



**Instytut Informatyki  
Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**  
**Programowanie portali biznesowych**

**Dokumentacja projektu:**

***Biblioteka uczelniana***

**Wykonał:**  
**Mateusz Molik**

**Prowadzący: dr inż. Piotr Lasek**

**Rzeszów 2020**

# Spis treści

1. Opis programu.....	
2. Technologie .....	
3. Diagram ERD .....	
4. Funkcjonalności .....	
5. Uruchomienie programu .....	

# 1.Opis programu

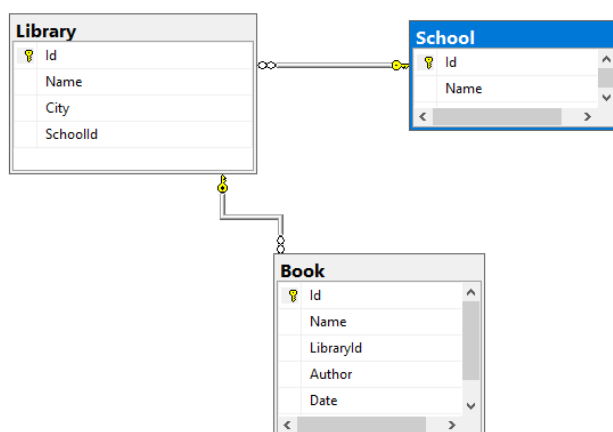
Biblioteka uczelniana to aplikacja webowa napisana w stylu REST udostępniająca API dla przyszłych potencjalnych klientów. API pozwala na dodawania szkół-uczelni oraz dodanie do nich bibliotek w której znajdują się książki. Wszystkie dane takie w systemie takie jak: Uczelnia, biblioteka oraz książki zostały zaprojektowane tak żeby przyszły klient mógł wykonać na nich operacje bazodanowe tj. CRUD.W celu pomocy przyszłym programistą do aplikacji został podpięty swagger który udostępnia wszystkie dostępne endpointy w systemie a także wzór modeli jakie może wysyłać i otrzymywać z systemu tj. Schemas.

## 2. Technologie

W projekcie zostały użyte następujące technologie:

- **NET Core 3.0**
- Visual Studio 2019
- **Microsoft SQL Server (MS SQL)** - Entity framework core
- Swagger

## 3. Diagram ERD



## 4. Funkcjonalności

- Dodawanie/Usuwanie/Updatowanie szkół(uczelni) oraz pobieranie bibliotek należących do danej szkoły.

Schools			▼
GET	/api/Schools	Pobierz wszystkie szkoły	🔒
POST	/api/Schools	Stwórz nową szkołę	🔒
GET	/api/Schools/{id}	Pobierz szkołę po id	🔒
PUT	/api/Schools/{id}	Aktualizacja szkoły po id	🔒
DELETE	/api/Schools/{id}	Usun szkołę po id	🔒
GET	/api/Schools/{id}/libraries	Pobierz biblioteki należące do danej szkoły/uczelni	🔒

- Dodawanie/Usuwanie/Updatowanie bibliotek oraz pobieranie książek należących do danej biblioteki oraz modyfikacja przynależności biblioteki do uczelni za pomocą metody PUT.

Libraries			▼
GET	/api/Libraries	Pobierz wszystkie biblioteki	🔒
POST	/api/Libraries	Stwórz nową bibliotekę	🔒
GET	/api/Libraries/{id}	Pobierz bibliotekę po id	🔒
PUT	/api/Libraries/{id}	Aktualizacja biblioteki po id	🔒
DELETE	/api/Libraries/{id}	Usun bibliotekę po id	🔒
GET	/api/Libraries/{id}/books	Pobierz książki które należą do biblioteki	🔒

- Dodawanie/Usuwanie/Updatowanie książki oraz modyfikacja przynależności książki do uczelni za pomocą metody PUT.

Books			▼
GET	/api/Books	Pobierz wszystkie książki	🔒
POST	/api/Books	Tworzenie nowej książki	🔒
GET	/api/Books/{id}	Pobierz książkę po id	🔒
PUT	/api/Books/{id}	Aktualizacja książki po id	🔒
DELETE	/api/Books/{id}	Usuwanie książki po ID	🔒

- Przykładowe odpowiedzi i żądania

```

BookRequest {
  id          integer($int32)
              Książka Id.

  name        string
              Książka nazwa.

  author       string
              Nazwa Autora książki.

  date         string
              Data powstania książki.

  side         integer($int32)
              Liczba stron książki.

  libraryId    integer($int32)
              Biblioteka Id.
}

```

```

LibraryResponse {
  id          integer($int32)
              Biblioteka Id.

  city         string
              Miejscowość w której jest biblioteka.

  name         string
              Biblioteka nazwa.

  schoolResponse SchoolResponse > {...}
}

```

```

LibraryRequest {
  id          integer($int32)
              Biblioteka Id.

  name         string
              Biblioteka nazwa.

  city         string
              Miejscowość biblioteki.

  schoolId     integer($int32)
              Szkoła nazwa Id.
}

```

## 5. Uruchomienie programu

Aby uruchomić projekt należy pobrać **Visual studio** (najlepiej najnowsze czyli 2019 ) oraz zaimportować projekt. Należy też pobrać rozszerzenie w programie Visual Studio tj **.Net Core 3.0**. Po zaimportowaniu projektu należy stworzyć bazę danych w Microsoft SQL Server (Można to zrobić za pomocą **Visual studio** lub dedykowanego do tego narzędzia np. Microsoft SQL Tools. Ostatnia krok jaki trzeba zrobić to zmienić connection string na swój.

