
	Uniwersytet Technologiczno-Przyrodniczy im. J. i J. Śniadeckich w Bydgoszczy Wydział Telekomunikacji, Informatyki i Elektrotechniki	
Przedmiot	Programowanie w środowisku Windows	
Prowadzący	Dr inż. J. Zdrojewski	
Projekt	System rejestracji i umawiania wizyt lekarskich	
Student	Łukasz Farulewski	
Data wykonania	25.01.2020	
Data oddania	25.01.2020	

1. Wstęp

Celem projektu jest przygotowanie aplikacji webowej w języku **C#** na platformie **ASP.NET CORE** pod nazwą „System rejestracji i umawiania wizyt lekarskich”. Do budowania wizualnej części aplikacji internetowej wykorzystano framework **Bootstrap**.

ASP.NET Core to międzyplatformowa struktura typu open source o wysokiej wydajności służąca do tworzenia nowoczesnych aplikacji internetowych opartych na chmurze. Platforma ASP.NET Core umożliwia:

- kompilowanie aplikacji i usług internetowych, aplikacji IoT i zapleczy mobilnych,
- używanie ulubionych narzędzi programistycznych w systemach Windows, macOS i Linux,
- wdrażanie w chmurze lub lokalnie,
- uruchamianie na platformie .NET Core lub .NET Framework. ¹

Aplikacja została zbudowana w architekturze Model-View-Controller (MVC). Wzorzec MVC umożliwia oddzielenie aplikacji od trzech głównych komponentów – modelu (Model), widoku (View) oraz kontrolera (Controller). Zadaniem **kontrolera** jest zarządzanie przychodzącymi zapytaniami http i odpowiadaniem na nie określonymi akcjami i metodami, np. kontroler zwraca wybrane widoki do użytkownika. **Model** odpowiada za logikę biznesową i w pewnym sensie stanowi reprezentację bazy danych, która jest właśnie odzwierciedleniem Modelu – nazwa tabeli to nazwa klasy Modelu, natomiast zawarte w klasie Modelu pola (properties) to kolumny. Nazwy kolumn można zmienić poprzez walidację – [Display(Name = "Imię")]. **Widok** w architekturze MVC reprezentuje graficzny interfejs użytkownika, czyli wszystkie te elementy, które są widoczne na monitorze i z którymi użytkownik wchodzi w interakcję. Są to najczęściej pliki html lub jak w przypadku przedstawionej aplikacji – widoki typu razor (Razor view) – czyli połączenie języka html oraz CS. Są to wszystkie widoki w aplikacji, począwszy od strony głównej po formularze do wypełnienia przez użytkownika – np. rejestracja i logowanie pacjenta, dodawanie Lekarza czy modyfikacja danych.

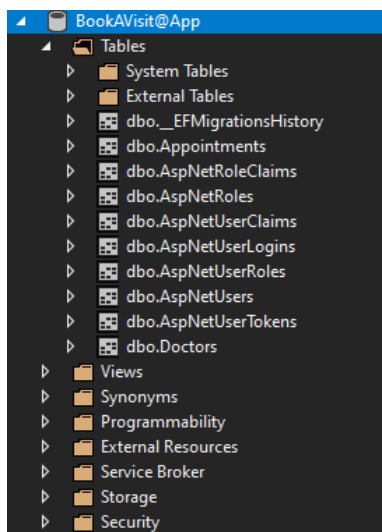
¹ <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>

2. Połączenie z bazą danych

W aplikacji wykorzystano SQL Server Local DB. Połączenie z bazą danych skonfigurowano w pliku appsettings.json

```
9  "AllowedHosts": "*",
10  "ConnectionStrings": {
11    "BookAVisitAppConnection": "Server=(localdb)\\mssqllocaldb;Database=BookAVisit@App;Trusted_Connection=True;MultipleActiveResultSets=true"
12  }
13  }
14
```

Widok 1. Plik konfiguracyjny appsettings.json



Widok 2. Baza danych BookAVisit@App

3. Wykaz tabel

Tabela: Appointments => Model.Appointment

	AppointmentID	DoctorID	UserName	AppointmentDate
▶	2	7	lfarulewski@yahoo.com	24.01.2020 06:00:00
	3	7	pit@gmail.com	28.01.2020 08:00:00
⬇	NULL	NULL	NULL	NULL

Tabela 1. Appointments

Tabela: Doctors => Model.Doctor

Max Rows: 1000							
	DoctorID	FirstName	LastName	Specialty	Email	Description	PhotoPath
▶	6	Dawid	Podsiadlo	Ortopeda	podsiadlo@klinika.com	Lekarz ortopeda zajmuje ...	1a5f4c53-fcd8-412f-b16c-aa7e0dfa6be7_doc-1.jpg
	7	Gary	Lineker	Laryngolog	lineker@klinika.com	Specjalizuje się w profila...	63943662-69c1-408b-963f-adc5c57ecede_doc-3.jpg
	8	Grzegorz	Nowakowski	Stomatolog	nowakowski@klinika.com	Zajmuje się funkcjonow...	cc99c5a0-b3b6-44d4-af39-9a027c977d4f_doc-8.jpg
	9	Małgorzata	Kowalska	Pediatra	kowalska@klinika.com	Specjalista w zakresie pe...	1439d61b-c0f7-4cc9-9077-4cfa57aba478_doc-5.jpg
	10	Katarzyna	Michalska	Okulista	michalska@klinika.com	Zajmuje się wadami wzr...	16d39ccf-ef44-45c1-9426-fde0ce92c960_doc-4.jpg
	11	Magdalena	Zawadzka	Dermatolog	zawadzka@klinika.com	Diagnostuje i leczy choro...	e6e01a0e-84c3-4462-96ca-20ef719da07a_doc-2.jpg
	12	Monika	Wiosna	Kardiolog	wiosna@klinika.com	Zajmuje się chorobami s...	50e30913-ff93-48a1-990e-bf7e74fc2fd4_team-4.jpg
	13	Tomasz	Kowalczyk	Ginekolog	kowalczyk@klinika.com	Ginekologia to dziedzina...	8c9a3c0b-66aa-4b60-b022-9e143822b053_team-3.jpg
⚙	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela 2. Doctors

W aplikacji nie zaimplementowano Modelu klasy Pacjent i jej odpowiednika w bazie danych – tabeli Pacjent. Do przechowywania danych Pacjentów wykorzystano tabelę Users. Wynika to z tego, że wraz z rejestracją użytkownika, np. pacjenta, od razu pojawia się on w bazie Users. Klasa IdentityUser została rozszerzona o klasę ApplicationUser, gdzie zostały zdefiniowane dodatkowe pola, takie jak: FirstName, LastName oraz Pesel.

Tabela: dbo.AspNetUsers

Max Rows: 1000																		
	ID	UserName	NormalizedUs...	Email	NormalizedEm...	Email...	PasswordHash	SecurityStamp	ConcurrencySt...	Phon...	Phon...	TwoFa...	LockoutEnd	LockoutEnabled	AccessFailedC...	Pesel	FirstName	LastName
▶	64649239-a579...	pit@gmail.com	PIT@GMAIL.CO...	pit@gmail.com	PIT@GMAIL.CO...	False	AQAAAAEAAC...	KWGPXCINXW...	fa8d140-229a...	NULL	False	False	NULL	True	0	7777777777	Bratt	Pit
▶	e47a6d92-ceb4...	lfarulewski@ya...	LFARULEWSKI...	lfarulewski@ya...	LFARULEWSKI...	False	AQAAAAEAAC...	MUXRHPGRGP...	f6ea84e-f22e-4...	NULL	False	False	NULL	True	0	1111111111	Lukasz	Farulewski
⊞	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabela 3. Użytkownicy (Users)

4. Założenia do diagramów E-R

1. Każdy pacjent może umówić więcej niż jedną wizytę, ale każda wizyta musi być umówiona przez jednego pacjenta.
2. Wizyta może być umówiona do więcej niż jednego lekarza, ale każdy lekarz może mieć jedną wizytę.
3. Funkcja pomocnicza Programista pozwala na modyfikowanie danych wielu lekarzy, ale dane lekarza mogą być modyfikowane przez 1 programistę.

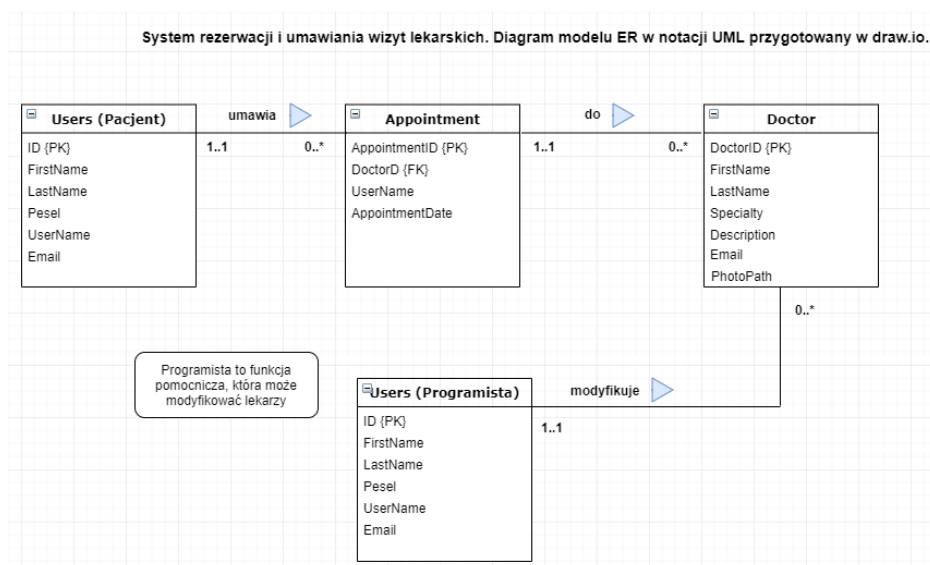


Diagram 1. Diagram modelu E-R w notacji UML

5. Aplikacja

Celem projektu jest przygotowanie aplikacji webowej w języku **C#** na platformie **ASP.NET CORE** pod nazwą „System rejestracji i umawiania wizyt lekarskich”. Aplikacja ma za zadanie odzwierciedlać stronę internetową przychodni lekarskiej wraz z jej podstawowymi funkcjonalnościami – ogólny opis przychodni i jej lekarzy, kontakt do przychodni, rejestrację i logowanie pacjentów, umawianie wizyt lekarskich oraz pomocniczo administrowanie stroną (Panel programisty). Administracja stroną internetową (aplikacją) jest funkcją pomocniczą, która służy do dodawania nowych lekarzy, edycji i usuwania już zapisanych lekarzy oraz sprawdzania wszystkich zapisanych wizyt.

W aplikacji wykorzystano mechanizm **Identity**. ASP.Net Core Identity to API, które wspiera graficzny interfejs użytkownika w procesie logowania a także zarządza użytkownikami, hasłami, danymi profilowymi, rolami, tokenami, potwierdzaniem emaili oraz wspiera programistę w wielu innych zadaniach.²

Aplikacja została przygotowana z perspektywy pacjenta, który po zapoznaniu się z ofertą przychodni, może zarejestrować się, zalogować i umówić wizytę.

Na stronie głównej aplikacji udostępniony został ogólny widok i podstawowe informacje – Home, O nas, Kontakt oraz stopka z informacjami kontaktowymi. Zaimplementowano tutaj pokaz slajdów (Carousel). W związku z tym, na niektórych przedstawianych niżej widokach strona główna może się różnić.

² <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-3.1&tabs=visual-studio>


Witamy w Centrum medycznym BookAVisit@App



© 2020 - Przychodnia medyczna BookAVisitApp, Bydgoszcz, ul. Gdańska 21, telefon: 347-764-238.

Widok 3. Ogólny widok na stronę główną

Po wybraniu zakładki „O nas” otrzymujemy podstawowe informacje o przychodni.



O nas


BookVisit@App to przychodnia lekarska istniejąca w Bydgoszczy. Każdego dnia w trosce o Państwa zdrowie i poczucie komfortu dbamy o najwyższą jakość naszych usług. Dzięki staraniom wykwalifikowanego, doświadczonego personelu mają Państwo dostęp do profesjonalnego i kompleksowego leczenia w obrębie jednej przychodni.

[Poznaj nas](#)

Widok 4. Widok zakładki „O nas”

Przejsie do widoku „Poznaj nas” przenosi użytkownika do listy dostępnych lekarzy wraz z opisem ich specjalizacji. Tutaj mamy do wyboru – „Umów wizytę” lub „Powrót” do strony głównej.

Wybierz lekarza



Ortopeda
lek. Dawid Podsiadło

Lekarz ortopeda zajmuje się diagnozowaniem i leczeniem chorób bądź urazów, które są związane z narządem ruchu. Lekarz ortopeda zajmuje się chorobami narządu ruchu.

[Umów wizytę](#) [Powrót](#)



Laryngolog
lek. Gary Lineker

Specjalizuje się w profilaktyce, diagnozowaniu i leczeniu chorób narządów głowy i szyi. Wśród nich znajdują się: objawy związane z uchem, w tym uczucie zatkanego ucha, szumy uszne, ból ucha, pogorszenie słuchu oraz krwawienie z nosa, przewlekły katar, chrapanie.

[Umów wizytę](#) [Powrót](#)

Widok 5. Widok zakładki „Poznaj nas” na przykładzie 2 lekarzy

Próba umówienia wizyty przed zalogowaniem – „Umów wizytę”, zwraca użytkownikowi widok z zachętą do rejestracji i logowania lub samego logowania, jeżeli użytkownik posiada już konto.



Przed umówieniem wizyty, proszę się zalogować.

Czy już się rejestrowałeś? Proszę potwierdzić rejestrację.

[Zarejestruj się](#)

[Zaloguj się](#)

[Powrót](#)

Widok 6. Zachęta do rejestracji lub logowania przed umówieniem wizyty

Jeżeli użytkownik nie posiada konta, to musi się zarejestrować. Po pomyślnej rejestracji może przejść do logowania.

Zarejestruj się i załóż konto

Pesel

2222222222

First Name

David

Last Name

Beckham

Email

beckham@gmail.com

Password

.....

Confirm password

.....

[Rejestruje się](#)

[Powrót](#)

Widok 7. Rejestracja pacjenta

Zaloguj się

Email

beckham@gmail.com

Password

.....

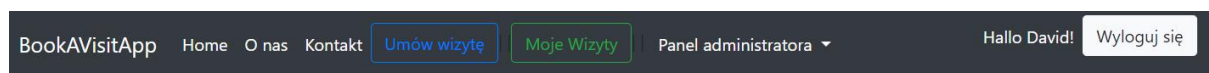
☐ Remember Me?

[Loguję się](#)

[Powrót](#)

Widok 8. Logowanie pacjenta

Po pomyślnym logowaniu użytkownik jest witany na stronie głównej.




Witamy w Centrum medycznym BookAVisit@App David



Widok 9. Strona główna po zalogowaniu pacjenta

W przedstawionym wyżej scenariuszu logowania – użytkownikiem jest David Beckham. Wszyscy pacjenci po zalogowaniu mają dostępne dwa moduły – „Umów wizytę” oraz „Moje wizyty”. Pierwszy pozwala na wybranie lekarza i umówienie wizyty a drugi na wyświetlenie wszystkich umówionych wizyt zalogowanego pacjenta. Po wybraniu opcji „Umów wizytę” użytkownik zostanie przeniesiony do widoku, który prezentowany był wyżej (widok 5), z tą różnicą, że teraz mechanizm umówienia wizyty jest odblokowany oraz użytkownik jest rozpoznany przez system i zachęcany imiennie do umówienia wizyty. Na tym etapie pacjent może oczywiście zrezygnować z umawiania wizyty i wrócić do strony głównej – „Powrót”.

Wybierz lekarza David




Ortopeda

lek. Dawid Podsiadło

Lekarz ortopeda zajmuje się diagnozowaniem i leczeniem chorób bądź urazów, które są związane z narządem ruchu. Lekarz ortopeda zajmuje się chorobami narządu ruchu.

[Umów wizytę](#)[Powrót](#)



Laryngolog

lek. Gary Lineker

Specjalizuje się w profilaktyce, diagnozowaniu i leczeniu chorób narządów głowy i szyi. Wśród nich znajdują się: objawy związane z uchem, w tym uczucie zatkanego ucha, szumy uszne, ból ucha, pogorszenie słuchu oraz krwawienie z nosa, przewlekły katar, chrapanie.

[Umów wizytę](#)[Powrót](#)

Widok 10. Moduł „Umów wizytę” po zalogowaniu

Jeżeli Pacjent zdecyduje się na umówienie wizyty – „Umów wizytę”, zostanie odesłany do kolejnego widoku, gdzie będzie musiał podać datę wizyty.



Umów wizytę David

Pacjent

Imie: David

Nazwisko: Beckham

Email: beckham@gmail.com

Pesel: 2222222222

UserName: beckham@gmail.com

Lekarz

LekarzID: 6

Imie: Dawid

Nazwisko: Podsiadło

Specjalizacja: Ortopeda

Data wizyty

Proszę podać datę i godzinę wizyty

dd.mm.rrrr --:--

[Umawiam](#) [Powrót](#)

Widok 11. Użytkownik David umawia wizytę

Jeżeli pacjent próbuje umówić wizytę do lekarza na już zajęty termin – datę i godzinę, to zostanie o tym poinformowany i wizyta nie zostanie umówiona.



Przykro nam, ale lekarz do którego próbujesz się umówić masz już umówioną wizytę w tym terminie.

Proszę wybrać inny dzień lub godzinę.

[Powrót](#)

Widok 12. Próba umówienia wizyty na już zajęty termin

W przeciwnym razie, jeżeli termin jest wolny, wizyta zostaje umówiona i zostaje zapisana w bazie danych. Widok bazy z poziomu aplikacji oraz serwera SQL.

Dzisiaj jest: 1/24/2020 10:23:55 AM

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	Edytuj Szczegóły Usuń

Widok 13. Zapis wizyty do bazy danych (Moje wizyty) – widok z aplikacji

	AppointmentID	DoctorID	UserName	AppointmentDate
▶	2	7	lfarulewski@yahoo.com	24.01.2020 06:00:00
	3	7	pit@gmail.com	28.01.2020 08:00:00
	4	6	pit@gmail.com	30.01.2020 08:00:00
	5	6	beckham@gmail.com	29.01.2020 11:00:00
⊞	NULL	NULL	NULL	NULL

Widok 14. Zapis wizyty do bazy danych (wszystkie wizyty) – widok z servera SQL

W powyższym widoku (widok 11) przekazuję do formularza appointment.cshtml pacjenta za pośrednictwem UserManagera (@UserManager.GetUserAsync(User)). Umożliwia on pobranie danych osoby zalogowanej i umawiającej wizytę. Dane lekarza pobierane są za pomocą @Model i implementacji @model BookAVisit_App.ViewModels.AppointmentViewModel do widoku. Pola do widoku muszą zostać wcześniej zaimplementowane albo w Modelu albo w klasie ViewModel. Zgodnie z dobrymi praktykami w aplikacji wykorzystano ViewModel – klasę, która pośredniczy między Modelem a widokiem i zawiera wszystkie te pola, które w Modelu nie zostały uwzględnione a są potrzebne w widoku. Należy jednak pamiętać, że tylko pola z klasy Model będą zapisane do bazy danych.

Pacjent może skorzystać z podglądu swoich wizyt poprzez funkcję – „Moje Wizyty”.


Dzisiaj jest: 1/24/2020 10:29:16 AM

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	Edytuj Szczegóły Usuń
6	8	beckham@gmail.com	2/11/2020 12:00:00 PM	Edytuj Szczegóły Usuń
7	10	beckham@gmail.com	2/18/2020 8:00:00 AM	Edytuj Szczegóły Usuń

Widok 15. Funkcja Moje Wizyty

Każdą z wizyt można następnie edytować, zapoznać się ze szczegółami oraz usunąć. Podczas edycji danych, można dokonać zmian.

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	Edytuj Szczegóły Usuń



Aktualizujesz dane David ?

Lekarz ID

Proszę podać ID Lekarza

6

Email pacjenta

Proszę podać Email pacjenta

beckham@gmail.com

Data wizyty

Proszę podać datę i godzinę wizyty

29.01.2020 11:00


Aktualizuje

Powrót

Widok 16. Edycja wizyty pacjenta

Szczegóły przenoszą użytkownika do szczegółów jego wizyty w której dostępna jest edycja a także możliwość zapisu podsumowania w pliku PDF.

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	<div>Edytuj</div> <div>Szczegóły</div> <div>Usuń</div>



Szczegóły Twojej wizyty David

DoctorID

6

UserName

beckham@gmail.com

Appointment Date

29-01-2020


Edytuj

Powrót

Wydruk do PDF

Widok 17. Szczegóły wizyty pacjenta

BookAVisitApp



Dziękujemy za umówienie wizyty David

Podsumowanie Twojej wizyty

DoctorID
6

UserName
beckham@gmail.com

Appointment Date
29-01-2020

Widok 18. Zapisanie wizyty do pliku PDF

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	Edytuj Szczegóły Usuń

Przed usunięciem użytkownik jest proszony o potwierdzenie usunięcia danych.

Komunikat ze strony localhost:44350

Czy na pewno chcesz usunąć wizytę: 5 ?

OK

Anuluj

Widok 19. Prośba o potwierdzenie usunięcia danych

Na stronie głównej aplikacji znajduje się również zakładka – Kontakt z danymi kontaktowymi do przychodni.



Przychodnia medyczna BookAVisit@App

Skontaktuj się z nami:

-> Bydgoszcz, ul. Gdańska 21

-> Poniedziałek - Sobota, 8.00 - 20.00

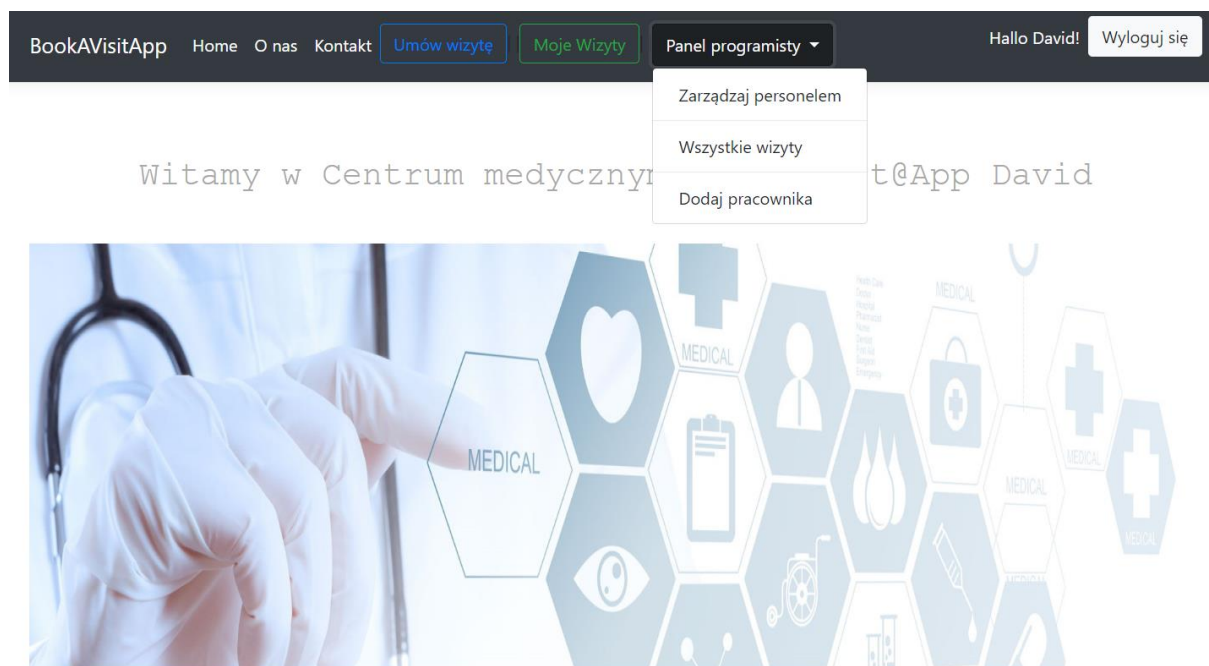
-> 347-764-238

-> bookvisit@kontakt.pl

[Powrót](#)

Widok 20. Kontakt do przychodni

Mimo, że aplikacja została przygotowana z perspektywy pacjenta, została dodana także funkcja programisty, która umożliwia administrowanie częścią danych – podgląd wszystkich wizyt, możliwość dodawania, edytowania i usuwania lekarzy.



Widok 21. Strona główna – Panel programisty

Panel programisty pozwala zobaczyć wszystkie umówione wizyty – edytować je, przejść do szczegółów oraz usunąć je. Wszystkie te funkcjonalności zostały opisane wyżej, przy omawianiu funkcji pacjenta, który ma także dostęp do swoich wizyt – może je edytować, usunąć lub zapisać podsumowanie w pliku PDF.

ID wizyty	ID lekarza	Email	Data wizyty	Co chcesz zrobić?
2	7	lfarulewski@yahoo.com	1/24/2020 6:00:00 AM	Edytuj Szczegóły Usun
3	7	pit@gmail.com	1/28/2020 8:00:00 AM	Edytuj Szczegóły Usun
4	6	pit@gmail.com	1/30/2020 8:00:00 AM	Edytuj Szczegóły Usun
5	6	beckham@gmail.com	1/29/2020 11:00:00 AM	Edytuj Szczegóły Usun
6	8	beckham@gmail.com	2/11/2020 12:00:00 PM	Edytuj Szczegóły Usun
7	10	beckham@gmail.com	2/18/2020 8:00:00 AM	Edytuj Szczegóły Usun


Widok 22. Panel programisty – Wszystkie wizyty

Funkcja „Zarządzaj personelem” wyświetla wszystkich dostępnych lekarzy oraz umożliwia dodanie lekarza, aktualizację danych oraz usunięcie pracownika z bazy.

Wykaz pracowników


[Dodaj lekarza](#)

Ortopeda
lek. Dawid Podsiadło




[Podgląd](#) [Edytuj](#) [Usuń](#)

Laryngolog
lek. Gary Lineker




[Podgląd](#) [Edytuj](#) [Usuń](#)

Stomatolog
lek. Grzegorz Nowakowski



[Podgląd](#) [Edytuj](#) [Usuń](#)

Pediatra
lek. Małgorzata Kowalska



[Podgląd](#) [Edytuj](#) [Usuń](#)

Widok 23. Panel programisty – Zarządzaj personelem

Dodaj nowego lekarza

First Name

LastName

Specjalty

Email

Description

Photo

Załącz zdjęcie...

Browse

[Dodaj](#)

[Powrót](#)

Widok 24. Panel programisty – Zarządzaj personelem – Dodaj lekarza

Dane konkretnego lekarza można edytować i aktualizować, zmieniając np. adres email, zdjęcie.

First Name


LastName

Specjalty

Email

Description

Photo



Widok 25. Panel programisty – Zarządzaj personelem – Edytuj

Próba usunięcia lek. Dawid Podsiadło, skutkuje zapytaniem ze strony serwera i prośbą o potwierdzenie decyzji.

Komunikat ze strony localhost:44350

Czy na pewno chcesz usunąć lekarza: Dawid Podsiadło ?

Widok 26. Prośba o potwierdzenie usunięcia lekarza z bazy

6. Wnoski i podsumowanie

Projekt bardzo ciekawy. ASP.NET Core to stosunkowo nowy framework, ale posiada bardzo dobre wsparcie ze strony Microsoftu jeżeli chodzi o dostępność dokumentacji, materiałów szkoleniowych oraz tutoriali.

Mechanizm Identity umożliwił sprawną implementację systemu rejestracji i logowanie wraz ze wsparciem bazodanowym. Rozszerzenie klasy IdentityUser o klasę ApplicationUser pozwolił zmodyfikować dostępne pola w tabeli SQL server. Serwis UserManager umożliwił pobranie z bazy

zalogowanych użytkowników. Do przekazania danych z kontrolera do widoku wykorzystano mechanizm @ViewBag oraz @Model.