****

**KOLEGIUM INFORMATYKI STOSOWANEJ**

**Kierunek: Informatyka**

**Specjalność: Technologie internetowe i mobilne**

Kinga Pszeniczna

57122

***Internetowy zbiór książek***

Promotor:

dr inż. Łukasz Piątek

**PRACA DYPLOMOWA INŻYNIERSKA**

**Rzeszów 2020**

Ja niżej podpisany/a oświadczam, że składana przeze mnie praca dyplomowa pt. „Internetowy zbiór książek” została przygotowana samodzielnie.

Oświadczam również, że praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna ze złożoną wersją elektroniczną.

.........................................................

data czytelny podpis autora

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

................................................................

data czytelny podpis promotora

Spis treści

[**WSTĘP** 4](#_Toc33131847)

[**1.** **WPROWADZENIE** 5](#_Toc33131848)

[**1.1.** **Opis technologii** 5](#_Toc33131849)

[**1.2.** **Opis środowiska programistycznego** 6](#_Toc33131850)

[**1.3.** **Porównanie wykorzystanych technologii do alternatywnych rozwiązań** 6](#_Toc33131851)

[**2.** **SPECYFIKACJA SYSTEMU** 7](#_Toc33131852)

[**2.1.** **Wymagania funkcjonalne i niefunkcjonalne** 7](#_Toc33131853)

[**2.2.** **Podział systemu** 7](#_Toc33131854)

[**3.** **CHARAKTERYSTYKA PROBLEMU** 9](#_Toc33131855)

[**3.1.** **Określenie problemu** 9](#_Toc33131856)

[**3.2.** **Cel projektu** 10](#_Toc33131857)

[**4.** **IMPLEMENTACJA** 11](#_Toc33131858)

[**5.** **PREZENTACJA SYSTEMU** 18](#_Toc33131859)

[**5.1.** **Część administracyjna** 18](#_Toc33131860)

[**5.2.** **Część użytkowa** 18](#_Toc33131861)

[**6.** **PROBLEMY** 26](#_Toc33131862)

[**STRESZCZENIE** 27](#_Toc33131863)

[**LITERATURA** 28](#_Toc33131864)

[**SPIS ILUSTRACJI** 29](#_Toc33131865)

[**ZAŁĄCZNIKI** 30](#_Toc33131866)

# **WSTĘP**

Aktualnie popularność czytania książek jest wypierana przez rozrywkę cyfrową, taką jak gry, filmy, seriale. W Polsce potencjalny czytelnik woli zasięgnąć do artykułów w Internecie czy telewizji. Jeśli natomiast czytelnik sięgnie po książkę, to coraz częściej jest to wersja elektroniczna.

Celem utworzonej aplikacji jest zbliżenie potencjalnego użytkownika komputera lub telefonu do przeczytania książki. Witryna internetowa umożliwia zalogowanym użytkownikom, przeglądanie informacji o autorach i książkach, oraz daje możliwość pobierania książki w formacie \*.pdf. Aplikacja została utworzona w wersji na przeglądarki internetowe. W stworzonej aplikacji zostały umieszczone książki wolne od praw autorskich.

W ramach niniejszej pracy utworzono aplikację internetową, pozwalającą na przeglądanie danych o autorach, ich książkach z możliwością pobrania książki. Pierwszy rozdział zawiera część teoretyczną, która została podzielona na 4 podrozdziały. W podrozdziale pierwszym zostało umieszczone wprowadzenie, czyli krótki opis technologii, jakie zostały wykorzystane do utworzenia aplikacji. Środowiska programistyczne jakie są wykorzystane, zostały opisane w podrozdziale drugim. W trzecim podrozdziale poddane zostały porównaniu użyte technologie oraz środowisko do innych alternatywnych rozwiązań. Podano również krótkie opisy innych popularnych technologii do tworzenia aplikacji mobilnych i internetowych. Czwarty podrozdział opisuje działanie metody REST API. Drugi rozdział zawiera specyfikację systemu, która jest podzielona na dwa podrozdziały. Pierwszy podrozdział zawiera spis wymagań funkcjonalnych oraz niefunkcjonalnych. W podrozdziale drugim zawarty został podział na część, którą zarządza administrator oraz na część użytkownika. Trzeci rozdział zawiera pełniejszą charakterystykę problemu, z podziałem na opis problemu oraz cel jaki ma zrealizować praca. Rozdział czwarty zawiera implementację aplikacji, w której przedstawione są fragmenty kodu z funkcjonalnościami tworzonego systemu. W rozdziale piątym zawarta jest prezentacja systemu z podziałem na część administratora oraz na część użytkownika systemu.

# **WPROWADZENIE**

Aplikacje internetowe mają na celu nie tylko przekazywanie informacji, ale również odbierać i przetwarzać dane, które wprowadzamy. Witryna internetowa, którą odwiedzamy i spełnia ona rolę aplikacji, powinna zachęcić użytkownika do interakcji. Aplikacji internetowej nie trzeba instalować na komputerze, powinna być uniwersalna oraz elastyczna.

* 1. **Opis technologii**

Aplikacja została napisana technologią ASP .NET MVC 5 z podejściem CodeFirst, poprzez wykorzystanie Entity Framework. Aplikacje można uruchamiać w przeglądarce internetowej na PC oraz na smartfonach. Widok aplikacji dostosowuje się do rozdzielczości ekranu, na którym jest wyświetlana. Jest to możliwe dzięki narzędziu Bootstrap, które powoduje, że wyświetlana aplikacja jest responsywna.

Aby wiedzieć czym tak naprawdę jest ASP .NET, trzeba wiedzieć też czym jest HTML. Jest to język opisowy, który za pomocą znaczników określa zawartość strony. HTML wykorzystuje hiperłącza, obrazki, pola tekstowe, tabele, przyciski, nagłówki, stopki, itp. Określany jest również tytuł strony, opis, skrypty, meta dane (np. język znaków), czy też arkusze stylów (CSS).

CSS jest kaskadowym arkuszem stylów, które odpowiadają za odpowiedni layout strony, która wyświetla się na ekranie. Z pomocą CSS można ustawić kolor, rozmiar oraz styl czcionki na stronie, tło, wielkość wyświetlanych obrazków, itp.

Czym więc tak właściwie jest ASP .NET? Jest to zbiór technologii, które oparte są na frameworkach firmy Microsoft. Na podstawie ASP .NET można budować aplikacje internetowe, pracujące na serwerach, komunikujące się z użytkownikiem za pomocą wybranej przeglądarki.

Model tworzenia aplikacji ASP .NET MVC 5 jest najpopularniejszym modelem, który wykorzystują programiści. Zapewnia on przejrzystą strukturę, intuicyjność zarządzania oraz odizolowaną logikę od interfejsu użytkownika. Skrót MVC oznacza Model-View-Controller, poniżej znajduje się opis składowych wzorca MVC:

* Model – w tworzonym projekcie, w katalogu Model znajduje się model bazy danych. Pliki, które się znajdują w tym katalogu są klasami. Podczas tworzenia aplikacji z podejściem code first, pliki z modelem bazy będą tworzone przez programistę. Natomiast przy wyborze podejścia database first, pliki z modelem bazy zostaną automatycznie dodane do projektu.
* View – pliki do projektu, umieszczone w katalogu Views, odpowiadają za prezentację danych, które poda kontroler.Ta część wzorca używa takich technologii jak, HTML, CSS oraz innych języków, potrzebnych do zarządzania widokiem strony.
* Controller – utworzone pliki w katalogu Controllers pozwalają na odbiór i przetwarzanie danych wprowadzonych przez użytkownika z poziomu strony, czy też tych pobranych z serwera bazodanowego. Kontroler może wywołać żądanie wprowadzenia danych, później wysłać je do modelu, z modelu do widoku, który wyświetli te dane użytkownikowi. Może zostać wywołany również inny kontroler.

Podejścia, które można wyszczególnić to:

* Code first – podejście to polega na tym, że nie jest tworzony schemat bazy wprost, lecz tworzony jest pewien szablon w języku C#, który z pomocą Entity Framework generuje model bazy danych.
* Database first – w tym podejściu trzeba najpierw utworzyć schemat bazy danych, następnie zimportować bazę do modelu, a Entity Framework wygeneruje odpowiednie klasy oraz je zmapuje.
* Model first – to podejście pozwala na utworzenie modelu bazy w wizualnym designerze. Następnie tworzona jest na tej podstawie baza danych.

Entity Framework jest narzędziem, które ułatwia programiście utworzenie bazy danych w środowisku .NET.

Aktualnie tworząc strony internetowe, czy aplikacje webowe, wykorzystuje się bibliotekę CSS, która powoduje, że widok strony dopasowuje się do rozdzielczości ekranu, na którym jest wyświetlany, czyli jest responsywny. Ta biblioteka CSS to Bootstrap. Działa on na zasadzie siatki, która dopasowuje się do wielkości ekranu i ustawia elementy na stronie według określonej szerokości oraz kolejności.

* 1. **Opis środowiska programistycznego**

Do utworzenia aplikacji zostało wykorzystane środowisko programistyczne Visual Studio 2019 Community. Dane przechowywane są na serwerze zdalnym oraz obsługiwane są poprzez Microsoft SQL Server Management Studio 17.

Visual Studio 2019 jest zintegrowanym środowiskiem, służącym do programowania, które zostało stworzone przez firmę Microsoft. Visual Studio używa się do tworzenia aplikacji typu: Windows Forms, WPF, Web Applications, Web Sites. Aplikacje mogą być tworzone na różne platformy.

Baza danych, która została utworzona w ramach projektu znajduje się na zdalnym serwerze Microsoft SQL Server. Baza jest zarządzana przez środowisko SQL Server Management Studio 2017.Aplikacja ta pozwala na zarządzanie wszystkimi komponentami, takimi jak, baza danych, usługi raportowe, analityczne, itp. Dzięki temu narzędziu można budować zapytania oraz skrypty. Aplikacja zawiera Object Explorer, za pomocą którego możemy przeglądać, wybierać oraz wykonywać różne działania na obiektach serwera. Do utworzonej aplikacji wykorzystana została bezpłatna wersja express, która jest ograniczona w porównaniu do wersji pełnej. Brak w tej wersji usług analitycznych, raportowych oraz integracji.

* 1. **Porównanie wykorzystanych technologii do alternatywnych rozwiązań**

Utworzona aplikacja została napisana w technologii ASP .NET MVC 5, co pozwala na uruchomienie aplikacji na platformach Windows. Alternatywnym rozwiązaniem jest technologia ASP .NET CORE, która pozwala na uruchomienie aplikacji na innych platformach. ASP .NET CORE najlepiej używać kiedy wymagana jest wieloplatformowość aplikacji, ale i również kiedy wymagana jest wysoka wydajność tworzonego oprogramowania. Wadą wieloplatformowej technologii jest brak wsparcia dla niektórych bibliotek.

# **SPECYFIKACJA SYSTEMU**

Każdy tworzony system ma określoną specyfikację. W tym rozdziale zostaną przedstawione wymagania funkcjonalne i niefunkcjonalne utworzonego systemu. Następnym podrozdziałem jest podział systemu na część administracyjną oraz na część użytkownika.

* 1. **Wymagania funkcjonalne i niefunkcjonalne**

Niniejszy rozdział przedstawia wymagania funkcjonalne, jakie ma zawierać system, aby spełnić podstawową funkcjonalność utworzonej aplikacji. W poniższej tabeli (Tabela 1) opisuje wspomniane wymagania.

Tabela 1 Wymagania funkcjonale

|  |  |
| --- | --- |
| **Wymaganie** | **Opis** |
| Logowanie i rejestracja | Użytkownik rejestruje się do systemu, podając odpowiednie dane (adres e-mail, hasło). Aby zalogować się do systemu trzeba podać adres e-mail oraz hasło |
| Wyszukiwanie wybranego autora | Wyszukiwanie odbywa się po zalogowaniu do systemu oraz przejściu do widoku listy autorów, następnie użytkownik może sortować listę bądź filtrować ją wpisując odpowiedni ciąg znaków. |
| Wyszukiwanie wybranej książki | Aby wyszukać dana książkę, użytkownik musi być zalogowany. Wyszukiwanie książki odbywa się za pomocą sortowania lub filtrowania. |
| Pobieranie książki | Zalogowany użytkownik po wybraniu książki i przejściu do szczegółów, może skopiować zamieszczony link i pobrać wybraną książkę. |
| Dodawanie danych o książkach i autorach | Osoba mająca dostęp do bazy danych może dodawać rekordy |
| Modyfikacja danych | Osoba z odpowiednimi uprawnieniami do bazy danych może edytować wprowadzone dane. |

Wymagania niefunkcjonalne określają jakość utworzonego systemu. Nie musi on spełniać tych wymagań, lecz muszą się one znajdować, aby spełnić wymagania interesariuszy. Poniższa tabela przedstawia wymaganie niefunkcjonalne, które zostały spełnione w aplikacji.

Tabela 2 Wymagania niefunkcjonalne

|  |  |
| --- | --- |
| **Wymaganie** | **Opis** |
| Responsywność aplikacji | Oznacza, że wyświetlana strona, za pomocą siatki, dopasowuje się do rozdzielczości ekranu. |
| Intuicyjny interfejs | Aplikacja powinna być prosta w obsłudze, a wyświetlane informacje czytelne oraz zrozumiałe dla użytkownika. |
| Bezpieczeństwo | Wprowadzane dane do systemu powinny być możliwe tylko dla uprawnionych osób. Dane wyświetlane powinny być dostępne tylko po zalogowaniu użytkownika. |

* 1. **Podział systemu**

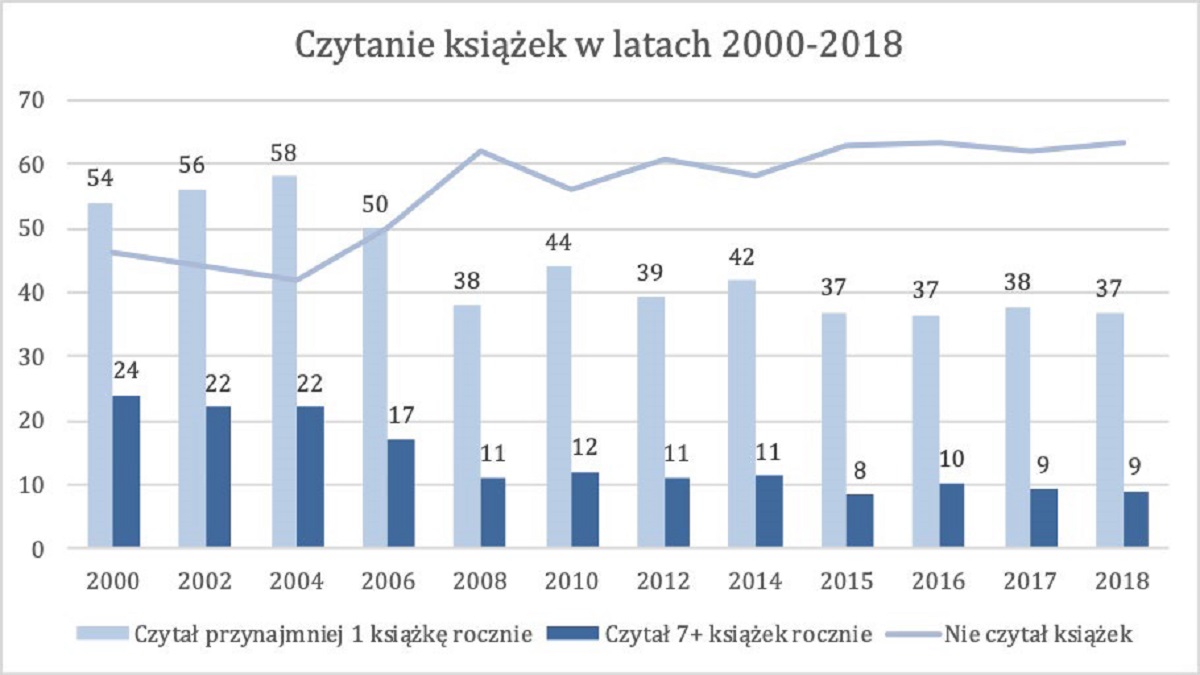
System został podzielony na część, którą zarządza administrator oraz na część, którą obsługuje użytkownik. W części administracyjnej zarządzanie odbywa się w SQL Server Management Studio, na serwerze zdalnym. W dalszym planie rozwoju systemu przewidziane jest utworzenie panelu administracyjnego bezpośrednio w aplikacji Internetowego zbioru książek. Po zalogowaniu się użytkownik ma możliwość przeglądania danych o autorach, książkach oraz pobierać wybrane książki. Aby wyszukać odpowiednie dane, użytkownik ma możliwość filtrowania oraz sortowania danych.

# **CHARAKTERYSTYKA PROBLEMU**

W tym rozdziale został przedstawiony problem czytelnictwa w Polsce oraz cel jaki ma zrealizować utworzona aplikacja.

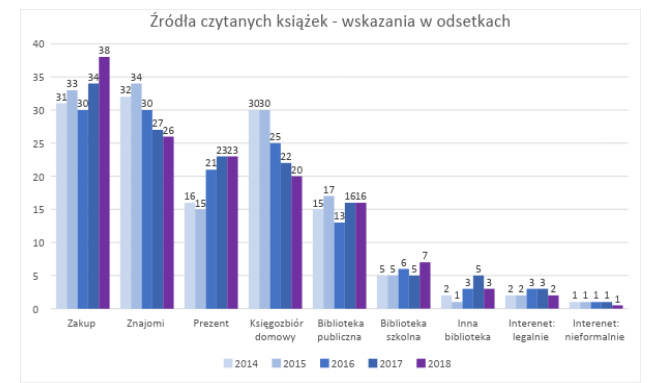
* 1. **Określenie problemu**

Od bardzo wielu lat ilość osób czytających regularnie książki maleje. Młodzież woli spędzić czas grając w gry komputerowe, które pochłaniają bardzo dużo czasu. Na dalszy plan schodzą nawet obowiązkowe lektury szkolne. Osoby pracujące zazwyczaj są pochłonięte życiem codziennym, w którym często brakuje czasu na przeczytanie czegoś dłuższego niż artykuł w gazecie, czy na popularnych stronach internetowych. Problemem jest nie tylko brak czasu na czytanie książek, lecz również postęp technologiczny. Sprawił on, że człowiek ma coraz więcej sposobów na spędzenie wolnego czasu. Można tutaj wymienić gry single player, multi player, wybijające się na rynku gry Virtual Reality, serwisy z filmami i serialami, np.: Netflix, HBO. Dużo wolnego czasu poświęcane jest również na przeglądanie portali społecznościowych. Na poniższym wykresie został pokazany spadek ilości czytanych książek na przestrzeni lat 2000 do 2018.



Rys. 1 Wykres ilości czytanych książek w latach 2000 do 2018

Aby nadążyć za postępem technologicznym potencjalny czytelnik, mimo elektronicznych czytników książek, woli ekranizowaną wersję książki, która jest bardzo często okrojona ze szczegółów. Zaletą czytania książek jest wzbogacanie języka oraz wczuwanie się w atmosferę, co pobudza wyobraźnię czytelnika. Wiele ludzi odchodzi jednak od czytania ze względu na możliwość pozyskania wiedzy w innych źródłach. Zazwyczaj w statystykach biorą udział głównie dzieci i młodzież, które nie zawsze są zachęcane do czytania. Z tego względu obwinianie rozwoju technologii czy Internetu mija się z celem. Od kilku lat organizowane są akcje czytania książek dla najmłodszych, dzięki czemu czytelnictwo wśród tej grupy podniosło swój poziom. Popularność Internetu wzrosła na tyle, że czytelnik często przeszuka wszystkie dostępne zasoby aby znaleźć książkę, nie zawsze w legalnych źródłach. Na poniższym zdjęciu został przedstawiony wykres, z jakich źródeł korzystają czytelnicy, aby pozyskać książkę.



Rys. 2 Źródła pozyskiwania książek w latach 2014 do 2018

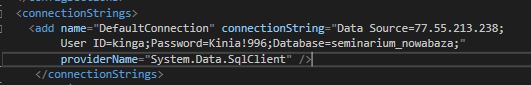
* 1. **Cel projektu**

Celem utworzonej aplikacji jest zbliżenie potencjalnego czytelnika, aby skłonił się do przeczytania każdego rodzaju książki. Aplikacja została utworzona z myślą o postępie technologicznym, aby odnalezienie szukanej książki stało się łatwiejsze oraz przyjemniejsze. Istnieją strony z tego rodzaju materiałami, lecz im więcej takich źródeł, tym większa szansa, że czytelnictwo ponownie wzbije się w górę.

# **IMPLEMENTACJA**

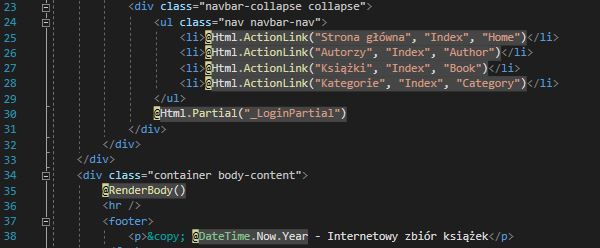
W tym rozdziale zostaną przedstawione fragmenty kodu z tworzenia aplikacji. Związane one będą z funkcjonalnościami utworzonymi, aby zrealizować cel projektu.

Na poniższym zdjęciu przedstawione zostało połączenie do bazy danych, która jest na zdalnym serwerze. Parametrami, które trzeba uzupełnić, są: adres serwera bazy danych, nazwa użytkownika, hasło oraz nazwa bazy, w której dane mają się znajdować.



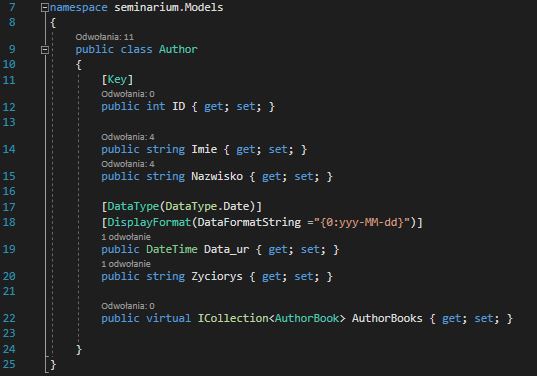
Rys. 3 Fragment kodu odpowiadający za połączenie z bazą danych

Na Rys. 4 znajduje się fragment kodu, który został utworzony domyślnie przez szablon tworzonej aplikacji ASP .NET. Na potrzeby tworzonej aplikacji został on zmodyfikowany. Do automatycznie utworzonego panelu menu zostały dodane odwołania do widoków listy autorów, książek oraz kategorii. W nagłówku oraz stopce strony została dodana nazwa aplikacji. Funkcja ActionLink, działa jako odnośnik do wskazanej w parametrach funkcji strony. Na ekranie użytkownik zobaczy tylko tekst, wskazany w funkcji jako nazwa akcji.



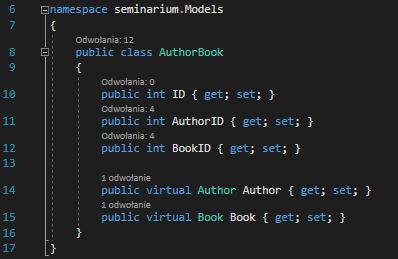
Rys. 4 Fragment kodu odpowiedzialnego za pasek menu

Fragment kodu na rysunku poniżej przedstawia model bazy tabeli Author. W tabeli znajdują się takie pola jak ID, czyli klucz główny, Imię, Nazwisko, Data urodzenia oraz Życiorys. Na końcu modelu ustawiona została zależność do tabeli AuthorBook. Parametr *public* oznacza, że dane są widoczne publicznie dla wszystkich klas. Parametry takie jak: *int*, *string*, *DateTime* oznaczają typy zmiennych. Zmienna ID przyjmuje typ *int*, czyli jest zmienna liczbową. Zmienne Imię, Nazwisko, Życiorys przyjmują typ *string*, czyli są zmiennymi tekstowymi. Typ zmiennej *DateTime*, przypisany do Daty\_urodzenia oznacza datę oraz czas. Na potrzeby aplikacji, zmienna została zmodyfikowana do typu *DateTime.Date*.



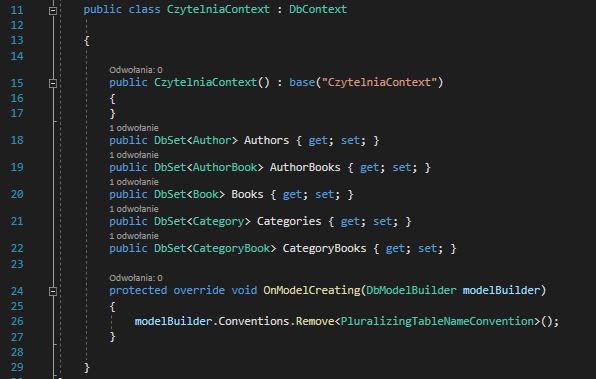
Rys. 5 Model bazy tabeli Author

Rys. 6 przedstawia model tabeli AuthorBook. Ta tabela odpowiada za połączenie dwóch tabel Author oraz Book. Zawiera ona tylko pola ID, AuthorID, BookID oraz powiązania do odpowiednich tabel. Linijki kodu 14 oraz 15 tworzą powiązanie z tabelami Author oraz Book. Utworzone zmienne przyjmują typ *int*, czyli są zmiennymi numerowanymi. Zmienna ID jest numerem unikalnym każdego rekordu w tej tabeli. Zmienna AuthorID zawiera numer ID autora, do którego ma zostać przypisana książka. Zmienna BookID zawiera numer ID książki, która ma być przypisana do autora.



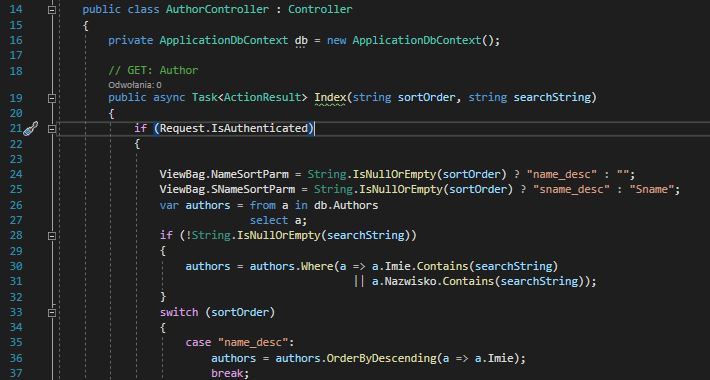
Rys. 6 Model tabeli AuthorBook

Po utworzeniu wszystkich modeli utworzono kontekst bazy. Fragment kodu został pokazany na Rys. 7. Kontekst zawiera odwołanie do wszystkich tabel utworzonych w modelu bazy.



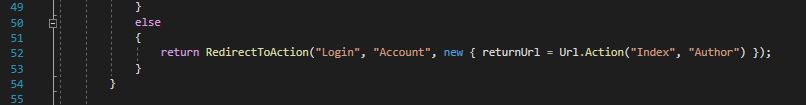
Rys. 7 Fragment kodu CzytelniaContext

Aby kontekst utworzonej bazy danych był spójny z kontekstem bazy utworzonym według schematu ASP .NET, kontrolery zostały stworzone na podstawie kontekstu szablonu projektu. Celem takiego zabiegu jest, aby dane wprowadzane do tabel Author, Book, Category, były w tej samej bazie, co użytkownicy rejestrujący się do serwisu. Fragment kodu z kontrolera AuthorController znajduje się na Rys. 8. Poniższy kod zawiera również fragment funkcji, która sprawdza czy użytkownik, który chce przeglądać listę autorów jest zalogowany, czy też nie. Odpowiada za to linijka nr 21. W utworzonej aplikacji użytkownik po zalogowaniu ma możliwość sortować dane, odpowiada za to funkcja *sortOrder.* Sortowanie danych jest możliwe według kolumny Imię oraz Nazwisko. Funkcja *searchString*, która również jest pokazana na poniższym rysunku, umożliwia użytkownikowi filtrowanie danych. Przedstawione na rysunku funkcje warunkowe *if* określają, jakie polecenia zostaną wykonane. Zależy to, od tego czy warunek zostanie spełniony czy też nie.



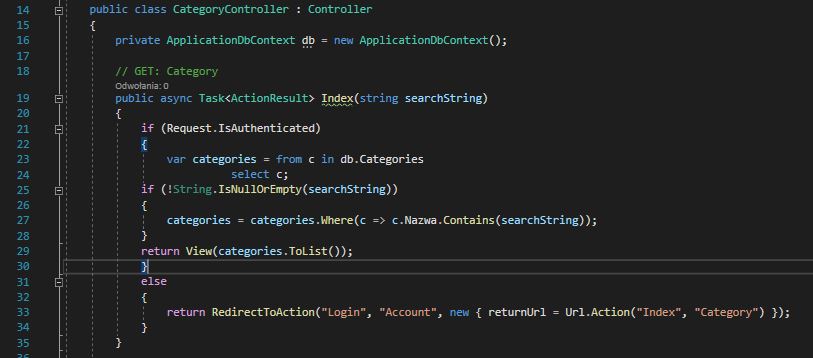
Rys. 8 Fragment kodu AuthorController

Na Rys. 9 pokazany został fragment kodu, który zamyka funkcję sprawdzania, czy użytkownik jest zalogowany. Jeśli użytkownik nie zalogował się do systemu, zostanie automatycznie przekierowany do ekranu logowania. Funkcja *RedirectToAction* przekieruje niezalogowanego użytkownika do ekranu logowania. Wskazują na to parametry wewnątrz funkcji.



Rys. 9 Zakończenie funkcji weryfikacji logowania użytkownika

Poniższy rysunek przedstawia zmodyfikowany kod pliku CategoryController.cs. Również znajduje się tutaj weryfikacja, czy użytkownik się zalogował. Występuje w tym kontrolerze tylko możliwość filtrowania kategorii po nazwie. Funkcja znajdująca się w linijkach 25 do 29 sprawdza, czy pole tekstowe jest różne od pustego. Następnie porównuje wprowadzony ciąg znaków, czy pasuje do rekordów znajdujących się w bazie danych.



Rys. 10 Fragment kodu pliku CategoryController.cs

Wraz z utworzeniem kontrolerów, utworzone zostały widoki dla każdego z nich. Utworzone katalogi zawierają standardowo pliki:

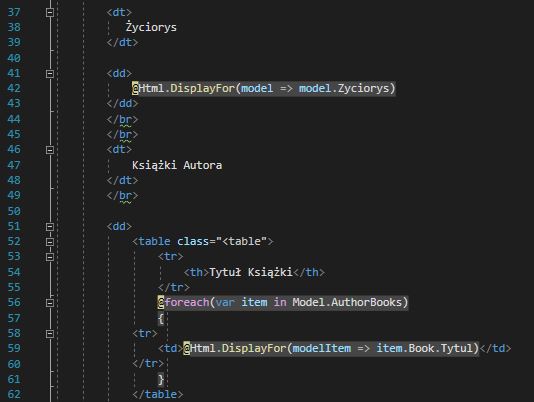
* Index.cshtml, który zawiera listę rekordów wprowadzonych do bazy,
* Create.cshtml, plik ten umożliwia utworzenie nowego rekordu,
* Details.cshtml, zawiera on szczegóły wybranego rekordu,
* Edit.cshtml, ten widok umożliwia edytowanie wybranego elementu,
* Delete.cshtml, ten plik pozwoli na usunięcie rekordu.

Na potrzeby utworzonej aplikacji, użytkownik widzi tylko dwa z tych widoków, Index oraz Details. Fragmenty kodu pliku Index.cshtml zostały przedstawione na Rys. 11. W przedstawionym fragmencie kodu znajdują się również pola umożliwiające filtrowanie danych (linijki od 10 do 13) oraz sortowanie danych (linijki 18 i 19). Parametr *TextBox* oznacza wprowadzenie do widoku pola tekstowego. Wewnątrz tego parametru znajduje się odwołanie do funkcji *searchString*, która jest zdefiniowana w kontrolerze. Znacznik *input* z parametrem *type=”submit”*, oznacza możliwość wysłania wprowadzonych danych do funkcji *searchString*. Funkcja *sortOrder* określona w linijkach 18 i 21 odwołuje się do definicji funkcji w kontrolerze. Funkcja ta pozwala na sortowanie rekordów w tabeli po kolumnach Imię i Nazwisko.



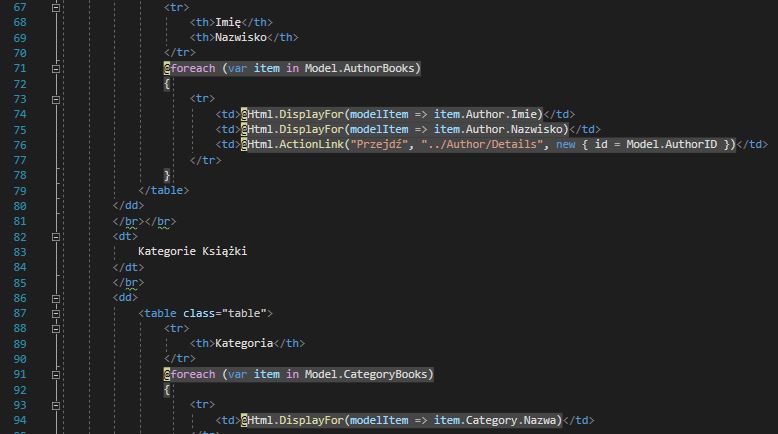
Rys. 11 Fragment kodu pliku Index.cshtml

Na Rys. 12 znajduje się fragment kodu pliku Details.cshtml, przedstawia on odwołanie do tabeli AuthorBook. Za pomocą tego odwołania, widok Details tabeli Author, ma możliwość pokazać Tytuły książek, które są powiązane z danym autorem.



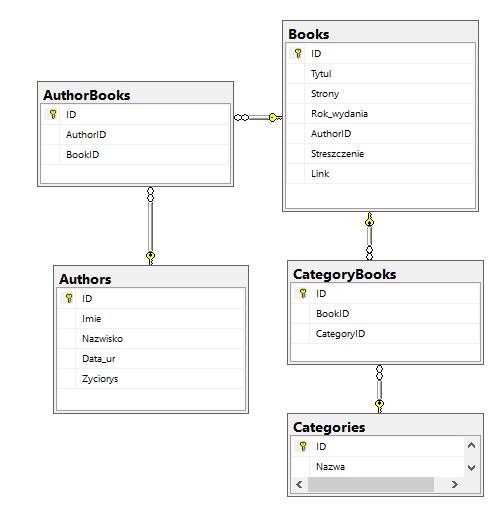
Rys. 12 Fragment kodu pliku Details.cshtml

Poniższy rysunek przedstawia fragment kodu odpowiedzialny za wyświetlanie szczegółów wybranej książki. Przedstawiony kod pozwala również, aby widok wyświetlił dane o autorze oraz kategorii książki. Jest to możliwe, dzięki zależnościom pomiędzy tabelami.



Rys. 13 Fragment kodu pliku Details.cshtml widoku Book

Na Rys. 14 przedstawiony został diagram zależności między tabelami. Aby utworzyć zależność wiele do wielu, potrzebna jest tabela pośrednia. W przypadku poniższego modelu bazy, tabelami pośrednimi są AuthorBook oraz CategoryBook.



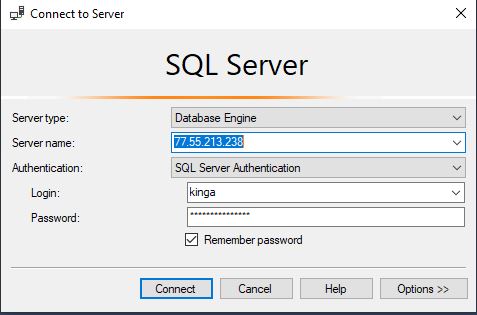
Rys. 14 Diagram zależności między tabelami

# **PREZENTACJA SYSTEMU**

W tym rozdziale przedstawiony zostanie sposób działania utworzonej aplikacji. Prezentacja systemu zostanie podzielona na dwie części: część administracyjną, w której administrator ma możliwość dodawać dane, oraz część przeznaczoną dla użytkownika, w której zalogowany czytelnik ma możliwość wyszukać wybraną książkę i ją pobrać.

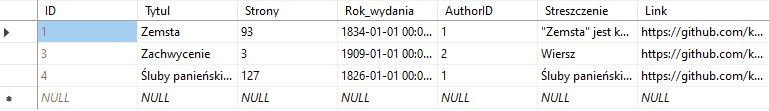
* 1. **Część administracyjna**

W części administracyjnej, osoba z uprawnieniami do zalogowania się do bazy danych ma możliwość edytowania wprowadzanych rekordów. Poniższy rysunek przedstawia okno logowania do bazy danych w programie SQL Server Management Studio.



Rys. 15 Ekran logowania do zdalnego serwera z bazą danych

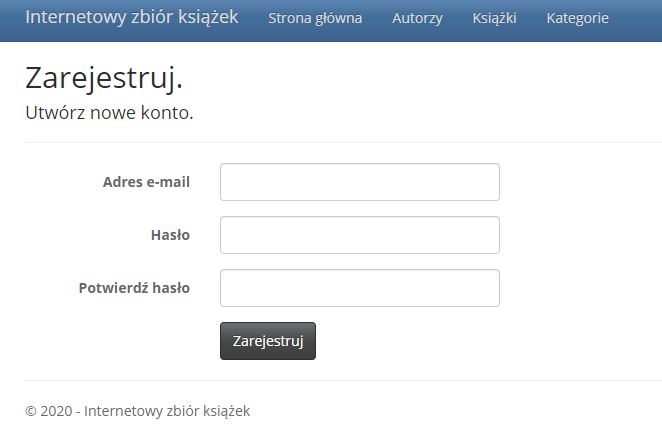
Rys. 16 przedstawia tabelę Books. Administrator ma możliwość dodawać, edytować oraz usuwać dane z bazy.



Rys. 16 Widok edycji rekordów w tabeli Books

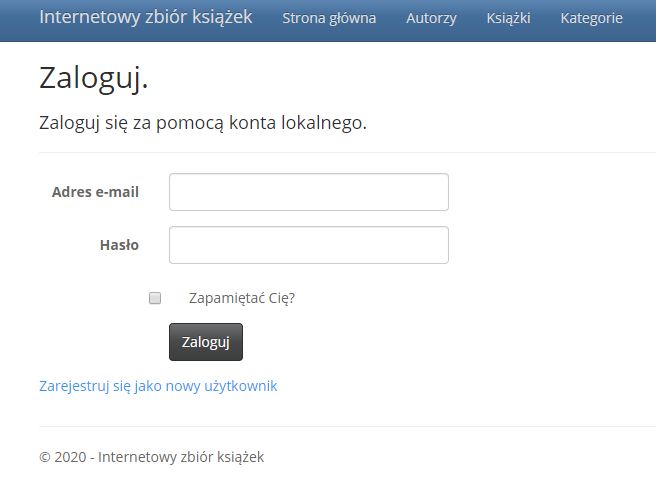
* 1. **Część użytkowa**

Aby czytelnik mógł korzystać z aplikacji internetowej musi się najpierw zarejestrować do systemu. Po zarejestrowaniu, system automatycznie zaloguje użytkownika. Ekran rejestracji przedstawia Rys. 17. Do rejestracji wymagany jest adres e-mail, hasło oraz dla celów walidacyjnych powtórzenie hasła.



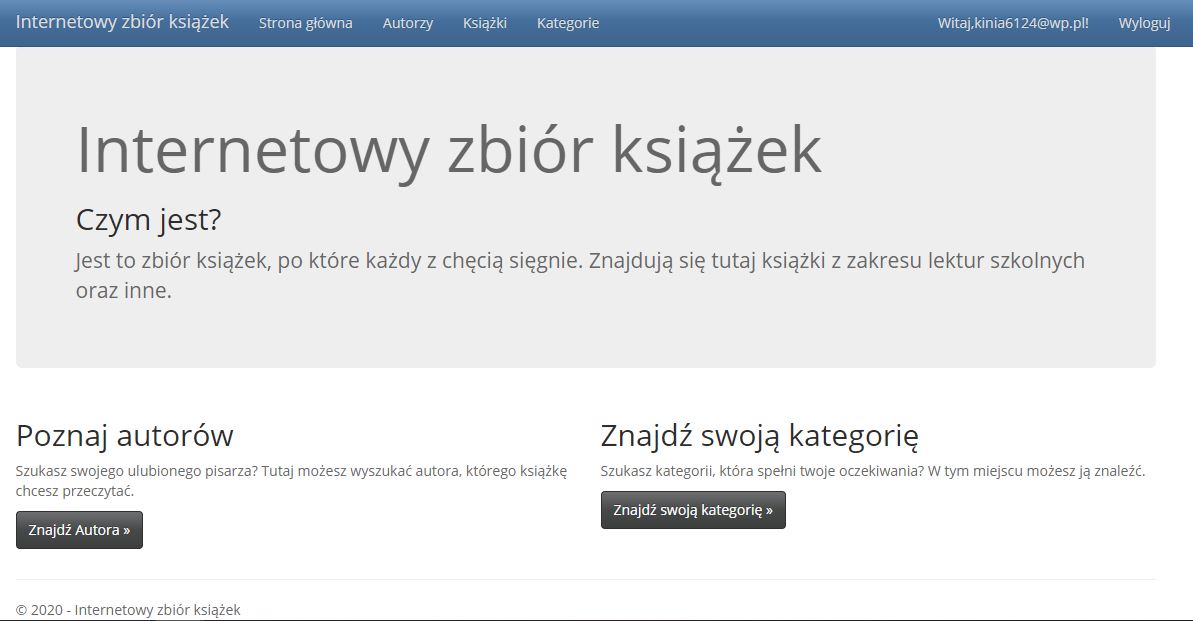
Rys. 17 Ekran rejestracji do systemu

Jeśli użytkownik jest już zarejestrowany, wystarczy że zaloguje się do systemu, co pokazuje Rys. 18. Do zalogowania potrzebny jest adres e-mail oraz hasło podane podczas rejestracji. Użytkownik ma możliwość zaznaczenia opcji zapamiętania. Oznacza to, że przy kolejnym logowaniu, jego dane logowania zostaną automatycznie uzupełnione.



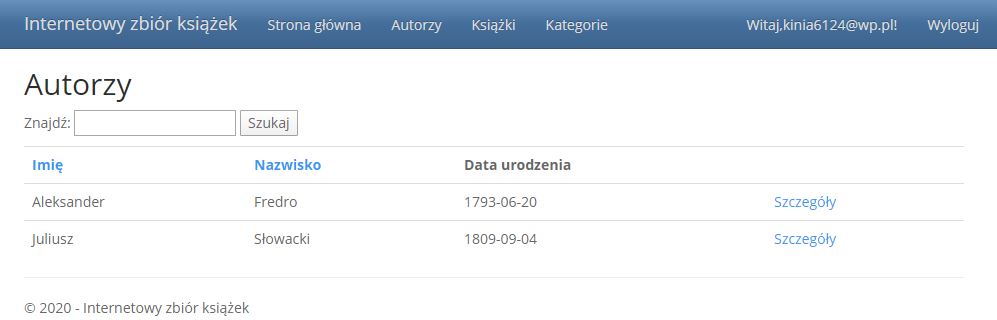
Rys. 18 Ekran logowania

Pierwszym ekranem jaki widzi czytelnik, jest ekran startowy aplikacji, przedstawiony na Rys. 19. Na ekranie widoczne są ogólne informacje, czego dotyczy aplikacja, oraz okna, w których jest informacja zachęcająca do przeglądania strony.



Rys. 19 Strona główna aplikacji

Na Rys. 20 umieszczony został widok listy autorów. Podświetlone kolumny pozwalają na sortowanie danych. Natomiast panel wyszukiwarki pozwoli na filtrowanie danych.



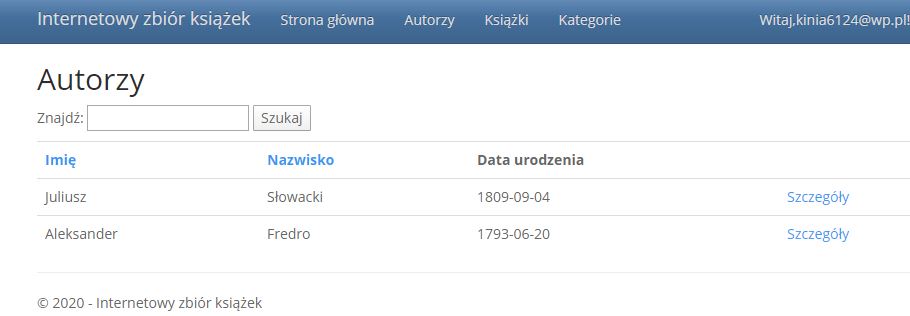
Rys. 20 Lista Autorów

Poniższe rysunki (Rys. 21, Rys. 22) przedstawiają listę autorów posortowaną według Imion od Z-A. Rys. 21 przedstawia pasek adresu, w którym określone zostało jakie jest sortowanie.



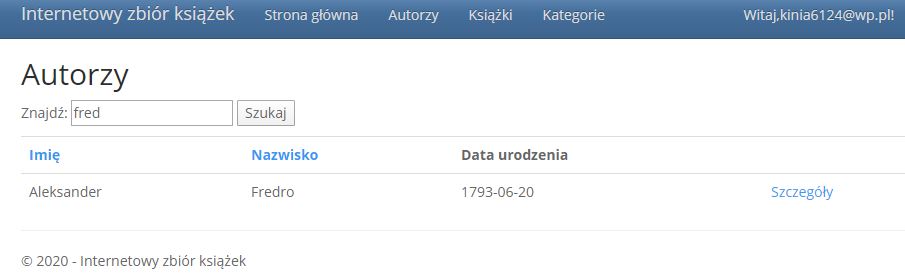
Rys. 21 Pasek adresu

Rys. 22 zawiera posortowaną listę autorów.



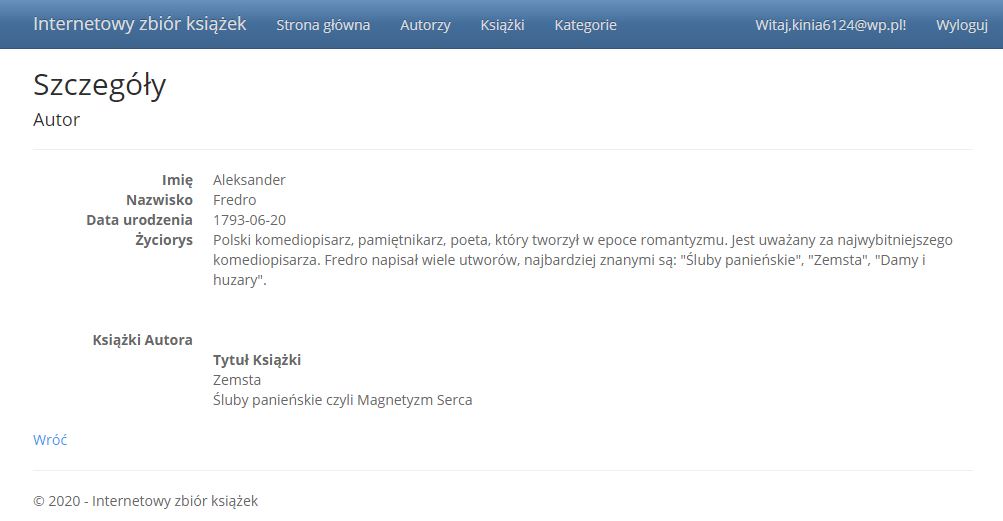
Rys. 22 Posortowana lista autorów

Na Rys. 23 widać jak działa filtrowanie danych. Filtrowanie jest możliwe według imienia i nazwiska autora.



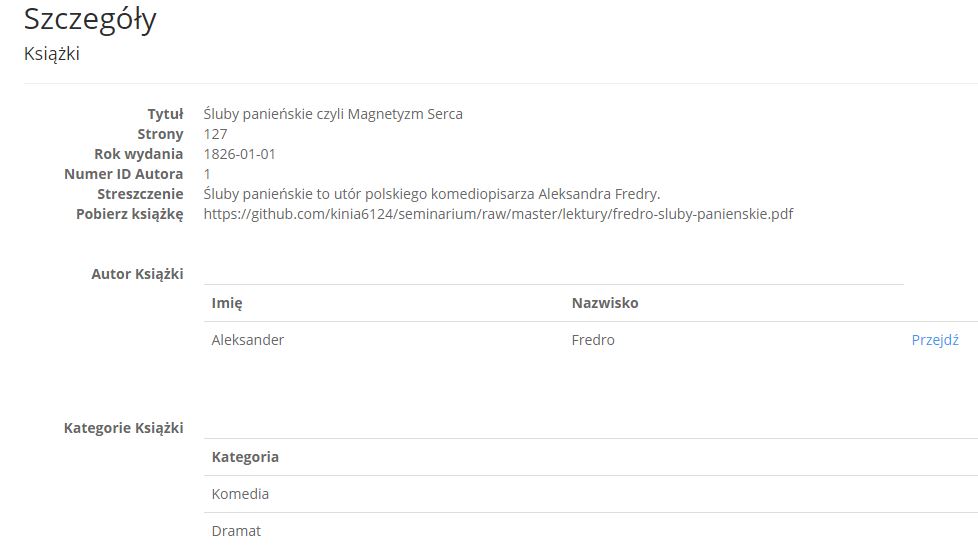
Rys. 23 Filtrowanie danych

Poniższy rysunek przedstawia widok szczegółów wybranego autora. Widać również listę jakie książki napisał autor.



Rys. 24 Widok szczegółów wybranego autora

Rys. 25 przedstawia szczegóły wybranej książki. Na tym ekranie widać również autora oraz kategorie książki.



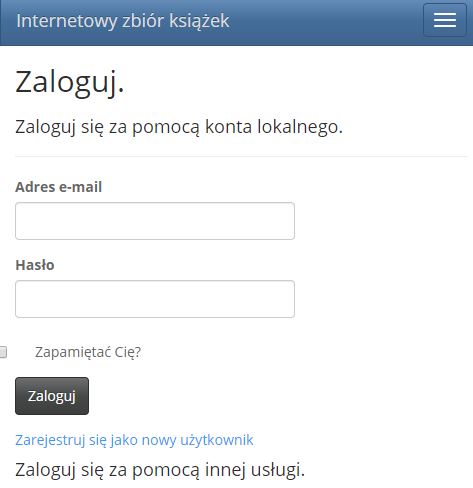
Rys. 25 Szczegóły wybranej książki

Poniższy rysunek przedstawia widok menu przy mniejszej rozdzielczości ekranu.



Rys. 26 Widok menu aplikacji w niskiej rozdzielczości

Zdjęcie poniżej przedstawia ekran logowania do systemu dla wersji mobilnej.



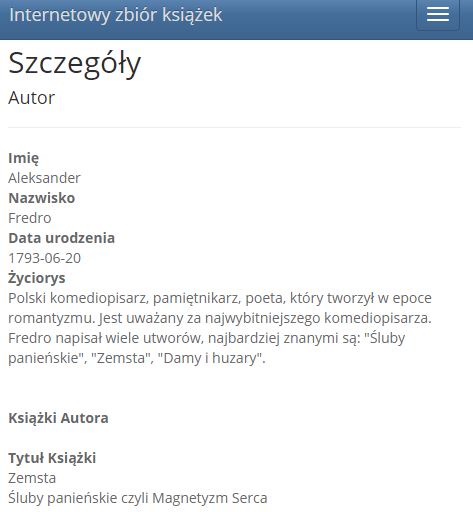
Rys. 27 Logowanie do systemu - wersja mobilna

Rys. 28 przedstawia listę autorów w mniejszej rozdzielczości ekranu.



Rys. 28 Lista autorów - wersja mobilna

Na poniższym rysunku widać szczegóły wybranego autora w rozdzielczości dla wersji mobilnej.



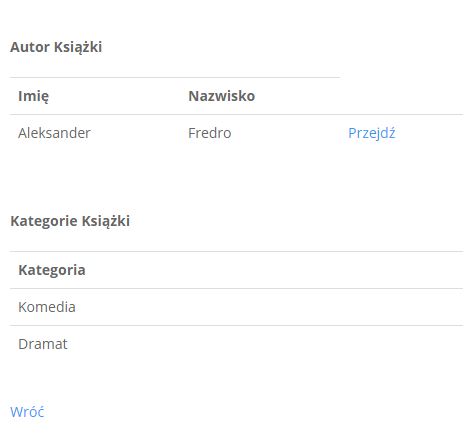
Rys. 29 Widok szczegółów dla wersji mobilnej

Rys. 29 oraz Rys. 30 przedstawiają ekran szczegółów wybranej książki w wersji mobilnej. Można przyuważyć jak Bootstrap dopasowuje wygląd strony do rozdzielczości ekranu.



Rys. 30 Szczegóły książek - wersja mobilna

Rysunek poniżej przedstawia również autora oraz kategorie książki.



Rys. 31 Szczegóły książki - wersja mobilna

# **PROBLEMY**

Problemy jakie pojawiały się podczas tworzenia aplikacji, to między innymi, błąd połączenia z bazą danych, rozwiązaniem okazała się modyfikacja funkcji *connectionString.* Kolejny problem był związany z modyfikowaniem modelu bazy danych podczas tworzenia aplikacji. Problem został rozwiązany poprzez utworzenie aplikacji z nowym modelem bazy, bez konieczności tworzenia migracji.

# **STRESZCZENIE**

**Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie**

**Kolegium Informatyki Stosowanej**

**Streszczenie pracy dyplomowej**

*Internetowy zbiór książek*

**Autor: Kinga Pszeniczna**

**Promotor: dr inż. Łukasz Piątek**

**Słowa kluczowe: aplikacja, czytelnia, czytelnik, książka, autor**

Utworzona aplikacja internetowa pozwala na przeglądanie informacji o autorach oraz książkach, z możliwością pobrania wybranej książki. Strona została napisana w technologii ASP .NET MVC 5 Entity Framework z wykorzystaniem Bootstrapa, dzięki któremu strona jest responsywna. Do przechowywania danych wykorzystano narzędzie bazodanowe SQL Server. Zarządzanie danymi odbywa się dzięki narzędziu SQL Server Management Studio. Aplikacja została napisana w środowisku Visual Studio Community 2019. Książki wykorzystane do projektu są wolne od praw autorskich. Wkładem własnym było napisanie aplikacji, utworzenie bazy danych oraz wykonanie dokumentacji do projektu.

# **LITERATURA**

https://www.spidersweb.pl/rozrywka/2019/03/25/czytelnictwo-w-polsce-2018-raport-opinia/

https://www.wirtualnemedia.pl/artykul/ponad-60-proc-polakow-nie-przeczytalo-w-ubieglym-roku-zadnej-ksiazki-czytelnictwo-biblioteka-narodowa

https://wiedzainformatyczna.pl/asp-net-entity-framework/#.Xk7zayhKjIW

http://lukaszkosiorowski.pl/programowanie/net/asp-mvc/kurs-asp-mvc-5/

https://psd2css.pl/baza-wiedzy/co-to-jest-bootstrap

https://www.plukasiewicz.net/Artykuly/NetFrameworkVsNetCore

# **SPIS ILUSTRACJI**

Rys. 1 Wykres ilości czytanych książek w latach 2000 do 2018 9

Rys. 2 Źródła pozyskiwania książek w latach 2014 do 2018 10

Rys. 3 Fragment kodu odpowiadający za połączenie z bazą danych 11

Rys. 4 Fragment kodu odpowiedzialnego za pasek menu 11

Rys. 5 Model bazy tabeli Author 12

Rys. 6 Model tabeli AuthorBook 12

Rys. 7 Fragment kodu CzytelniaContext 13

Rys. 8 Fragment kodu AuthorController 14

Rys. 9 Zakończenie funkcji weryfikacji logowania użytkownika 14

Rys. 10 Fragment kodu pliku CategoryController.cs 14

Rys. 11 Fragment kodu pliku Index.cshtml 15

Rys. 12 Fragment kodu pliku Details.cshtml 16

Rys. 13 Fragment kodu pliku Details.cshtml widoku Book 16

Rys. 14 Diagram zależności między tabelami 17

Rys. 15 Ekran logowania do zdalnego serwera z bazą danych 18

Rys. 16 Widok edycji rekordów w tabeli Books 18

Rys. 17 Ekran rejestracji do systemu 19

Rys. 18 Ekran logowania 19

Rys. 19 Strona główna aplikacji 20

Rys. 20 Lista Autorów 20

Rys. 21 Pasek adresu 20

Rys. 22 Posortowana lista autorów 21

Rys. 23 Filtrowanie danych 21

Rys. 24 Widok szczegółów wybranego autora 21

Rys. 25 Szczegóły wybranej książki 22

Rys. 26 Widok menu aplikacji w niskiej rozdzielczości 22

Rys. 27 Logowanie do systemu - wersja mobilna 23

Rys. 28 Lista autorów - wersja mobilna 23

Rys. 29 Widok szczegółów dla wersji mobilnej 24

Rys. 30 Szczegóły książek - wersja mobilna 24

Rys. 31 Szczegóły książki - wersja mobilna 25

# **ZAŁĄCZNIKI**