

Dokumentacja Techniczna Projektu

Autorzy Projektu: Jakub Jankowski, Alicja Nowacka

System Rezerwacji Biletów

Opis Działania Projektu:

System Rezerwacji Filmów jest aplikacją internetową, która umożliwia użytkownikom przeglądanie repertuaru filmów, ocenianie filmów oraz rezerwację biletów do kina. Aplikacja dostarcza funkcjonalności zarówno dla zwykłych użytkowników (gości), jak i dla użytkowników zarejestrowanych, zapewniając bardziej zaawansowane opcje zarządzania rezerwacjami oraz ocenami

1. Elementy projektu

Funkcjonalności

1. Strona główna:

- Dostępna dla wszystkich użytkowników, prezentuje informacje o aplikacji i wybranych filmach.

2. System zarządzania użytkownikami:

- Możliwość rejestracji i logowania użytkowników.

3. Repertuar:

- Informacje o dostępnych filmach.

4. Obsługa bazy danych:

- Dane użytkowników, rezerwacji i repertuaru przechowywane w bazie SQL Server.

5. Moduł oceniania filmów:

- Użytkownicy mogą oceniać filmy w skali od 1 do 5 gwiazdek.

6. System rezerwacji biletów:

- Rezerwacja miejsc na seanse z wyborem godziny oraz daty.

2. Instrukcja uruchomienia

Wymagania wstępne

- **Środowisko:** System operacyjny Windows 10/11, Visual Studio 2022.
- **Technologie:**
 - .NET 8.0
 - SQL Server (localdb)

Kroki uruchomienia

1. **Pobranie kodu:**
 - Pobierz projekt z repozytorium.
2. **Otwórz projekt:**
 - Uruchom plik WebApplication3.sln w Visual Studio.
3. **Przygotowanie bazy danych:**
 - Wykonaj migracje bazy poleceniem w Konsoli Menedżera Pakietów:
 - Add-Migration <nazwa>
 - Update-Database.
4. **Uruchomienie projektu:**
 - Ustaw WebApplication3 jako projekt startowy.
 - Kliknij **Start** lub naciśnij **F5** w Visual Studio.
5. **Testowanie dostępu:**
 - Przeglądarka z projektem otworzy się automatycznie.

3. Opis struktury projektu

Kluczowe foldery i pliki

1. **Pliki projektu:**
 - WebApplication3.sln: Główny plik rozwiązania.
 - WebApplication3.csproj: Konfiguracja projektu .NET.
2. **Foldery:**
 - **Controllers/:** Kontrolery obsługujące logikę aplikacji:
 1. Repertuar.cs - Zawiera skrypty do dodawania ocen oraz rezerwacji biletów.
 2. Home.cs - Podstawowe ustawienia strony głównej.

3. ReservationController.cs - Zawiera skrypty pozwalające na wyświetlenie i edycje własnych rezerwacji i ocen.
 4. Privacy.cs - Polityka prywatności.
- **Models/**: Definicje modeli danych:
 1. Movie.cs - Model danych filmu.
 2. MovieRating.cs - Model danych oceny filmu.
 3. Reservations.cs - Model danych rezerwacji.
 4. UserRating.cs - Model danych oceny użytkownika.
 - **Views/**: Widoki aplikacji:
 1. Repertuar.cs - Wyświetla stronę /Repertuar
 2. Home.cs - Strona główna.
 3. ReservationController.cs - Wyświetla stronę z rezerwacjami i ocenami użytkownika.
 4. Privacy.cs - Polityka prywatności.
 - **wwwroot/**: Pliki statyczne, takie jak plakaty filmów.
 - **Areas/Identity/**: Moduł zarządzania tożsamością i uwierzytelnianiem.
 - **Services/**: Pliki z danymi początkowymi
 1. MovieServices - Zawiera listę filmów ładowaną przy każdym uruchomieniu.

3. Pliki konfiguracyjne:

- appsettings.json: Ustawienia bazy danych i logowania.

Przepływ danych

1. Użytkownik wysyła żądanie HTTP (np. otwarcie strony głównej).
2. Kontroler (np. HomeController) przetwarza żądanie.
3. Widok generuje interfejs użytkownika i zwraca go do przeglądarki.
4. W przypadku operacji na danych (np. rezerwacja) wykorzystywana jest komunikacja z bazą SQL Server.

4. Opis systemu użytkowników

Zarządzanie użytkownikami

1. Rejestracja:

- Formularz rejestracji umożliwia tworzenie kont użytkowników.
- Walidacja danych przed zapisaniem w bazie.

2. Logowanie:

- Uwierzytelnianie na podstawie danych przechowywanych w tabeli `AspNetUsers`.
- Zarządzanie sesjami użytkownika z wykorzystaniem plików cookie.

3. Role i uprawnienia:

- **Użytkownik:**
 - Może przeglądać repertuar, oceniać filmy i dokonywać rezerwacji.
- **Gosć:**
 - Może przeglądać repertuar i sprawdzać szczegóły filmu.

Komponenty systemu użytkowników

1. Baza danych:

- Tabela `AspNetUsers`: Przechowuje dane użytkowników.
- Tabela `AspNetRoles`: Definiuje role i uprawnienia.
- Tabela `Reservations`: Zarządza danymi rezerwacji.

2. Warstwa aplikacji:

- Moduł **Identity** obsługuje uwierzytelnianie i autoryzację.

3. Widoki:

- Formularze rejestracji (`Register`), logowania (`Login`) oraz panel użytkownika (`Manage`).

5. Uwagi końcowe

- Dokumentacja została opracowana na podstawie kodu projektu i jego konfiguracji.
- W przypadku pytań lub problemów technicznych prosimy o kontakt z autorem projektu.
- Prosimy zgłaszać wszelkie niezgodności funkcjonalne, które wymagają poprawy lub aktualizacji.