

**Dokumentacja projektu**

Przedmiot: **Programowanie**

Tytuł projektu: Movie Rent

Prowadzący: Wykonawca: Bohdan Filimonyuk

*Dr Marek Jaszuk* w58960

Grupa: SP02

Rzeszów, 2019

Spis treści

Opis założeń projektu(wymagania biznesowe)3

Specyfikacja wymagań – funkcjonalne3

Specyfikacja wymagań – niefunkcjonalne 3

Diagram przypadków użycia4

Harmonogram realizacji projektu5

Opis techniczny projektu6

Prezentacja warstwy użytkowej projektu 8

System kontroli wersji 10

Raporty z testów 12

Materiały źródłowe 12

Dokumentacja w postaci Html 12

**1.** Opis założeń projektu(wymagania biznesowe)

Aplikacja Movie Rent jest to aplikacja webowa oparta na technologii ASP .NET Core MVC, która ogarnia rezerwacje filmów do wypożyczenia przez użytkowników oraz kontrole tych rezerwacji przez pracowników. Jest realizowany zapis zamówień do bazy danych oraz rejestracja/logowanie użytkowników.

Strona jest łatwa w użyciu dla pracowników oraz zwykłych użytkowników.

Aplikacja może być wykorzystana dla wypożyczenia filmów oraz sterowaniem nimi.

**2.1** Specyfikacja wymagań – funkcjonalne

1) Najpierw użytkownik może zobaczyć pierwszą stronę ”Home” gdzie jest tylko pozdrowienia dla użytkowników.

2) Jest możliwość zobaczyć listę filmów w drugiej zakładce ”MvcMovie”.

3) Są dwie zakładki „Log in” oraz „Register” gdzie można zalogować się, jeżeli użytkownik ma założone konto lub założyć go.

4) Po logowaniu, pojawia się możliwość do wypożyczenia filmów w zakładce „MvcMovie”.

5) Pojawia się zakładka „My Rents” gdzie można zobaczyć listę wypożyczonych filmów zalogowanym użytkownikiem.

6) Jeżeli zalogowany użytkownika ma rolę „Admin”, to może on zobaczyć wszystkie zamówienia w zakładce „All Rents” oraz w zakładce „Admin MovieMvc” można zobaczyć listę filmów, ale z możliwością edytowania, usunięcia oraz stworzenia nowych wpisów do listy filmów.

**2.2** Specyfikacja wymagań – niefunkcjonalne

1) Język interfejsu – angielski.

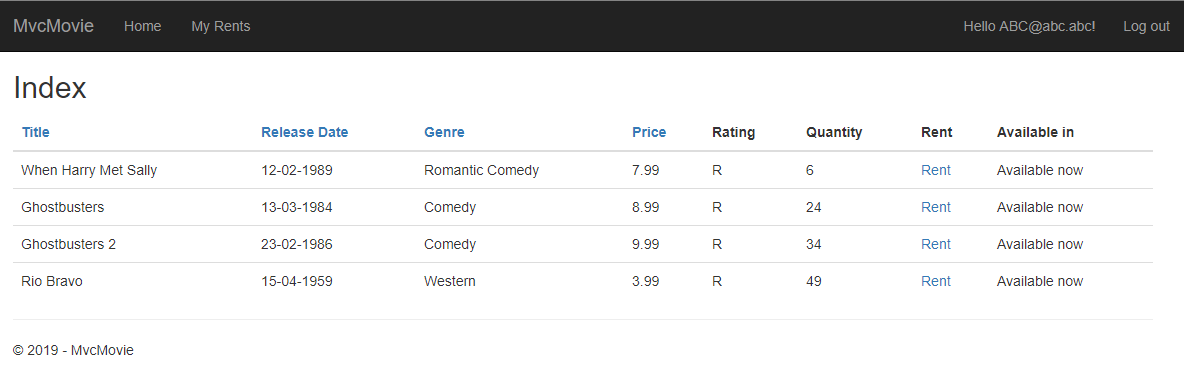
2) Bardzo prosty interfejs użytkownika z menu na górze strony, oraz zawartością w innych częściach ekranu. **Rysunek 1**

3) Modalne okna dla niektórych funkcyjności(„Details”, ”Delete”, ”Edit”).

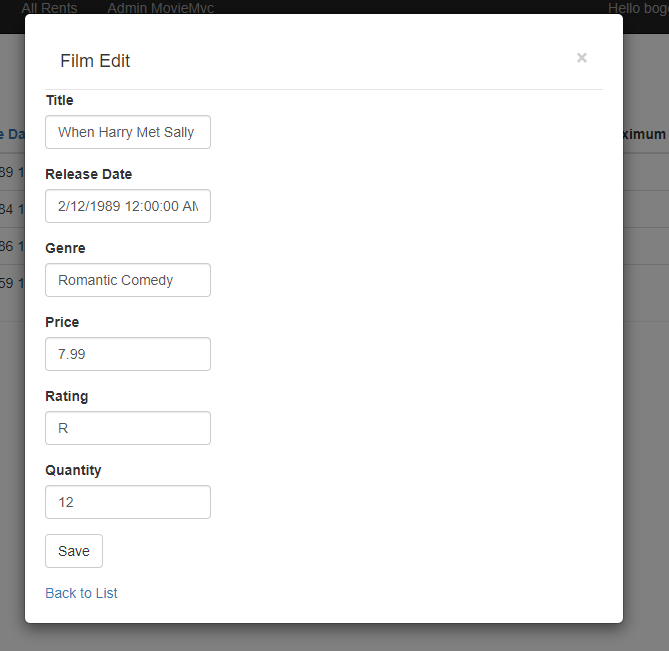
**Rysunek 2**

4) Okno dla logowania użytkowników oraz rejestracji.

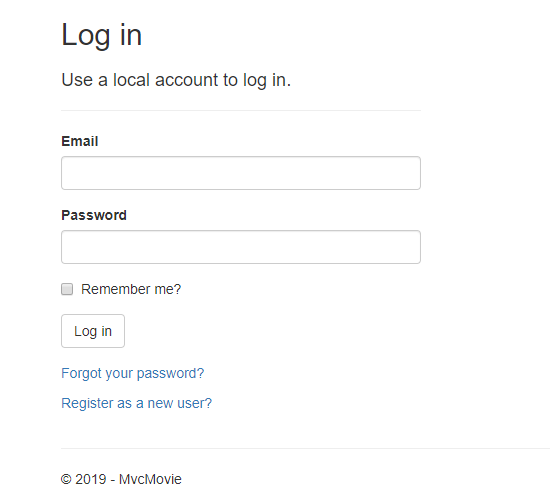
**Rysunek 3**



Rysunek 1



Rysunek 2



Rysunek 3

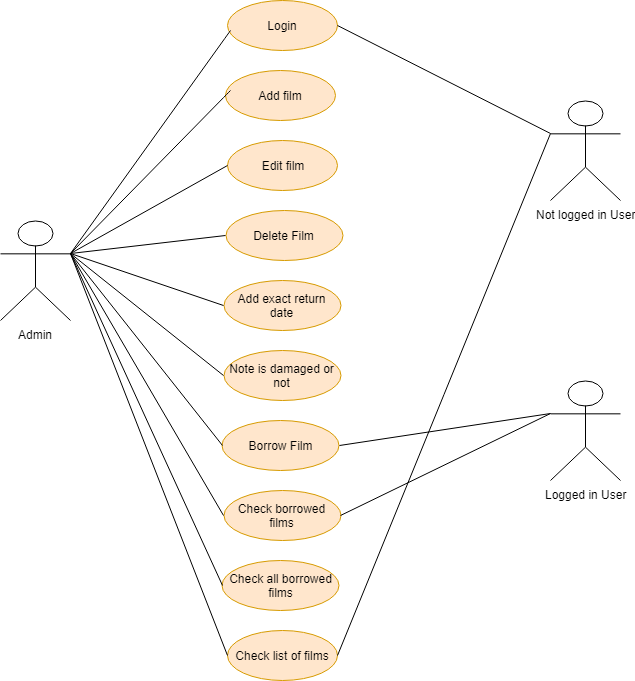
**3.** Diagram przypadków użycia

W web aplikacji „Movie Rent” są dwa typy użytkowników: Customer oraz Admin.

**Customer** – jest to użytkownik docelowy, jaki rejestruje się i wypożycza filmy. Może on także zobaczyć co wypożyczył.

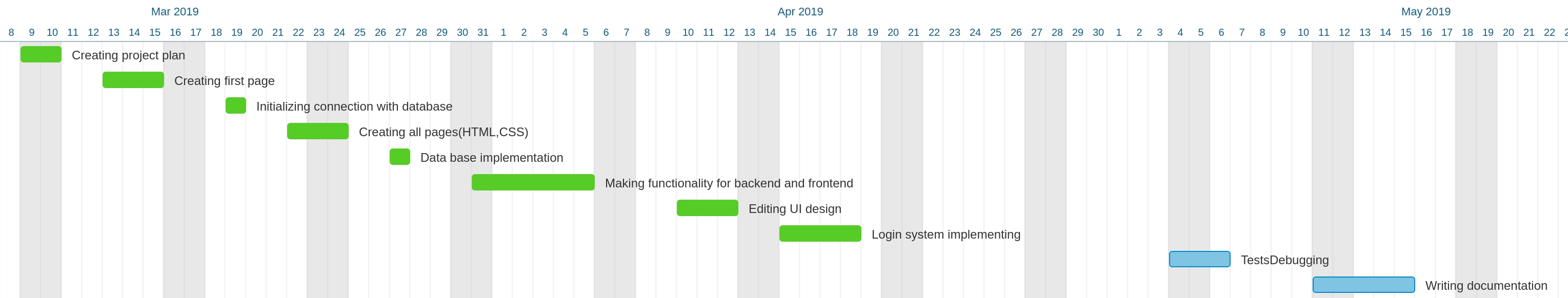
**Admin** – jest to użytkownik, jaki steruje wypożyczeniem, oraz może dodawać filmy, edytować oraz usuwać. Także zapisuje kiedy dokładnie był zwrócony film oraz czy był on uszkodzony. Ma takie same możliwości co i **Customer.**

Diagram użycia – **Rysunek 4.**



Rysunek 4

**4.** Harmonogram realizacji projektu



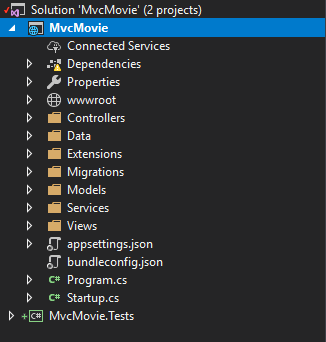
Rysunek 5

**5.** Opis techniczny projektu

Projekt jest realizowany za pomocą Asp .Net Core. Struktura aplikacji jest zaprojektowana za pomocą MVC(Model View Controller).

**Struktura aplikacji**

Solution składa się z 2 projektów: MvcMovie(aplikacja) oraz MvcMovie.Tests(projekt dla testów jednostkowych).

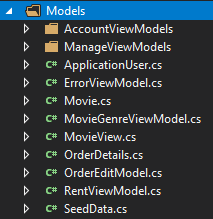


Rysunek 6

W związku z wykorzystaniem architektury MVC, kod był podzielony na Models, Controllers, Views, oraz pomocnicze Data, Services, Migrations, Extensions(**Rysunek 6**).

**Models**

W folderze z nazwą *Models*, znajdują się Modele danych, a dokładnie: Movie, *MovieGenreViewModel*, *MovieView*, *OrderDetails*, *OrderEditModel*, *RentViewModel*, *SeedData*. Pozostałę są wygenerowane automatycznie przez wykorzystanie frameworku **Identity**.(**Rysunek 7**).

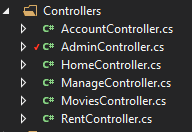


Rysunek 7

**Controllers**

W folderze z nazwą *Controllers* znajdują się kontrolerzy: *AccountController*(wygenerowany przez **Identity**)*, AdminController, HomeController, ManageController, MoviesController, RentController.*

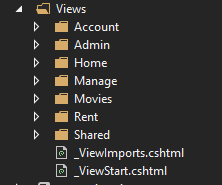
(**Rysunek 8**)



Rysunek 8

**Views**

W folderze z nazwą *Views* znajdują się foldery z widokami dla każdego kontrolera(*Account* oraz *Manage* był wygenerowany przez **Identity**): Admin(Create, Delete, Details, Edit, EditOrder, Index), Home(About, Contact, Index), Movies(Index), Rent(Index, MyRent, MyRentAdmin).



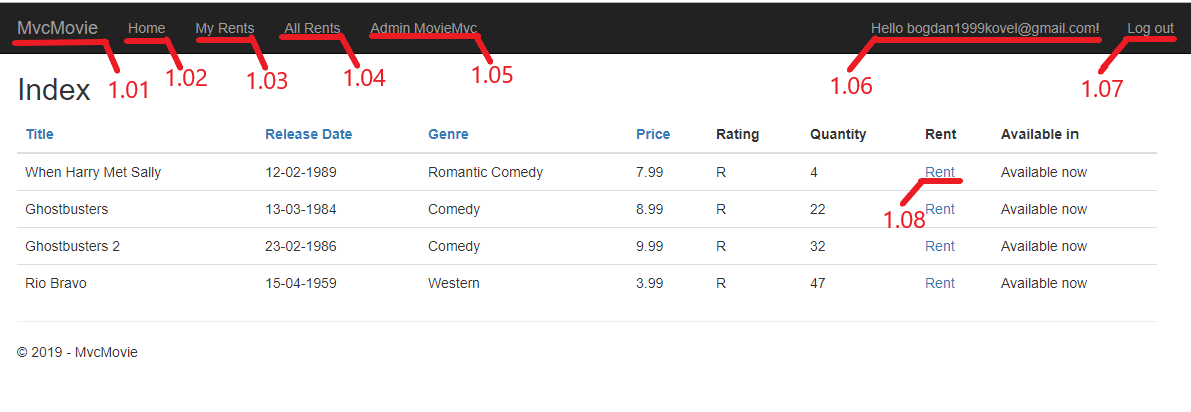
Rysunek 9

**Inne elementy**

W zakładce *wwwroot* znajdują się wszystkie styli, skrypty JavaScript oraz inne pliki potrzebne dla strony. W wwwroot w folderze Views są foldery z skryptami dla każdego widoku gdzie oni były stworzone.

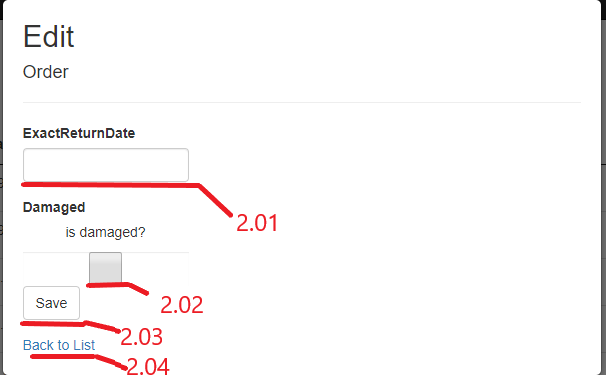
**6.** Prezentacja warstwy użytkowej projektu

**Interfejs użytkownika**



* 1. – link na główną stronę z listą filmów.
  2. – link na pierwszą stronę.
  3. – link na stronę z listą wypożyczeni zalogowanego użytkownika.
  4. – link na stronę z listą zamówień wszystkich użytkowników.
  5. – link na stronę z listą filmów, w jakiej można edytować filmy.

**Modalne okienko edycji zamówień**

****

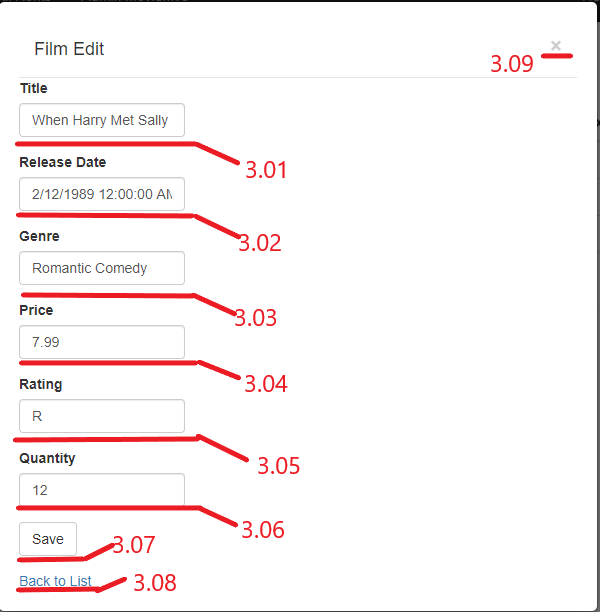
2.01 – pole do wpisania daty zwrotu filmu.

2.02 – checkbox, jaki oznacza czy był film uszkodzony czy nie.

2.03 – przycisk do zapisania zmian.

2.04 – link na stronę z listą zamówień.

**Modalne okienko edycji filmów**



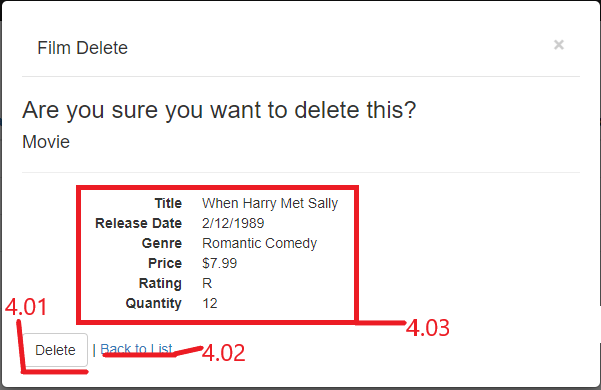
Od 3.01 do 3.06 – są to pola do wpisania danych filmów.

3.07 – przycisk do zapisania zmian.

3.08 – link na stronę z listą filmów admina.

3.09 – przycisk zamknięcia okna modalnego.

**Modalne okienko usunięcia filmów**

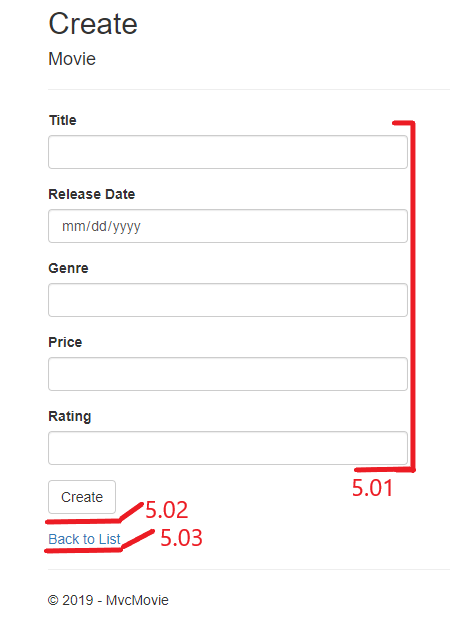


4.01 – przycisk dla usunięcia filma.

4.02 - link na stronę z listą filmów admina.

4.03 – informacja o filmie, jaki użytkownik chcę usunąć.

**Modalne okienko usunięcia filmów**



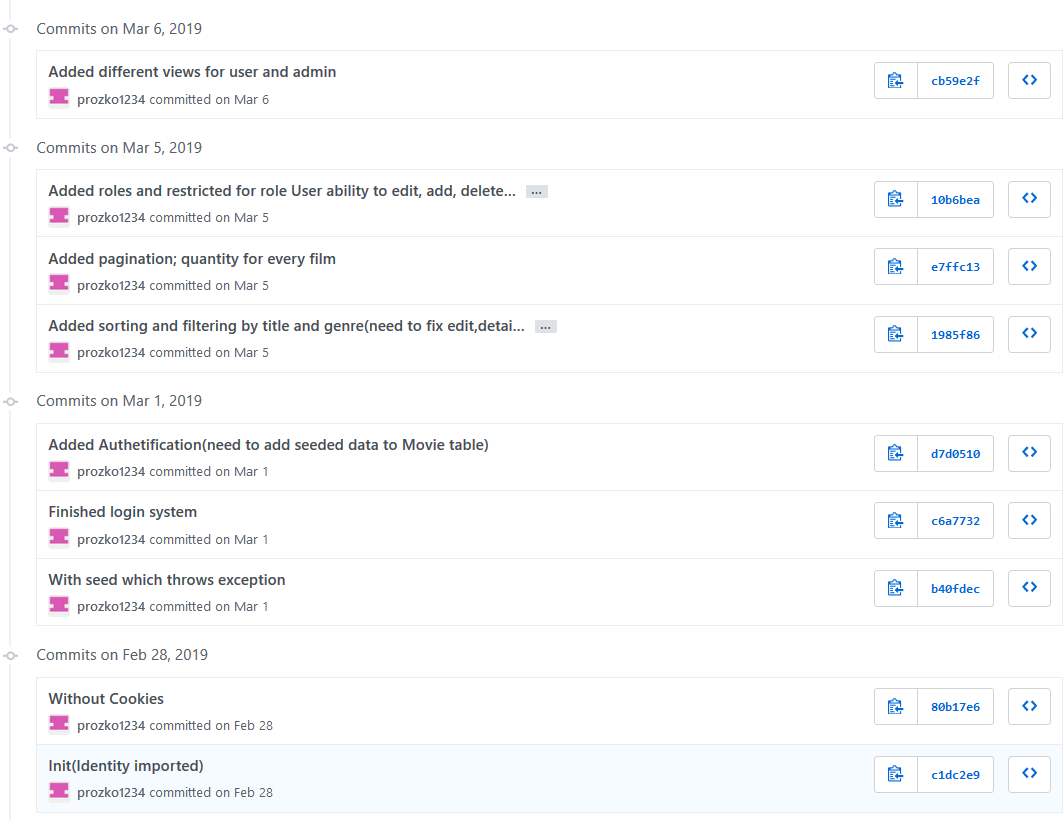
5.01 – pola dla wpisania danych o filmie.

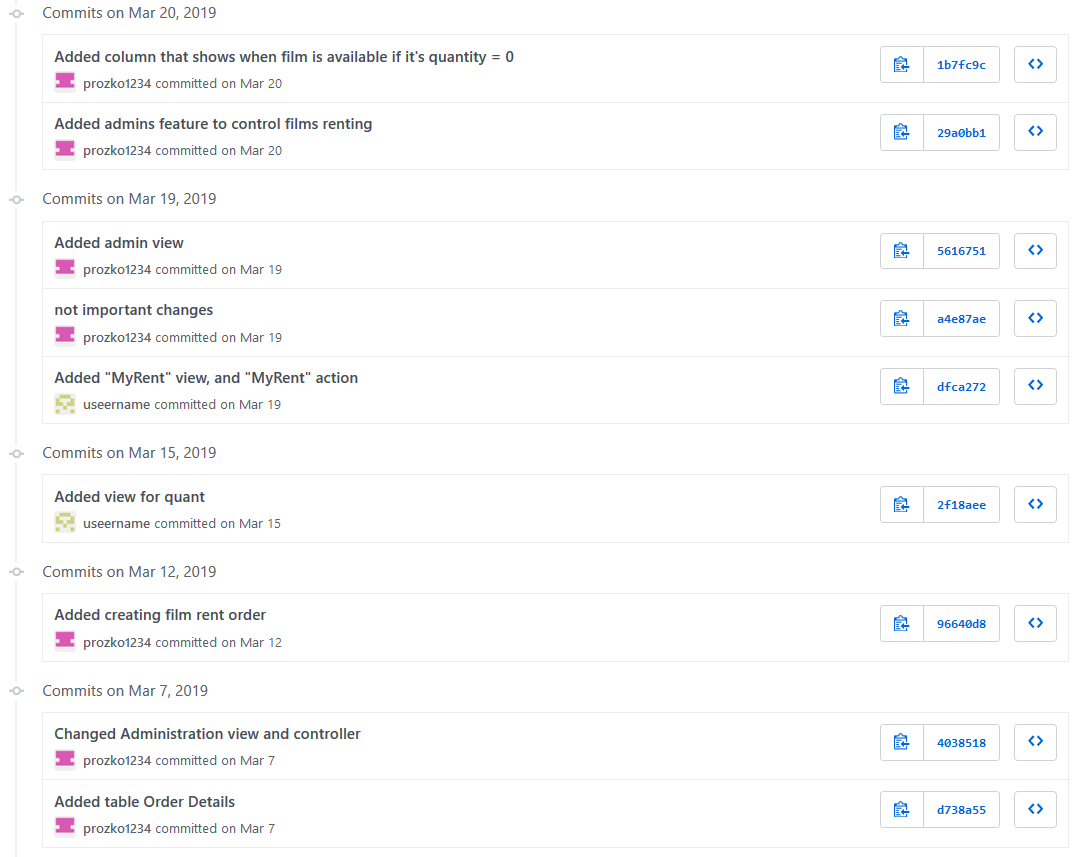
5.02 – przycisk dla dodania filmu.

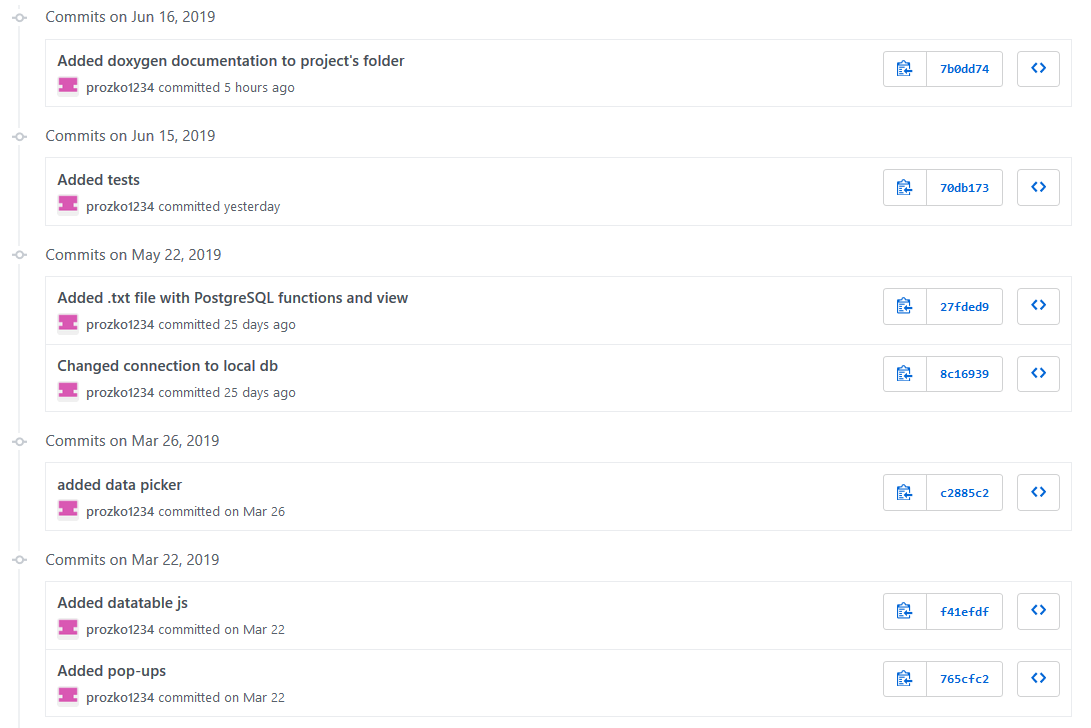
5.03 – link na stronę z listą filmów admina.

**7.** System kontroli wersji

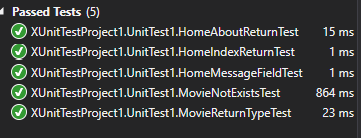
Projekt był tworzony z wykorzystaniem systemu kontroli wersji git. Hostingowy serwis – GitHub w prywatnym repozytorium. Link: <https://github.com/prozko1234/MvcMovie> .







**8.** Raporty z testów



**9.** Materiały źródłowe

1. <https://docs.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/?view=aspnetcore-2.2> (16.06.2019)

# 2. Programming C# 5.0: Building Windows 8, Web, and Desktop Applications for the .NET 4.5 Framework - 1st Edition, Kindle Edition.

# 3. <https://docs.microsoft.com/en-us/dotnet/csharp/>(16.06.2019)

**10.** Dokumentacja w postaci Html

<https://drive.google.com/open?id=1NzSRVln29UwjTYv9vgJWAgJZ-FPyGofO> (16.06.2019)