

Programowanie Aplikacji Webowych(PAW)

Krzysztof Makówka 173664

System do obsługi filmów

1. Wstęp

- **Opis projektu:** Aplikacja webowa dla bazy danych z filmami, która pozwala kontu administratora uzupełnić bazę filmów dla wszystkich. Dla użytkownika jest przygotowana funkcja szukania filmu po tytule. Aplikacja służy do wyszukiwania filmów i najważniejszych informacji o nim.
- **Cele aplikacji:**
 - Umożliwia użytkownikom szybkie przeglądanie filmów
 - Wyszukiwanie filmów po tytule
 - Zarządzanie zbiorem filmów dla konta administratora
- **Technologie:**
 - **Backend:** C# .NET, ASP.NET
 - **Frontend:** Cshtml, Razor pages
 - **Baza danych:** MySQL

2. Opis

Modele:

1. LoginViewModel

Model używany do obsługi logowania użytkowników.

- **UserNameOrEmail:** Wymagane pole dla nazwy użytkownika lub e-maila. Maksymalna długość to 20 znaków. Oznaczone atrybutem DisplayName jako "Username or Email".
- **Password:** Wymagane pole dla hasła. Maksymalna długość to 20 znaków. Użyto atrybutu DataType.Password, aby wskazać na poufny charakter danych.

2. RegistrationViewModel

Model używany do rejestracji nowych użytkowników.

- **Id:** Klucz główny.
- **FirstName:** Wymagane pole dla imienia. Maksymalna długość to 50 znaków.
- **LastName:** Wymagane pole dla nazwiska. Maksymalna długość to 50 znaków.
- **Email:** Wymagane pole dla adresu e-mail. Zawiera wyrażenie regularne do walidacji poprawnego formatu.
- **Username:** Wymagane pole dla nazwy użytkownika. Maksymalna długość to 20 znaków.
- **Password:** Wymagane pole dla hasła. Maksymalna długość to 20 znaków.
- **Role:** Opcjonalne pole dla roli użytkownika. Domyślna wartość to "USER", maksymalnie 5 znaków.
- **ConfirmPassword:** Pole do potwierdzenia hasła. Musi być zgodne z Password. Minimalna długość to 5 znaków, maksymalna 20.

3. Movie

Model reprezentujący film.

- Id: Unikalny identyfikator filmu.
- Title: Tytuł filmu, maksymalna długość to 100 znaków.
- Description: Opis filmu, maksymalna długość to 100 znaków.
- Author: Autor filmu, maksymalna długość to 100 znaków.
- Genre: Gatunek filmu, maksymalna długość to 100 znaków.
- Price: Cena filmu, przechowywana jako wartość dziesiętna z precyzją 16,2.
- ImageFileName: Nazwa pliku graficznego związanego z filmem, maksymalna długość to 100 znaków.
- CreatedAt: Data utworzenia wpisu o filmie.

4. MovieDTO

Data Transfer Object (DTO) dla filmów.

- Title: Tytuł filmu, wymagany, maksymalnie 100 znaków.
- Description: Opis filmu, wymagany, maksymalnie 100 znaków.
- Author: Autor filmu, wymagany, maksymalnie 100 znaków.
- Genre: Gatunek filmu, wymagany, maksymalnie 100 znaków.
- Price: Cena filmu, wymagane pole.
- ImageFile: Opcjonalny plik graficzny (przechowywany jako IFormFile).

5. Model: UserAccount

UserAccount to encja przeznaczona do reprezentowania danych użytkownika w bazie danych. Zawiera szczegóły takie jak imię, nazwisko, adres e-mail, nazwa użytkownika, hasło i rola użytkownika. Model używa atrybutów walidacyjnych i indeksów dla unikalności.

Wszystkie z poniższych pól posiadają walidację: takie jak, że są wymagane, czy ograniczenie dotyczące długości, oraz dodana jest unikalność.

- Id
- FirstName
- LastName
- Email
- UserName
- Password
- Role

Kontrolery:

AccountController.cs

1. Index()

- **Typ:** GET
- **Opis:** Zwraca listę wszystkich użytkowników zapisanych w kolekcji UserAccountsCollection i przekazuje ją do widoku.
- **Zastosowanie:** Wyświetlanie listy zarejestrowanych użytkowników.

2. Registration()

- **Typ:** GET
- **Opis:** Renderuje formularz rejestracji użytkownika.
- **Zastosowanie:** Wyświetlanie formularza rejestracyjnego.

3. Registration(RegistrationViewModel model)

- **Typ:** POST
- **Opis:**
 - Przyjmuje dane z formularza rejestracyjnego.
 - Tworzy nowy obiekt UserAccount na podstawie danych wejściowych.
 - Waliduje dane i zapisuje nowego użytkownika w bazie danych.
 - Jeśli wystąpi błąd (np. duplikat emaila lub username), dodaje komunikat o błędzie do ModelState.
- **Zastosowanie:** Obsługa procesu rejestracji użytkownika.

4. Login()

- **Typ:** GET
- **Opis:** Renderuje formularz logowania.
- **Zastosowanie:** Wyświetlanie formularza logowania.

5. Login(LoginViewModel model)

- **Typ:** POST
- **Opis:**
 - Przyjmuje dane logowania z formularza.
 - Wyszukuje użytkownika w bazie danych na podstawie nazwy użytkownika/emaila oraz hasła.
 - Jeśli dane są poprawne, tworzy **ciasteczko uwierzytelniające** zawierające informacje o użytkowniku (claims).
 - Jeśli dane są niepoprawne, dodaje komunikat o błędzie do ModelState.
- **Zastosowanie:** Obsługa procesu logowania i tworzenie sesji użytkownika.

6. Logout()

- **Typ:** GET
- **Opis:**
 - Wylogowuje użytkownika poprzez usunięcie ciasteczka uwierzytelniającego.
 - Przekierowuje użytkownika na stronę główną (metoda Index).
- **Zastosowanie:** Obsługa procesu wylogowywania.

7. SecurePage()

- **Typ:** GET
- **Opis:**
 - Strona chroniona, dostępna tylko dla zalogowanych użytkowników.
 - Wyświetla informacje o zalogowanym użytkowniku, takie jak nazwa (email).
- **Zastosowanie:** Wyświetlanie treści tylko dla użytkowników posiadających autoryzację.

MoviesController.cs

1. Index(string searchString)

- **Typ:** GET
- **Opis:** Pobiera listę filmów z bazy danych. Filtruje wyniki na podstawie przekazanej frazy searchString (szuka w tytule filmu).
- **Zastosowanie:** Wyświetlanie listy filmów z opcjonalnym filtrowaniem.

2. Create()

- **Typ:** GET
- **Opis:** Renderuje formularz do dodawania nowego filmu.
- **Zastosowanie:** Wyświetlanie formularza tworzenia filmu.

3. Create(MovieDTO movieDTO)

- **Typ:** POST
- **Opis:**
 - Waliduje dane wejściowe.
 - Zapisuje obraz filmu na serwerze.
 - Tworzy nowy obiekt Movie, zapisuje go w bazie danych.
- **Zastosowanie:** Obsługa dodawania nowego filmu.

4. Edit(int id)

- **Typ:** GET

- **Opis:**
 - Pobiera film o podanym id.
 - Jeśli film istnieje, przygotowuje dane do edycji i renderuje formularz.
- **Zastosowanie:** Wyświetlanie formularza edycji filmu.

5. Edit(int id, MovieDto movieDto)

- **Typ:** POST
- **Opis:**
 - Aktualizuje dane filmu o podanym id na podstawie przesłanych danych.
 - Jeśli dodano nowy obraz, zastępuje stary (usuwa poprzedni plik z serwera).
- **Zastosowanie:** Obsługa edycji filmu.

6. Delete(int id)

- **Typ:** GET
- **Opis:**
 - Usuwa film o podanym id z bazy danych.
 - Usuwa również powiązany plik obrazu z serwera.
- **Zastosowanie:** Obsługa usuwania filmu.

HomeController

1. Index()

- **Typ:** GET
- **Opis:** Renderuje stronę główną aplikacji.
- **Zastosowanie:** Wyświetlanie głównej strony.

Widoki

1. Index (dla kolekcji UserAccount)

Cel widoku:

- Wyświetlenie listy użytkowników (UserAccount) w formie tabeli.

Elementy widoku:

- **Nagłówek:** Tytuł "Index".
- **Przycisk "Create New":** Link do akcji Create, pozwalający na dodanie nowego użytkownika.
- **Tabela:**
 - Wyświetla dane użytkowników (Id, FirstName, LastName, Email, UserName).

- Dane są generowane dynamicznie w pętli foreach, iterującej po modelu (IEnumerable<UserAccount>).

2. Login

Cel widoku:

- Formularz logowania użytkownika.

Elementy widoku:

- **Nagłówek:** Tytuł "Login".
- **Formularz logowania:**
 - **Pola:**
 - UserNameOrEmail (nazwa użytkownika lub e-mail).
 - Password (hasło).
 - **Przycisk "Login":** Przesyła dane formularza do akcji Login.
 - **Walidacja:** Wyświetla komunikaty o błędach, jeśli walidacja modelu nie powiedzie się.
- **Link do rejestracji:** Przekierowanie do akcji Registration, jeśli użytkownik nie ma konta.

3. Registration

Cel widoku:

- Formularz rejestracji nowego użytkownika.

Elementy widoku:

- **Nagłówek:** Tytuł "Registration".
- **Formularz rejestracyjny:**
 - **Pola:**
 - FirstName, LastName, Email, Username, Password, ConfirmPassword.
 - **Przycisk "Register":** Przesyła dane formularza do akcji Registration.
 - **Komunikaty:**
 - Wyświetla wiadomość z ViewBag.Message, jeśli rejestracja zakończy się sukcesem.
 - Pokazuje błędy walidacji, jeśli dane są nieprawidłowe.
- **Link do logowania:** Przekierowanie do akcji Login, jeśli użytkownik ma już konto.

4. SecurePage Widok powitalny (Hello @ViewBag.Name)

Cel widoku:

- Wyświetlenie powitania z nazwą użytkownika (@ViewBag.Name) na stronie, dostępnej po zalogowaniu.

Elementy widoku:

- Nagłówek z dynamicznie generowanym powitaniem (np. "Hello Krzysztof").

5. New Movie (Dodawanie filmu)

Cel widoku:

- Umożliwienie użytkownikowi dodania nowego filmu do systemu.

Elementy widoku:

- **Nagłówek:** Tytuł "New Movie".
- **Formularz:**
 - Pola do wprowadzania danych:
 - Title (tytuł filmu).
 - Description (opis filmu).
 - Genre (gatunek filmu - wybierany z rozwijanej listy).
 - Price (cena filmu).
 - Author (autor filmu).
 - ImageFile (zdjęcie wgrywane jako plik).
 - Walidacja: Wyświetlane są komunikaty o błędach przy nieprawidłowych danych.
 - Przycisk "Submit": Zapisuje dane filmu.
 - Przycisk "Cancel": Przekierowuje użytkownika z powrotem do listy filmów.

6. Edit Movie (Edycja filmu)

Cel widoku:

- Umożliwienie użytkownikowi edytowania istniejących danych filmu.

Elementy widoku:

- **Nagłówek:** Tytuł "Edit Movie".
- **Formularz:**
 - Pola do edycji danych:
 - Title, Description, Genre, Price, Author.
 - Wyświetlane tylko do odczytu:
 - Id (identyfikator filmu).
 - Created At (data utworzenia).

- Sekcja podglądu:
 - Obraz filmu (ImageFileName).
- Przycisk "Submit": Zapisuje zmiany.
- Przycisk "Cancel": Powrót do listy filmów.

7. List of Movies (Lista filmów)

Cel widoku:

- Wyświetlenie wszystkich dostępnych filmów w systemie oraz udostępnienie opcji zarządzania (dla administratora).

Elementy widoku:

- **Nagłówek:** Tytuł "List of Movies".
- **Sekcja wyszukiwania:**
 - Pole wyszukiwania po nazwie filmu.
 - Link do powrotu do pełnej listy.
- **Przycisk "New Movie"** (dla administratorów): Przekierowuje do formularza dodawania nowego filmu.
- **Tabela z danymi:**
 - Kolumny:
 - Id, Title, Description, Author, Genre, Price, Created At, Action.
 - Wiersze:
 - Dynamiczne wyświetlanie danych o filmach z modelu.
 - Kolumna "Action" (dla administratorów):
 - Przycisk "Edit": Przekierowanie do edycji filmu.
 - Przycisk "Delete": Usunięcie filmu.
- **Podgląd zdjęć:**
 - Obrazy są wyświetlane w tabeli w kolumnie ImageFileName.

3. Link do repozytorium:

<https://github.com/makovvka/MoviesRental>

