

Home Library – Dokumentacja

Wprowadzenie:

Home Library to aplikacja internetowa w architekturze typu klient-serwer. Każda osoba może:

- Założyć własne konto użytkownika i zalogować się
- Przeglądać i zarządzać listą gier dostępną dla wszystkich zalogowanych użytkowników
- Przeglądać i zarządzać przypisanymi do siebie bibliotekami
- Wyświetlać i modyfikować zawartość przypisanych do siebie bibliotek

Ogólne opis techniczny:

Z technicznego punktu widzenia, jest to aplikacja internetowa będąca projektem ASP.NET Core (MVC) napisanym w IDE Visual Studio 2022. Przy tworzeniu projektu w Visual Studio wybrano:

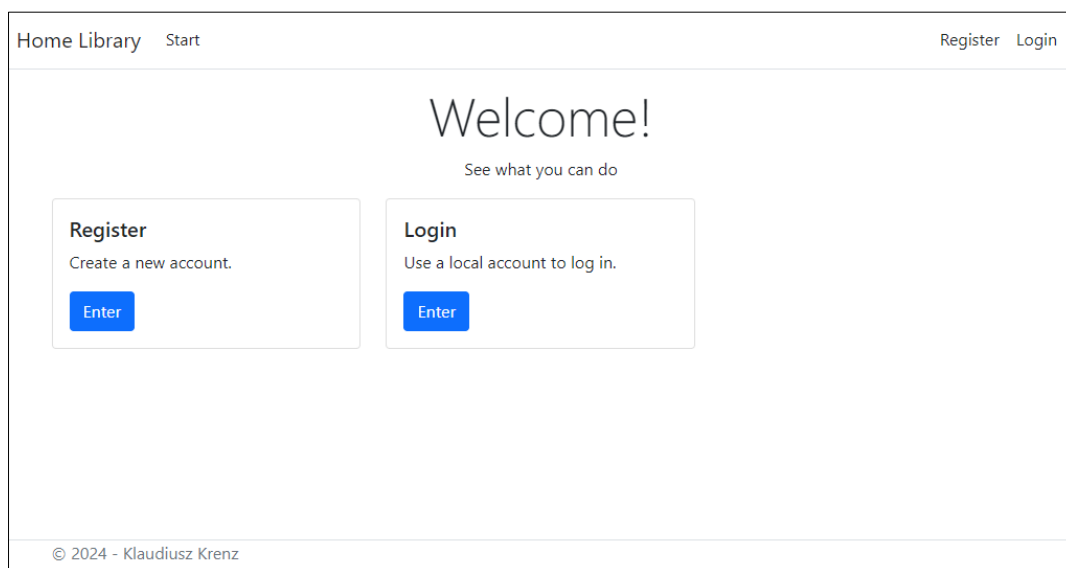
- Framework .NET 8.0
- Typ autoryzacji - Pojedyncze konta

Projekt korzysta z EntityFramework Core do obsługi mapowania obiektowo-relacyjnego (ORM).

W procesie tworzenia kontrolerów i widoków na podstawie modeli użyto opcji "Scaffold".

Rozpoczęcie korzystania z aplikacji:

Po otwarciu solucji, skompilowaniu i poprawnemu uruchomieniu aplikacji ukaże się nam strona startowa, jej zawartością są kafelki będące linkami do innych funkcjonalności.



Home Library Start Register Login

Welcome!

See what you can do

Register

Create a new account.

Enter

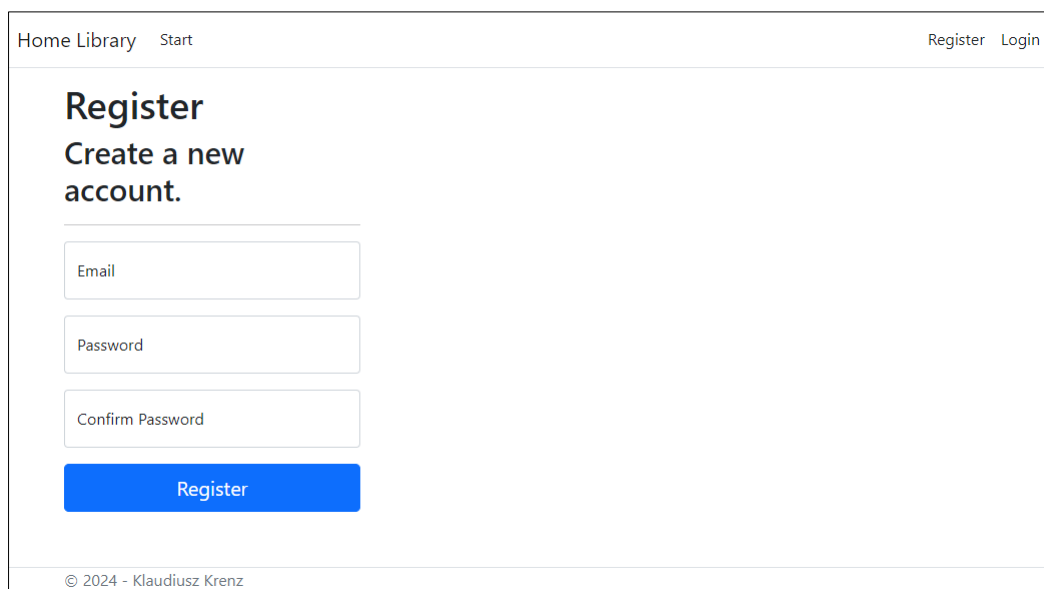
Login

Use a local account to log in.

Enter

© 2024 - Klaudiusz Krenz

Poza możliwością rejestracji i zalogowania jest to **jedyna** zakładka do której ma dostęp niezalogowany użytkownik.



Home Library Start Register Login

Register

Create a new account.

Email

Password

Confirm Password

Register

© 2024 - Klaudiusz Krenz

Opis techniczny:

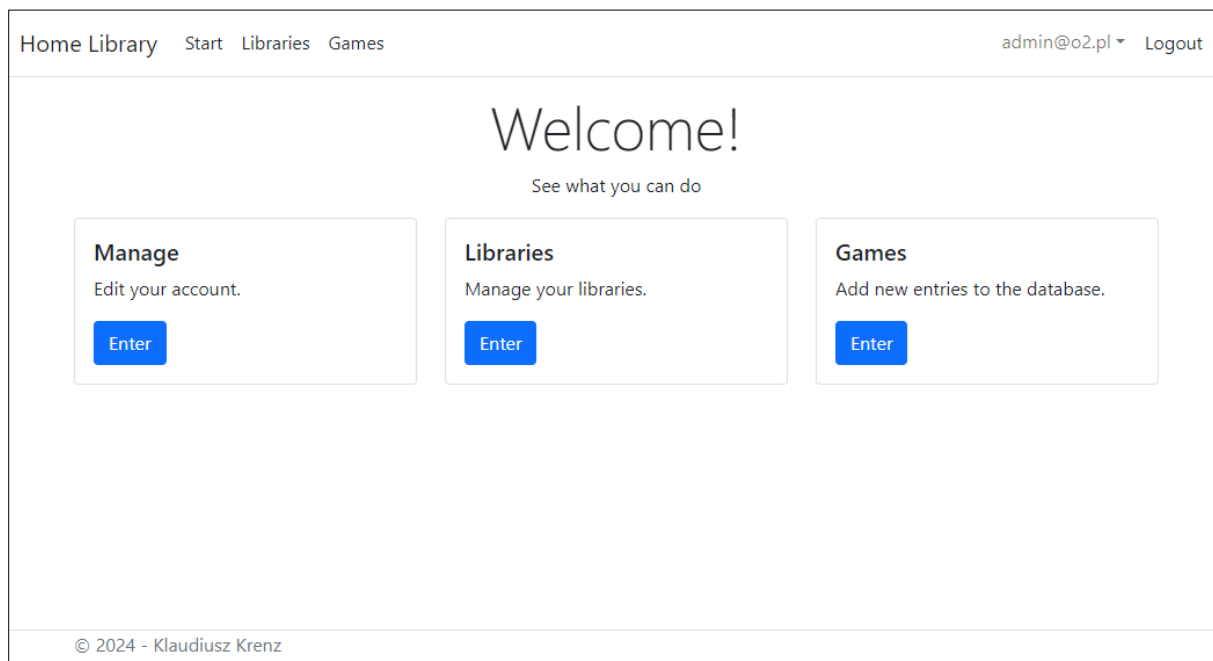
W widoku **Index** dla kontrolera **HomeController** oraz we współdzielonych w całej aplikacji plikach **_Layout.cshtml** i **_LoginPartial.cshtml** dodano warunek sprawdzający czy użytkownik jest zalogowany, jeśli tak to wyświetlana jest inna zawartość na stronie startowej oraz w nagłówku.

Poza samym logowaniem pierwsze uruchomienie powinno także wypełnić tabelę Game przykładowymi danymi, jest to wykonywane przez klasę **SeedData** który dodaje dane przy uruchomieniu aplikacji pod warunkiem że w tabeli nie ma żadnych danych.

Użytkownicy:

Aby korzystać ze wszystkich funkcji aplikacji, należy stworzyć konto klikając na opcję **'Register'**.

Po zalogowaniu, zmienia się zawartość strony startowej oraz zostają dodane linki do innych zakładki aplikacji.



Opis techniczny:

Dzięki skorzystaniu z odpowiedniej metody autoryzacji w projekcie, obsługa rejestracji, logowania i wylogowania jest już obsługiwana.

Aby móc dodać relację "jeden do wielu" w bazie danych pomiędzy Użytkownikiem a jego bibliotekami, stworzono model **'ApplicationUser'** będący rozszerzeniem domyślnego **'IdentityUser'**. **ApplicationUser** jako property nawigacyjną posiada kolekcję typu **<Library>**.

Odniesienia do **<IdentityUser>** w klasie Program.cs, modelu Library i kontrolerze LibrariesController zostały zastąpione przez **<ApplicationUser>**.

We wszystkich kontrolerach **poza** HomeController.cs została dodana linijka **'[Authorize]'**, niezalogowany użytkownik nie będzie miał dostępu do wszystkich widoków obsługiwanych przez te kontrolery.

Games

W tej zakładce pojawia się lista wszystkich zapisanych gier w formie tabeli. Gry można dodawać, edytować oraz usuwać. W ramach dodawania i edycji można **opcjonalnie** wgrać plik okładki.

Home Library	Start	Libraries	Games	admin@o2.pl ▾	Logout
<h3>Index</h3> Create New					
Title	Developer	Publisher	Platform	Release Date	
Sonic Frontiers	Sonic Team	SEGA	PS4	06.08.2022	Edit Details Delete
The World Ends with You	Jupiter	Square Enix	NDS	18.04.2008	Edit Details Delete
Persona 4 Golden	ATLUS	SEGA	PC	14.06.2020	Edit Details Delete
Kid Icarus Uprising	Sora Ltd	Nintendo	3DS	22.03.2012	Edit Details Delete

Opis techniczny:

W modelu dla każdej propertki poza Id (bo jest kluczem głównym) i związanymi z okładką dodano "[Required]" dzięki czemu na poziomie kontrolera i widoków jest odpowiednia **walidacja**.

Propertka '**ReleaseDate**' obok "[Required]" ma także **Name** (stosowane jako nazwa wyświetlana nagłówka kolumny) i **DataType** (aby ograniczyć wartość zmiennej do samej daty)

Okładka jest obsługiwana przez propertki '**CoverPath**' i '**Cover**', pierwsza jest ciągiem znaków będącym ścieżką do pliku w folderze '**wwwroot**', druga jest używana w procesie zapisywania pliku okładki wybranego w formularzu tworzenia/edycji gry.

Kod obsługujący plik został dodany do metod **Create**, **Edit** i **DeleteConfirmed** kontrolera **GamesController**, na przykładzie kodu w metodzie Edit (najbardziej rozwinięty przykład):

- No poziomie widoku można wgrać tylko pliku typu obraz
- Kod odpowiedzialny za obsługę jest uruchamiany tylko jeśli podano nowy plik w formularzu i ma on odpowiednie rozszerzenie, w przeciwnym wypadku stara wartość zostaje zachowana, stara wartość jest zapisana w zmiennej "oldGameCover" i jest przekazywana przez formularz
- Najpierw zapisywane jest rozszerzenie, używane w nazwie pliku docelowego
- Jeśli gra miała zapisaną jakąś wartość ścieżki okładki to poprzedni plik okładki jest usuwany
- coverPath = ścieżka do pliku zapisywana w bazie danych, złożona z podfolderów, nazwy pliku w formie unikalnego identyfikatora (GUID) i rozszerzenia
- filePath - używana w procesie zapisywania pliku, przed ścieżką w coverPath dodaje nazwę folderu 'wwwroot', wymagane aby zapis się powiódł
- game.CoverPath = nowa ścieżka do pliku

Kod w metodzie **Create** wykonuje analogiczne czynności co powyżej pomijając usuwanie poprzedniego pliku, w metodzie **DeleteConfirmed** jest usuwany plik wraz z rekordem w bazie danych (jeśli plik istnieje).

Wszędzie gdzie są wyświetlane okładki sprawdza się czy plik istnieje, jeśli nie to pojawia się puste tło.

Libraries

W tej zakładce mamy możliwość przeglądania listy stworzonych przez nas bibliotek gier, dodania nowej, edycji, usunięcia i przede wszystkim przejścia do zawartości danej biblioteki przez opcję **List**.

Każda biblioteka ma swoją nazwę, opcjonalny opis i jest przypisana do **obecnie zalogowanego** użytkownika.

Home Library Start Libraries Games

admin@o2.pl Logout

Index

[Create New](#)

Name	Description	
Game Library 1	In progress	Edit Details Delete List
Game Library 2	Backlog	Edit Details Delete List
Game Library 3	Completed	Edit Details Delete List

© 2024 - Klaudiusz Krenz

Opis techniczny

W kontrolerze znajduje się zmienna **_userManager** dzięki której można w metodach używać użytkownika, np. Przy tworzeniu zapytania do bazy danych.

Podstawą działania jest relacja bazodanowa “jeden do wielu” między modelami **ApplicationUser** i **Library** stworzona przez EntityFramework na podstawie propertek nawigacyjnych **ApplicationUser.Libraries** i **Library.User**.

Na podstawie Id biblioteki wykonuję zapytanie do bazy które zwraca **listę bibliotek** powiązanych z id zalogowanego użytkownika.

Dla każdego widoku który jako parametr **w adresie URL** przyjmuje Id biblioteki wykonywane jest sprawdzenie w kontrolerze czy ID zalogowanego użytkownika jest faktycznie równe ID użytkownika przypisanego do biblioteki, jeśli nie to **nie ma dostępu** do tego widoku.

Zawartość biblioteki


Tutaj wyświetla się zawartość biblioteki do której możemy przypisać dodane wcześniej tytuły. Dostępne formy widoku to tabela lub siatka.

Home LibraryStartLibrariesGames


admin@o2.plLogout

Game Library 1: In progress


View:TableGridActions:ListAssignUnassign



Sonic Frontiers
Sonic Team \ SEGA
Release date: 2022-08-06
Platform: PS4



The World Ends with You
Jupiter \ Square Enix
Release date: 2008-04-18
Platform: NDS



Kid Icarus Uprising
Sora Ltd \ Nintendo
Release date: 2012-03-22
Platform: 3DS

© 2024 - Klaudiusz Krenz

Aby dodać lub usunąć tytuł z biblioteki klikamy na przycisk **Assign** lub **Unassign**, następnie wyświetli się tabela gdzie zaznaczamy wybrane pozycje, potwierdzamy przyciskiem **Confirm**.

Home LibraryStartLibrariesGames

admin@o2.plLogout

Game Library 1: In progress

Actions:ListAssignUnassign

Title	Developer	Publisher	Platform	Release Date	Unassign
Sonic Frontiers	Sonic Team	SEGA	PS4	2022-08-06	<input type="checkbox"/>
The World Ends with You	Jupiter	Square Enix	NDS	2008-04-18	<input type="checkbox"/>
Kid Icarus Uprising	Sora Ltd	Nintendo	3DS	2012-03-22	<input checked="" type="checkbox"/>

Confirm

© 2024 - Klaudiusz Krenz

Opis techniczny

Za wszystkie czynności związane z zawartością biblioteki odpowiadają metody **List**, **Assign** i **Unassign** w kontrolerze **LibrariesController**. We wszystkich trzech zakładkach część zachowań jest identyczna:

- Nazwa obecnie wybranej akcji jest zapisana w zmiennej **ViewData["actionName"]**
- Każdy widok ładuje na początku wspólny plik **_header.cshtml** która odpowiada za podnagłówki, przyciski aktualnego widoku i akcji są wyłączone przez prosty kod JS który dodaje klasę HTML "disabled" na podstawie nazwy akcji i parametru
- Jeśli nie ma żadnych pozycji do wyświetlania pojawia się informacja "No data available"
- Każdy widok używa modelu **LibraryViewModel**, jedną z jego zmiennych jest "**Games**" która zawiera listę gier będącą wynikiem zapytania

Podstawą działania jest relacja bazodanowa "wiele do wielu" między modelami **Game** i **Library** tworząca tabelę **GameLibrary** na podstawie propertek nawigacyjnych **Game.Libraries** i **Library.Games**, gdzie w każdym wierszu znajdują się pary ID wybranej biblioteki i ID powiązanej gry.

List

Jeśli zmienna "**Style**" przekazywana metodą HTTP GET będzie miała wartość "**Table**" lub w ogóle nie zostanie przekazana to w widoku pojawi się tabela, jeśli będzie miała wartość "**Grid**" to pojawi się siatka.

W kontrolerze wykonywane jest zapytanie zwracające listę powiązanych gier na podstawie ID biblioteki.

Assign

Metoda kontrolera dla GET wykonuje dwa zapytania – pierwsze jest używane do pobrania wszystkich gier przypisanych do danej biblioteki a drugie zwraca listę wszystkich gier w bazie **oprócz** tych które już są przypisane.

W widoku w tabeli generowana jest lista wszystkich pozycji które możemy przypisać, na końcu każdego wiersza jest checkbox którego wartością jest ID danej pozycji, po kliknięciu na "Confirm" do kontrolera przekazywana jest lista pod nazwą "**selectedItemIds**".

Kontroler dla POST następnie wyszukuje każdą pozycję po jej ID i w pętli zapisuje je, pod koniec następuje powrót do zawartości biblioteki.

Unassign

Kontroler dla GET w odróżnieniu od List i Assign zapisuje do zmiennej "**library**" także pełną zawartość kolekcji Games, jest to wymagane aby przy usuwaniu EntityFramework wiedział jakie pozycje zostały usunięte w porównaniu z bazą danych i by przy wykonywaniu metody "**SaveChangesAsync()**" zmiany te by były naniesione także do niej.

Metoda wykonuje zapytanie którego wynikiem jest lista gier powiązanych z daną biblioteką, podobnie jak w Assign używana jest zmienna "**selectedItemIds**", tym razem jest ona używana do usunięcia powiązania.