**Dokumentacja Techniczna**

**Tytuł Projektu:**

[**FilmTicketShop**](https://github.com/Rozoom7721/FilmTicketShop)

**Krótki Opis Działania Projektu:**

Jest to aplikacja internetowa do kupowania biletów do kin na wybrany film.  
Aplikacja pozwala na dodawanie biletów do koszyka a następnie kupowanie ich.  
Administrator może dodawać nowe filmy, kina, aktorów i reżyserów.

**Autor Projektu:**

Mateusz Rozumek

**Specyfikacja Wykorzystanych Technologii:**

* Wersja .NET: 8
* Baza danych: SqlServer
* Inne technologie: Bootstrap

**Instrukcje Pierwszego Uruchomienia Projektu:**

1. Skonfiguruj połączenie do bazy danych w pliku appsettings.json.
2. Wykonaj polecenie Update-Database w konsoli menedżera pakietów
3. Uruchom aplikację za pomocą Visual Studio lub polecenia dotnet run.
4. Dane do logowania jako administrator : Email: [admin@admin.pl](mailto:admin@admin.pl) Hasło:@Admin123, dane do logowania dla użytkownika: Email:user@user.pl Hasło:@User12345

**Opis Struktury Projektu:**

**Model**

1. **Model: Actor**
   * **Opis:** Model posiadający atrybuty aktora
   * **Pola**
     + Id: int, Unikalny
     + **ProfilePictureURl:** string, wartość wymagana
     + **FullName:** string, wartość wymagana, długość między 3 a 50 znaków
     + **Bio:** string, wartość wymagana
2. **Model: Actor\_Movie**
   * **Opis:** Model dodający relacje między filmami a aktorami
   * **Pola**
     + **MovieId**: int
     + **Movie**: **Movie**
     + **ActorId**: int
     + **Actor**: Actor
3. **Model: Cinema**
   * **Opis:** Model posiadający atrybuty kina
   * **Pola**
     + Id: int, Unikalny
     + **LogoURL:** string, wartość wymagana
     + **Name:** string, wartość wymagana, długość między 3 a 50 znaków
     + **Description:** string, wartość wymagana
     + **Movies:** List<Movie>
4. **Model: Producer**
   * **Opis:** Model posiadający atrybuty reżysera
   * **Pola**
     + Id: int, Unikalny
     + **ProfilePictureURl:** string, wartość wymagana
     + **FullName:** string, wartość wymagana, długość między 3 a 50 znaków
     + **Bio:** string, wartość wymagana
5. **Model: Movie**
   * **Opis:** Model posiadający atrybuty filmu
   * **Pola**
     + Id: int, Unikalny
     + Title : string, wymagane
     + Description: string wymagane
     + Price: double, wymagane
     + ImageURL: string, wymagane
     + StartDate: DateTime, wymagane
     + EndDate: DateTime, wymagane
     + MovieCategory: MovieCategory, wymagane
     + Actors\_Movies: List<Actor\_Movie>, wymagane
     + CinemaId: int , klucz obcy
     + Cinema: Cinema, relacja nawigacyjna
     + ProducerId: int, klucz obcy
     + Producer: Producer, relacja nawigacyjna
6. **Model: Order**
   * **Opis:** Model posiadający dane pojedynczego zamówienia
   * **Pola**
     + Id: int, Unikalny
     + **Email:** string,
     + **UserId:** string
     + **User:** AppliactionUser, klucz obcy
     + **OrderItems: List<OrderItems>**
7. **Model: OrderItems**
   * **Opis:** Model posiadający pojedyncze przedmioty
   * **Pola**
     + Id: int, Unikalny
     + **Amount:** int , wymagana
     + **Price:** double
     + **MovieId:** int, wymagana, klucz obcy
     + **Movie:** Movie, relacja nawigacyjna
     + **OrderId:** int, wymagana, klucz obcy
     + **Order:** Order, relacja nawigacyjna
8. **Model: ShoppingCartItem**
   * **Opis:** Model posiadający dane przedmiotów w koszyku
   * **Pola**
     + Id: int, Unikalny
     + Movie: Movie
     + Amount: int
     + ShoppingCartId: string

**Kontrolery:**

1. **AccountController**

* **Metoda: Users**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera listę wszystkich użytkowników i przekazuje ją do widoku.
  + **Zwracane dane:** Widok z listą użytkowników.
* **Metoda: Login (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok logowania.
  + **Zwracane dane:** Widok logowania.
* **Metoda: Login (POST)**
  + **HTTP:** POST
  + **Parametry:** LoginVM loginVM
  + **Opis:** Przetwarza formularz logowania. Sprawdza, czy użytkownik istnieje i czy hasło jest poprawne. Jeśli tak, przekierowuje do strony głównej.
  + **Zwracane dane:** Przekierowanie do strony głównej lub widok logowania z błędem.
* **Metoda: Register (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok rejestracji.
  + **Zwracane dane:** Widok rejestracji.
* **Metoda: Register (POST)**
  + **HTTP:** POST
  + **Parametry:** RegisterVM registerVM
  + **Opis:** Przetwarza formularz rejestracji. Sprawdza, czy użytkownik już istnieje. Jeśli nie, tworzy nowego użytkownika i przypisuje mu rolę User.
  + **Zwracane dane:** Widok potwierdzenia rejestracji lub widok rejestracji z błędem.
* **Metoda: Logout**
  + **HTTP:** POST
  + **Parametry:** Brak
  + **Opis:** Wylogowuje użytkownika i przekierowuje na stronę główną.
  + **Zwracane dane:** Przekierowanie do strony głównej.
* **Metoda: AccessDenied**
  + **HTTP:** GET
  + **Parametry:** string ReturnUrl
  + **Opis:** Renderuje widok informujący o odmowie dostępu.
  + **Zwracane dane:** Widok odmowy dostępu.

1. **ActorsController**

* **Metoda: Index**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera listę wszystkich aktorów i przekazuje ją do widoku.
  + **Zwracane dane:** Widok z listą aktorów.
* **Metoda: Create (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok tworzenia nowego aktora.
  + **Zwracane dane:** Widok tworzenia nowego aktora.
* **Metoda: Create (POST)**
  + **HTTP:** POST
  + **Parametry:** Actor actor
  + **Opis:** Przetwarza formularz tworzenia nowego aktora. Sprawdza, czy dane są prawidłowe, a następnie dodaje aktora do bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy aktorów lub widok tworzenia nowego aktora z błędami.
* **Metoda: Details**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego aktora na podstawie jego identyfikatora.
  + **Zwracane dane:** Widok szczegółów aktora lub widok informujący o braku aktora.
* **Metoda: Edit (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego aktora na podstawie jego identyfikatora i renderuje widok edycji.
  + **Zwracane dane:** Widok edycji aktora lub widok informujący o braku aktora.
* **Metoda: Edit (POST)**
  + **HTTP:** POST
  + **Parametry:** int id, Actor actor
  + **Opis:** Przetwarza formularz edycji aktora. Sprawdza, czy dane są prawidłowe, a następnie aktualizuje dane aktora w bazie danych.
  + **Zwracane dane:** Przekierowanie do widoku listy aktorów lub widok edycji aktora z błędami.
* **Metoda: Delete (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego aktora na podstawie jego identyfikatora i renderuje widok usuwania.
  + **Zwracane dane:** Widok usuwania aktora lub widok informujący o braku aktora.
* **Metoda: DeleteConfirmed**
  + **HTTP:** POST
  + **Parametry:** int id
  + **Opis:** Potwierdza usunięcie aktora na podstawie jego identyfikatora i usuwa aktora z bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy aktorów.

1. **CinemasController**

* **Metoda: Index**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera listę wszystkich kin i przekazuje ją do widoku.
  + **Zwracane dane:** Widok z listą kin.
* **Metoda: Create (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok tworzenia nowego kina.
  + **Zwracane dane:** Widok tworzenia nowego kina.
* **Metoda: Create (POST)**
  + **HTTP:** POST
  + **Parametry:** Cinema cinema
  + **Opis:** Przetwarza formularz tworzenia nowego kina. Sprawdza, czy dane są prawidłowe, a następnie dodaje kino do bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy kin lub widok tworzenia nowego kina z błędami.
* **Metoda: Details**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego kina na podstawie jego identyfikatora.
  + **Zwracane dane:** Widok szczegółów kina lub widok informujący o braku kina.
* **Metoda: Edit (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego kina na podstawie jego identyfikatora i renderuje widok edycji.
  + **Zwracane dane:** Widok edycji kina lub widok informujący o braku kina.
* **Metoda: Edit (POST)**
  + **HTTP:** POST
  + **Parametry:** int id, Cinema cinema
  + **Opis:** Przetwarza formularz edycji kina. Sprawdza, czy dane są prawidłowe, a następnie aktualizuje dane kina w bazie danych.
  + **Zwracane dane:** Przekierowanie do widoku listy kin lub widok edycji kina z błędami.
* **Metoda: Delete (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego kina na podstawie jego identyfikatora i renderuje widok usuwania.
  + **Zwracane dane:** Widok usuwania kina lub widok informujący o braku kina.
* **Metoda: DeleteConfirm**
  + **HTTP:** POST
  + **Parametry:** int id
  + **Opis:** Potwierdza usunięcie kina na podstawie jego identyfikatora i usuwa kino z bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy kin.

1. **MoviesController**

* **Metoda: Index**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera listę wszystkich filmów i przekazuje ją do widoku.
  + **Zwracane dane:** Widok z listą filmów.
* **Metoda: Details**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego filmu na podstawie jego identyfikatora.
  + **Zwracane dane:** Widok szczegółów filmu.
* **Metoda: Create (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok tworzenia nowego filmu. Pobiera dane potrzebne do wypełnienia pól wyboru (kino, producent, aktorzy).
  + **Zwracane dane:** Widok tworzenia nowego filmu.
* **Metoda: Create (POST)**
  + **HTTP:** POST
  + **Parametry:** NewMovieVM movie
  + **Opis:** Przetwarza formularz tworzenia nowego filmu. Sprawdza, czy dane są prawidłowe, a następnie dodaje film do bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy filmów lub widok tworzenia nowego filmu z błędami.
* **Metoda: Edit (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego filmu na podstawie jego identyfikatora i renderuje widok edycji. Pobiera dane potrzebne do wypełnienia pól wyboru (kino, producent, aktorzy).
  + **Zwracane dane:** Widok edycji filmu.
* **Metoda: Edit (POST)**
  + **HTTP:** POST
  + **Parametry:** int id, NewMovieVM movie
  + **Opis:** Przetwarza formularz edycji filmu. Sprawdza, czy dane są prawidłowe, a następnie aktualizuje dane filmu w bazie danych.
  + **Zwracane dane:** Przekierowanie do widoku listy filmów lub widok edycji filmu z błędami.
* **Metoda: Filter**
  + **HTTP:** GET
  + **Parametry:** string searchString
  + **Opis:** Filtruje listę filmów na podstawie podanego ciągu znaków. Szuka filmów, których tytuł lub opis zawiera podany ciąg.
  + **Zwracane dane:** Widok z przefiltrowaną listą filmów.

1. **OrdersController**

* **Metoda: Index**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera zamówienia dla aktualnie zalogowanego użytkownika na podstawie jego identyfikatora i roli, a następnie przekazuje je do widoku.
  + **Zwracane dane:** Widok z listą zamówień.
* **Metoda: ShoppingCart**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera elementy koszyka zakupów, oblicza całkowitą wartość koszyka i przekazuje te dane do widoku.
  + **Zwracane dane:** Widok koszyka zakupów z jego zawartością i całkowitą wartością.
* **Metoda: AddItemToShoppingCart**
  + **HTTP:** POST
  + **Parametry:** int id
  + **Opis:** Dodaje wybrany film do koszyka zakupów.
  + **Zwracane dane:** Przekierowanie do widoku koszyka zakupów.
* **Metoda: RemoveItemFromShoppingCart**
  + **HTTP:** POST
  + **Parametry:** int id
  + **Opis:** Usuwa wybrany film z koszyka zakupów.
  + **Zwracane dane:** Przekierowanie do widoku koszyka zakupów.
* **Metoda: OrderComplet**
  + **HTTP:** POST
  + **Parametry:** Brak
  + **Opis:** Finalizuje zamówienie, przechowuje je w bazie danych i czyści koszyk zakupów.
  + **Zwracane dane:** Widok potwierdzenia zamówienia.

1. **ProducersController**

* **Metoda: Index**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Pobiera listę wszystkich producentów i przekazuje ją do widoku.
  + **Zwracane dane:** Widok z listą producentów.
* **Metoda: Details**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego producenta na podstawie jego identyfikatora.
  + **Zwracane dane:** Widok szczegółów producenta lub widok informujący o braku producenta.
* **Metoda: Create (GET)**
  + **HTTP:** GET
  + **Parametry:** Brak
  + **Opis:** Renderuje widok tworzenia nowego producenta.
  + **Zwracane dane:** Widok tworzenia nowego producenta.
* **Metoda: Create (POST)**
  + **HTTP:** POST
  + **Parametry:** Producer producer
  + **Opis:** Przetwarza formularz tworzenia nowego producenta. Sprawdza, czy dane są prawidłowe, a następnie dodaje producenta do bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy producentów lub widok tworzenia nowego producenta z błędami.
* **Metoda: Edit (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego producenta na podstawie jego identyfikatora i renderuje widok edycji.
  + **Zwracane dane:** Widok edycji producenta lub widok informujący o braku producenta.
* **Metoda: Edit (POST)**
  + **HTTP:** POST
  + **Parametry:** int id, Producer producer
  + **Opis:** Przetwarza formularz edycji producenta. Sprawdza, czy dane są prawidłowe, a następnie aktualizuje dane producenta w bazie danych.
  + **Zwracane dane:** Przekierowanie do widoku listy producentów lub widok edycji producenta z błędami.
* **Metoda: Delete (GET)**
  + **HTTP:** GET
  + **Parametry:** int id
  + **Opis:** Pobiera szczegóły wybranego producenta na podstawie jego identyfikatora i renderuje widok usuwania.
  + **Zwracane dane:** Widok usuwania producenta lub widok informujący o braku producenta.
* **Metoda: DeleteConfirmed**
  + **HTTP:** POST
  + **Parametry:** int id
  + **Opis:** Potwierdza usunięcie producenta na podstawie jego identyfikatora i usuwa producenta z bazy danych.
  + **Zwracane dane:** Przekierowanie do widoku listy producentów.

**System Użytkowników:**

* **Role w systemie:**
  + Istniejące role: User, Admin
  + **Nadawanie roli:**
    - User: Nadawana po rejestracji każdemu użytkownikowi
    - Admin: Nadawana w kodzie lub w bazie danych
* **Możliwości użytkowników:**
  + **Zalogowani użytkownicy:** Kupowanie biletów, wyświetlanie swoich zamówień, wylogowywanie się, te same co Goście (poza logowaniem i rejestracją)
  + **Goście:** Przeglądanie listy filmów, aktorów, kin oraz reżyserów, możliwość logowania i rejestracji, możliwość wyświetlania szczegółów filmów aktorów kin i reżyserów , możliwość wyszukiwania filmów po tytule lub opisie
  + **Administrator:** to samo co gość(poza logowaniem i rejestracją), możliwość edytowania i dodawania: filmów, kin, aktorów, reżyserów, możliwość wyświetlania wszystkich zamówień, możliwość wyświetlania wszystkich użytkowników, wylogowywanie się
* **Informacje powiązane z :**
  + Użytkownik: Id, FullName, Name, Email, Password, Orders, Role,
* **Informacje globalne:**
  + Lista filmów, data filmów, informacje o kinach, informacje o reżyserach, informacje o aktorach

**Najciekawsze Funkcjonalności:**

1. **Funkcjonalność: Wyszukiwanie filmów po tytule**
   * **Opis:** Dodaje możliwość wyszukiwania filmów po ich tytule lub po ich opisie