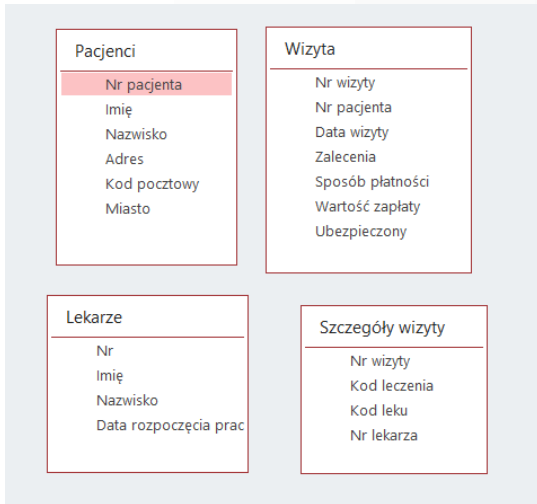


1. Schemat bazy danych – Clinic (Przychodnia) / Dokumentacja

Kacper Kędzior 2t

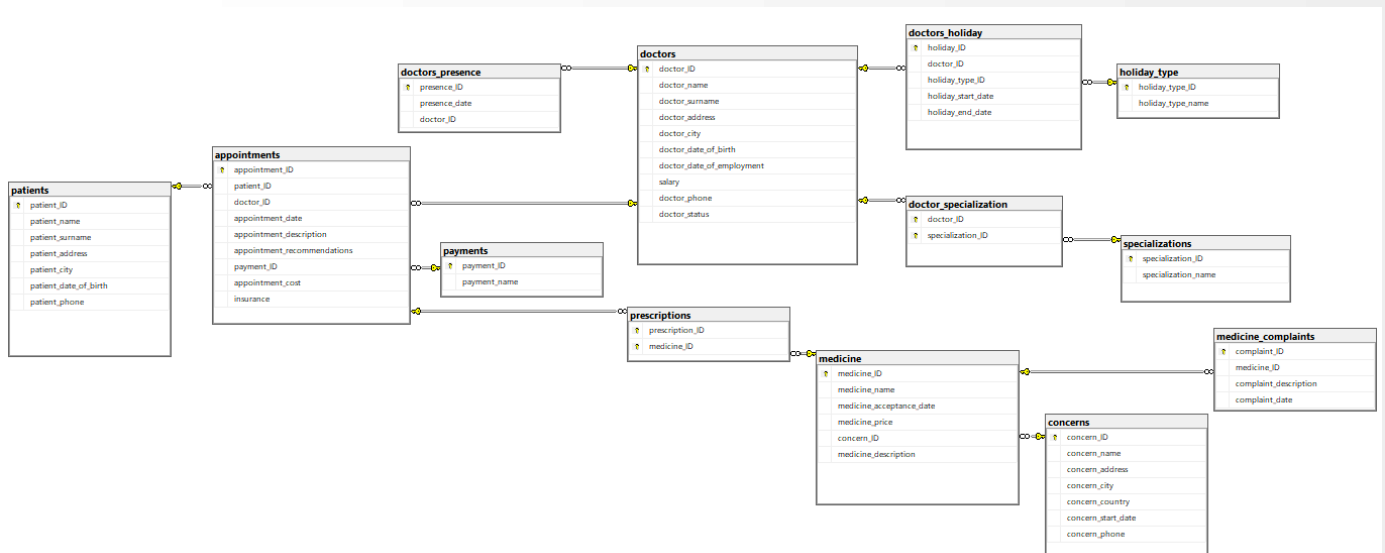
1.1 Tabele przed moją ingerencją

Zrzut ekranu z programu Microsoft Access:



1.2 Końcowy schemat bazy danych bez widocznych szczegółów pól

Zrzut ekranu z programu Microsoft SQL Server Management Studio 18:



1.3 Końcowy schemat bazy danych z widocznymi szczegółami pól

Zrzut ekranu z programu Microsoft SQL Server Management Studio 18:



1.4 Opis tabel:

Nazwa tabeli (kol. alfabetycznie)	Opis tabeli (bez uwzględnienia pól z kluczami podstawowymi)
<i>appointments</i>	Zawiera informacje o wizytach u pacjentów u lekarzy (ID pacjenta, ID lekarza, data wizyty, opis, wskazania lekarza, ID sposobu płatności, koszt wizyty, ubezpieczenie- tak czy nie)
<i>concerns</i>	Zawiera informacje o koncernach wprowadzających leki na rynek (nazwa, adres, miasto, kraj, data wejścia firmy na rynek, telefon)
<i>doctor_specialization</i>	Łączy lekarza do specjalizacji
<i>doctors</i>	Zawiera informacje o lekarzach pracujących w przychodni (imię, nazwisko, adres, miasto, data ur., data zatr., płaca, telefon, status – czy pracuje)
<i>doctors_holiday</i>	Zawiera informacje o urlopach lekarzy (ID lekarza, ID rodzaju urlopu, data rozp. urlopu, data zak., urlopu)
<i>doctors_presence</i>	Zawiera wpisy o obecności w pracy lekarzy (data obecności, ID lekarza)
<i>holiday_type</i>	Rodzaje urlopów (nazwa rodzaju urlopu)
<i>medicine</i>	Zawiera informacje o lekach (nazwa produktu, data akceptacji leku, cena, ID koncernu, opis leku)
<i>medicine_complaints</i>	Skargi / zażalenia składane odnośnie leku
<i>patients</i>	Zawiera informacje o pacjentach (imię, nazwisko, adres, miasto, data ur., telefon)
<i>payments</i>	Rodzaje płatności (nazwa rodzaju płatności)
<i>prescriptions</i>	Łączy przepisane leki do konkretnej wizyty /na recepcie/
<i>specializations</i>	Specjalizacje lekarzy (nazwa specjalizacji)

1.5 Relacje:

Nazwa tabeli (kol. według schematu)	Relacje
<i>patients</i> (PK: patient_ID AUTOINCREMENT)	- <i>appointments</i> (FK: patient_ID / jeden do wielu / jeden pacjent może mieć wiele wizyt)
<i>appointments</i> (PK: appointment_ID AUTOINCREMENT, FK: patient_ID, doctor_ID, payment_ID)	- <i>prescriptions</i> (FK: prescription_ID / jeden do wielu / w trakcie pojedynczej wizyty lekarz może przepisać wiele leków)
<i>payments</i> (PK: payment_ID AUTOINCREMENT)	- <i>prescriptions</i> (FK: payment_ID / jeden do wielu / jedna metoda płatności może być przypisana do wielu wizyt)
<i>prescriptions</i> (PK: prescription_ID, medicine_ID)	- <i>appointments</i> i <i>medicine</i> (spełnia między tymi tabelami relację wiele do wielu)
<i>medicine</i> (PK: medicine_ID AUTOINCREMENT, FK: concern_ID)	- <i>medicine_complaints</i> (FK: medicine_ID / jeden do wielu / jeden produkt może mieć wiele skarg) - <i>prescriptions</i> (FK: medicine_ID / jeden do wielu / jeden produkt może być przepisany wiele razy)
<i>concerns</i> (PK: concern_ID AUTOINCREMENT)	- <i>medicine</i> (FK: concern_ID / jeden do wielu / jeden koncern może wydać wiele produktów)
<i>medicine_complaints</i> (PK: complaint_ID AUTOINCREMENT, FK: medicine_ID)	
<i>doctors</i> (PK: doctor_ID AUTOINCREMENT)	- <i>doctors_presence</i> (FK: doctor_ID / jeden do wielu / jeden lekarz może mieć wiele wpisów obecności) - <i>appointments</i> (FK: doctor_ID / jeden do wielu / jeden lekarz może mieć wiele wizyt) - <i>doctors_holiday</i> (FK: doctor_ID / jeden do wielu / jeden lekarz może mieć wiele urlopów) - <i>doctor_specialization</i> (FK: doctor_ID / jeden do wielu / jeden lekarz może mieć wiele specjalizacji)
<i>doctors_presence</i> (PK: presence_ID AUTOINCREMENT, FK: doctor_ID)	
<i>doctors_holiday</i> (PK: holiday_ID AUTOINCREMENT, FK: doctor_ID, holiday_type_ID)	
<i>holiday_type</i> (PK: holiday_type_ID AUTOINCREMENT)	- <i>doctors_holiday</i> (FK: holiday_type_ID / jeden do wielu / jeden typ urlopu może być przypisany do wielu urlopów)
<i>doctor_specialization</i> (PK: doctor_ID, specialization_ID)	- <i>doctors</i> i <i>specializations</i> (spełnia między tymi tabelami relację wiele do wielu)
<i>specializations</i> (PK: specialization_ID AUTOINCREMENT)	- <i>doctor_specialization</i> (FK: specialization_ID / jeden do wielu / jedna specjalizacja może być przypisana do wielu lekarzy)

1.6 Widoki:

Łącznie aktualnie w bazie znajduje się 8 widoków.

Nazwa widoku	Opis widoku
patients_print	Wypisanie informacji o pacjencie
doctors_print_spec	Wypisanie lekarzy z ich specjalizacjami
doctors_print	Wypisanie lekarzy bez ich specjalizacji
specialization_print	Wypisanie specjalizacji potrzebnych do <i>selecta</i> w aplikacji
appointments_print	Wypisanie wizyt z podstawowymi informacjami pacjenta oraz lekarza
leaves_print	Wypisanie urlopów lekarzy
holiday_type_print	Wypisanie typów urlopu potrzebnych do <i>selecta</i> w aplikacji
presence_print	Wypisanie obecności lekarza

1) patients_print:

```
CREATE VIEW patients_print(patient_ID,patient_name,patient_surname,patient_date_of_birth,patient_phone)
AS
SELECT
p.patient_ID,p.patient_name,p.patient_surname,patient_date_of_birth,p.patient_phone
FROM patients as p
```

2) doctors_print_spec:

```
CREATE VIEW
doctors_print_spec(doctor_ID,doctor_name,doctor_surname,doctor_date_of_birth,doctor_phone,doctor_date_o
f_employment,doctor_status,doctor_specialization_ID,doctor_specialization_name) AS
SELECT
d.doctor_ID,d.doctor_name,d.doctor_surname,d.doctor_date_of_birth,d.doctor_phone,d.doctor_date_of_emplo
yment,d.doctor_status,s.specialization_ID,s.specialization_name
FROM doctors as d JOIN doctor_specialization AS ds ON ds.doctor_ID=d.doctor_ID JOIN specializations AS
s ON s.specialization_ID=ds.specialization_ID
```

3) doctors_print:

```
CREATE VIEW
doctors_print(doctor_ID,doctor_name,doctor_surname,doctor_date_of_birth,doctor_phone,doctor_date_of_emp
loyment,doctor_status) AS
SELECT
d.doctor_ID,d.doctor_name,d.doctor_surname,d.doctor_date_of_birth,d.doctor_phone,d.doctor_date_of_emplo
yment,d.doctor_status
FROM doctors as d
```

4) specialization_print:

```
create VIEW specialization_print(specialization_id,specialization_name) AS
SELECT
s.specialization_ID,
s.specialization_name
FROM specializations as s
```

5) appointments_print:

```
CREATE VIEW
appointments_print(appointment_ID,appointment_date,appointment_description,patient_ID,patient_name,patient_surname,doctor_ID,doctor_name,doctor_surname) AS
SELECT
a.appointment_ID,a.appointment_date,a.appointment_description,a.patient_ID,p.patient_name,p.patient_surname,a.doctor_ID,d.doctor_name,d.doctor_surname
FROM appointments AS a JOIN patients AS p ON a.patient_ID=p.patient_ID JOIN doctors AS d ON
a.doctor_ID=d.doctor_ID
```

6) leaves_print:

```
CREATE VIEW
leaves_print(holiday_ID,holiday_type_ID,holiday_type_name,holiday_start_date,holiday_end_date,doctor_ID,doctor_name,doctor_surname) AS
SELECT
dh.holiday_ID,dh.holiday_type_ID,ht.holiday_type_name,dh.holiday_start_date,dh.holiday_end_date,dh.doctor_ID,d.doctor_name,d.doctor_surname FROM doctors_holiday AS dh JOIN holiday_type AS ht ON
ht.holiday_type_ID=dh.holiday_type_ID JOIN doctors AS d ON d.doctor_ID=dh.doctor_ID
```

7) holiday_type_print:

```
create VIEW holiday_type_print(holiday_type_ID,holiday_type_name) AS
SELECT ht.holiday_type_ID, ht.holiday_type_name
FROM holiday_type AS ht
```

8) presence_print:

```
CREATE VIEW presence_print(presence_ID,presence_date,doctor_ID,doctor_name,doctor_surname) AS
SELECT dp.presence_ID,dp.presence_date,dp.doctor_ID,d.doctor_name,d.doctor_surname FROM
doctors_presence AS dp JOIN doctors AS d ON d.doctor_ID=dp.doctor_ID
```

1.7 Procedury i funkcje:

Łącznie aktualnie w bazie danych znajduje się 17 procedur.

Nazwa procedury	Opis procedury
patient_show_details	Wyświetlanie szczegółów pacjenta
patient_edit	Edycja pacjenta
patient_delete	Usuwanie pacjenta
patient_add	Dodawanie pacjenta
doctor_show_details	Wyświetlanie szczegółów lekarza
doctor_show_specializations	Wyświetlanie specjalizacji lekarza
leave_edit	Edycja urlopu lekarza
leave_add	Dodawanie urlopu lekarza
leave_delete	Usuwanie urlopu lekarza
presence_add	Dodawanie obecności lekarza
presence_delete	Usuwanie obecności lekarza
doctor_edit	Edycja lekarza
doctor_specialization_add	Dodawanie specjalizacji doktorowi
doctor_specialization_delete	Usuwanie specjalizacji doktorowi
doctor_add_with_specialization	Dodawanie doktora ze specjalizacją
doctor_add_without_specialization	Dodawanie doktora bez specjalizacji
doctor_delete	Usuwanie doktora

1) patient_show_details:

```
CREATE PROCEDURE patient_show_details
    @id int
AS
BEGIN
    SELECT p.patient_name AS patient_name, p.patient_surname AS patient_surname, p.patient_date_of_birth
    AS patient_date_of_birth, p.patient_phone AS patient_phone, p.patient_address AS patient_address,
    p.patient_city AS patient_city FROM patients AS p WHERE p.patient=@id
END
```

2) patient_edit:

```
create PROCEDURE patient_edit
    @patient_id int,
    @patient_name varchar(50),
    @patient_surname varchar(50),
    @patient_date_of_birth date,
    @patient_phone varchar(30),
    @patient_city varchar(50),
    @patient_address varchar(100)
AS
BEGIN
    UPDATE patients SET patient_name=@patient_name, patient_surname=@patient_surname,
    patient_date_of_birth=@patient_date_of_birth, patient_phone=@patient_phone, patient_city=@patient_city,
    patient_address=@patient_address WHERE patient_ID=@patient_id
END
```

3) patient_delete:

```
create PROCEDURE patient_delete
    @patient_id int
AS
BEGIN
    DELETE FROM patients WHERE patient_ID=@patient_id
END
GO
```

4) patient_add:

```
create PROCEDURE patient_add
    @patient_name varchar(50),
    @patient_surname varchar(50),
    @patient_date_of_birth date,
    @patient_phone varchar(30),
    @patient_city varchar(50),
    @patient_address varchar(100)
AS
BEGIN
    INSERT INTO patients
(patient_name,patient_surname,patient_date_of_birth,patient_phone,patient_city,patient_address) VALUES
(@patient_name,@patient_surname,@patient_date_of_birth,@patient_phone,@patient_city,@patient_address)
END
```

5) doctor_show_details:

```
CREATE PROCEDURE doctor_show_details
    @id int
AS
BEGIN
    SELECT d.doctor_name AS doctor_name, d.doctor_surname AS doctor_surname,d.doctor_date_of_birth AS
doctor_date_of_birth,d.doctor_phone AS doctor_phone, d.doctor_address AS doctor_address, d.doctor_city
AS doctor_city, d.doctor_status AS doctor_status, d.doctor_date_of_employment AS
doctor_date_of_employment, d.salary AS doctor_salary FROM doctors AS d WHERE d.doctor_ID=@id
END
```

6) doctor_show_specializations:

```
CREATE PROCEDURE doctor_show_specializations
    @id int
AS
BEGIN
    SELECT d.doctor_ID, s.specialization_ID AS doctor_specialization_ID, s.specialization_name AS
doctor_specialization_name FROM doctors AS d
    JOIN doctor_specialization AS ds ON d.doctor_ID=ds.doctor_ID JOIN specializations AS s ON
s.specialization_ID=ds.specialization_ID
    WHERE d.doctor_ID=@id
END
```

7) leave_edit:

```
CREATE PROCEDURE leave_edit
    @holiday_ID int,
    @doctor_ID int,
    @holiday_type_ID int,
    @holiday_start_date date,
    @holiday_end_date date

AS
BEGIN
    UPDATE doctors_holiday SET
    doctor_ID=@doctor_ID,holiday_type_ID=@holiday_type_ID,holiday_start_date=@holiday_start_date,holiday_end_date=@holiday_end_date WHERE holiday_ID=@holiday_ID
END
```

8) leave_add:

```
create PROCEDURE leave_add
    @doctor_ID int,
    @holiday_type_ID int,
    @holiday_start_date date,
    @holiday_end_date date

AS
BEGIN
    INSERT INTO doctors_holiday(doctor_ID,holiday_type_ID,holiday_start_date,holiday_end_date) VALUES
    (@doctor_ID,@holiday_type_ID,@holiday_start_date,@holiday_end_date)
END
```

9) leave_delete:

```
create PROCEDURE leave_delete
    @leave_id int

AS
BEGIN
    DELETE FROM doctors_holiday WHERE holiday_ID = @leave_id
END
```

10) presence_add:

```
create PROCEDURE presence_add
    @presence_date date,
    @doctor_ID int

AS
BEGIN
    INSERT INTO doctors_presence (presence_date,doctor_ID) VALUES (@presence_date,@doctor_ID)
END
```


11) presence_delete:

```
create PROCEDURE presence_delete
    @presence_id int
AS
BEGIN
    DELETE FROM doctors_presence WHERE presence_ID = @presence_id
END
```

12) doctor_edit:

```
create PROCEDURE doctor_edit
    @doctor_id int,
    @doctor_name varchar(50),
    @doctor_surname varchar(50),
    @doctor_address varchar(100),
    @doctor_city varchar(50),
    @doctor_date_of_birth date,
    @doctor_date_of_employment date,
    @salary money,
    @doctor_phone varchar(30),
    @doctor_status int
AS
BEGIN
    UPDATE doctors
    SET doctor_name = @doctor_name,
        doctor_surname = @doctor_surname,
        doctor_address = @doctor_address,
        doctor_city = @doctor_city,
        doctor_date_of_birth = @doctor_date_of_birth,
        doctor_date_of_employment = @doctor_date_of_employment,
        salary = @salary,
        doctor_phone = @doctor_phone,
        doctor_status = @doctor_status
    WHERE doctor_id = @doctor_id
END
```

13) doctor_specialization_add:

```
create PROCEDURE doctor_specialization_add
    @doctor_id int,
    @specialization_id int
AS
BEGIN
    INSERT INTO doctor_specialization (doctor_ID,specialization_ID) VALUES
    (@doctor_id,@specialization_id)
END
```

14) doctor_specialization_delete:

```
create PROCEDURE doctor_specialization_delete
    @doctor_id int,
    @specialization_id int
AS
BEGIN
    DELETE FROM doctor_specialization WHERE doctor_ID=@doctor_id and
specialization_ID=@specialization_id
END
```

15) doctor_add_with_specialization:

```
CREATE PROCEDURE doctor_add_with_specialization
    @doctor_name varchar(50),
    @doctor_surname varchar(50),
    @doctor_address varchar(100),
    @doctor_city varchar(50),
    @doctor_date_of_birth date,
    @doctor_date_of_employment date,
    @salary money,
    @doctor_phone varchar(30),
    @doctor_status int,
    @specialization_id int
AS
BEGIN
    INSERT INTO doctors
        (doctor_name,doctor_surname,doctor_address,doctor_city,doctor_date_of_birth,doctor_date_of_employment,s
        alary,doctor_phone,doctor_status)
        VALUES
        (@doctor_name,@doctor_surname,@doctor_address,@doctor_city,@doctor_date_of_birth,@doctor_date_of_employ
        ment,@salary,@doctor_phone,@doctor_status)

    DECLARE @doctor_id INT
    SET @doctor_id = SCOPE_IDENTITY()

    INSERT INTO doctor_specialization
        (doctor_ID,specialization_ID)
        VALUES (@doctor_id, @specialization_id)
END
```

16) doctor_add_without_specialization:

```
CREATE PROCEDURE doctor_add_without_specialization
    @doctor_name varchar(50),
    @doctor_surname varchar(50),
    @doctor_address varchar(100),
    @doctor_city varchar(50),
    @doctor_date_of_birth date,
    @doctor_date_of_employment date,
    @salary money,
    @doctor_phone varchar(30),
    @doctor_status int
AS
BEGIN
    INSERT INTO doctors
        (doctor_name,doctor_surname,doctor_address,doctor_city,doctor_date_of_birth,doctor_date_of_employment,s
        alary,doctor_phone,doctor_status)
        VALUES
        (@doctor_name,@doctor_surname,@doctor_address,@doctor_city,@doctor_date_of_birth,@doctor_date_of_employ
        ment,@salary,@doctor_phone,@doctor_status)
END
```

17) doctor_delete:

```
create PROCEDURE doctor_delete
    @doctor_id int
AS
BEGIN
    DELETE FROM doctors WHERE doctor_ID=@doctor_id
END
```

2. Aplikacja – Clinic (Przychodnia)

2.1. Ogólne informacje o aplikacji:

Aplikacja została stworzona w celu umożliwienia łatwego i intuicyjnego zarządzania przychodnią, w tym danymi dotyczącymi lekarzy, pacjentów, wizyt, recept oraz leków.

Dzięki dostępnemu w aplikacji pojedynczemu widokowi admina, osoba upoważniona jest w stanie w pełni zarządzać danymi (dodawanie, edycja, usuwanie).

Aplikacja nie posiada żadnych zabezpieczeń ani mechanizmów autoryzacji, co oznacza, że każda osoba, która ma dostęp do jej adresu internetowego, ma również dostęp do wszystkich funkcjonalności i danych. Z należy zachować ostrożność w udostępnianiu dostępu do aplikacji.

Uwaga! Obecnie aplikacja obejmuje funkcjonalności związane tylko z zarządzaniem lekarzami oraz pacjentami, więc zarządzanie wizytami, receptami i lekami zostało pominięte na tym etapie.

Schemat bazy danych, który obecnie obejmuje aplikacja:



2.2. Technologie, z których aplikacja korzysta:

Aplikacja została stworzona przy użyciu języka PHP (v. 8.0.13) i rozszerzenia sqlsrv (v. 5.11.0), które umożliwiają połączenie z serwerem baz danych MSSQL. Do uzupełnienia funkcjonalności, użyto elementów napisanych w języku JavaScript z użyciem biblioteki jQuery. Wygląd strony internetowej został opisany za pomocą plików języka arkuszy stylów CSS.

Inne rozwiązania: Ikony pochodzą z Google Fonts / Icons.

2.3. Środowisko działania aplikacji:

Aplikacja została zaprojektowana z myślą o urządzeniach stacjonarnych, a minimalna szerokość ekranu wymagana do jej poprawnego wyświetlania wynosi 1600 pikseli. Z tego powodu, jej funkcjonalność może być ograniczona na urządzeniach mobilnych, tabletach lub innych urządzeniach o mniejszej rozdzielczości ekranu.

2.4. Połączenie z bazą danych:

Za połączenie z bazą danych odpowiada zawartość pliku *connection.php*:

```
<?php
$serverName = "localhost";
$connectionOptions = array(
    "Database" => "clinic",
    "Uid" => "",
    "PWD" => "",
    "CharacterSet" => "UTF-8"
);
$conn = sqlsrv_connect($serverName, $connectionOptions);
?>
```

Domyślnie aplikacja łączy się z *localhost*, wykorzystując Windows Authentication (brak nazwy użytkownika oraz brak hasła), a następnie z bazą danych *clinic*.

Opcja *"CharacterSet" => "UTF-8"* umożliwia na wprowadzanie np. polskich znaków do bazy

2.5. Działanie aplikacji / Podstrony:

W aplikacji dostępny jest tylko jeden widok admina, a osoba upoważniona jest w stanie w pełni zarządzać danymi (dodawanie, edycja, usuwanie).

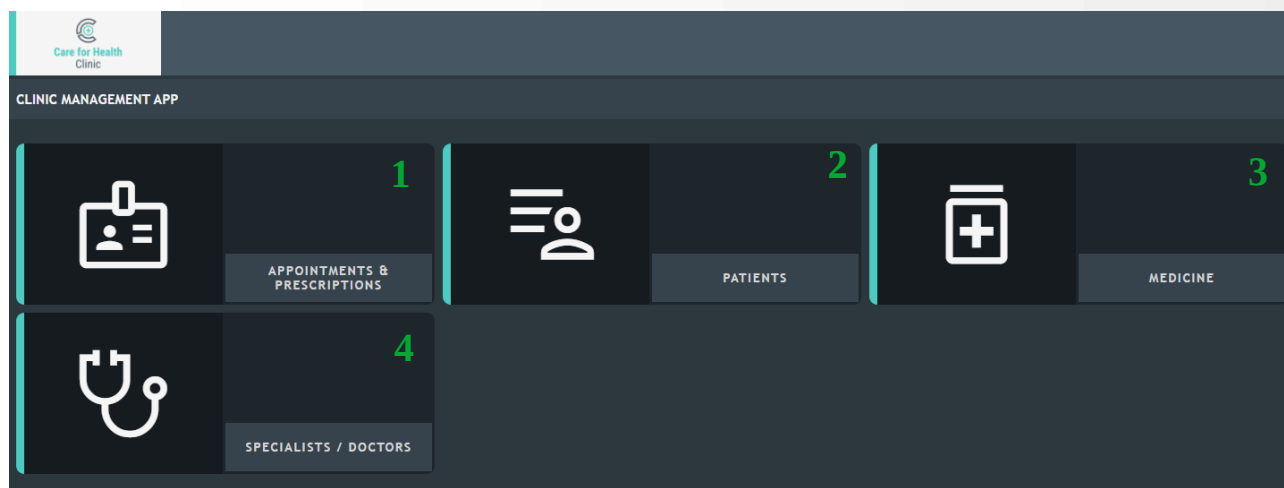
Aplikacja na ten moment składa się ze strony głównej *main.html*, na której wyświetlane będą w modalu (okienku) inne podstrony (w sumie liczba ich wynosi 18).

1) main.html:

Jest to strona główna. Stanowi „menu”, w którym do wyboru mamy 4 opcje zarządzania:

- zarządzanie wizytami i receptami (**niedostępne na tym etapie**)
- zarządzanie pacjentami
- zarządzanie lekami (**niedostępne na tym etapie**)
- zarządzanie lekarzami

a) Wygląd podstrony:



b) Przyciski (Akcje/Eventy):

-Po kliknięciu **2** zostaniemy przekierowani do *patients/patients_browse.php*

-Po kliknięciu **4** zostaniemy przekierowani do *doctors/doctors_browse.php*

Wszystkie podstrony uruchamiane są w modalu (okienku).

c) Dostęp do innych plików:

- *main.css*

- *main.js*, dzięki niemu modal jest prawidłowo wyświetlany i zamykany

2) Lekarze:

I. doctors_browse.php:

Tutaj wyświetlana jest lista lekarzy z bazy danych, w której zawarte są informacje o: ID lekarza, jego imieniu, nazwisku, dacie urodzenia, telefonie kontaktowym, dacie zatrudnienia i stanie zatrudnienia (czy jest obecnie zatrudniony) oraz o liczbie znalezionych rekordów. Jest także dostępna możliwość filtrowania wyświetlanych rekordów, dokładniej opisana w jednym z poniższych punktów.

a) Wygląd podstrony:

The screenshot shows the 'DOCTORS LIST' interface. At the top, there's a header 'DOCTORS LIST (click HERE to refresh)' with a refresh icon (6). Below it are search filters: 'Find Doctor By Name' (1), 'Find Doctor By Surname' (2), 'Order: From The Latests (F v)' (3), 'Doctor Status: Working' (4), and 'Doctor Specialization:' (5). A search button (5) and an 'ADD NEW DOCTOR' button (4) are also present. The main table lists doctors with columns: ID, NAME, SURNAME, DATE OF BIRTH, PHONE, DATE OF EMPLOYMENT, WORKS, and actions (1, 2, 3). The table contains 7 rows of data. The total count 'DOCTORS(628)' is shown on the right.

ID	NAME	SURNAME	DATE OF BIRTH	PHONE	DATE OF EMPLOYMENT	WORKS	1	2	3
1203	Janusz	Lekarzyna	04.04.1978	+48 890 279 123	22.04.2023	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1200	Yoshia	Hood	04.12.1974	(819) 216-8464	21.04.2023	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1199	Nita	Ray	26.12.1982	(328) 372-6950	01.12.2011	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1198	Kylee	Burks	01.10.2017	(844) 783-7250	21.04.2023	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1197	Connor	Howard	04.01.1997	1-545-417-3239	11.08.2019	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1196	Nyssa	Newman	18.06.2017	1-745-936-1612	29.07.2017	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE
1195	Rae	Fitzpatrick	21.02.1975	1-250-856-9415	07.04.2018	☑	SHOW DETAILS	ADD LEAVE	ADD PRESENCE

b) Przyciski (Akcje/Eventy):

-Po kliknięciu **1** zostaniemy przekierowani do *doctors_details.php?id=id_lekarza*, gdzie zobaczymy szczegółowe informacje na temat lekarza

-Po kliknięciu **2** zostaniemy przekierowani do *doctors_add_leave.php?id=id_lekarza*, gdzie będziemy mogli dodać wpis o urlopie dla lekarza

-Po kliknięciu **3** zostaniemy przekierowani do *doctors_add_presence.php?id=id_lekarza*, gdzie będziemy mogli dodać wpis o obecności dla lekarza

-Po kliknięciu **4** zostaniemy przekierowani do *doctors_add.php*, gdzie będziemy mogli dodać nowego lekarza

-Po kliknięciu **5** wcześniej wybrane filtry zostaną zastosowane

-Po kliknięciu **6** strona zostanie odświeżona, a wraz z nią wszystkie filtry

c) Filtry:

- **1** – filtr po imieniu lekarza (wyszukuje wpisanej frazy na początku imienia lekarza)

- **2** – filtr po nazwisku lekarza (wyszukuje wpisanej frazy na początku nazwiska lekarza)

- **3** – filtr, który uporządkowuje rekordy na trzy sposoby: na podstawie ID lekarza (od najmniejszego do największego i vice versa) oraz po nazwisku w kolejności alfabetycznej
- **4** – filtr po statusie lekarza, na 3 sposoby: wyświetla lekarzy którzy obecnie są zatrudnieni, którzy nie są zatrudnieni oraz wszystkich lekarzy
- **5** – filtr wyświetlający lekarzy o danej specjalizacji

d) Dostęp do innych plików:

- `css/doctors_browse.css`

II. doctors_details.php:

Tutaj wyświetlane są wszystkie informacje na temat konkretnego lekarza, o konkretnym ID przekazanym jako parametr URL metodą GET.

a) Wygląd podstrony:

DOCTOR'S DETAILS - Graiden Mendez

1 ADD LEAVE 2 ADD PRESENCE 3 EDIT DOCTOR 7

ID: 1191
 NAME: Graiden
 SURNAME: Mendez
 DATE OF BIRTH: 16.03.1989
 PHONE: (375) 940-8097
 ADDRESS: Enns, Ap #122-7356 Adipiscing Ave
 SPECIALIZATIONS: Dermatology, Infectious disease,
 DOCTOR STATUS: WORKING
 DATE OF EMPLOYMENT: 14.12.2013
 SALARY: 6,123.00\$

4 DOCTOR'S APPOINTMENTS 5 DOCTOR'S LEAVES 6 DOCTOR'S PRESENCE

b) Przyciski (Akcje/Eventy):

- Po kliknięciu **1** zostaniemy przekierowani do `doctors_add_leave.php?id=id_lekarza`, gdzie będziemy mogli dodać wpis o urlopie dla lekarza
- Po kliknięciu **2** zostaniemy przekierowani do `doctors_add_presence.php?id=id_lekarza`, gdzie będziemy mogli dodać wpis o obecności dla lekarza
- Po kliknięciu **3** zostaniemy przekierowani do `doctors_edit.php?id=id_lekarza`, gdzie będziemy mogli edytować lekarza
- Po kliknięciu **4** zostaniemy przekierowani do `doctors_details_appointments.php?id=id_lekarza`, gdzie będziemy mogli przeglądać wizyty lekarza
- Po kliknięciu **5** zostaniemy przekierowani do `doctors_details_leaves.php?id=id_lekarza`, gdzie będziemy mogli przeglądać historię urlopów lekarza
- Po kliknięciu **6** zostaniemy przekierowani do `doctors_details_presence.php?id=id_lekarza`, gdzie będziemy mogli przeglądać historię obecności lekarza
- Po kliknięciu **7** zostaniemy przekierowani do `doctors_browse.php`, gdzie zobaczymy listę lekarzy, a jeśli będzie ona naszą ostatnią stroną w historii, to po powrocie zapiszą się także ustawione wcześniej filtry, dzięki skryptowi zawartemu w `js/doctors_details.js`.

c) Dostęp do innych plików:

- *css/doctors_details.css*

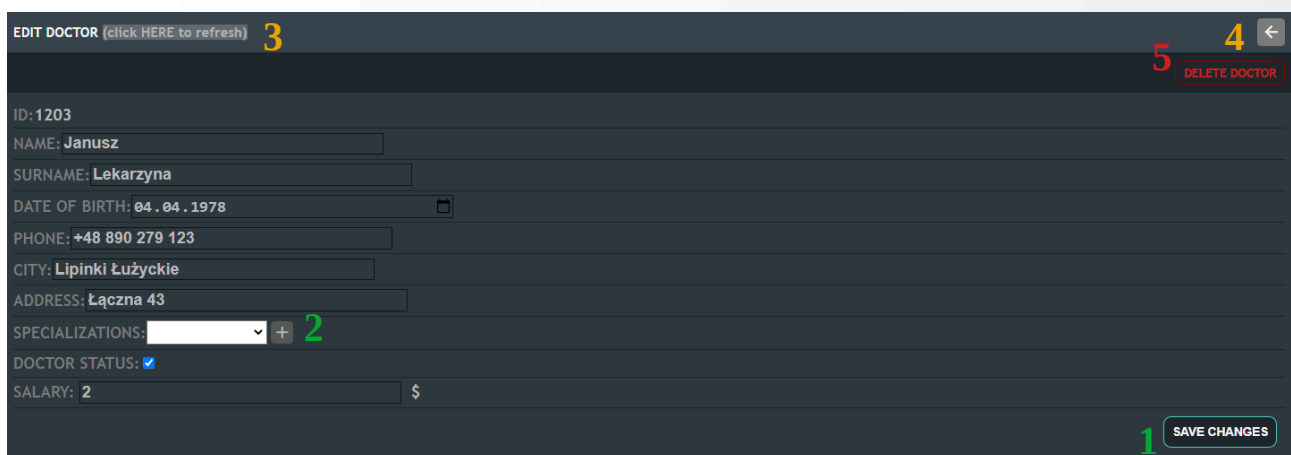
- *js/doctors_details.js*, który umożliwia powrót do adresu strony *doctors_browse.php* z zapisanymi filtrami

III. doctors_edit.php:

Tutaj będziemy mogli edytować konkretnego lekarza o ID przekazanym jako parametr URL metodą GET oraz dodawać i usuwać jego specjalizacje. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przesłaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *imię, nazwisko, data urodzenia, telefon kontaktowy, miasto, adres, pensja* nie będą puste
- Wartości pola: *imię, nazwisko, miasto* mają max. 50 znaków
- Wartość pola *telefon* ma max. 30 znaków
- Wartość pola *adres* ma max. 100 znaków
- Wartość pola *data urodzenia* składa się ze wszystkich składników (rok, miesiąc, dzień) i jest równa lub dawniejsza od dzisiejszej daty
- Wartość pola *pensja* będzie większa od 1

a) Wygląd podstrony:



b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 dane z formularza zostaną wysłane metodą POST i poddane walidacji.

Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *doctors_details.php?id=id_lekarza*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu 2 pokaże się okienko, z którego będziemy mogli wybrać specjalizację, jaką chcemy dodać lekarzowi. Po ponownym kliknięciu okienko się zamknie. Skrypt obsługujący to zdarzenie znajduje się w pliku *js/doctor_edit.js*

-Po kliknięciu 3 strona zostanie odświeżona, a zawartość pól formularza również (zostaną uzupełnione aktualnymi danymi o lekarzu)

-Po kliknięciu 4 zostaniemy przekierowani do *doctors_details.php?id=id_lekarza*, gdzie zobaczymy szczegółowe informacje na temat lekarza

-Po kliknięciu 5 zostaniemy przekierowani do *doctors_delete.php?id=id_lekarza*, gdzie zostanie wykonane polecenie usuwające lekarza z bazy danych. Przycisk pojawia się tylko gdy nie istnieją żadne powiązane z tym lekarzem rekordy w innych tabelach (specjalizacje, wizyty, obecności, urlopy).

c) Dostęp do innych plików:

- *css/doctor_edit.css*

- *js/doctor_edit.js*

IV. doctors_add.php:

Tutaj będziemy mogli dodawać nowego lekarza. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przesłaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *imię, nazwisko, data urodzenia, telefon kontaktowy, miasto, adres* nie będą puste

- Wartości pola: *imię, nazwisko, miasto* mają max. 50 znaków

- Wartość pola *telefon* ma max. 30 znaków

- Wartość pola *adres* ma max. 100 znaków

- Wartość pola *data urodzenia* składa się ze wszystkich składników (rok, miesiąc, dzień) i jest równa lub dawniejsza od dzisiejszej daty

-Wartość pola *pensja* będzie większa od 1

a) Wygląd podstrony:

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 dane z formularza zostaną wysłane metodą POST i poddane walidacji. Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *doctors_browse.php*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu 2 pokaże się okienko, z którego będziemy mogli wybrać specjalizację, jaką chcemy dodać lekarzowi. Po ponownym kliknięciu okienko się zamknie. Skrypt obsługujący to zdarzenie znajduje się w pliku *js/doctor_add.js*

-Po kliknięciu 3 strona zostanie odświeżona, a zawartość pól formularza również

-Po kliknięciu 4 zostaniemy przekierowani do *doctors_browse.php*, gdzie zobaczymy listę lekarzy

c) Dostęp do innych plików:

- *css/doctor_add.css*

- *js/doctor_add.js*

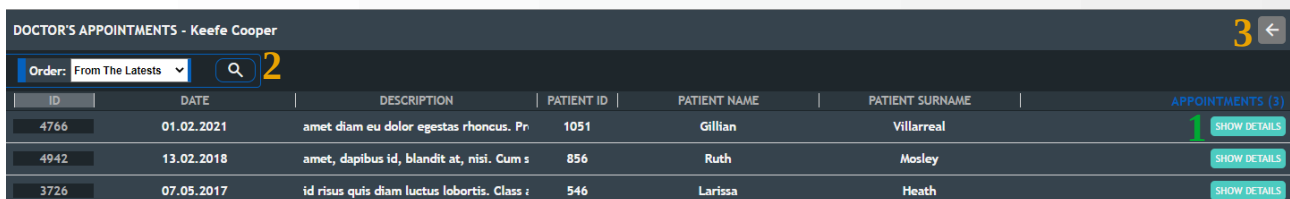
V. doctors_delete.php:

Zadaniem tej podstrony będzie usunięcie konkretnego lekarza o ID przekazanym jako parametr URL metodą GET. Ta strona nie ma wyglądu. Bezpośrednio po wykonaniu zapytania do bazy danych zostaniemy przekierowani do *doctors_browse.php*.

VI. doctors_details_appointments.php:

Na tej podstronie będziemy mogli przeglądać historię wizyt konkretnego lekarza o ID przekazanym jako parametr URL metodą GET. Ukazane dane będziemy mogli też posortować od najnowszych, bądź najstarszych wizyt (po dacie wizyty), a całkowita ich liczba rekordów zostanie wyświetlona.

a) Wygląd podstrony:



ID	DATE	DESCRIPTION	PATIENT ID	PATIENT NAME	PATIENT SURNAME	APPOINTMENTS (3)
4766	01.02.2021	amet diam eu dolor egestas rhoncus. Pr	1051	Gillian	Villarreal	SHOW DETAILS
4942	13.02.2018	amet, dapibus id, blandit at, nisi. Cum s	856	Ruth	Mosley	SHOW DETAILS
3726	07.05.2017	id risus quis diam luctus lobortis. Class i	546	Larissa	Heath	SHOW DETAILS

b) Przyciski (Akcje/Eventy):

-Po kliknięciu **1** BRAK FUNKCJONALNOŚCI

-Po kliknięciu **2** zastosowane zostanie sortowanie

-Po kliknięciu **3** zostaniemy przekierowani do *doctors_details.php*, gdzie zobaczymy szczegółowe informacje na temat lekarza

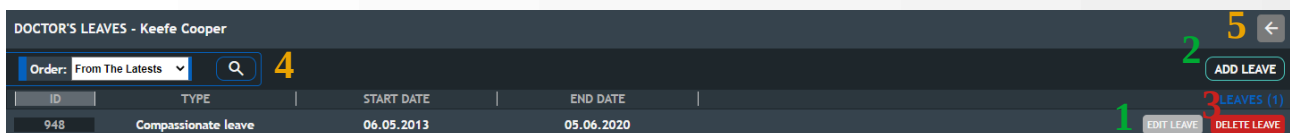
c) Dostęp do innych plików:

- *css/doctor_details_appointments.css*

VII. doctors_details_leaves.php:

Na tej podstronie będziemy mogli przeglądać historię urlopów konkretnego lekarza o ID przekazanym jako parametr URL metodą GET. Ukazane dane będziemy mogli też posortować od najnowszych, bądź najstarszych wizyt (po dacie wizyty), a całkowita ich liczba rekordów zostanie wyświetlona.

a) Wygląd podstrony:



ID	TYPE	START DATE	END DATE
948	Compassionate leave	06.05.2013	05.06.2020

b) Przyciski (Akcje/Eventy):

-Po kliknięciu **1** zostaniemy przekierowani do *doctors_leaves_edit.php?id=id_urlopu*, gdzie będziemy mogli edytować wpis urlopu lekarza

-Po kliknięciu **2** zostaniemy przekierowani do *doctors_leaves_add.php*, gdzie będziemy mogli dodać nowy urlop dla lekarza

-Po kliknięciu **3** zostaniemy przekierowani do *doctors_leaves_delete.php?id=id_urlopu*, co spowoduje usunięcie wpisu urlopu lekarza

-Po kliknięciu **4** zastosowane zostanie sortowanie

-Po kliknięciu 5 zostaniemy przekierowani do *doctors_details.php*, gdzie zobaczymy szczegółowe informacje na temat lekarza

c) Dostęp do innych plików:

- *css/doctor_details_leaves.css*

VIII. doctors_leaves_edit.php:

Tutaj będziemy mogli edytować konkretny wpis urlopu lekarza o ID przekazanym jako parametr URL metodą GET. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przestaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *ID lekarza, typ urlopu, data rozpoczęcia i zakończenia urlopu* nie będą puste

- Wartości pól: *data zakończenia urlopu* będzie dalsza od *daty rozpoczęcia urlopu*

a) Wygląd podstrony:

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 dane z formularza zostaną wysłane metodą POST i poddane walidacji. Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *doctors_details_leaves.php?id=id_lekarza*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu 2 zostaniemy przekierowani do *doctors_leaves_delete.php?id=id_urlopu*, co spowoduje usunięcie wpisu urlopu lekarza

-Po kliknięciu 3 strona zostanie odświeżona, a zawartość pól formularza również (zostaną uzupełnione aktualnymi danymi o urlopie)

-Po kliknięciu 4 zostaniemy przekierowani do *doctors_details_leaves.php?id=id_lekarza*, gdzie będziemy mogli przeglądać historię urlopów lekarza

c) Dostęp do innych plików:

- *css/doctor_leaves_edit.css*

IX. doctors_leaves_add.php:

Na tej podstronie będziemy mogli dodać nowy wpis o urlopie lekarza. Pole ID lekarza zostanie uzupełnione jeśli w GET będzie ustawiona zmienna ID. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przestaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *ID lekarza, typ urlopu, data rozpoczęcia i zakończenia urlopu* nie będą puste

- Wartości pól: *data zakończenia urlopu* będzie dalsza od *daty rozpoczęcia urlopu*

a) Wygląd podstrony:

ADD NEW LEAVE (click HERE to refresh) 2

DOCTOR ID: 1168

HOLIDAY TYPE: ▾

START DATE: dd.mm.yyyy

END DATE: dd.mm.yyyy

1 SAVE CHANGES

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 dane z formularza zostaną wysłane metodą POST i poddane walidacji. Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *doctors_details_leaves.php?id=id_lekarza* , jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu 2 strona zostanie odświeżona, a zawartość pól formularza również

-Po kliknięciu 3 zostaniemy przekierowani do *doctors_details_leaves.php?id=id_lekarza*, gdzie będziemy mogli przeglądać historię urlopów lekarza

c) Dostęp do innych plików:

- *css/doctor_leaves_add.css*

X. doctors_leaves_delete.php:

Zadaniem tej podstrony będzie usunięcie konkretnego wpisu urlopu lekarza o ID urlopu przekazanym jako parametr URL metodą GET. Ta strona nie ma wyglądu. Bezpośrednio po wykonaniu zapytania do bazy danych zostaniemy przekierowani do *doctors_details_leaves.php?id=id_lekarza*, które również będzie w GET.

XI. doctors_details_presence.php:

Na tej podstronie będziemy mogli przeglądać historię obecności konkretnego lekarza o ID przekazanym jako parametr URL metodą GET. Ukazane dane będziemy mogli też posortować od najnowszych, bądź najstarszych obecności (po dacie obecności), a całkowita ich liczba rekordów zostanie wyświetlona.

a) Wygląd podstrony:

DOCTOR'S PRESENCE - Keefe Cooper

Order: From The Latests

1 ADD PRESENCE

2 DELETE PRESENCE

PRESENCES (8)

ID	DATE
3939	31.03.2021
4939	31.03.2021
6559	16.07.2016

DELETE PRESENCE

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 zostaniemy przekierowani do *doctors_presence_add.php*, gdzie będziemy mogli dodać nowy wpis obecności dla lekarza

-Po kliknięciu 2 zostaniemy przekierowani do *doctors_presence_delete.php?id=id_obecnosci*, gdzie wpis obecności zostanie usunięty

-Po kliknięciu 3 strona zostanie odświeżona, a zawartość pól formularza również

-Po kliknięciu **4** zostaniemy przekierowani do *doctors_details.php*, gdzie zobaczymy szczegółowe informacje na temat lekarza

c) Dostęp do innych plików:

- *css/doctor_details_presence.css*

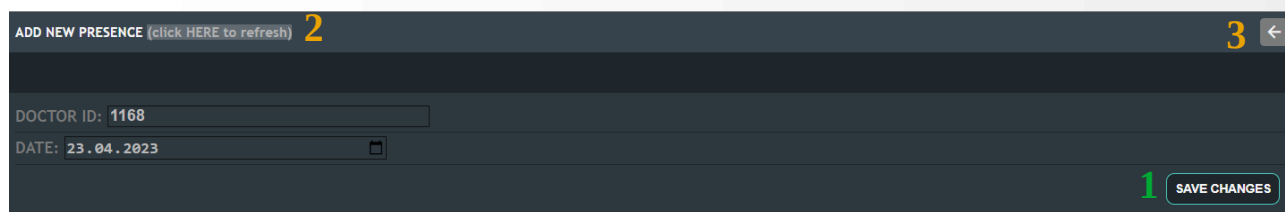
XII. doctors_presence_add.php:

Na tej podstronie będziemy mogli dodać nowy wpis o obecności lekarza. Pole ID lekarza zostanie uzupełnione jeśli w GET będzie ustawiona zmienna ID. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przestaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *ID lekarza*, *data obecności* nie będą puste

- Wartość pola *data obecności* będzie wcześniejsza bądź równa dzisiejszej

a) Wygląd podstrony:



b) Przyciski (Akcje/Eventy):

-Po kliknięciu **1** dane z formularza zostaną wysłane metodą POST i poddane walidacji. Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *doctors_details_presence.php?id=id_lekarza*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu **2** strona zostanie odświeżona, a zawartość pól formularza również

-Po kliknięciu **3** zostaniemy przekierowani do *doctors_details_presence.php?id=id_lekarza*, gdzie będziemy mogli przeglądać historię obecności lekarza

c) Dostęp do innych plików:

- *css/doctor_presence_add.css*

XIII. doctors_presence_delete.php:

Zadaniem tej podstrony będzie usunięcie konkretnego wpisu obecności lekarza o ID obecności przekazanym jako parametr URL metodą GET. Ta strona nie ma wyglądu. Bezpośrednio po wykonaniu zapytania do bazy danych zostaniemy przekierowani do *doctors_details_presence.php?id=id_lekarza*, które również będzie w GET.

3) Pacjenci:

I. patients_browse.php:

Tutaj wyświetlana jest lista pacjentów z bazy danych, w której zawarte są informacje o: ID pacjenta, jego imieniu, nazwisku, dacie urodzenia, telefonie kontaktowym oraz o liczbie znalezionych rekordów. Jest także dostępna możliwość filtrowania wyświetlanych rekordów, dokładniej opisana w jednym z poniższych punktów.

The screenshot shows a web interface for a patient list. At the top, there's a header 'PATIENTS' LIST' with a link '(click HERE to refresh)'. Below this is a search bar with three input fields: 'Find Patient By Name' (annotated with 1), 'Find Patient By Surname' (annotated with 2), and 'Order: From The Latests (F)' (annotated with 3). There is also a search button with a magnifying glass icon (annotated with 4) and an 'ADD NEW PATIENT' button (annotated with 3). The main part of the interface is a table with columns: ID, NAME, SURNAME, DATE OF BIRTH, and PHONE. The table contains 8 rows of patient data. To the right of the table, there's a link 'PATIENTS(2996)' (annotated with 2) and two buttons: 'SHOW DETAILS' (annotated with 1) and 'NEW APPOINTMENT'.

ID	NAME	SURNAME	DATE OF BIRTH	PHONE
3000	Len	Rhodes	28.02.2009	(137) 478-7213
2999	Kimberly	Melendez	05.08.1981	(637) 277-7033
2998	Connor	English	20.03.1968	(585) 266-5042
2997	Jamalia	Nieves	15.03.1999	(675) 985-4747
2996	Josephine	Porter	04.09.2015	1-890-571-0713
2995	Madeson	Wallace	03.11.2003	(327) 658-6134
2992	Quynn	Glover	25.05.1952	1-451-355-1441

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 zostaniemy przekierowani do *patients_details.php?id=id_pacjenta*, gdzie zobaczymy szczegółowe informacje na temat pacjenta

-Po kliknięciu 2 **BRAK FUNKCJONALNOŚCI**

-Po kliknięciu 3 zostaniemy przekierowani do *patients_add.php*, gdzie będziemy mogli dodać nowego lekarza

-Po kliknięciu 4 wcześniej wybrane filtry zostaną zastosowane

-Po kliknięciu 5 strona zostanie odświeżona, a wraz z nią wszystkie filtry

c) Filtry:

- 1 – filtr po imieniu pacjenta (wyszukuje wpisanej frazy na początku imienia pacjenta)

- 2 – filtr po nazwisku pacjenta (wyszukuje wpisanej frazy na początku nazwiska pacjenta)

- 3 – filtr, który uporządkowuje rekordy na trzy sposoby: na podstawie ID pacjenta (od najmniejszego do największego i vice versa) oraz po nazwisku w kolejności alfabetycznej

c) Dostęp do innych plików:

- *css/patients_browse.css*

II. patients_details.php:

Tutaj wyświetlane są wszystkie informacje na temat konkretnego pacjenta, o konkretnym ID przekazanym jako parametr URL metodą GET. Poniżej także znajdziemy historię wizyt pacjenta i ich liczbę.

a) Wygląd podstrony:

PATIENT'S DETAILS - Len Rhodes

1 NEW APPOINTMENT 2 EDIT PATIENT 4 <

ID: 3000
NAME: Len
SURNAME: Rhodes
DATE OF BIRTH: 28.02.2009
PHONE: (137) 478-7213
ADDRESS: San Fernando, P.O. Box 828, 2070 Cursus Avenue

APPOINTMENTS HISTORY

ID	DATE	DESCRIPTION	DOCTOR ID	DOCTOR NAME	DOCTOR SURNAME	APPOINTMENTS (3)
2027	14.10.2020	arcu. Vivamus sit amet risus. Donec ege	215	Stone	Stark	3 SHOW DETAILS
1953	02.07.2020	gravida. Praesent eu nulla at sem moles	14	Logan	Poole	SHOW DETAILS
1807	29.09.2012	sagittis lobortis mauris. Suspendisse alic	63	Molly	Mitchell	SHOW DETAILS

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 **BRAK FUNKCJONALNOŚCI**

-Po kliknięciu 2 zostaniemy przekierowani do *patients_edit.php?id=id_pacjenta*, gdzie będziemy mogli edytować pacjenta

-Po kliknięciu 3 **BRAK FUNKCJONALNOŚCI**

-Po kliknięciu 4 zostaniemy przekierowani do *patients_browse.php*, gdzie zobaczymy listę pacjentów, a jeśli będzie ona naszą ostatnią stroną w historii, to po powrocie zapiszą się także ustawione wcześniej filtry, dzięki skryptowi zawartemu w *js/patient_details.js*

c) Dostęp do innych plików:

- *css/patients_details.css*

- *js/patients_details.js*, który umożliwia powrót do adresu strony *patients_browse.php* z zapisanymi filtrami

III. patients_edit.php:

Tutaj będziemy mogli edytować konkretnego pacjenta o ID przekazanym jako parametr URL metodą GET. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przesłaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *imię*, *nazwisko*, *data urodzenia* nie będą puste

- Wartości pola: *imię*, *nazwisko*, *miasto* mają max. 50 znaków

- Wartość pola *telefon* ma max. 30 znaków

- Wartość pola *adres* ma max. 100 znaków

- Wartość pola *data urodzenia* składa się ze wszystkich składników (rok, miesiąc, dzień) i jest równa lub dawniejsza od dzisiejszej daty

a) Wygląd podstrony:

b) Przyciski (Akcje/Eventy):

-Po kliknięciu **1** dane z formularza zostaną wysłane metodą POST i poddane walidacji.

Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *patients_details.php?id=id_pacjenta*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu **2** strona zostanie odświeżona, a zawartość pól formularza również (zostaną uzupełnione aktualnymi danymi o pacjencie)

-Po kliknięciu **3** zostaniemy przekierowani do *patients_details.php?id=id_pacjenta*, gdzie zobaczymy szczegółowe informacje na temat pacjenta

-Po kliknięciu **4** zostaniemy przekierowani do *patients_delete.php?id=id_pacjenta*, gdzie zostanie wykonane polecenie usuwające pacjenta z bazy danych. Przycisk pojawia się tylko gdy nie istnieją żadne powiązane z tym pacjentem rekordy w innych tabelach (wizyty).

c) Dostęp do innych plików:

- *css/patients_edit.css*

IV. patients_add.php:

Tutaj będziemy mogli dodać nowego pacjenta. Polecenie zostanie wykonane do bazy danych jeżeli walidacja danych z formularza, po ich przestaniu metodą POST przebiegnie pomyślnie, według schematu:

- Pola: *imię*, *nazwisko*, *data urodzenia* nie będą puste
- Wartości pola: *imię*, *nazwisko*, *miasto* mają max. 50 znaków
- Wartość pola *telefon* ma max. 30 znaków
- Wartość pola *adres* ma max. 100 znaków
- Wartość pola *data urodzenia* składa się ze wszystkich składników (rok, miesiąc, dzień) i jest równa lub dawniejsza od dzisiejszej daty

a) Wygląd podstrony:

ADD NEW PATIENT (click HERE to refresh) 2

NAME:

SURNAME:

DATE OF BIRTH:

PHONE:

CITY:

ADDRESS:

1 ADD

b) Przyciski (Akcje/Eventy):

-Po kliknięciu 1 dane z formularza zostaną wysłane metodą POST i poddane walidacji. Jeśli przebiegnie pomyślnie zapytanie zostanie wykonane do bazy danych i zostaniemy przekierowani do *patients_browse.php*, jeśli nie- zostanie wyświetlony błąd

-Po kliknięciu 2 strona zostanie odświeżona, a zawartość pól formularza również

-Po kliknięciu 3 zostaniemy przekierowani do *patients_browse.php*, gdzie zobaczymy listę pacjentów

c) Dostęp do innych plików:

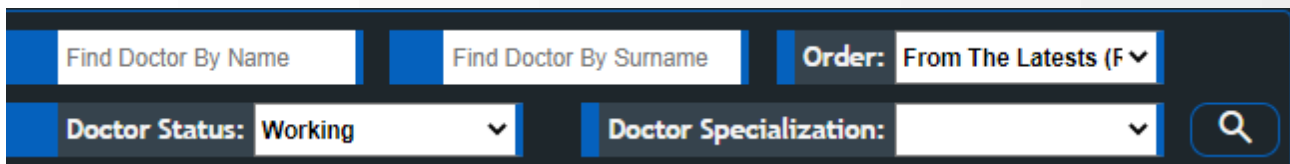
- *css/patients_add.css*

V. patients_delete.php:

Zadaniem tej podstrony będzie usunięcie konkretnego pacjenta o ID przekazanym jako parametr URL metodą GET. Ta strona nie ma wyglądu. Bezpośrednio po wykonaniu zapytania do bazy danych zostaniemy przekierowani do *patients_browse.php*.

2.6. „Smaczki”:

- Rozbudowane filtrowanie na podstronie *doctors_browse.php*:

A dark-themed search bar with two input fields: "Find Doctor By Name" and "Find Doctor By Surname". To the right is a dropdown menu labeled "Order:" with the selected option "From The Latests (F". Below these are two more dropdowns: "Doctor Status:" with "Working" selected, and "Doctor Specialization:" which is empty. A magnifying glass icon is on the far right.

- Powrót do *doctors_browse.php* i *patients_browse.php* z zapisanymi filtrami, jeśli strony te były ostatnio otwarte w historii

- System dodawania i usuwania specjalizacji w *doctors_edit.php* i *doctors_add.php*:

A dark-themed bar labeled "SPECIALIZATIONS:". It contains two checked checkboxes: "Diagnostic radiology," and "Oncology," followed by an empty dropdown menu. To the right is a plus icon and a message: "You can add only one specialization at once!".

2.7. Możliwości rozwoju:

- Rozbudowa aplikacji o zarządzanie wizytami, receptami oraz lekarstwami
- Zrobienie przy powrocie na stronę do przeglądania pacjentów lub lekarzy automatyczne przescrollowanie do rekordu, którego szczegóły wcześniej otworzyliśmy i zapisywanie filtrów w Local Storage
- Zrobienie dwóch dodatkowych podstron do ogólnego wyświetlania urlopów i obecności lekarzy w danym dniu
- Zrobienie paginacji