **Brak użytkowników aplikacji (np. klienci, zawodnicy)**
    * Aplikacja posiada funkcjonalność pozwalającą na zarządzanie kontami użytkowników przez administratora, co pozwala na łatwe dodawanie i edytowanie użytkowników.
13. **Użytkownicy aplikacji, mogą się zalogować, zarządzać swoimi zasobami, zarządzać swoimi danymi**
    * Użytkownicy mogą się logować do aplikacji, zarządzać swoimi danymi osobowymi oraz przeglądać historię wypożyczeń.
14. **Nowi użytkownicy mogą się samodzielnie zarejestrować**
    * Aplikacja oferuje formularz rejestracyjny, który umożliwia nowym użytkownikom samodzielne założenie konta.
15. **W aplikacji dostępne są funkcjonalności typu CRUD, przeprowadzany jest pełny CRUD ogólnie dostępnego zasobu**
    * Administratorzy mają pełen dostęp do operacji CRUD na zasobach, takich jak filmy, użytkownicy i zamówienia.
16. **Występują obrazki zasobów oraz użytkowniki nimi zarządza**
    * Filmy w aplikacji posiadają przypisane obrazki, które są zarządzane przez administratorów przy dodawaniu lub edycji filmów.
17. **W aplikacji dostępne są funkcjonalności oprogramowane logiką biznesową**
    * Logika biznesowa, taka jak obliczanie kosztów wypożyczeń, naliczanie punktów lojalnościowych oraz zarządzanie dostępnością filmów, jest zaimplementowana i automatyzuje wiele procesów.
18. **Nie występuje problem N+1**
    * Problem N+1 został rozwiązany dzięki odpowiedniemu stosowaniu eager loadingu w zapytaniach bazodanowych.
19. **Walidacja danych po stronie backendu**
    * Aplikacja wykorzystuje Laravel Requests do walidacji danych wprowadzanych przez użytkowników, co zapewnia poprawność i integralność danych.
20. **Walidacja danych po stronie frontend**
    * Formularze w aplikacji posiadają wbudowaną walidację po stronie klienta, co zwiększa responsywność i poprawność wprowadzanych danych.
21. **Aplikacja posiada schludny responsywny GUI, spójny dla całej aplikacji**
    * Interfejs użytkownika jest zaprojektowany zgodnie z zasadami responsywnego designu, zapewniając spójne doświadczenie na różnych urządzeniach.
22. **GUI aplikacji zawiera inne kontrolki interfejsu (nie tylko przyciski, pola tekstowe) np. slidery, pickery, toggle**
    * Interfejs użytkownika zawiera różnorodne elementy interaktywne, takie jak slidery, pickery dat, przyciski toggle.
23. **GUI aplikacji prezentuje wyliczone informacje np. statystyki, wykresy**
    * Aplikacja prezentuje dane statystyczne, takie jak ilość wypożyczeń, najczęściej wypożyczane filmy, itp.
24. **W kodach błędów HTTP zwracane są widoki z informacją o błędzie**
    * W przypadku wystąpienia błędów aplikacja zwraca odpowiednie widoki błędów z informacjami dla użytkownika.
25. **Przygotowanie skryptu startowego „stawiającego” bazę i aplikację**
    * Aplikacja zawiera skrypt startowy **start.bat**, który automatyzuje proces konfiguracji bazy danych, instalacji zależności i uruchomienia aplikacji.

# **Podsumowanie**

Projekt "Wypożyczalnia Filmów VOD" jest kompleksową aplikacją internetową, zaprojektowaną w celu umożliwienia użytkownikom wypożyczania filmów online. Aplikacja została stworzona przy użyciu najnowszych technologii webowych, zapewniając zarówno funkcjonalność, jak i bezpieczeństwo oraz łatwość użytkowania.

## **Założenia Projektu**

Aplikacja ma na celu obsługę procesu wypożyczania filmów na żądanie (VOD), co obejmuje rejestrację i logowanie użytkowników, przeglądanie dostępnych filmów, zarządzanie wypożyczeniami oraz recenzowanie filmów. Administratorzy mają pełny dostęp do zarządzania treściami i użytkownikami, co pozwala na efektywne utrzymanie platformy.

## **Narzędzia i Technologie**

Do realizacji projektu wykorzystano najnowszą wersję frameworka Laravel 11.x, który zapewnia solidne podstawy do budowy aplikacji webowych. Baza danych została zrealizowana przy użyciu MySQL, z pełnym wsparciem dla migracji i seederów, co ułatwia zarządzanie strukturą danych i wstępnym zasilaniem bazy. Interfejs użytkownika został zaprojektowany zgodnie z zasadami responsywnego designu, dzięki czemu aplikacja jest dostępna na różnych urządzeniach.

## **Baza Danych**

Projekt bazy danych został starannie zaplanowany, aby odzwierciedlać wszystkie niezbędne relacje między danymi. Zostały stworzone tabele takie jak **users**, **movies**, **loans**, **categories**, **opinions**, które są ze sobą powiązane za pomocą kluczy obcych. Integralność danych jest zapewniona poprzez odpowiednie więzy, a dane są wypełniane przykładowymi wartościami za pomocą seederów.

## **Interfejs Użytkownika (GUI)**

Interfejs aplikacji został zaprojektowany z myślą o intuicyjności i łatwości obsługi. Użytkownicy mogą przeglądać katalog filmów, dodawać filmy do koszyka, finalizować zamówienia oraz zarządzać swoimi danymi osobowymi. Administratorzy mają dostęp do panelu zarządzania, który pozwala na dodawanie, edytowanie i usuwanie filmów oraz użytkowników.

## **Uruchomienie Aplikacji**

Proces uruchamiania aplikacji został maksymalnie uproszczony. Wymaga on zainstalowania odpowiednich narzędzi takich jak XAMPP, Composer oraz Visual Studio Code. Skrypt startowy automatyzuje proces konfiguracji, tworzenia bazy danych, instalacji zależności oraz uruchomienia serwera aplikacji.

## **Funkcjonalności Aplikacji**

Aplikacja oferuje szeroki zakres funkcjonalności zarówno dla zwykłych użytkowników, jak i administratorów.

### **Użytkownicy mogą:**

* Rejestrować się i logować,
* Przeglądać katalog filmów,
* Dodawać filmy do koszyka,
* Finalizować zamówienia,
* Zarządzać swoimi danymi oraz historią wypożyczeń.

### **Administratorzy mają możliwość:**

* Zarządzania filmami (dodawanie, edycja, usuwanie),
* Zarządzania użytkownikami (dodawanie, edycja, usuwanie),
* Przeglądania i zarządzania zamówieniami,
* Obsługi zgłoszeń problemów od użytkowników.

## **Walidacja Danych**

Walidacja danych jest kluczowym elementem zapewniającym poprawność i bezpieczeństwo aplikacji. Zaimplementowane reguły walidacji zarówno po stronie serwera, jak i klienta, chronią przed wprowadzeniem nieprawidłowych danych. Laravel Requests umożliwiają definiowanie kompleksowych reguł walidacji dla różnych operacji, takich jak rejestracja, logowanie, dodawanie filmów czy aktualizacja danych użytkownika.

## **Wnioski**

Projekt "Wypożyczalnia Filmów VOD" to zaawansowana aplikacja webowa, która integruje nowoczesne technologie z intuicyjnym interfejsem użytkownika. Aplikacja jest w pełni funkcjonalna, bezpieczna i łatwa w obsłudze, co sprawia, że jest gotowa do wdrożenia i użytkowania. Dzięki szczegółowo zaplanowanej architekturze i solidnemu fundamentowi technologicznemu, projekt ten stanowi doskonały przykład nowoczesnej aplikacji internetowej, spełniającej wszelkie wymagania współczesnych użytkowników oraz administratorów.