

## Spis treści

1.	Opis projektu.....	2
1.1.	Założenia i cele projektu.....	2
1.2.	Wykorzystane technologie.....	2
2.	Struktura bazy danych.....	3
2.1.	Diagram bazy .....	3
2.2.	Encje i relacje.....	<b>Błąd! Nie zdefiniowano zakładki.</b>
2.3.	Obiekty transferu danych .....	<b>Błąd! Nie zdefiniowano zakładki.</b>
2.4.	Paginacja, filtrowanie i sortowanie wyników .....	<b>Błąd! Nie zdefiniowano zakładki.</b>
3.	Kontrolery .....	<b>Błąd! Nie zdefiniowano zakładki.</b>
4.	Prezentacja aplikacji za pomocą Swagger .....	4
5.	Podsumowanie projektu.....	<b>Błąd! Nie zdefiniowano zakładki.</b>
6.	Netografia .....	<b>Błąd! Nie zdefiniowano zakładki.</b>

**To tylko FRAGMENTY dokumentacji, całość może zostać przesłana na prośbę podczas rekrutacji.**

# **1. Opis projektu**

## **1.1. Założenia i cele projektu**

Projekt ma na celu stworzenie API do obsługi systemu biblioteki szkolnej. API ma umożliwiać zarządzanie zasobami książkowymi oraz umożliwiać dodawanie danych dodatkowych np. egzemplarze książek.

Celem projektu jest:

- Utworzenie modelu bazy danych
- Stworzenie API do zarządzania danymi
- Prezentacja wizualna rozwiązania przy pomocy Swagger

## **1.2. Wykorzystane technologie**

Projekt utworzono w technologii ASP.NET Core 9 API. Całość działa w środowisku lokalnym z lokalną bazą danych Microsoft SQL Server. Projekt rozwijany był w Visual Studio 2022.

Do obsługi połączenia z bazą zainstalowano package *Entity Framework Core*. Jest to uproszczona, rozszerzalna i wieloplatformowa wersja open source popularnej technologii dostępu do danych Entity Framework.

Do prezentacji działania aplikacji użyto narzędzia *Swagger* - interaktywnego narzędzia służącego do dokumentowania i testowania REST API. Automatycznie generuje czytelny interfejs użytkownika na podstawie kodu aplikacji, umożliwiając przeglądanie dostępnych endpointów, ich parametrów oraz wykonywanie zapytań bezpośrednio z poziomu przeglądarki.

W projekcie zaimplementowano trzy kontrolery: BookController, CatalogController oraz CopyController. Odpowiadają one kolejno za obsługę operacji CRUD na książkach, operacje GET wykorzystywane w widoku katalogu książek (dla czytelnika) oraz operacje CRUD na kopiach (fizycznych egzemplarzach) książek. Każdy z kontrolerów realizuje określone metody http.

## 2. Struktura bazy danych

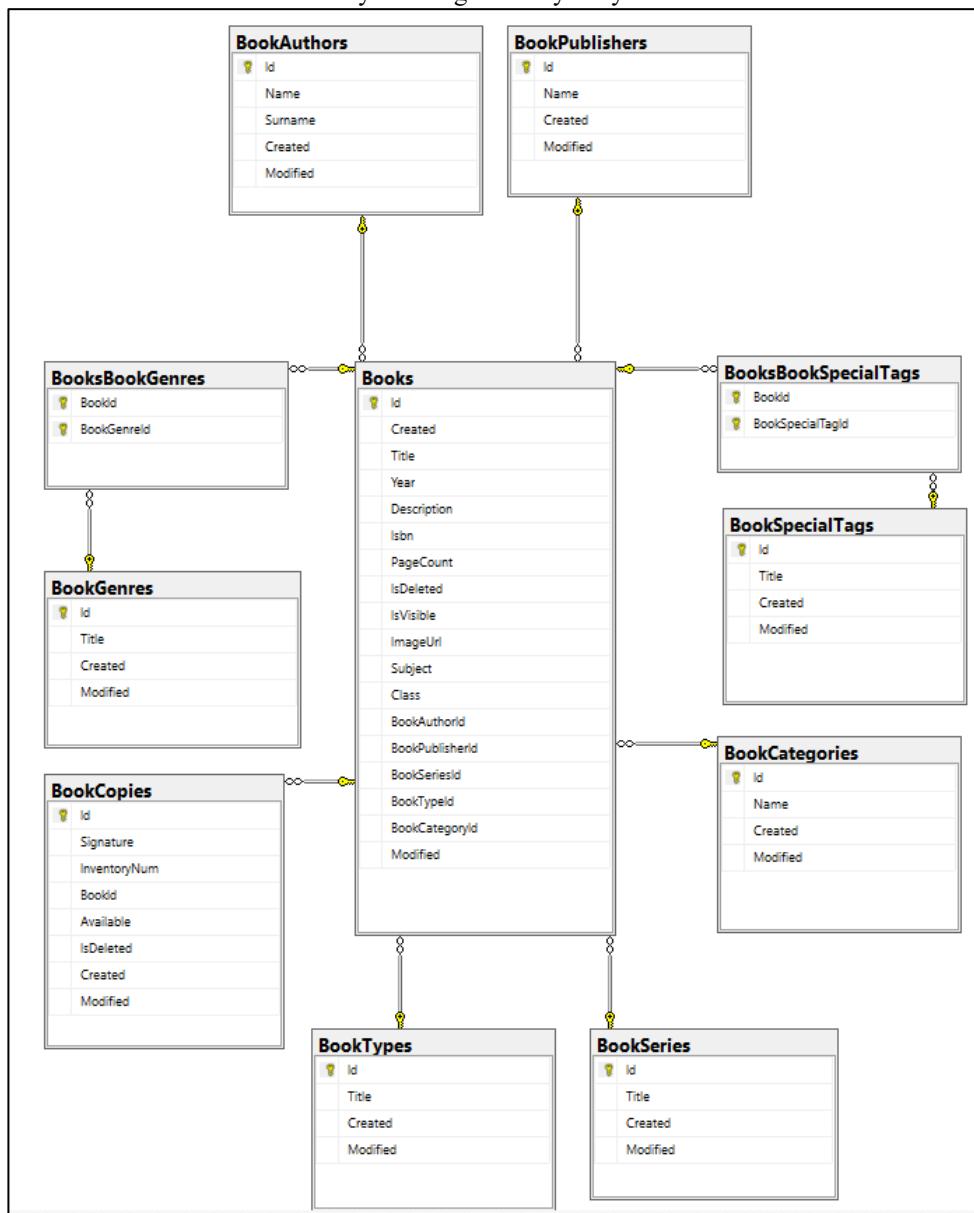
Baza danych zawiera tabele i relacje dotyczące zasobów bibliotecznych (książki, autorzy, egzemplarze, gatunki itd.). Struktura bazy dot. zasobów została utworzona programowo, korzystając z kodu C#.

Dostęp do danych opiera się o obiekty transferu danych DTO (Data Transfer Object) – dzięki temu użytkownik nie ma bezpośredniego dostępu do bazy oraz ma dostęp tylko do wybranej zawartości modelu.

### 2.1. Diagram bazy

Diagram został wygenerowany w *SQL Server Management Studio 20* i przedstawia tabele oraz ich relacje między sobą.

Rys. 1 Diagram bazy danych



Źródło 1: Praca własna

## 4. Prezentacja aplikacji za pomocą Swagger

Działanie API można zaprezentować za pomocą Swaggera, który oferuje prosty interfejs do obrazowania działania i przesyłanych danych. Poniżej przykłady wybranych metod GET.

Zdj. 1 Prezentacja dostępnych kontrolerów i schematów API

The screenshot shows the Swagger UI interface for the **BibliotekaAPI v1**. The interface is organized into sections corresponding to different controllers:

- Book**: Contains endpoints for managing books, including creating, editing, and deleting them.
- Catalog**: Contains endpoints for managing catalog details.
- Copy**: Contains endpoints for managing copy details.
- Schemas**: Lists the data transfer objects (DTOs) used in the API, such as BookCreateDto, BookEditDto, BookGetDetailedDto, BookGetDto, CopyCreateDto, CopyEditDto, CopyGetDetailedDto, and CopyGetDto.

Each endpoint is represented by a row with a colored button indicating the HTTP method (e.g., GET, POST, PUT, DELETE) and the corresponding URL path. The rows are color-coded: Book (light blue), Catalog (light blue), Copy (light blue), and Schemas (light gray).

Źródło 2 Praca własna

**Metoda GET dla wszystkich elementów *Books*** – pobiera wszystkie obiekty *Book* oraz oferuje filtry i sortowanie. Zwraca po kilka wyników na stronę (domyślnie 10). Poniżej pogląd na metodę oraz przykładowe wyniki z filtrem i bez.

Zdj. 2 Główna metoda GET pobierająca wszystkie książki

Code	Description	Links
200	OK Media type text/plain	No links

```
{
  "id": 0,
  "title": "string",
  "year": 0,
  "description": "string",
  "isbn": "string",
  "pageCount": 0,
  "isUsed": false,
  "isVisible": true,
  "imageId": "string",
  "subject": "string",
  "categ": "string",
  "bookAuthor": "string",
  "bookPublisher": "string",
  "bookSeries": "string",
  "bookCategory": "string",
  "bookType": "string",
  "bookCopies": [
    {
      "id": 0,
      "signature": "string",
      "inventoryId": 0,
      "available": true
    }
  ],
  "bookSpecialTags": [
    "string"
  ],
  "bookCopies": [
    {
      "id": 0,
      "signature": "string",
      "inventoryId": 0,
      "available": true
    }
  ]
}
```

Źródło 3 Praca własna

### Zdj. 3 Przykładowe wyniki bez filtrowania

Curl

```
curl -X 'GET' \
'https://localhost:7098/api/manage/Book/books' \
-H 'accept: text/plain'
```

Request URL

<https://localhost:7098/api/manage/Book/books>

Server response

Code Details

200 Response body

```
{
  "pageNumber": 1,
  "pageSize": 10,
  "totalCount": 63,
  "totalPages": 7,
  "items": [
    {
      "id": 49,
      "title": "Akademia Pana Kleksa",
      "year": 2023,
      "description": "Powie\u0144 ba\u0144na o naukach ch\u0142opca Adasia Niezg\u0144dki w dziwej, trochę smiesznej i zwariowanej akademii prowadzonej przez sympatycznego d\u0142iwaka Pana Kleksa. Ilustracje Jana Marcina Szancera.",
      "isbn": "978-83-7272-432-8",
      "pageCount": 120,
      "isDeleted": false,
      "isVisible": true,
      "imageUrl": "/images/books/10013759-9026-4886-b0a7-306b82bcf12..jpg",
      "class": "DV-VI",
      "bookAuthor": "Brzezina, Jan",
      "bookPublisher": "GPP",
      "bookSeries": "brak",
      "bookCategory": "Tekstura",
      "bookType": "Powie\u0144",
      "bookGenres": [
        "Fantasyka",
        "Miedziowa",
        "Dzieciaca",
        "Science fiction"
      ]
    }
  ]
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 01 Jul 2025 10:32:39 GMT
server: Kestrel
```

#### Źródło 4 Praca własna

Zdj. 4 Przykładowe wyniki z filtrem "Rowling" w *BookAuthor*

```
Curl
curl -X 'GET' \
'https://localhost:7098/api/manage/Book/books?BookAuthor=Rowling' \
-H 'accept: text/plain'

Request URL
https://localhost:7098/api/manage/Book/books?BookAuthor=Rowling

Server response
Code Details

200 Response body
{
  "pageNumber": 1,
  "pageSize": 10,
  "totalCount": 7,
  "totalPages": 1,
  "items": [
    {
      "id": 4,
      "title": "Harry Potter i Czara Ognia",
      "year": 2001,
      "description": "W Hogwarcie rozgrywa się Turniej Trójmagiczny, na który przybywają uczniowie szkół z Francji i Bułgarii. Zgodnie ze stara tradycja, do turnieju mają być wybrani trzej przedstawiciele, kazdej ze szkół. Jednak dziwnym zbiegiem okoliczności do turnieju zostaje wybranych czterech uczestników, w tym i Harry Potter.",
      "isbn": "978-83-7278-021-8",
      "pageCount": 766,
      "isDeleted": false,
      "isVisible": true,
      "imageUrl": "/images/books/94cb3604-46aa-4fa4-a979-12bddf3e87cd..jpeg",
      "bookAuthor": "Rowling, Joanne K.",
      "bookPublisher": "Media Rodzina",
      "bookSeries": "Harry Potter",
      "bookCategory": "pozostale",
      "bookType": "Powieść",
      "bookGenres": [
        "Fantasyka",
        "Przygodowa",
        "Młodzieżowa"
      ],
    }
  ]
}

Response headers
content-type: application/json; charset=utf-8
date: Tue, 01 Jul 2025 10:27:58 GMT
server: Kestrel
```

## Źródło 5 Praca własna

**Metoda GET pobierająca wybraną książkę po ID** – zwraca wybraną książkę z listą egzemplarzy i innymi informacjami.

Zdj. 5 Przykład pobrania danej książki po ID

The screenshot shows a REST API testing interface. At the top, there's a form with a single input field labeled 'Id \* required' containing the value '5'. Below the form are two buttons: 'Execute' on the left and 'Clear' on the right. The main area is titled 'Responses' and contains several sections:

- Curl:** A code block showing a cURL command to fetch the book details: `curl -X 'GET' '\https://localhost:7098/api/manage/Book/details/5*' \H 'accept: text/plain'`.
- Request URL:** A text box showing the URL: `https://localhost:7098/api/manage/Book/details/5`.
- Server response:** A table with two columns: 'Code' and 'Details'. Under 'Code' is '200'. Under 'Details' is a large JSON object representing the book's details, including its title, author, and genres.
- Response body:** A large text area showing the JSON response from the server. The JSON includes fields like id, title, year, description, isbn, pageCount, isDeleted, isVisible, imageUrl, bookAuthor, bookPublisher, bookSeries, bookCategory, bookType, bookGenres, and bookCopies.
- Response headers:** A text area showing the HTTP headers: content-type: application/json; charset=utf-8, date: Tue, 01 Jul 2025 10:35:43 GMT, and server: Kestrel.

Źródło 6 Praca własna