

# Dokumentacja projektu – MvcAppMovies (ASP.NET MVC)

Autor: Patryk Figas

Data: 02.01.2026

## SPIS TREŚCI:

1. Informacje ogólne.....	2
2. Architektura aplikacji (MVC) .....	3
3. Struktura projektu.....	3
4. Model danych.....	3
5. Warstwa dostępu do danych – Entity Framework Core.....	4
6. Kontroler MoviesController.....	5
7. Widoki aplikacji .....	6
8. Routing .....	9
9. Baza danych.....	9
10. Funkcjonalności aplikacji .....	10
11. Wnioski końcowe.....	12
12. Aktualizacje i rozszerzenia projektu (Update 1.0) .....	12
12.1 Ulepszenia interfejsu użytkownika (UI).....	13
12.2 Ulepszenia tabeli z listą filmów .....	14
12.3 Modal potwierdzania usuwania.....	14
12.4 Ulepszenia formularzy.....	15
12.5 Spełnienie wymagań projektowych.....	16
12.5.1 Potwierdzenie usunięcia filmu .....	17
12.5.2 Wywołanie funkcji przy przycisku „Usuń” .....	17
12.5.3 Funkcje opcjonalne (dodatkowe punkty).....	17
Walidacja formularzy .....	18
Komunikat po zapisaniu danych.....	18
12.5.4 Responsywność aplikacji .....	18
12.5.5 Podsumowanie końcowe.....	19

# 1. Informacje ogólne

**Nazwa projektu:** MvcAppMovies

**Technologie:**

- C#
- ASP.NET Core MVC
- Entity Framework Core
- SQLite

**Opis projektu:**

Projekt MvcAppMovies to aplikacja webowa umożliwiająca zarządzanie listą filmów.

Użytkownik może dodawać nowe filmy, edytować istniejące, usuwać je oraz przeglądać szczegóły wybranego filmu.

Dane są zapisywane w relacyjnej bazie danych SQLite.

# 2. Architektura aplikacji (MVC)

Aplikacja została wykonana zgodnie ze wzorcem architektonicznym **MVC (Model–View–Controller)**.

**Model**

Odpowiada za reprezentację danych oraz logikę biznesową aplikacji.

**View**

Odpowiada za prezentację danych użytkownikowi (interfejs graficzny).

**Controller**

Pośredniczy pomiędzy modelem a widokiem, obsługuje żądania użytkownika.

## 3. Struktura projektu

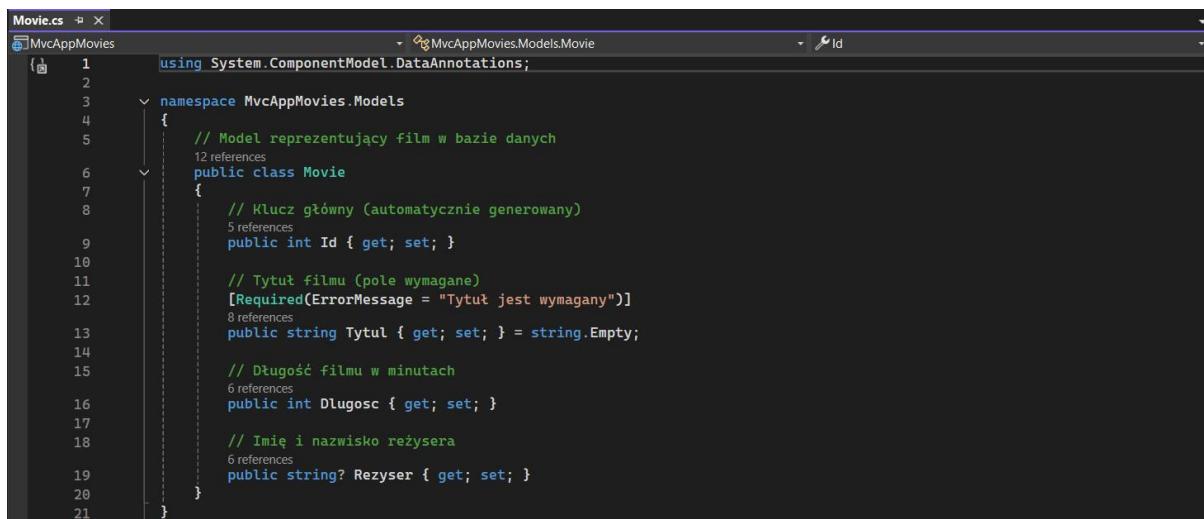
Projekt składa się z następujących katalogów:

- **Controllers** – zawiera kontrolery aplikacji
- **Models** – zawiera klasy modeli danych
- **Data** – zawiera konfigurację bazy danych
- **Views** – zawiera widoki Razor (.cshtml)
- **wwwroot** – pliki statyczne (CSS, JS, biblioteki)
- **Migrations** – migracje Entity Framework Core

## 4. Model danych

### Klasa Movie

Model Movie reprezentuje pojedynczy film w bazie danych.



```
Movie.cs -> X
MvcAppMovies <-> MvcAppMovies.Models.Movie <-> Id
1   using System.ComponentModel.DataAnnotations;
2
3   namespace MvcAppMovies.Models
4   {
5       // Model reprezentujący film w bazie danych
6       // 12 references
7       public class Movie
8       {
9           // Klucz główny (automatycznie generowany)
10          // 5 references
11          public int Id { get; set; }
12
13          // Tytuł filmu (pole wymagane)
14          // [Required(ErrorMessage = "Tytuł jest wymagany")]
15          // 8 references
16          public string Titul { get; set; } = string.Empty;
17
18          // Długość filmu w minutach
19          // 6 references
20          public int Dlugosc { get; set; }
21
22          // Imię i nazwisko reżysera
23          // 6 references
24          public string? Rezyser { get; set; }
25      }
26  }
```

### Pola modelu:

- Id – klucz główny (automatycznie generowany)
- Tytuł – tytuł filmu

- Dlugosc – długość filmu w minutach
- Rezyser – imię i nazwisko reżysera

Model wykorzystuje atrybuty walidacji, np. [Required], aby zapewnić poprawność danych wprowadzanych przez użytkownika.

## 5. Warstwa dostępu do danych – Entity Framework Core

Aplikacja korzysta z **Entity Framework Core** jako warstwy dostępu do danych.

### AppDbContext

```

AppDbContext.cs  ✘
MvcAppMovies
  1  using Microsoft.EntityFrameworkCore;
  2  using MvcAppMovies.Models;
  3  using System.Collections.Generic;
  4
  5  namespace MvcAppMovies.Data
  6  {
  7      // Klasa kontekstu bazy danych (Entity Framework Core)
  8      public class AppDbContext : DbContext
  9      {
 10          // Konstruktor przyjmujący opcję kontekstu
 11          public AppDbContext(DbContextOptions<AppDbContext> options)
 12              : base(options)
 13          {
 14          }
 15
 16          // Reprezentacja tabeli Movies w bazie danych
 17          public DbSet<Movie> Movies => Set<Movie>();
 18      }
 19  }
 20

```

Klasa AppDbContext dziedziczy po DbContext i odpowiada za:

- konfigurację połączenia z bazą SQLite,
- mapowanie modelu Movie na tabelę w bazie danych.

Dane są przechowywane w pliku **movies.db**.

## 6. Kontroler MoviesController

Kontroler MoviesController odpowiada za obsługę operacji CRUD na filmach.

Dostępne akcje:

- Index – wyświetla listę filmów

```
18
19
20     // Wyświetlenie listy filmów
21     public async Task<IActionResult> Index()
22     {
23         return View(await _context.Movies.ToListAsync());
24     }
```

- Details – wyświetla szczegóły filmu

```
70
71
72     // Wyświetlenie szczegółów filmu
73     public async Task<IActionResult> Details(int id)
74     {
75         var movie = await _context.Movies.FindAsync(id);
76         if (movie == null)
77             return NotFound();
78
79         return View(movie);
80     }
```

- Add – dodawanie nowego filmu

```
24
25
26     // Formularz dodawania nowego filmu
27     public IActionResult Add() => View();
28
29     // Obsługa dodawania filmu
30     [HttpPost]
31     [ValidateAntiForgeryToken]
32     public async Task<IActionResult> Add(Movie movie)
33     {
34         if (!ModelState.IsValid) return View(movie);
35
36         _context.Movies.Add(movie);
37         await _context.SaveChangesAsync();
38         return RedirectToAction(nameof(Index));
39     }
```

- Edit – edycja istniejącego filmu

```
39      // Formularz edycji filmu
40      0 references
41      public async Task<IActionResult> Edit(int id)
42      {
43          var movie = await _context.Movies.FindAsync(id);
44          if (movie == null) return NotFound();
45          return View(movie);
46      }
47
48      // Obsługa edycji filmu
49      [HttpPost]
50      [ValidateAntiForgeryToken]
51      0 references
52      public async Task<IActionResult> Edit(Movie movie)
53      {
54          if (!ModelState.IsValid) return View(movie);
55
56          _context.Movies.Update(movie);
57          await _context.SaveChangesAsync();
58          return RedirectToAction(nameof(Index));
59      }
```

- Delete – usuwanie filmu

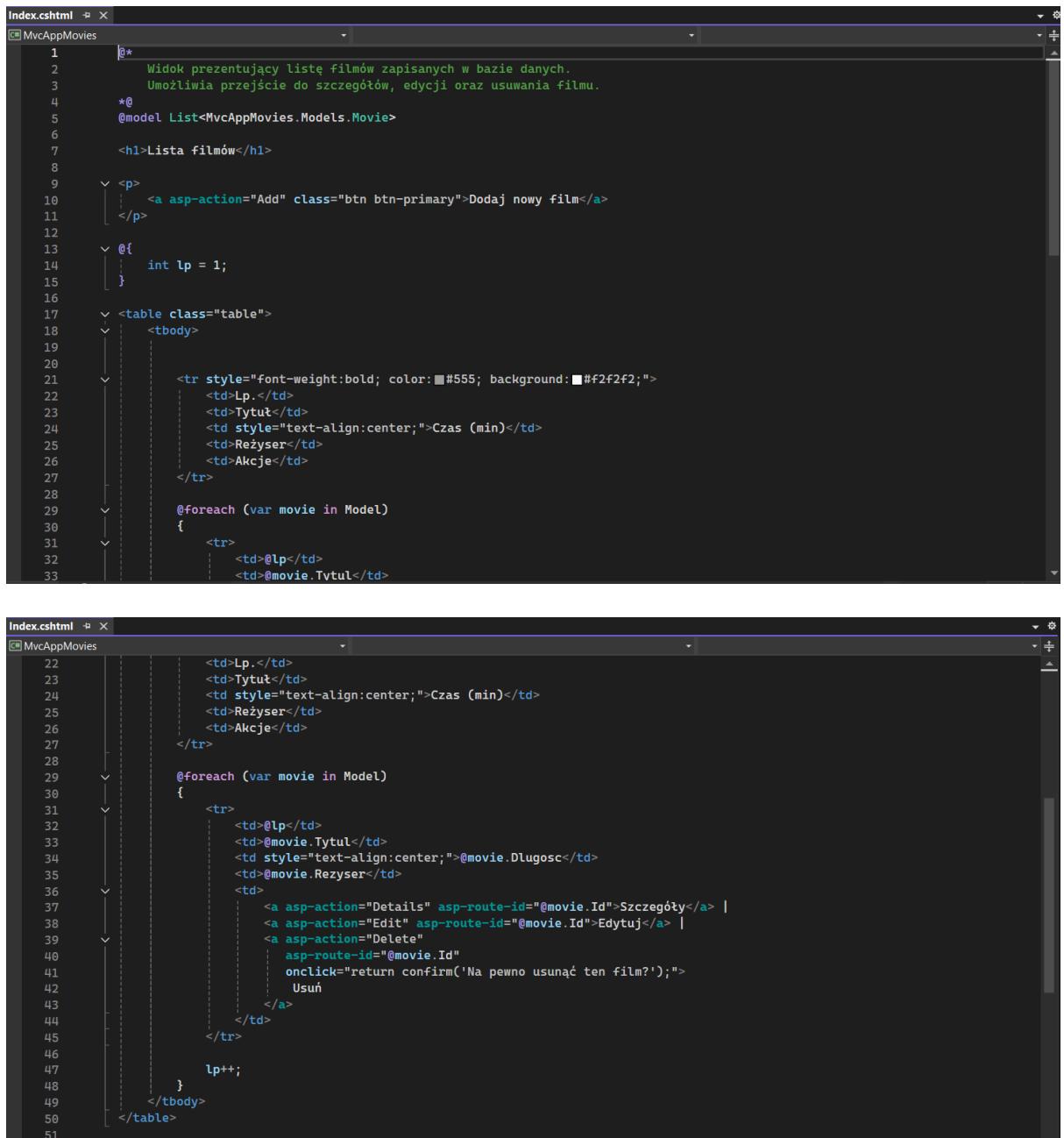
```
59
60      // Usunięcie filmu
61      0 references
62      public async Task<IActionResult> Delete(int id)
63      {
64          var movie = await _context.Movies.FindAsync(id);
65          if (movie == null) return NotFound();
66
67          _context.Movies.Remove(movie);
68          await _context.SaveChangesAsync();
69          return RedirectToAction(nameof(Index));
70      }
```

Kontroler korzysta bezpośrednio z kontekstu AppDbContext.

## 7. Widoki aplikacji

Aplikacja zawiera następujące widoki:

- **Index.cshtml** – lista filmów w formie tabeli



```
Index.cshtml  ✘ ×
MvcAppMovies
1  @*
2      Widok prezentujący listę filmów zapisanych w bazie danych.
3      Umożliwia przejście do szczegółów, edycji oraz usuwania filmu.
4  *@
5  @model List<MvcAppMovies.Models.Movie>
6
7  <h1>Lista filmów</h1>
8
9  <p>
10     <a href="#" asp-action="Add" class="btn btn-primary">Dodaj nowy film</a>
11 </p>
12
13  @{
14      int lp = 1;
15  }
16
17  <table class="table">
18      <tbody>
19
20          <tr style="font-weight:bold; color:#555; background-color:#f2f2f2;">
21              <td>Lp.</td>
22              <td>Tytuł</td>
23              <td style="text-align:center;">Czas (min)</td>
24              <td>Reżyser</td>
25              <td>Akcje</td>
26          </tr>
27
28          @foreach (var movie in Model)
29          {
30              <tr>
31                  <td>@lp</td>
32                  <td>@movie.Tytul</td>
```

```
Index.cshtml  ✘ ×
MvcAppMovies
22          |          |          |          |          |          |
23          |          |          |          |          |          |
24          |          |          |          |          |          |
25          |          |          |          |          |          |
26          |          |          |          |          |          |
27          |          |          |          |          |          |
28          |          |          |          |          |          |
29          |          |          |          |          |          |
30          |          |          |          |          |          |
31          |          |          |          |          |          |
32          |          |          |          |          |          |
33          |          |          |          |          |          |
34          |          |          |          |          |          |
35          |          |          |          |          |          |
36          |          |          |          |          |          |
37          |          |          |          |          |          |
38          |          |          |          |          |          |
39          |          |          |          |          |          |
40          |          |          |          |          |          |
41          |          |          |          |          |          |
42          |          |          |          |          |          |
43          |          |          |          |          |          |
44          |          |          |          |          |          |
45          |          |          |          |          |          |
46          |          |          |          |          |          |
47          |          |          |          |          |          |
48          |          |          |          |          |          |
49          |          |          |          |          |          |
50          |          |          |          |          |          |
51          |          |          |          |          |          |
```

- **Add.cshtml** – formularz dodawania filmu

```

Add.cshtml □ X
MvcAppMovies
1   @model MvcAppMovies.Models.Movie
2
3   <h1>Dodaj film</h1>
4
5   <form asp-action="Add" method="post">
6     <div>
7       <label asp-for="Tytul"></label><br />
8       <input asp-for="Tytul" />
9       <span asp-validation-for="Tytul"></span>
10      </div>
11
12      <div>
13        <label asp-for="Dlugosc"></label><br />
14        <input asp-for="Dlugosc" type="number" />
15      </div>
16
17      <div>
18        <label asp-for="Rezyser"></label><br />
19        <input asp-for="Rezyser" />
20      </div>
21
22      <button type="submit">Zapisz</button>
23    </form>
24

```

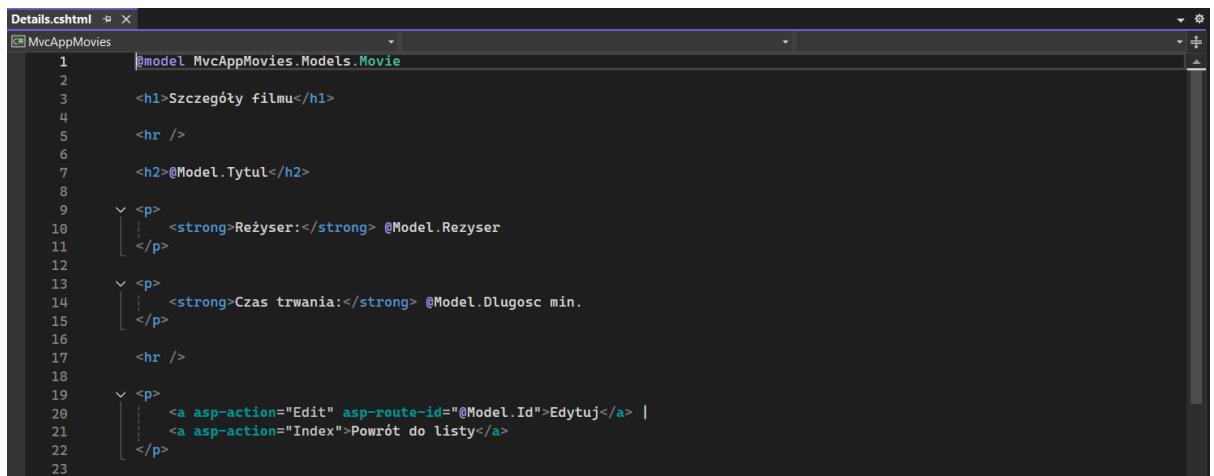
- **Edit.cshtml** – formularz edycji filmu

```

Edit.cshtml □ X
MvcAppMovies
1   @model MvcAppMovies.Models.Movie
2
3   <h1>Edytuj film</h1>
4
5   <form asp-action="Edit" method="post">
6     <input type="hidden" asp-for="Id" />
7
8     <div>
9       <label asp-for="Tytul"></label><br />
10      <input asp-for="Tytul" />
11      <span asp-validation-for="Tytul"></span>
12    </div>
13
14    <div>
15      <label asp-for="Dlugosc"></label><br />
16      <input asp-for="Dlugosc" type="number" />
17    </div>
18
19    <div>
20      <label asp-for="Rezyser"></label><br />
21      <input asp-for="Rezyser" />
22    </div>
23
24    <button type="submit">Zapisz zmiany</button>
25
26

```

- **Details.cshtml** – szczegóły wybranego filmu



```

1  @model MvcAppMovies.Models.Movie
2
3  <h1>Szczegóły filmu</h1>
4  <hr />
5  <h2>@Model.Tytul</h2>
6
7  <p>
8    <strong>Reżyser:</strong> @Model.Rezyser
9  </p>
10 <p>
11   <strong>Czas trwania:</strong> @Model.Dlugosc_min.
12 </p>
13 <hr />
14 <p>
15   <a href="#" asp-action="Edit" asp-route-id="@Model.Id">Edytuj</a> |
16   <a href="#" asp-action="Index">Powrót do listy</a>
17 </p>
18
19
20
21
22
23

```

Widoki zostały wykonane przy użyciu Razor Pages oraz HTML.

## 8. Routing

Domyślna trasa aplikacji została skonfigurowana w pliku Program.cs:

{controller=Movies}/{action=Index}/{id?}

```

39  app.MapControllerRoute(
40    name: "default",
41    pattern: "{controller=Movies}/{action=Index}/{id?}");
42
43    app.Run();
44
45
46
47
48

```

Oznacza to, że po uruchomieniu aplikacji użytkownik zostaje automatycznie przekierowany do listy filmów.

## 9. Baza danych

Aplikacja wykorzystuje **SQLite** jako relacyjną bazę danych. Dane są zapisywane trwale w pliku movies.db

Struktura bazy jest zarządzana za pomocą migracji EF Core.

## 10. Funkcjonalności aplikacji

Aplikacja umożliwia:

- dodawanie nowych filmów,

The screenshot shows a web page titled "Dodaj film" (Add movie). It contains three input fields: "Tytuł" (Title) with the value "Pulp Fiction", "Długość" (Duration) with the value "154", and "Reżyser" (Director) with the value "Quentin Tarantino". Below the inputs is a green "Zapisz" (Save) button. At the top left of the page, there is a header with the application name "MvcAppMovies" and links to "Home" and "Privacy". At the bottom center of the page, there is a copyright notice: "© 2025 - MvcAppMovies - [Privacy](#)".

- edytowanie istniejących filmów,

MvcAppMovies Home Privacy

Edytuj film

Tytuł  
Gladiator

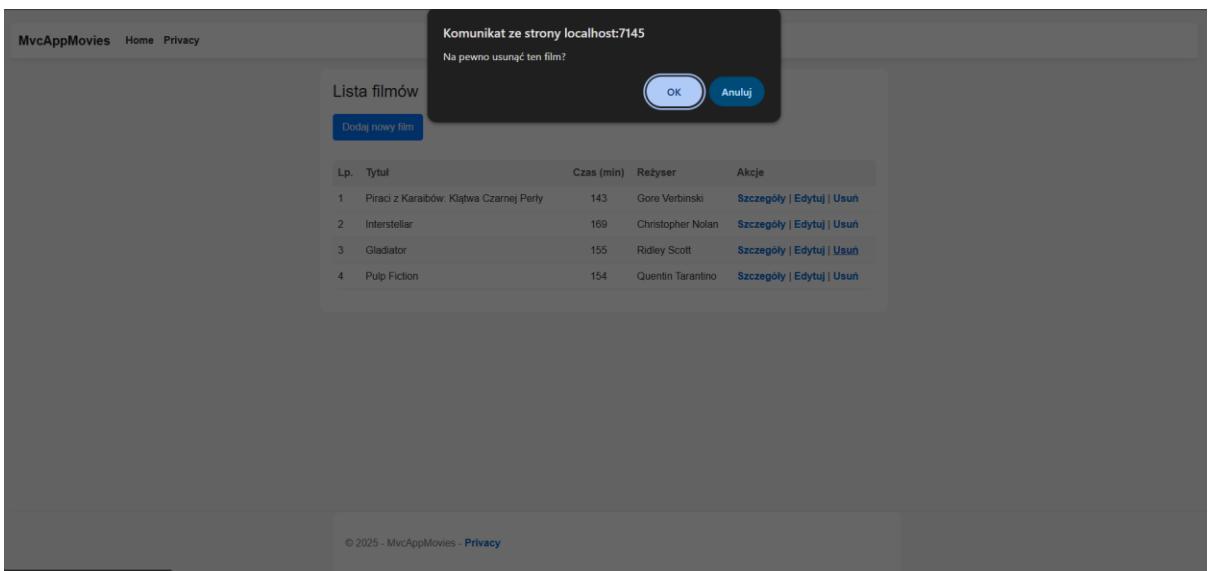
Długość  
155

Reżyser  
Ridley Scott

Zapisz zmiany

© 2025 - MvcAppMovies - [Privacy](#)

- usuwanie filmów,



- wyświetlanie listy filmów,

The screenshot shows a web application interface for managing movies. At the top, there's a navigation bar with links for 'MvcAppMovies', 'Home', and 'Privacy'. Below the navigation is a section titled 'Lista filmów' (List of movies) with a blue button labeled 'Dodaj nowy film' (Add new movie). A table displays four movies with columns: 'Lp.' (Index), 'Tytuł' (Title), 'Czas (min)' (Duration), 'Reżyser' (Director), and 'Akcje' (Actions). The movies listed are: 'Piraci z Karaibów: Klątwa Czarnej Perły' (Index 1, Director Gore Verbinski), 'Interstellar' (Index 2, Director Christopher Nolan), 'Gladiator' (Index 3, Director Ridley Scott), and 'Pulp Fiction' (Index 4, Director Quentin Tarantino). Each movie entry includes a link to 'Szczegóły | Edytuj | Usuń' (Details | Edit | Delete).

- podgląd szczegółów filmu.

The screenshot shows a detailed view of a movie record. At the top, it says 'Szczegóły filmu' (Movie details) for 'Piraci z Karaibów: Klątwa Czarnej Perły'. It lists the director as 'Reżyser: Gore Verbinski' and the duration as 'Czas trwania: 143 min.'. At the bottom of this section are two blue links: 'Edytuj' (Edit) and 'Powrót do listy' (Return to list). The footer of the page includes the copyright notice '© 2025 - MvcAppMovies - Privacy'.

## 11. Wnioski końcowe

Projekt MvcAppMovies został zrealizowany zgodnie z założeniami architektury MVC. Zastosowanie Entity Framework Core oraz bazy danych SQLite umożliwia trwałe przechowywanie danych oraz łatwe zarządzanie strukturą bazy.

## 12. Aktualizacje i rozszerzenia projektu (Update 1.0)

W trakcie realizacji projektu aplikacja **MvcAppMovies** została rozbudowana o dodatkowe elementy, które poprawiają estetykę interfejsu oraz komfort użytkowania.

Technologie:

- HTML5,
- CSS3,
- JavaScript,
- Bootstrap 5.

## 12.1 Ulepszenia interfejsu użytkownika (UI)

W projekcie wprowadzono własny motyw graficzny, który:

- wykorzystuje zmienne CSS do zarządzania kolorami,
- zapewnia spójny wygląd wszystkich widoków,
- poprawia czytelność treści oraz estetykę aplikacji.

Zastosowano m.in.:

- ciemne tło aplikacji,
- jasny kontrastowy tekst,
- wyraźne kolory akcji (dodawanie, edycja, usuwanie).

The screenshot shows a dark-themed user interface for a movie application. At the top, there's a navigation bar with the title 'MvcAppMovies'. Below it is a button labeled '+ Dodaj nowy film'. The main content area is titled 'Lista filmów' and contains a table with the following data:

Lp.	Tytuł	Czas (min)	Reżyser	Akcje
1	Piraci z Karaibów: Kąpiel Czarnej Perły	143	Gore Verbinski	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
2	Interstellar	169	Christopher Nolan	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
3	Gladiator	155	Ridley Scott	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
4	Pulp Fiction	154	Quentin Tarantino	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
5	Incepja	148	Christopher Nolan	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
6	Skazani na Shawshank	142	Frank Darabont	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
7	Matrix	136	Lana i Lilly Wachowski	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
8	Ojciec chrzestny	175	Francis Ford Coppola	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
9	Fight Club	139	David Fincher	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
10	Forrest Gump	142	Robert Zemeckis	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
11	Zielona mila	189	Frank Darabont	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
12	Django	165	Quentin Tarantino	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
13	Leon zawodowiec	110	Luc Besson	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
14	Siedem	127	David Fincher	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
15	Avatar	162	James Cameron	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
16	Titanic	195	James Cameron	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
17	Joker	122	Todd Phillips	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
18	Władca Pierścieni: Drużyna Pierścienia	178	Peter Jackson	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
19	Władca Pierścieni: Dwie Wieże	179	Peter Jackson	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>
20	Opowieść	172	Christopher Nolan	<button>Szczegóły</button> <button>Edytuj</button> <button>Usuń</button>

## 12.2 Ulepszenia tabeli z listą filmów

Widok listy filmów (Index.cshtml) został rozbudowany o:

- tabelę z ustalonymi szerokościami kolumn (colgroup),
- numerację wierszy (Lp.),

- poprawione wyrównanie nagłówków i danych,
- efekt hover zwiększający czytelność,
- komunikat informujący o pustej bazie danych.

Dzięki temu tabela jest bardziej czytelna.

## 12.3 Modal potwierdzania usuwania

Proces usuwania filmu został zabezpieczony poprzez zastosowanie **modala Bootstrap**:

- użytkownik musi potwierdzić operację usunięcia,
- w modalu wyświetlany jest tytuł usuwanego filmu, co zapobiega przypadkowemu usunięciu danych.

Rozwiążanie to poprawia bezpieczeństwo i UX aplikacji.

Gladiator	155	Ridley Scott	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Pulp Fiction	154	Quentin Tarantino	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Incepcja	148	Christopher Nolan	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Skazani na Shawshank			<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Matrix			<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Ojciec chrzestny			<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Fight Club			<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Forrest Gump			<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Zielona mila	189	Frank Darabont	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Django	165	Quentin Tarantino	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Leon zawodowiec	110	Luc Besson	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Według Hiedla	127	David Fincher	<a href="#">Szczegóły</a>	<a href="#">Edytuj</a>	<a href="#">Usuń</a>

Potwierdzenie usunięcia

Czy na pewno chcesz usunąć ten film?

Tytuł: Piraci z Karaibów: Klatwa Czarnej Perły

Anuluj
Usuń

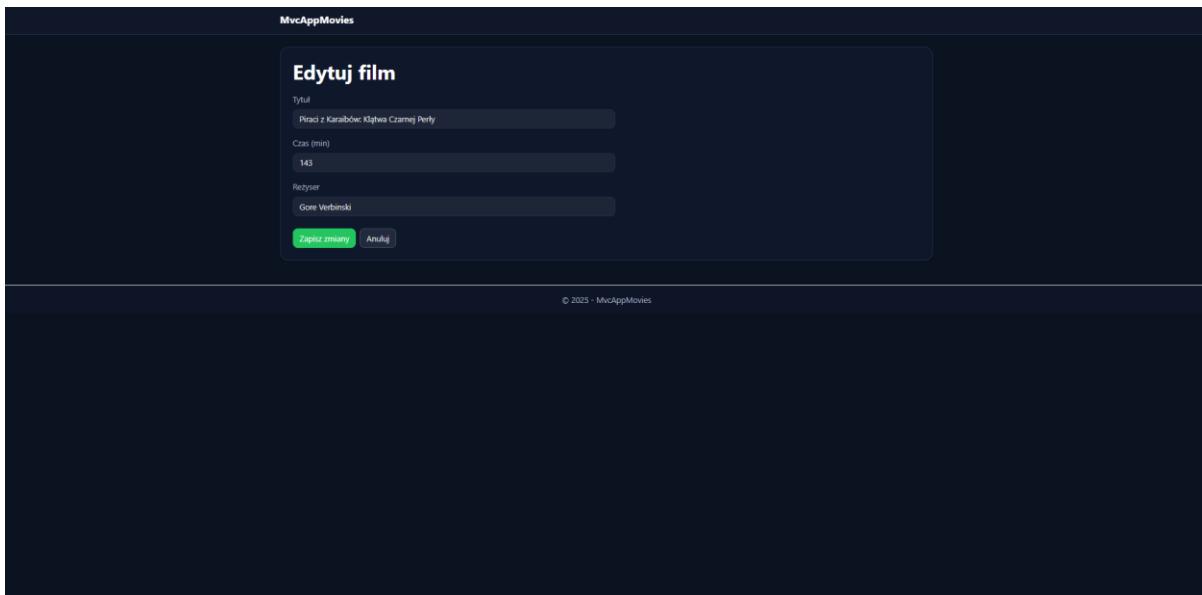
## 12.4 Ulepszenia formularzy

Formularze dodawania i edycji filmu zostały dopracowane poprzez:

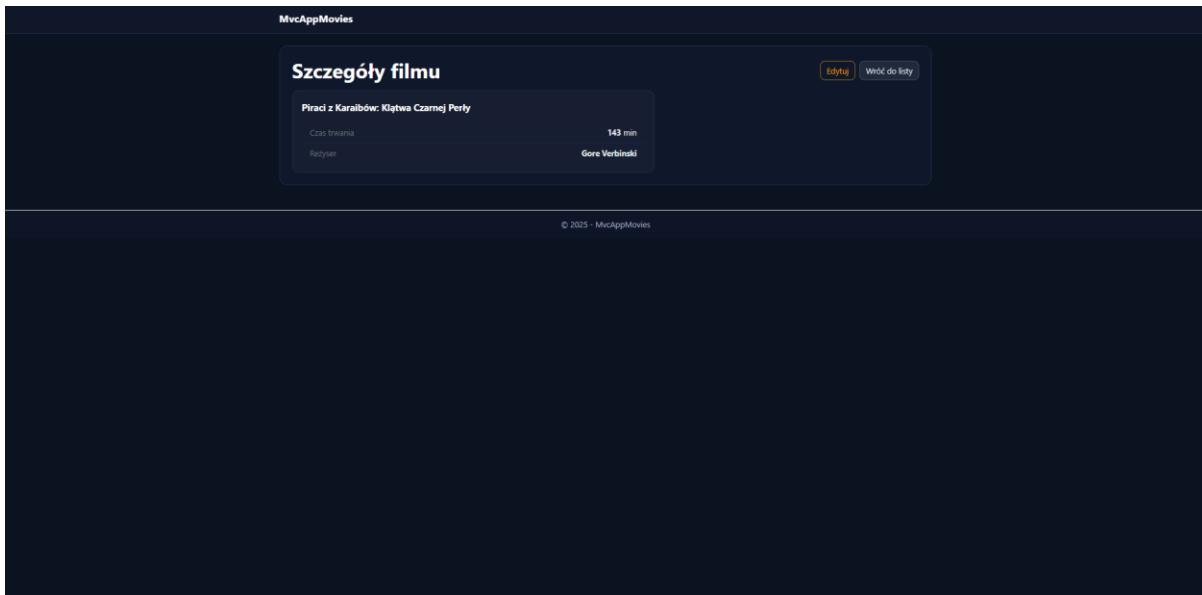
- validację danych po stronie klienta i serwera,
  - zabezpieczenie CSRF (AntiForgeryToken),
  - ograniczenie szerokości formularzy dla lepszej ergonomii,
  - spójny styl zgodny z motywem aplikacji.
- 
- Dodawanie filmu

The screenshot shows a modal dialog titled "Dodaj film" (Add movie) centered on a dark-themed web page. The form contains three input fields: "Tytuł" (Title), "Czas (min)" (Duration), and "Reżyser" (Director). Below the fields are two buttons: a green "Zapisz" (Save) button and a grey "Anuluj" (Cancel) button. At the bottom of the page, a small copyright notice reads "© 2025 - MvcAppMovies".

- Edycja filmu



- Szczegóły



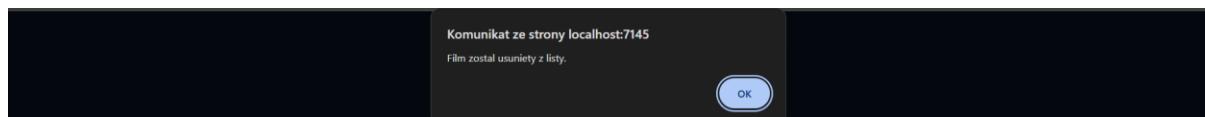
Dodano również dostęp do edycji z poziomu okna szczegółów filmu.

## 12.5 Spełnienie wymagań projektowych

Aplikacja **MvcAppMovies** spełnia wszystkie wymagania określone w założeniach projektu.

### 12.5.1 Potwierdzenie usunięcia filmu

Proces usuwania filmu jest zabezpieczony poprzez zastosowanie **okna modalnego**. Po kliknięciu przycisku „Usuń” użytkownik musi potwierdzić operację, co zapobiega przypadkowemu usunięciu danych, a po potwierdzeniu dostaje komunikat potwierdzający, jeśli akcja udała się.



### 12.5.2 Wywołanie funkcji przy przycisku „Usuń”

Przycisk „Usuń” wywołuje funkcję po stronie klienta (JavaScript), która:

- pobiera identyfikator filmu,
  - wyświetla okno potwierdzenia,
  - po akceptacji wykonuje akcję usunięcia rekordu.

```
100      |          |          |      <!--
101      |          |          |          Usuwanie filmu:
102      |          |          |          zamiast bezpośredniego usunięcia
103      |          |          |          otwierany jest modal z potwierdzeniem
104      |          |          |      -->
105      |          |          |          <button type="button"
106      |          |          |          class="btn btn-sm btn-outline-danger"
107      |          |          |          data-bs-toggle="modal"
108      |          |          |          data-bs-target="#deleteModal"
109      |          |          |          data-id="@movie.Id"
110      |          |          |          data-title="@movie.Tytul">
111      |          |          |          Usuń
112      |          |          |      </button>

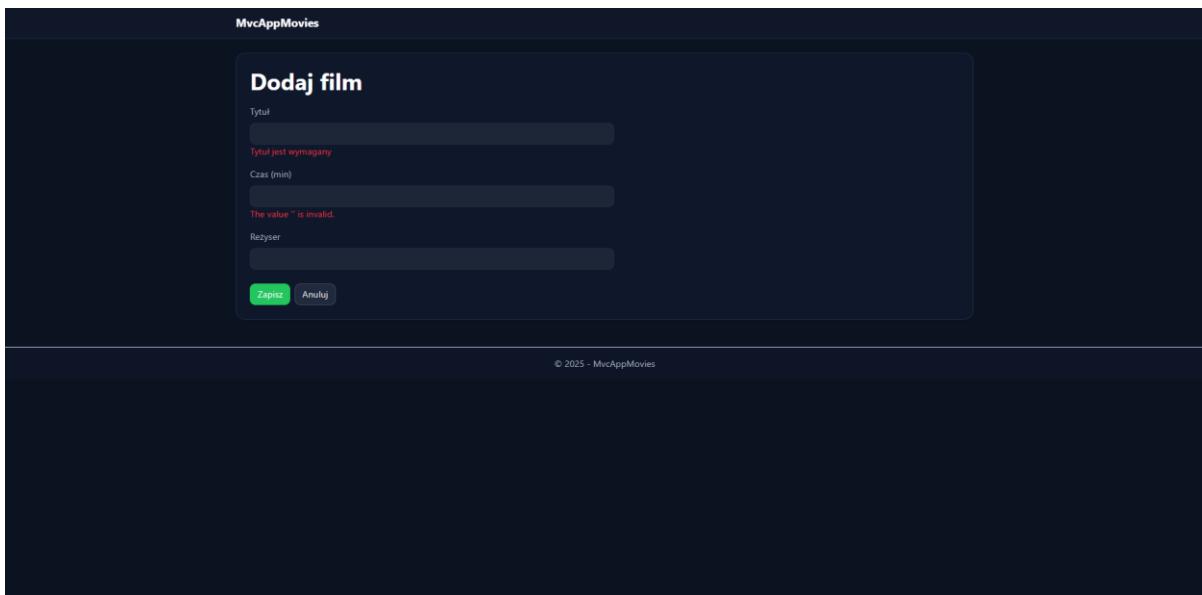
174      |          |          // Wykonuje się w momencie otwarcia modala
175      |          |          deleteModal.addEventListener('show.bs.modal', function (event) {
176      |          |          const button = event.relatedTarget;
177      |          |
178      |          |          // Pobranie danych filmu z atrybutów data-
179      |          |          const movieId = button.getAttribute('data-id');
180      |          |          const movieTitle = button.getAttribute('data-title') || '';
181      |          |
182      |          |          // Generowanie poprawnego linku do akcji Delete w kontrolerze
183      |          |          const baseUrl = '@Url.Action("Delete", "Movies", new { id = "_id_" })';
184      |          |          confirmBtn.href = baseUrl.replace('_id_', movieId);
```

### 12.5.3 Funkcje opcjonalne (dodatkowe punkty)

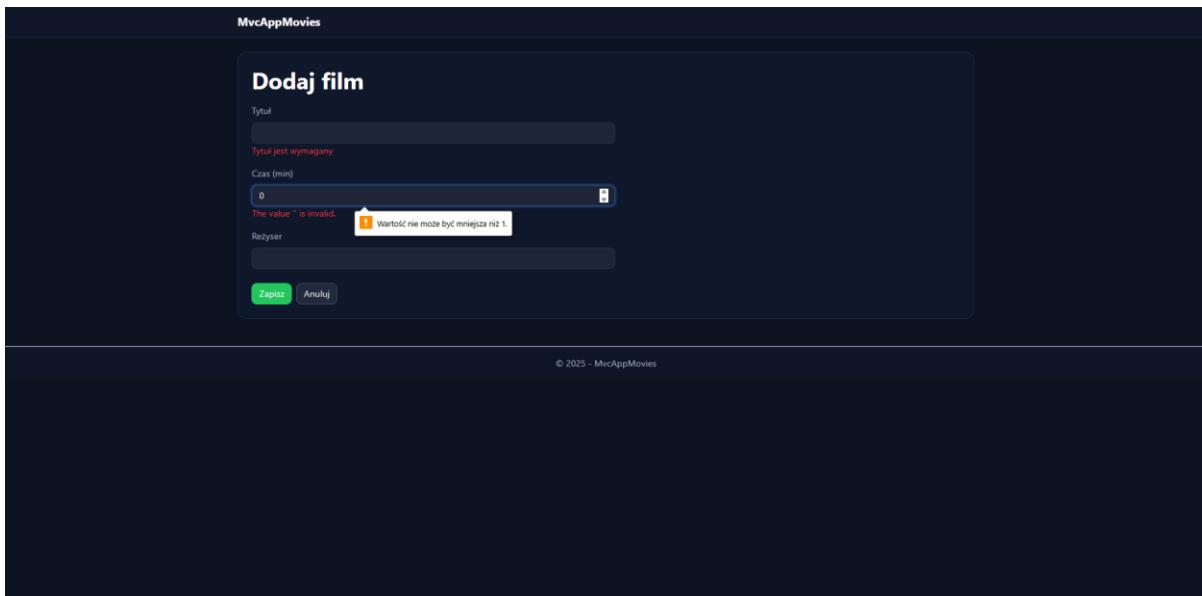
#### Walidacja formularzy

Formularze dodawania i edycji filmu posiadają walidację danych:

- sprawdzana jest poprawność danych po stronie klienta (HTML, JavaScript),
- walidacja wykonywana jest również po stronie serwera (ASP.NET MVC),
- przykładowo długość filmu musi być większa od zera.



The screenshot shows the 'Dodaj film' (Add movie) form in the MvcAppMovies application. The 'Tytuł' (Title) field is empty and highlighted in red, with the error message 'Tytuł jest wymagany' (Title is required) displayed below it. The 'Czas (min)' (Duration) field contains '0' and has a tooltip 'The value "0" is invalid.' with the additional message 'Wartość nie może być mniejsza niż 1.' (Value cannot be less than 1.) displayed below it. The 'Reżyser' (Director) field is empty. At the bottom are 'Zapisz' (Save) and 'Anuluj' (Cancel) buttons.

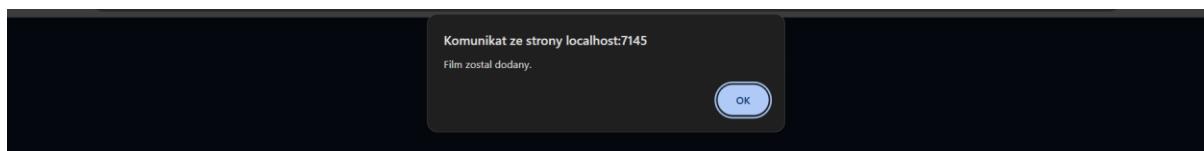


The screenshot shows the 'Dodaj film' (Add movie) form in the MvcAppMovies application. The 'Tytuł' (Title) field is empty and highlighted in red, with the error message 'Tytuł jest wymagany' (Title is required) displayed below it. The 'Czas (min)' (Duration) field contains '0' and has a tooltip 'The value "0" is invalid.' with the additional message 'Wartość nie może być mniejsza niż 1.' (Value cannot be less than 1.) displayed below it. The 'Reżyser' (Director) field is empty. At the bottom are 'Zapisz' (Save) and 'Anuluj' (Cancel) buttons.

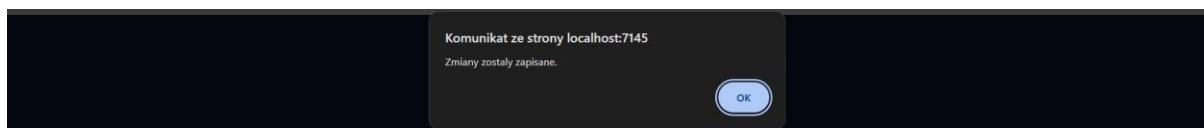
### *Komunikat po zapisaniu danych*

Po dodaniu lub edycji filmu użytkownik otrzymuje komunikat typu **alert()**, informujący o poprawnym zapisaniu danych.

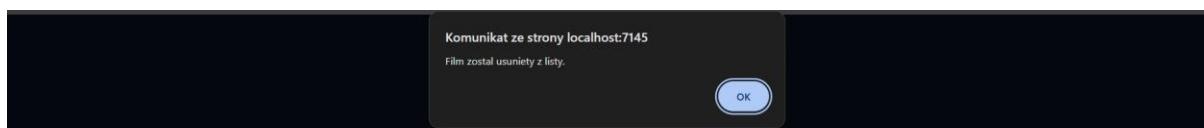
- Dodanie nowego filmu



- Zapis zmian



- Usunięcie filmu



### 12.5.4 Responsywność aplikacji

Aplikacja została zaprojektowana w sposób **responsywny**, dzięki czemu poprawnie wyświetla się na:

- komputerach stacjonarnych,
- tabletach,
- telefonach komórkowych.

Responsywność została osiągnięta poprzez wykorzystanie:

- frameworka **Bootstrap 5**,
- elastycznego układu (container, row, col),
- responsywnych komponentów interfejsu.

## 12.5.5 Podsumowanie końcowe

Projekt **MvcAppMovies** został rozwinięty o dodatkowe funkcje oraz poprawiony został wygląd aplikacji.

Kod aplikacji został:

- uporządkowany zgodnie z architekturą MVC,
- opatrzony czytelnymi komentarzami,
- podzielony logicznie na komponenty (Model, View, Controller),
- przygotowany w sposób umożliwiający dalszą rozbudowę projektu.