****

**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](#_Toc33044547)

[**1.** **WPROWADZENIE** 5](#_Toc33044548)

[**1.1.** **Opis technologii** 5](#_Toc33044549)

[**1.2.** **Opis środowiska programistycznego** 6](#_Toc33044550)

[**1.3.** **Porównanie wykorzystanych technologii do alternatywnych rozwiązań** 6](#_Toc33044551)

[**2.** **SPECYFIKACJA SYSTEMU** 7](#_Toc33044552)

[**2.1.** **Wymagania funkcjonalne i niefunkcjonalne** 7](#_Toc33044553)

[**2.2.** **Podział systemu** 7](#_Toc33044554)

[**3.** **CHARAKTERYSTYKA PROBLEMU** 8](#_Toc33044555)

[**3.1.** **Określenie problemu** 8](#_Toc33044556)

[**3.2.** **Cel projektu** 9](#_Toc33044557)

[**4.** **IMPLEMENTACJA** 10](#_Toc33044558)

[**5.** **PREZENTACJA SYSTEMU** 13](#_Toc33044559)

[**STRESZCZENIE** 14](#_Toc33044560)

[**LITERATURA** 15](#_Toc33044561)

[**SPIS ILUSTRACJI** 16](#_Toc33044562)

[**ZAŁĄCZNIKI** 17](#_Toc33044563)

# **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. Mobilna aplikacja 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 funkcjonalności 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**

Wymagania funkcjonalne:

1. Użytkownik loguje się do systemu
2. Użytkownik odszukuje wybranego autora
3. Użytkownik odszukuje wybraną książkę
4. Użytkownik może filtrować dane
5. Użytkownik może pobrać wybraną książkę
6. Administrator dodaje dane o autorach w bazie danych
7. Administrator dodaje dane o książkach w bazie danych

Wymagania niefunkcjonalne:

1. System powinien mieć intuicyjny interfejs graficzny
2. System powinien się rozwijać
   1. **Podział systemu**

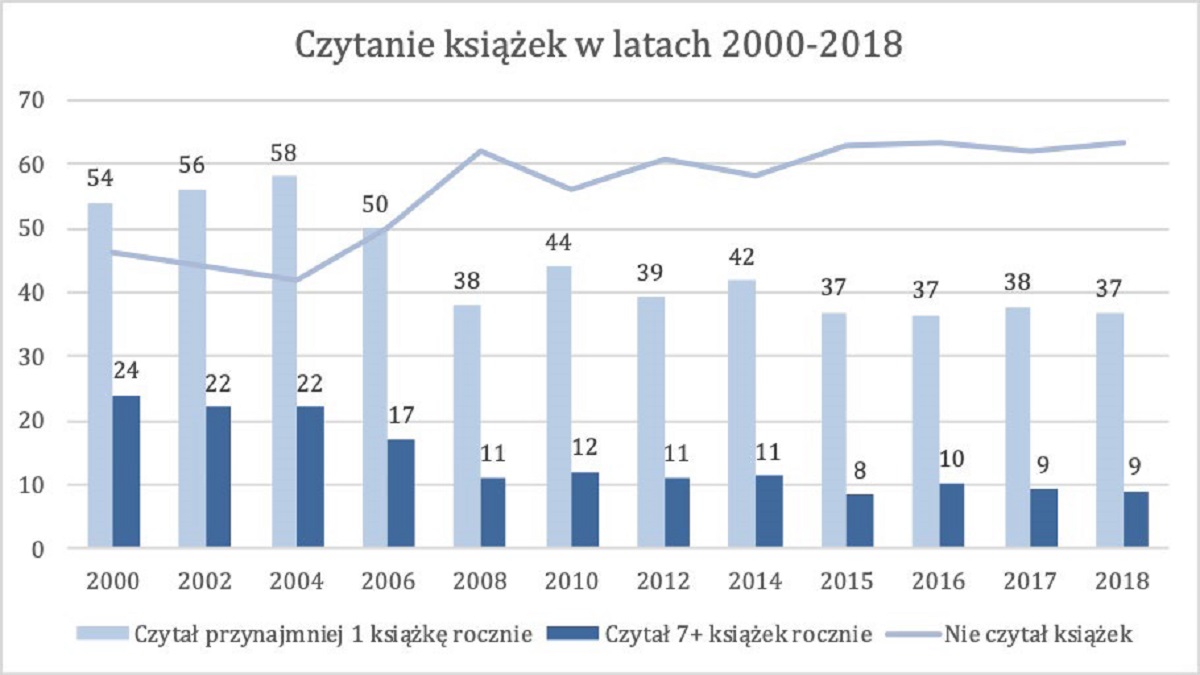
System został podzielony na część, którą zarządza administrator oraz na część, którą obsługuje użytkownik. W części administracyjnej zarządzanie odbywa się w SQL Server Management Studio, na serwerze zdalnym. W dalszym planie rozwoju systemu przewidziane jest utworzenie panelu administracyjnego bezpośrednio 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.

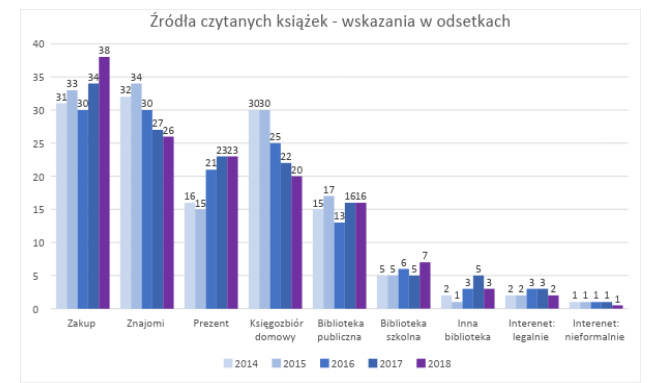
* 1. **Określenie problemu**

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

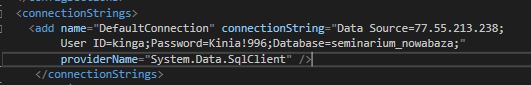
* 1. **Cel projektu**

Celem utworzonej aplikacji jest zbliżenie potencjalnego człowieka, aby otworzył się na wszelkiego 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**

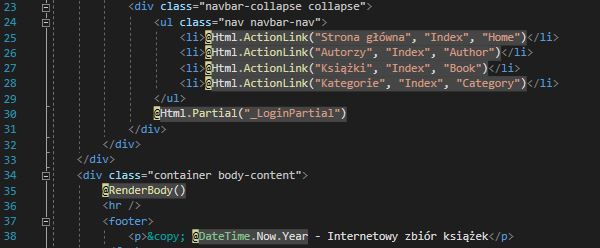
W tym rozdziale zostaną przedstawione fragmenty kodu z tworzenia aplikacji. Związane one będą z funkcjonalnościami utworzonymi, aby zrealizować cel projektu.

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.



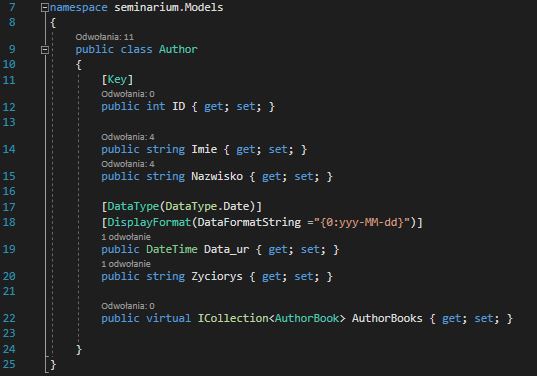
Rys. 3 Fragment kodu odpowiadający za połączenie z bazą danych

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.



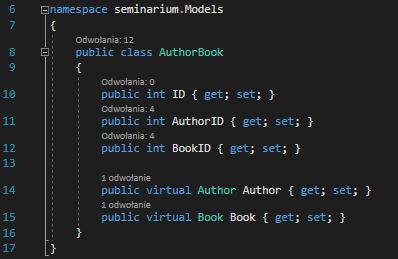
Rys. 4 Fragment kodu odpowiedzialnego za pasek menu

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.



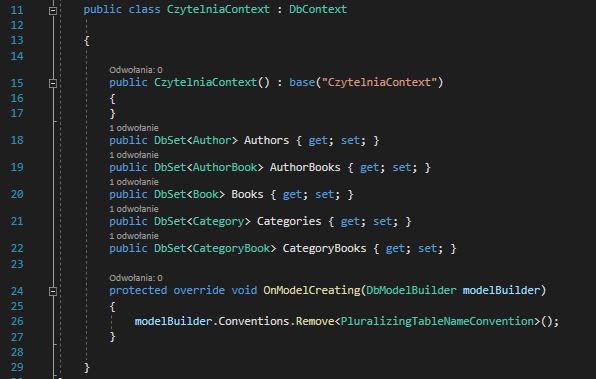
Rys. 5 Model bazy tabeli Author

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.



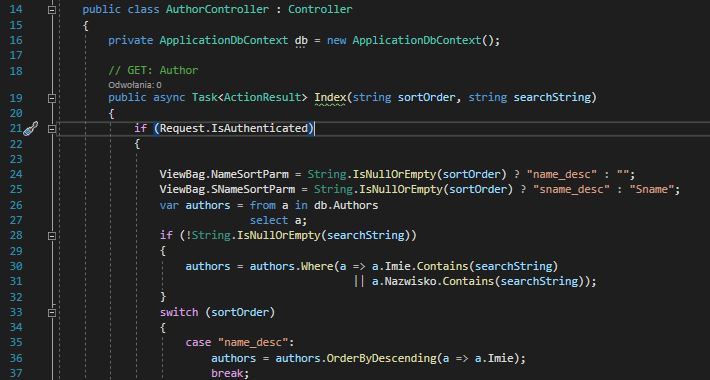
Rys. 6 Model tabeli AuthorBook

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



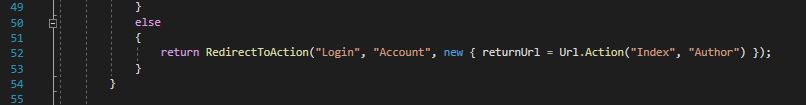
Rys. 7 Fragment kodu CzytelniaContext

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.



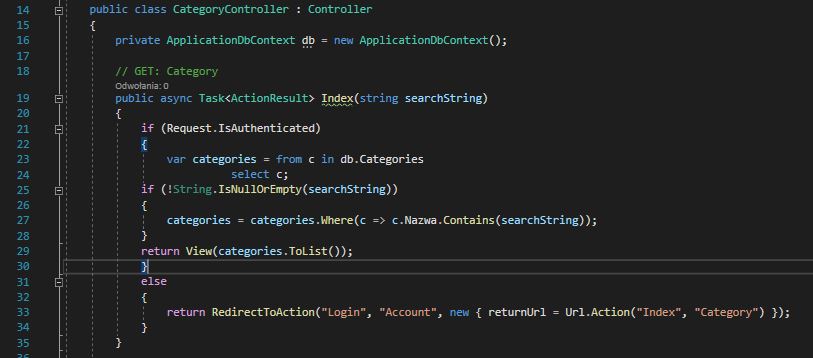
Rys. 8 Fragment kodu AuthorController

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.



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.



Rys. 10 Fragment kodu pliku CategoryController.cs

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

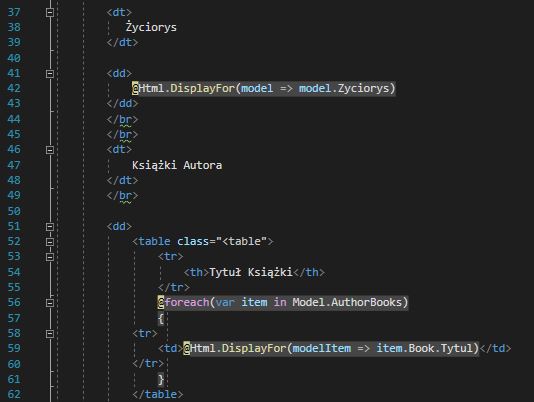
* Index.cshtml, który zawiera listę rekordów wprowadzonych do bazy,
* Create.cshtml, plik ten umożliwia utworzenie nowego rekordu,
* Details.cshtml, zawiera on szczegóły wybranego rekordu,
* Edit.cshtml, ten widok umożliwia edytowanie wybranego elementu,
* Delete.cshtml, ten plik pozwoli na usunięcie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



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

# **PREZENTACJA SYSTEMU**

# **STRESZCZENIE**

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

**Kolegium Informatyki Stosowanej**

**Streszczenie pracy dyplomowej**

*Internetowy zbiór książek*

**Autor:**

**Promotor:**

**Słowa kluczowe:**

# **LITERATURA**

Alexander M., Kusleika D. *„Microsoft Access 2013. Biblia. Kompendium wiedzy każdego użytkownika!”*, tłum. Walczak T., Meryk R., Helion, 2014

Górczyński J., Górczyński P. *„Projektowanie baz danych w MS Access, wykorzystanie VBA”*, Wyższa Szkoła Zarządzania i Marketingu w Sochaczewie, 2005

Rockoff L. *„The Language of SQL, Second Edition”*, tłum. Błaszczyk B., Helion 2017

# **SPIS ILUSTRACJI**

Rys. 1 Przykładowa relacja pomiędzy tabelami 5

Rys. 2 Tabela Produkty 12

Rys. 3 Tabela Kolory 13

Rys. 4 Tabela Producenci 13

Rys. 5 Tabela Biura\_sprzedazy 14

Rys. 6 Widok projektu tabeli Klienci 14

Rys. 7 Tabela Stawka\_VAT 15

Rys. 8 Widok projektu tabeli Zamówienia 15

Rys. 9 Widok projektu tabeli Zaliczki 16

Rys. 10 Widok projektu tabeli Faktury 16

Rys. 11 Relacje zachodzące pomiędzy tabelami 17

Rys. 12 Projekt formularza Wprowadzanie\_faktur 18

Rys. 13 Projekt formularza Wprowadzanie\_zamowien 18

Rys. 14 Widok projektu kwerendy Zamówienia 19

Rys. 15 Projekt raportu Dla\_klienta 19

Rys. 16 Projekt kwerendy Sprzedaz 20

Rys. 17 Projekt raportu Sprzedaz 20

Rys. 18 Lista klientów zapisanych do tabeli Klienci podczas tworzenia zamówienia 21

Rys. 19 Przykładowe zamówienie 21

Rys. 20 Lista dostępnych produktów 22

Rys. 21 Lista dostępnych kolorów 22

Rys. 22 Gotowy do wydruku raport z zamówienia 23

Rys. 23 Widok tabeli Zamowienia 24

Rys. 24 Ustawienia kwerendy Zamówienie 24

Rys. 25 Wynik wyszukiwania kwerendy Zamówienia 24

Rys. 26 Widok tabeli Faktury 25

Rys. 27 Widok wyszukiwania kwerendy Sprzedaż 25

Rys. 28 Raport ze sprzedaży w miesiącu grudniu 2018r. w rozbiciu na poszczególne biura sprzedaży 26

Rys. 29 Raport podsumowujący IV kwartał 2018 roku 26

Rys. 30 Wynik wyszukiwania blachy modułowej w tabeli Produkty 27

Rys. 31 Dostępność kolorów u poszczególnego producenta w tabeli Kolory 27

# **ZAŁĄCZNIKI**