

Politechnika Świętokrzyska w Kielcach Wydział Elektrotechniki, Automatyki i Informatyki	
MiniProjekt Bazy Danych 1	
Temat: Przychodnia	
Autorzy: Maciej Bandura Marcin Ślusarczyk Mariusz Śmistek Filip Tosnowiec grupa: 2ID14B	Data wykonania: 24.01.2022

1. Opis Projektu

Nasza baza danych została stworzona z myślą o serwisie internetowym na zasadzie wiele klientów – jeden serwer, gdzie każdy klient posiada rozbudowany interfejs graficzny. Wiele mechanizmów zostało zaimplementowanych w taki sposób aby łatwo można było obsłużyć poszczególne operacje na bazie. Podczas projektowania bazy postawiliśmy sobie, że będziemy unikać operacji na kluczach głównych w kontekście użytkownika pracującego na bazie. Sama struktura bazy już operuje na kluczach głównych i wartościach unikatowych. W celu optymalizacji i automatyzacji poszczególnych procesów stosujemy:

-sekwencje: do generowania kluczy głównych w wybranych tabelach, dzięki temu możemy uniknąć problemów związanych z integralnością kluczy głównych oraz zautomatyzować operację **INSERT** na wielu tabelach. W naszym projekcie zastosowaliśmy 7 sekwencji.

-widoki: w związku z tym, że na naszej bazie będzie pracowało trzy rodzaje użytkowników musieliśmy zastosować wiele perspektyw, aby uniknąć redundancji danych oraz problemów z niepożądanym dostępem do wybranych danych. Plik odpowiedzialny za funkcje wywoływane po stronie serwera PHP (req.php) w pewnym momencie stał się bardzo rozbudowany i mało czytelny, w celu poprawy przejrzystości pliku postanowiliśmy wiele operacji **SELECT** opartych na kilku tabelach przerobić właśnie na widoki dzięki czemu polecenie wywoływane na bazie mogliśmy skrócić do postaci: **SELECT * FROM Nazwa_widoku**. Najważniejszymi widokami w naszym projekcie o których należy wspomnieć to: **Pacjenci_View** oraz **Lekarze_View**. Widoki te decydują o każdej instancji pacjenta i lekarza w naszej bazie. Każdy pacjent ma swój adres z tabeli **Adresy**, swój kontakt z tabeli **Kontakty**, swoje konto użytkownika w tabeli **Konta**, oraz swoją kartę pacjenta w tabeli **Pacjenci**. Jeżeli chcemy dodać jakiegoś pacjenta to wykonujemy operację **INSERT** na tym widoku (Szczegółowy opis jak wykonujemy jeden insert na kilku tabelach zostanie opisany później w sekcji wyzwalaczy.). Jeżeli chcemy wyświetlić informacje o danym pacjencie to wykonujemy operację **SELECT** na tym widoku, a jeżeli chcemy usunąć pacjenta to wykonujemy operację **DELETE** na tym widoku. W przypadku lekarza schemat wygląda analogicznie, dodatkowo lekarz jest przypisany do poszczególnej specjalizacji z tabeli **Specjalizacje**. Nasz projekt zawiera 14 widoków.

-wyzwalacze: są bardzo ważną mechaniką wykorzystywaną w naszym projekcie, to właśnie dzięki triggerom możemy wykonywać operacje DML na widokach. Tu znów można wyróżnić dwa podstawowe wyzwalacze użyte w naszym projekcie, czyli: **Pacjent_Trigger** oraz **Lekarze_Trigger**. Te wyzwalacze uruchamiane są za pomocą polecenia **INSTEAD OF** gdy wykonujemy operacje DML na poszczególnych widokach przypisanych do tych triggerów. Działają one na wspomnianych już sekwencjach. Służą one do „rozbicia” jednej dyrektywy **INSERT/UPDATE/DELETE** na mniejsze polecenia operujące na tabelach wchodzących w skład danego widoku. Dzięki temu możemy za pomocą jednej instrukcji dodać, zaktualizować lub usunąć informacje z kilku tabel. Jeżeli któraś operacja wskaże błąd to zgodnie z zasadami ACID zostanie wykonany rollback a zmienione lub wprowadzone dane zostaną cofnięte. Wyzwalacze stosujemy również do

sprawdzania poprawności wprowadzanych dat. Na przykład: W tabeli **Osoby** wprowadzana data urodzenia nie może być większa od aktualnej. W tabeli **Wizyty**: wprowadzana data dla zaplanowanej wizyty nie może być mniejsza od aktualnej. W skład naszego projektu wchodzi 6 wyzwalaczy.

-procedury: większość operacji na naszej bazie zostało opartych o procedury. Jedne są bardziej rozbudowane drugie wykonują pojedyncze polecenia. Każda procedura jest nazwana w charakterystyczny sposób, w swojej nazwie zawierają nazwę użytkownika dla którego dana operacja jest przewidziana. Parametry przekazywane do procedury posiadają przedrostek „p_” a zmienne deklarowane w procedurach posiadają przedrostek „v_”. Procedury pomagają nam również skrócić polecenia wykonywane w pliku odpowiedzialnym za wykonywane funkcje po stronie serwera PHP. Większość operacji DML jest wywoływanych właśnie z poziomu procedur. Nasz projekt zawiera 20 procedur.

-funkcje: wykorzystujemy je w przypadku gdy potrzebujemy uzyskać jakąś konkretną informację zwrotną od serwera bazodanowego. Pomagają nam „wyciągać” informacje generowane przez kursory i przekazywać jako np. ciąg znaków. W naszym projekcie zastosowaliśmy 6 funkcji.

-kursory: pomagają nam generować poszczególne statystyki na stronę główną. Problem z wykorzystaniem kursorów na większą skalę pojawił się gdy próbowaliśmy pobierać informacje do klienta z kursora. Niestety ale nasze polecenie `dbRequire()` które używamy do wywoływania operacji na bazie, nie chciało pobrać informacji z kursora. Dlatego też kursory wykorzystujemy tylko w minimalnie wymaganym stopniu.

-scheduler: dodatkowo w naszym projekcie zastosowaliśmy funkcjonalność tak zwanego planisty. Jest on odpowiedzialny za usuwanie przestarzałych sesji. Jak to działa? Użytkownik (pacjent, lekarz, admin) logując się na naszą stronę ma generowany token logowania, który go uwierzytelnia. Jest on zapisany w tabeli **Sesje** która przechowuje wszystkie aktywne sesje. Dana instancja sesji jest aktywna przez 24 godziny od najnowszego logowania. I właśnie wspomniany planista obsługuje usuwanie przestarzałych sesji. Działa on następująco: co godzinę uruchamiana jest procedura która usuwa wszystkie przestarzałe sesje w danym momencie.

Nasz projekt składa się z 14 plików .sql. Podzielone są na zasadzie operacji SQL jakie wywołują (CREATE, DROP...), perspektyw i tabel których dotyczą (Pacjenci.sql, Wizyty.sql) oraz funkcjonalności za jakie odpowiadają (logowanie.sql, rejestracja.sql). W celu automatyzacji procesu instalacji naszej bazy stworzyliśmy skrypt (ladowanie.sh) który uruchamia poszczególne pliki .sql w odpowiedniej kolejności. Dzięki czemu nie musimy się martwić kolejnością uruchamiania plików czy spójnością danych. Zaleca się aby ten skrypt uruchamiać w konsoli obsługującej polskie znaki. Aby skrypt działał poprawnie, należy go edytować i wprowadzić poprawne dane logowania do bazy.

Komunikacja klient - serwer - baza danych

Cały interfejs użytkownika wyświetlany jest jako dokument w formacie .html. Interakcje z serwerem (bezpośrednio: PHP, pośrednio: SQL) następują poprzez język skryptowy strony (dokumentu html) - javascript. W języku tym opisaliśmy sobie proste API do komunikacji z serwerem (zawarte w pliku js/api.js).

W zasadzie cała komunikacja (po stronie JS) owita jest jedną funkcją: `dbReq(callback, "zadanieDoWykonania", ["argument", "wartość ..])`. Serwer PHP wykona zadanie o nazwie "zadanieDoWykonania" podając mu argumenty z ostatniej tablicy. Po wykonaniu zadania zwróci pakiet JSON który JavaScript poda jako jedyny argument do wywoływanej funkcji "callback".

Komunikacja odbywa się w sposób następujący: Klient poprzez JS'a wysyła do serwera (pliku "req.php") prośbę o wykonanie jakiejś funkcji.

W tym celu wysyłane jest zapytanie z odpowiednimi parametrami GET. Każde poprawne zapytanie zawsze zawiera następujące parametry/argumenty: (token - uniwersalny ciąg znaków, uwierzytelniający klienta, cmd - nazwę polecenia jakie serwer ma wykonać). Jako odpowiedź klient otrzymuje od serwera PHP, dane w formacie JSON. Taki pakiet JSON zawiera obowiązkowo boolowskie pole "success" będące flagą informującą czy powiodło się wykonanie zadanego polecenia po stronie serwera czy też nie. Jeżeli zadanie się nie powiodło to dodatkowo pakiet będzie składał się z pola "err" zawierającego dokładniejszy

opis błędu, np. błąd zwrócony przez serwer SQL. Jeżeli jednak zapytanie do serwera PHP się powiedzie, pakiet najprawdopodobniej będzie zawierać pole "db" będące tablicą (często tablicą tablic) której poszczególne elementy będą zawierać dane zwrócone przez serwer SQL (o ile serwer PHP wykonał zapytanie do serwera SQL i zapisał wynik tego zapytania w pakiecie odesłanym do klienta). Dodatkowo pakiety JSON zawierają pole o nazwie "acType", zawierające nazwę zalogowanego użytkownika (lub też niezalogowanego, wtedy naszą nazwą użytkownika będzie wartość "brak").

Istotną częścią serwera jest uwierzytelnianie osób (klientów) przesyłających zapytania. Serwer musi wiedzieć z kim rozmawia, czy z zalogowanym pacjentem, czy też z administratorem a może nawet i z nie zalogowanym gościem. Przy każdym zapytaniu, należy klienta uwierzytelnić - tj. potwierdzić jego autentyczność. Wysyłanie danych logowania (loginu i hasła) za każdym razem byłoby bardzo niebezpieczne i naraziłoby te dane na przechwycenie przez osoby niepożądane. W tym celu dane logowania uwierzytelnia się tylko raz. Jeżeli ich autentyczność się potwierdzi, klientowi generuje się nowy, tymczasowy klucz - poprzez który będzie się ów klient komunikował z serwerem, nie narażając swoich wrażliwych danych logowania. Od strony technicznej, po każdym sukcesywnym zalogowaniu się jakiejś osoby, generuje się nową sesję tej osobie z kluczem sesji - tzw. "tokenem" uwierzytelniającym, właśnie tą osobę. Jako token używamy ciąg 32 znaków (małe litery i cyfry) - który jest relatywnie ciężki do odgadnięcia przez hakera. Informacje o nowo otwartej sesji (token, czas ważności sesji oraz klucz obcy identyfikujący daną osobę po stronie serwera SQL) są przechowywane w tabeli **Sesje**, aż do czasu wylogowania się lub do czasu wygaśnięcia sesji (usunięcia jej przez procedurę uruchomioną przez planistę).

Komunikacja z serwerem bazodanowym od strony serwera PHP wygląda bardzo prosto. Podobnie jak we wcześniejszym przykładzie owita jest jedną funkcją `dbRequire("query")`. Funkcja ta jako argument przyjmuje zapytanie do serwera bazodanowego w języku SQL, po wykonaniu zapytania zwraca wynik w postaci i tablicy tablic. To czy ów wynik zostanie przekazany do klienta czy też nie zależy tylko i wyłącznie od serwera PHP. Jeżeli serwer bazodanowy zwróci jakiś błąd to zostanie on zawsze przekazany do klienta w postaci ciągu znaków i flagi `success` ustawionej na `false`.

Przykładowe zapytanie do serwera PHP:

```
req.php?token=l9x32fcw079wg50qsg93t8cs80207452&cmd=ping
```

Przykładowa odpowiedź z serwera PHP:

```
{
  "success": true,
  "token": "l9x32fcw079wg50qsg93t8cs80207452",
  "acType": "brak",
  "ip": ":::1",
  "db": []
}
```

Obsługa zapytań i odpowiedzi po stronie JS:

```
dbReq((e) => {
  console.log(`Odpowiedź na zadanie "ping"`, e);
}, "ping", []);
```

2. Opis użytych technologii

W naszym projekcie posługujemy się:

- Standardem Języka HTML w wersji 5,
- Językiem JavaScript zgodnym ze standardem ES6,
- językiem php,
- Systemem Zarządzania Bazą danych - Oracle Database 21c Express Edition Release 21.0.0.0.0,
- Serwerem Apache w zintegrowanym wieloplatformowym pakiecie XAMPP,
- Connector OCI,
- Biblioteką JQuery dla JavaScript'a (sporadycznie),
- Biblioteką Bootstrap (sporadycznie).
- Kontrola źródła VCS, repozytorium online na serwerze github.com
- lokalnie przy pomocy oprogramowania GIT

3. Opis zastosowanych elementów

Nasz Projekt składa się z 14 tabel.

Są to:

1. Adresy - zawiera wszystkie informacje dotyczące adresu oraz jego identyfikator

Nr_Adresu	Unikalny identyfikator wystąpień adresu. Używany jako klucz główny tabeli.
Miasto	Zawiera nazwę miasta konkretnego wystąpienia adresu.
Ulica	Zawiera nazwę ulicy konkretnego wystąpienia adresu.
Nr_Domu	Zawiera nr domu konkretnego wystąpienia adresu.
Nr_Mieszkania	Zawiera nr mieszkania konkretnego wystąpienia adresu. Nie jest wymagany.
Kod_Pocztowy	Zawiera kod pocztowy konkretnego wystąpienia adresu.

2. Kontakty - zawiera wszystkie informacje dotyczące kontaktu oraz jego identyfikator

Nr_Kontaktu	Unikalny identyfikator wystąpień adresu. Używany jako klucz główny tabeli.
Telefon	Zawiera nr telefonu konkretnego wystąpienia kontaktu .
Email	Zawiera Email konkretnego wystąpienia kontaktu .

3. Osoby - zawiera wszystkie informacje dotyczące osoby oraz jej identyfikator. Łączy się z tabelą Adres oraz Kontakty.

Nr_Osoby	Unikalny identyfikator wystąpień osób. Używany jako klucz główny tabeli.
Nazwisko	Zawiera nazwisko konkretnego wystąpienia osoby.
Imie	Zawiera imię konkretnego wystąpienia osoby.
Data_Urodzenia	Zawiera datę urodzenia konkretnego wystąpienia osoby.
PESEL	Zawiera unikalny PESEL konkretnego wystąpienia osoby.
Adres_Nr	Klucz obcy tabeli Osoba łączący ją z Adresem.
Kontakt_Nr	Klucz obcy tabeli Osoba łączący ją z Kontaktem.

4. Lekarze - zawiera wszystkie informacje dotyczące lekarza oraz jej identyfikator. Łączy się z tabelą Osoby, Specjalizacje oraz Recepty.

Nr_Lekarza	Unikalny identyfikator wystąpień lekarzy. Używany jako klucz główny tabeli.
Osoba_Nr	Klucz obcy tabeli Lekarze łączący ją z Osobą.
Specjalizacja_Nr	Klucz obcy tabeli Lekarze łączący ją z Specjalizacją.
Ostatnia_Recepta	Klucz obcy tabeli Lekarze łączący ją z Receptami.

5. Specjalizacje - zawiera identyfikator specjalizacji, jej nazwę oraz opis. Ściśle związana z lekarzami.

Nr_Specjalizacji	Unikalny identyfikator specjalizacji. Używany jako klucz główny tabeli.
Nazwa_Specjalizacji	Zawiera nazwę specjalizacji.
Opis	Zawiera opis specjalizacji.

6. Pacjenci- zawiera identyfikator pacjenta.

Nr_Karty_Pacjenta	Unikalny identyfikator Pacjenta. Używany jako klucz główny tabeli.
Osoba_Nr	Klucz obcy tabeli Pacjenci łączący ją z Osobą.

7. Recepty - Zawiera wszystkie informacje na temat recept.

Nr_Recepty	Unikalny identyfikator Recepty. Używany jako klucz główny tabeli.
Wizyta_Nr	Klucz obcy tabeli Recepty łączący ją z Wizytą.
Data_Wystawienia	Zawiera datę wystawienia recepty.
Data_Waznosci	Zawiera datę ważności recepty.
Zalecenia	Zawiera zalecenia wprowadzone przez lekarza w trakcie jej wystawiania.

8. Sesje- zawiera identyfikator sesji oraz jej parametry..

ID_Sesji	Unikalny identyfikator Sesji. Używany jako klucz główny tabeli.
Token	Zawiera token sesji.
EXPR	Zawiera datę wygaśnięcia sesji.
Osoba_Nr	Klucz obcy tabeli Sesja łączący ją z Osobą..

9. Konta - zawiera identyfikator konta oraz wszystkie jego informacje.

Nr_Producenta	Unikalny identyfikator Producenta. Używany jako klucz główny tabeli.
Nazwa	Zawiera Nazwę Producenta.
Adres_Nr	Klucz obcy tabeli Producent łączący ją z Adresem.
Kontakt_Nr	Klucz obcy tabeli Producent łączący ją z Kontaktem.

10. Leki- zawiera identyfikator leku oraz wszystkie jego informacje.

Nr_Leku	Unikalny identyfikator Leku. Używany jako klucz główny tabeli.
Nazwa_Leku	Zawiera Nazwę leku.
Producent_Nr	Klucz obcy tabeli Leki łączący ją z Producentem.
Opis	Zawiera Opis leku.

11. Leki_z_Apteki - zawiera szczegółowe informacje na temat leku..

Lek_Nr	Unikalny identyfikator Leku. Używany jako klucz główny tabeli.
Cena	Zawiera Cenę leku.
Zdjęcie	Zawiera URL do zdjęcia leku.
Odnosnik	Zawiera URL do strony na której lek się znajduje.

12. Lek_na_Recepte - zawiera identyfikator recepty i leki na niej zawarte.

Recepta_Nr	Klucz obcy tabeli Lek_i_na_Recepte łączący ją z Receptą.
Lek_Nr	Klucz obcy tabeli Lek_i_na_Recepte łączący ją z Lekiem.

13. Wizyty - zawiera identyfikator wizyty, daty i odwołania do pacjenta oraz lekarza.

Nr_wizyty	Unikalny identyfikator Wizyty. Używany jako klucz główny tabeli.
Lekarz_Nr	Klucz obcy tabeli Wizyty łączący ją z Lekarzem.
Pacjent_Nr	Klucz obcy tabeli Wizyty łączący ją z Pacjentem.
Data_Wizyty	Zawiera datę wizyty.
Opis	Zawiera opis wizyty wprowadzony przez lekarza.
Czy_Odbyta	Zawiera status wizyty.

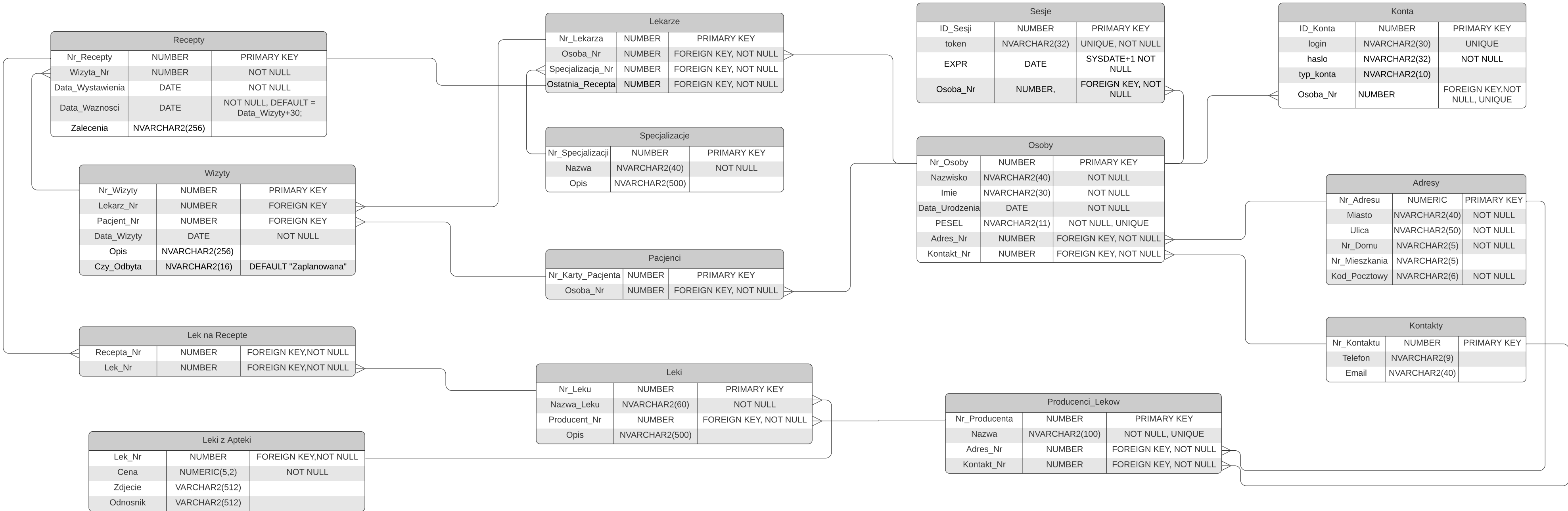
14. Konta - zawiera identyfikator konta, jego dane oraz odwołanie do osoby jako użytkownika.

ID_Konta	Unikalny identyfikator Konta. Używany jako klucz główny tabeli.
login	Zawiera unikalny login Konta.
haslo	Zawiera hasło Konta.
typ_konta	Zawiera jeden z trzech typów konta.
Osoba_Nr	Klucz obcy tabeli Konta łączący ją z Osobą.

4. Diagramy (diagram związku encji dla projektu)

Projekt Bazy Danych 1

| January 24, 2022



5. Opis widoków, procedur, kursorów, wyzwalaczy

1. Widoki

Nasz projekt posiada 14 widoków.

Oto one:

1. ADMINVIEW_ADMIN - jest to widok który wyświetla wszystkie informacje na temat kont adminów w panelu administratora. Widok ten łączy informacje z 4 tabeli. Widok ten jest użyty do pobierania danych do tabeli w panelu administratora.
2. ADMINVIEW_PACJENT- jest to widok który wyświetla wszystkie informacje na temat kont pacjentów w panelu administratora. Widok ten łączy informacje z 5 tabeli. Widok ten jest użyty do pobierania danych do tabeli w panelu administratora.
3. ADMINVIEW_SPECJALIZACJE- jest to widok który wyświetla informacje o specjalizacjach w panelu administratora.
4. ADMINVIEW_WIZYTA - jest to widok który wyświetla wszystkie informacje o wizycie, jej lekarzu i pacjencie. Widok ten łączy informacje z 6 tabeli.
5. LEKARZ_WIZYTY - jest to widok który wyświetla wszystkie informacje o wszystkich wizytach i ich pacjentach dla konkretnego lekarza. Widok ten łączy informacje z 3 tabeli.
6. LEKARZE_VIEW- jest to widok który wyświetla wszystkie informacje o lekarzach. Widok ten łączy informacje z 6 tabeli. Jest używany do dodawania, aktualizacji i usuwania lekarzy.
7. LEKI_VIEW - jest to widok który wyświetla wszystkie informacje o lekach. Widok ten łączy informacje z 2 tabeli.
8. PACJENCI_VIEW - jest to widok który wyświetla wszystkie informacje o pacjentach. Widok ten łączy informacje z 5 tabeli.
9. PACJENT_RECEPTY - jest to widok który wyświetla wszystkie recepty i ich dane dla jednego pacjenta. Widok ten łączy informacje z 6 tabeli.
10. PACJENT_WIZYTY - jest to widok który wyświetla wszystkie wizyty i ich dane dla jednego pacjenta. Widok ten łączy informacje z 5 tabeli.
11. PACJENTLEKARZAINFO - jest to widok który wyświetla wszystkich pacjentów i ich dane dla jednego lekarza. Widok ten łączy informacje z 4 tabeli.
12. PRODUCENCI_LEKOW_VIEW - jest to widok który wyświetla wszystkie informacje na temat producentów leków. Widok ten łączy informacje z 3 tabeli.
13. RECEPTYLEKARZA - jest to widok który wyświetla wszystkie recepty danego lekarza. Widok ten łączy informacje z 3 tabeli.
14. REQINFO - jest to widok który wyświetla wszystkie informacje na temat kont i ich użytkowników. Widok ten łączy informacje z 4 tabeli.

2. Procedury

Nasz projekt posiada 20 procedur.

Oto kilka z nich:

1. DODAJLEKARZA_ADMIN - jest to procedura która pozwala na dodanie nowego lekarza. Jest ona używana w panelu admina.
2. DODAJRECEPTE_Z_WIZYTY - jest to procedura która pozwala na dodanie nowej recepty do wizyty. Jest ona używana w panelu lekarza.
3. EDYTUJRECEPTE_LEKARZ - jest to procedura która pozwala na edycje recept. Jest ona używana w panelu lekarza.
4. ODWOLAJWIZYTELEK - jest to procedura która pozwala na odwołanie wizyty. Jest ona używana w panelu lekarza.
5. ODWOLAJWIZYTEPAC - jest to procedura która pozwala na odwołanie wizyty. Jest ona używana w panelu pacjenta.
6. UMOW_WIZYTE - jest to procedura która pozwala na umówienie wizyty. Jest ona używana w panelu pacjenta.
7. ADMINEDYTUJ_LEKARZA - jest to procedura która pozwala na edycję lekarza. Jest ona używana w panelu pacjenta. Projekt posiada również inne, bardzo podobne do tej procedury takie jak: ADMINEDYTUJ_PACJENTA, ADMINEDYTUJ_PRODUCENTA, ADMINEDYTUJ_WIZYTE.
8. ADMINUSUN_LEKARZA - jest to procedura która pozwala na usunięcie lekarza. Jest ona używana w panelu admina. Projekt posiada również inne, bardzo podobne do tej procedury takie jak: ADMINUSUN_ADMINA, ADMINUSUN_PACJENTA.

3. Kursory

Nasz projekt posiada 5 kursorów.

Oto kilka z nich:

1. NAZWSPEC - kursor ten przechowuje nazwy specjalizacji. Jest używany w funkcji dost_spec zwracającej specjalizacje dostępnych lekarzy.
2. LEKARZIMIENAZSPEC - kursor ten przechowuje imię, nazwisko i specjalizację lekarzy. Jest używany w funkcji dwaj_lekarze_info zwracającej dane 2 lekarzy.
3. PACIMIENAZODW - kursor ten przechowuje imię, nazwisko i ilość odbytych wizyt pacjentów. Jest używany w funkcji NajczesciejOdwiedzajacyPacjent zwracającej pacjenta z największą ilością wizyt.
4. LEKARZEIMIENAZSPECODW - kursor ten przechowuje imię, nazwisko, specjalizację i ilość odbytych wizyt lekarzy. Jest używany w funkcji NajczesciejOdwiedzaniLekarze zwracającej 4 lekarzy z największą ilością wizyt.
5. LEKCENA - kursor ten przechowuje nazwę i cenę leków. Jest używany w funkcji NajdrozszyLek zwracającej najdroższy lek w bazie.

4. Wyzwalacze

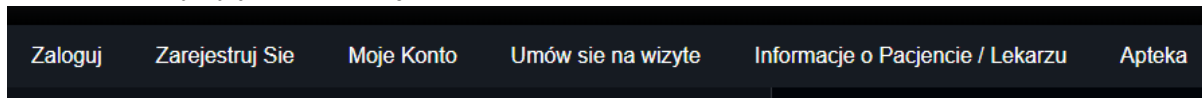
Nasz projekt posiada 7 wyzwalaczy.

Oto one:

6. CHECK_DATES_TRIGGER - wyzwalacz ten sprawdza czy data urodzenia jest mniejsza od sysdate. Używany przy dodawaniu i aktualizacji tabeli Osoby.
7. CHECK_WIZYTY_DATES_TRIGGER - wyzwalacz ten sprawdza czy data wizyty jest większa od sysdate. Używany przy dodawaniu do tabeli Wizyty.
8. LEKARZ_ADD_TRIGGER - wyzwalacz ten pozwala dodawać nowego lekarza za pomocą jednego insertu. Używany jest przy dodawaniu, aktualizacji lub usuwaniu na widoki LEKARZE_VIEW, który przechowuje wszystkie informacje na temat lekarzy.
9. LEKARZ_WIZYTY_TRIGGER - wyzwalacz ten pozwala na dodawanie nowych wizyt za pomocą widoku LEKARZ_WIZYTY. Używany jest przy dodawaniu, aktualizacji lub usuwaniu.
10. LEKI_TRIGGER - wyzwalacz ten jest bardzo podobny do wyzwalacza LEKARZ_ADD_TRIGGER, tylko zamiast lekarzy operuje na pacjentach.
11. PACJENT_TRIGGER - wyzwalacz ten pozwala na dodawanie i usuwanie leków za pomocą widoku PACJENCI_VIEW.
12. PRODUCENT_LEKOW_ADD_TRIGGER - wyzwalacz ten pozwala na dodawanie, aktualizację i usuwanie producentów leków za pomocą widoku PRODEUCENCI_LEKOW_VIEW.

13. Lista funkcjonalności klienta oraz przedstawienie działania aplikacji w formie zrzutów ekranu

Klient bazy danych możemy podzielić na kilka modułów: strony głównej, panelu moje konto, wyszukiwarki osób będących w naszej bazie oraz apteki.

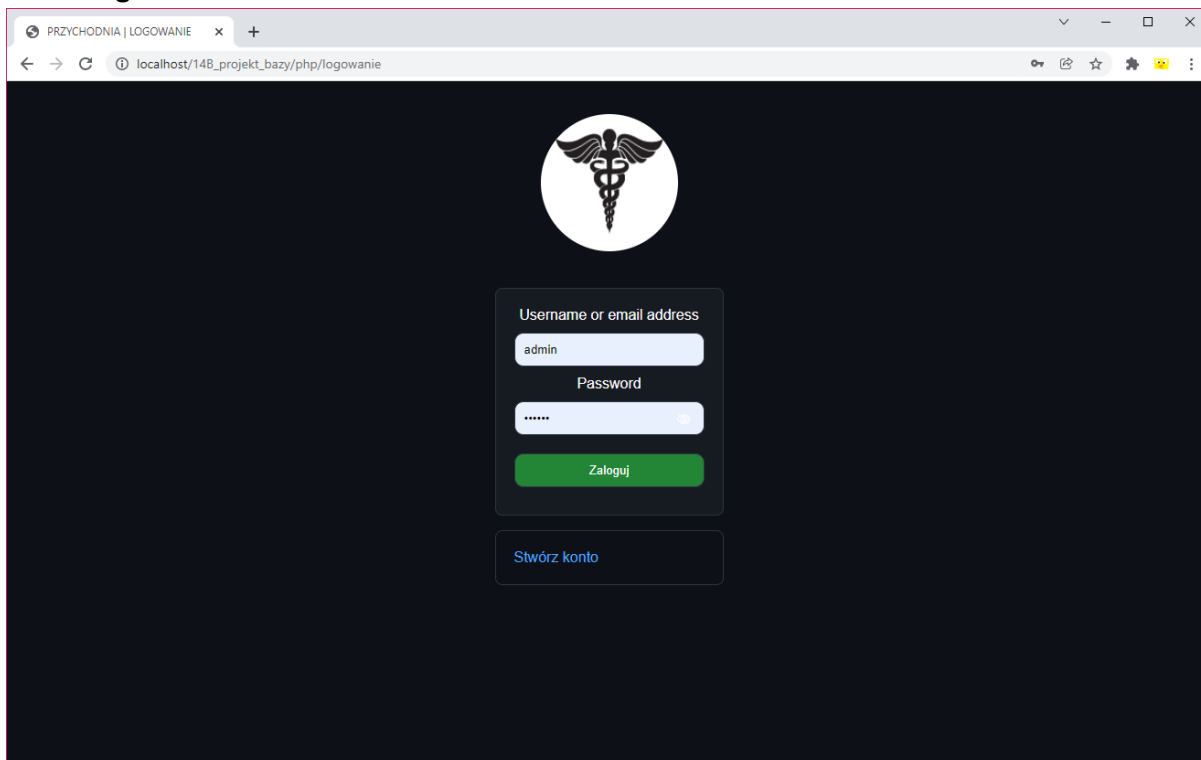


1. Strona główna



Strona główna zawiera dane statystyczne odnośnie naszej bazy. Na tej stronie możemy poznać między innymi najczęściej wybraną godzinę wizyt przez pacjentów oraz ilość wizyt o tej godzinie. Ranking lekarzy najczęściej wybieranych przez pacjentów. Ilość wizyt, leków, pacjentów i lekarzy. Spis specjalizacji lekarzy jacy pracują w naszej przychodni. Listę najlepszych lekarzy. Możliwe godziny przyjęć. Oraz pacjenta z największą ilością wizyt. Strona ta zawiera odnośniki do wszystkich kluczowych podstron. Każdy użytkownik widzi tę stronę z tej samej perspektywy. Dane statystyczne są dynamicznie pobierane i aktualizowane z serwera bazodanowego poprzez serwer PHP.

2. Panel logowania

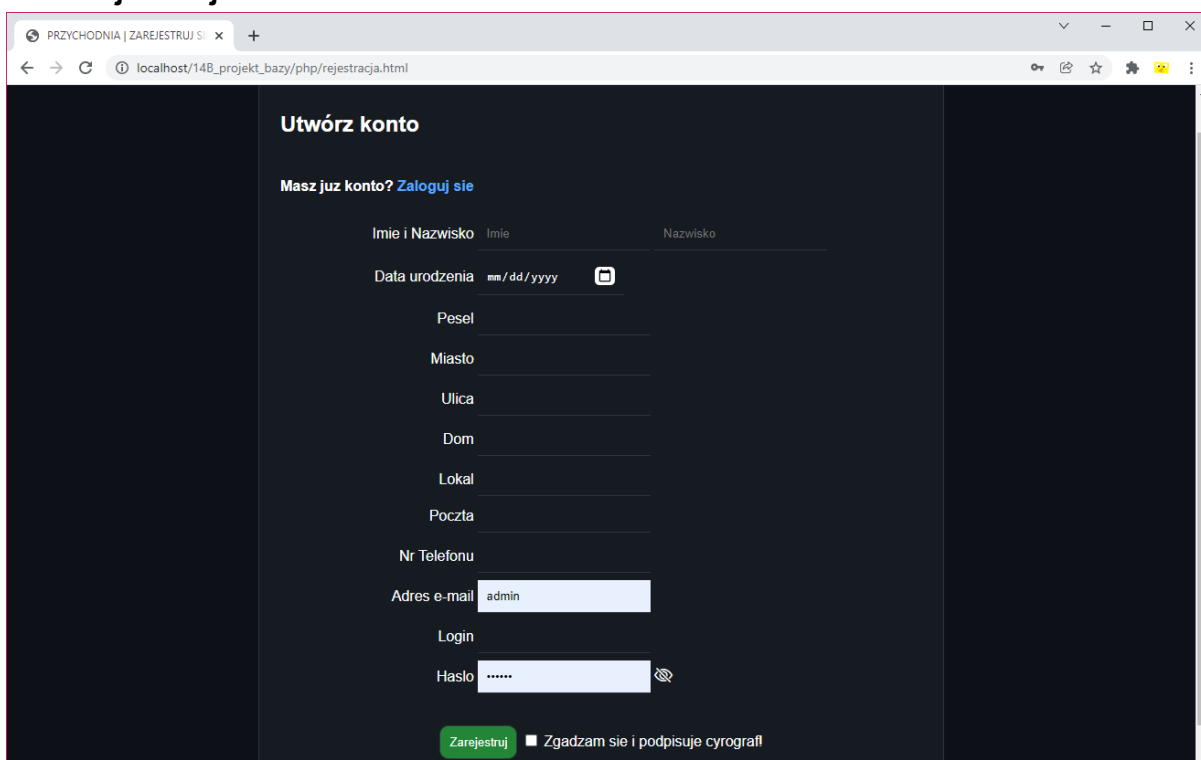


Strona ta obsługuje logowanie do konta użytkownika. W przypadku gdy użytkownik jest już zalogowany automatycznie przekierowuje do **panelu użytkownika**. Funkcja z naszego API:

```
dbRestrict("Jestes juz zalogowany/zalogowana!", "../konto", ["brak"]);
```

Należy zaznaczyć, że w profesjonalnych systemach logowania stosuje się tak zwane haszowanie (z soleniem) już po stronie klienta, w celu zabezpieczenia danych przed ich możliwym przechwyceniem. W celach prezentacyjnych, ze względu na przejrzystość informacji w naszej bazie postanowiliśmy nie stosować tej funkcjonalności. Użytkowników testujących uprasza się o nie podawania prawdziwych haseł.

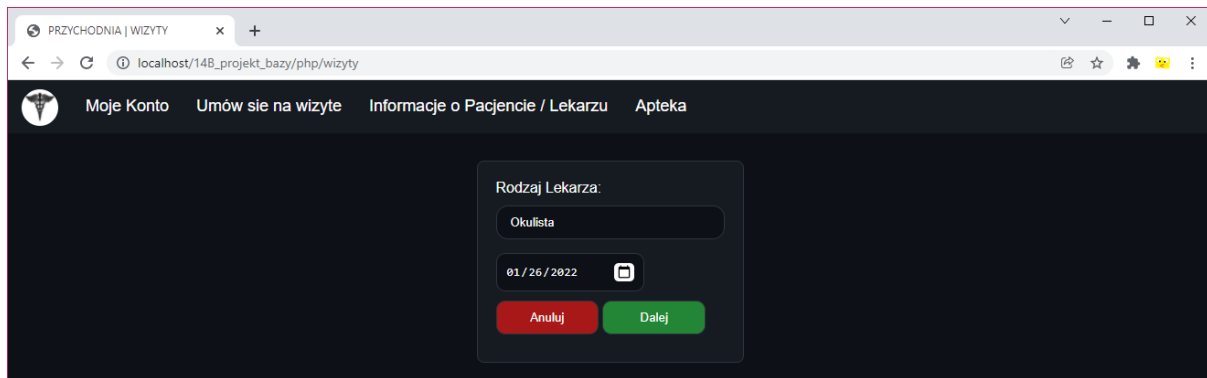
3. Panel rejestracji



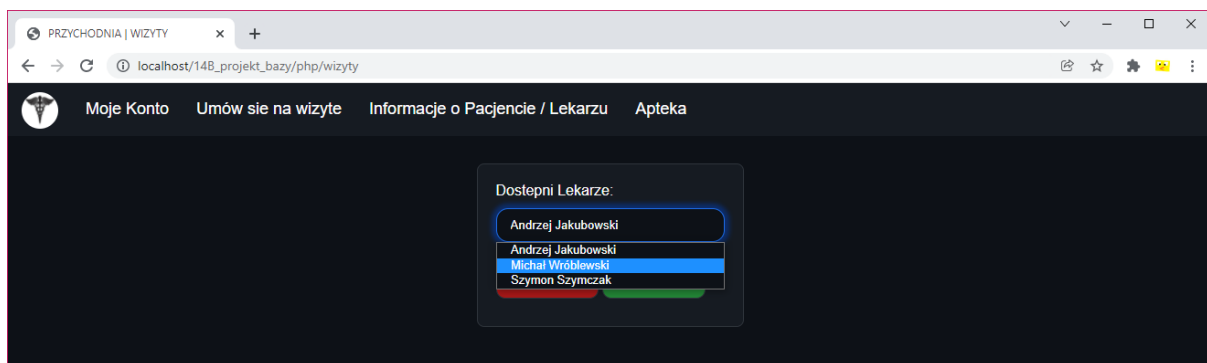
Panel rejestracji zawiera formularz umożliwiający rejestrację nowego pacjenta do przychodni. Przyjmuje on wszystkie wymagane dane potrzebne do stworzenia konta pacjenta w bazie danych. JavaScript pilnuje poprawności danych po stronie klienta, jeżeli jednak klient „oszuka” skrypt, serwer bazodanowy i tak nie przyjmie niepoprawnych danych. Po kliknięciu przycisku Zarejestruj JS serializuje wszystkie wprowadzone dane i wysyła je do serwera PHP prosząc go o wykonanie zadania „zarejestruj”. Zabezpieczenia w bazie danych zakładają, że: data urodzenia wprowadzona w tym formularzu nie może być z przyszłości (większa niż dzisiejsza), numer pesel może się składać tylko z 11 cyfr. Jeżeli zapytanie zwróci w polu success prawdę, to użytkownik zostanie automatycznie przekierowany do panelu logowania.

4. Panel umawiania się na wizytę

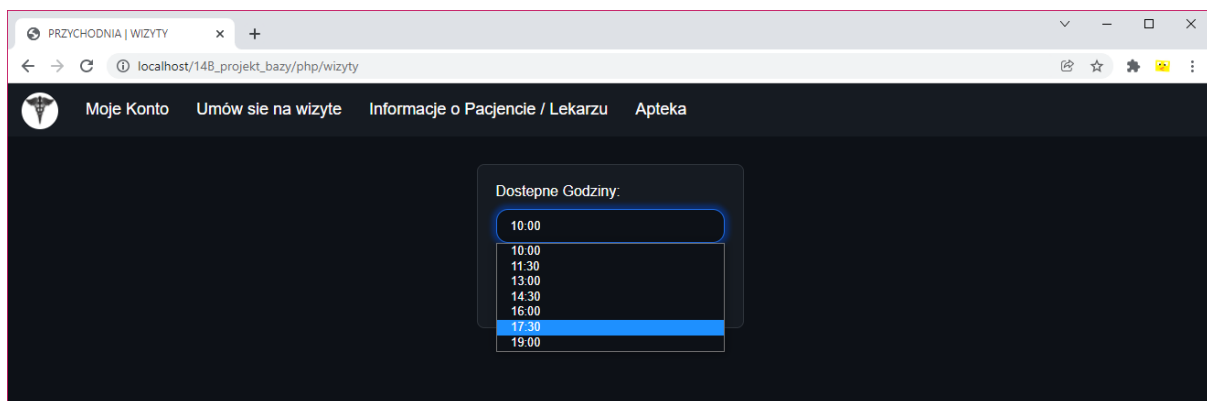
Funkcjonalność tego panelu odbywa się w kilku etapach, każdy etap zostanie zaprezentowany na osobnych zrzutach ekranu .



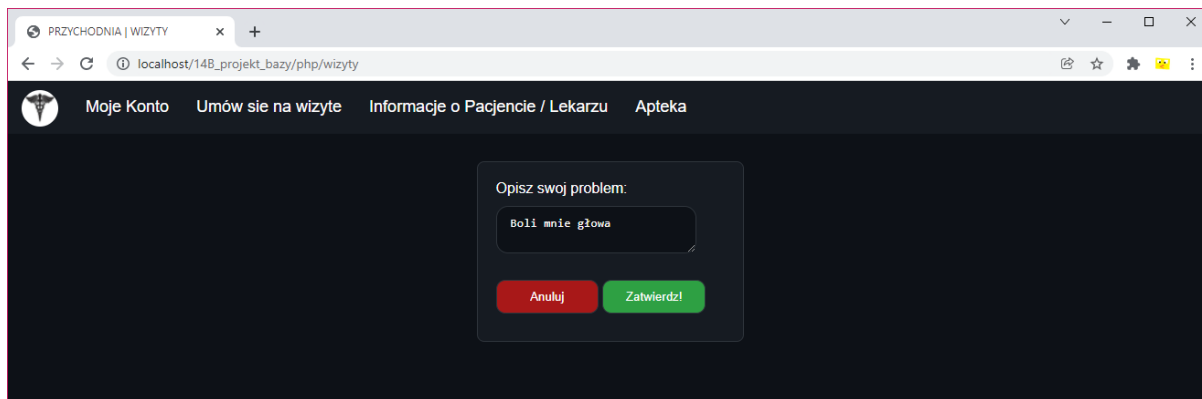
Na początku pacjent wybiera do lekarza jakiej specjalizacji chce się udać, oraz datę wizyty.



Następnie z serwera pobierana jest lista lekarzy wybranej specjalizacji i dynamicznie generowana wyświetla się na ekranie, tutaj pacjent wybiera pożądanego lekarza.

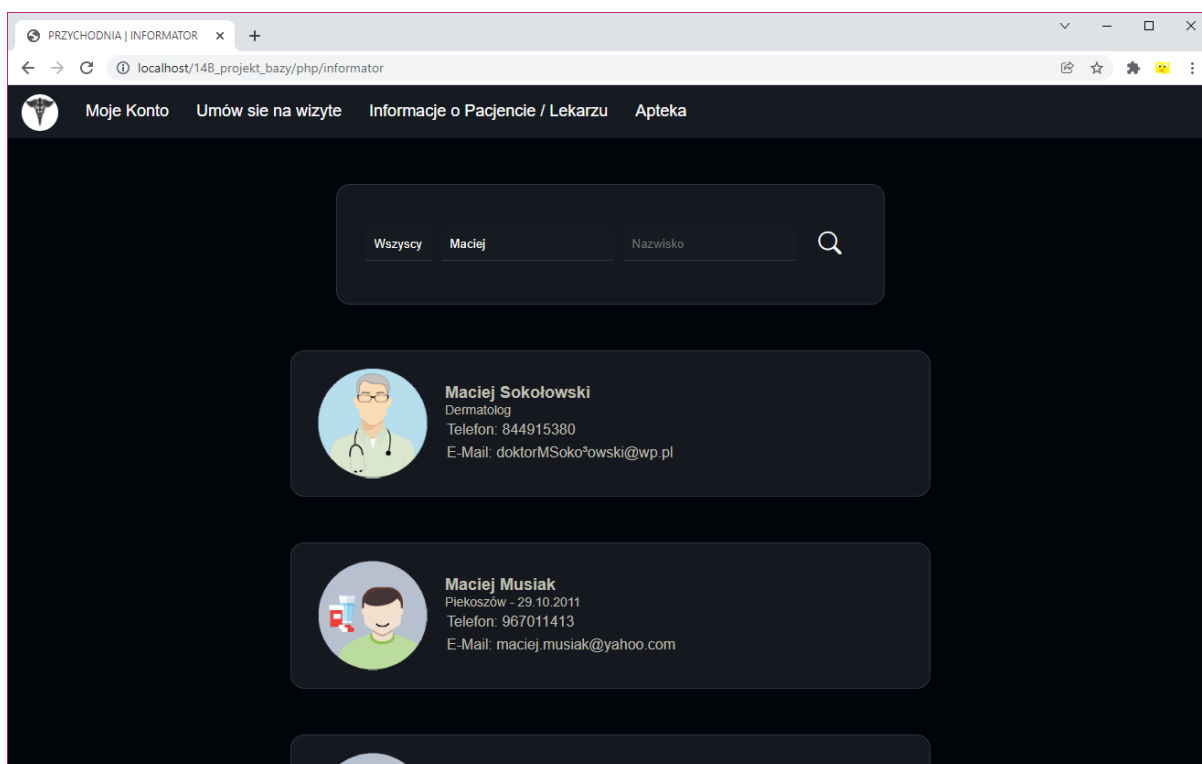


Po wyborze lekarza, zostaje wysłane zapytanie na serwer w celu sprawdzenia dostępności godzin wizyt danego lekarza. Zajęte godziny są odrzucane z listy wyboru. Pacjent wybiera odpowiadającą mu godzinę.



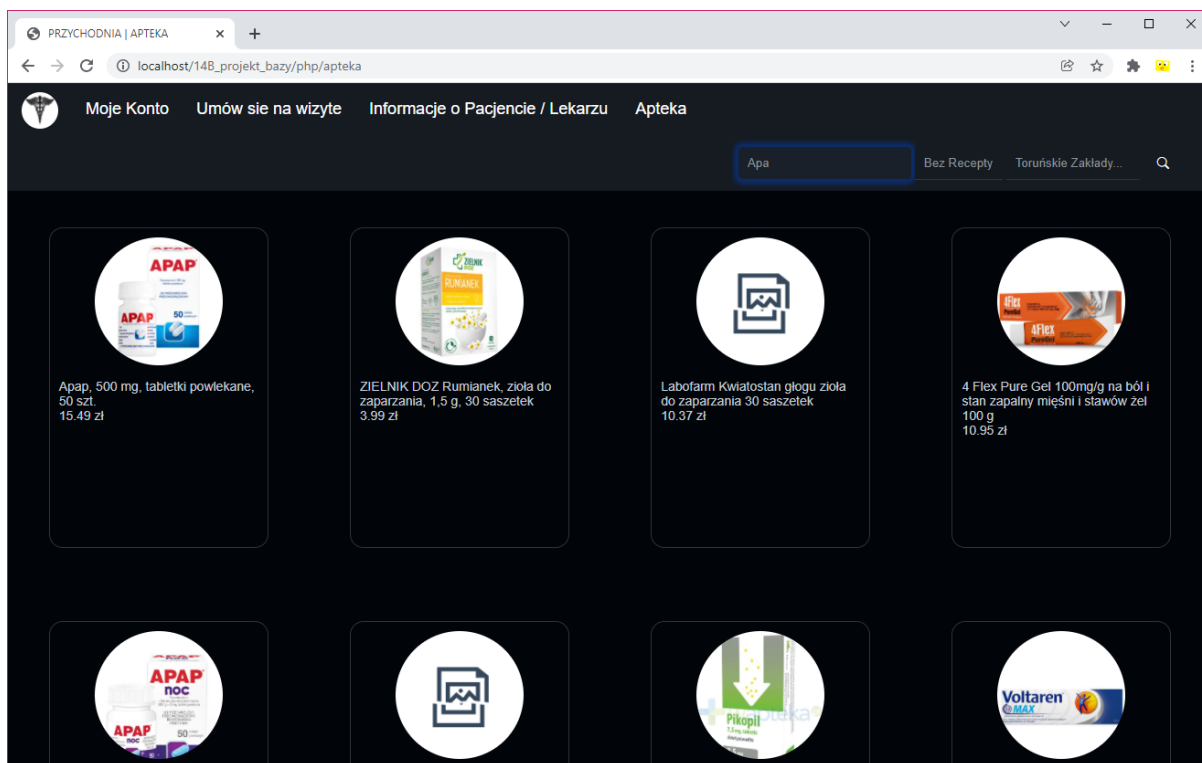
Ostatnim krokiem jest opisanie problemu pacjenta (w celu poinformowania lekarza o celu wizyty). Jeżeli wszystkie kroki przejdą pomyślnie pacjentowi wyświetli się komunikat informujący o poprawnym dodaniu wizyty.

5. Panel informacji o użytkownikach



Panel ten zawiera spis wszystkich użytkowników naszej bazy danych. Dodatkowo została tu zastosowana wyszukiwarka dzięki której możemy wybierać danych użytkowników imieniem lub nazwiskiem. Dodatkowym kryterium wyszukiwania jest typ konta użytkownika, możemy wyświetlić wszystkich użytkowników jednocześnie lub też wybraną grupę użytkowników.

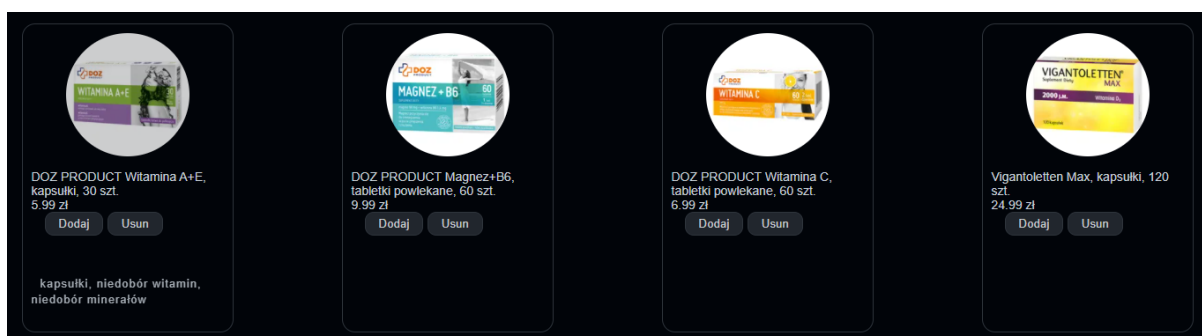
6. Panel apteki



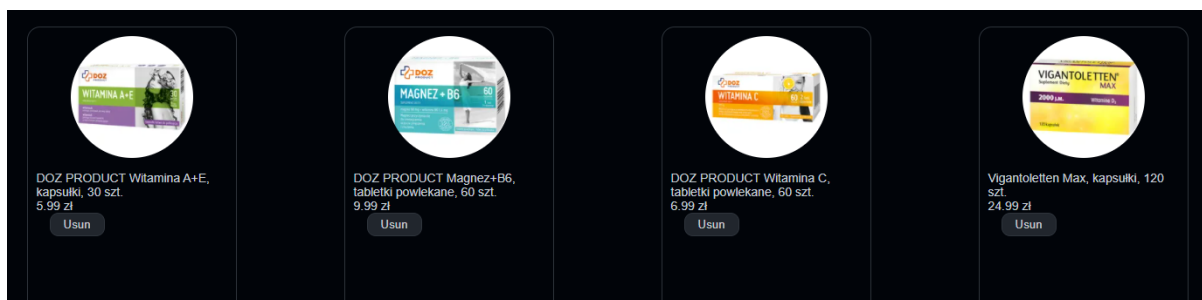
Działanie apteki różni się od typu zalogowanego użytkownika:

-osoba niezalogowana ma podgląd do leków będących w naszej aptece, może je wyszukiwać(nawet ograniczając wyszukiwanie po producencie leku), nie może wybrać recepty, gdyż osoba niezalogowana nie może mieć przypisanej recepty.

-pacjent ma podobną funkcjonalność co osoba niezalogowana, dodatkowo, ma możliwość ograniczenia wyszukiwania leków do tych z wybranej (przypisanej mu) recepty.



-lekarz może wybrać receptę do której będzie przypisywał leki, bądź też je usuwał.



-konto administratora posiada możliwość usuwania leków z bazy. Nie ma on możliwości wyszukiwania leków po receptach.

7. Panel pacjenta

Panel pacjenta umożliwia aktualizację danych oraz usunięcie konta. Jedyną własnością jakiej pacjent nie może zaktualizować jest login. Pozostałe dane można aktualizować.

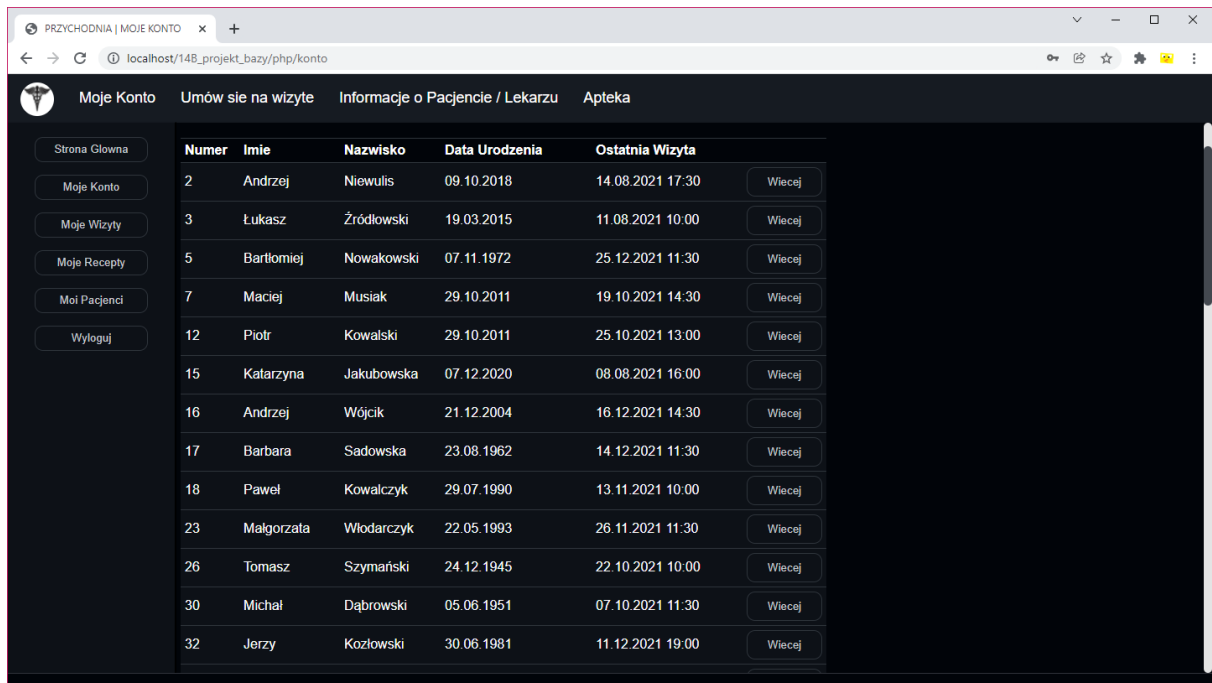
Numer	Imie	Nazwisko	Specjalizacja	Data	Opis	Status	Pacjent
432	Artur	Sadowski	Pediatra	2021-03-06 19:00	Pacjent zmarł	Odbyta	69
487	Agata	Dąbrowska	Urolog	2021-10-05 19:00	Zalecana dalsza hospitalizacja	Odbyta	69

W panelu wizyt pacjent ma możliwość wyświetlenia informacji o swoich wizytach, możliwość odwołania wizyty(jeżeli ma status zaplanowanej).

Nr Recepty	Wizyta	Lek / Lek	Zalecenia	Imie	Nazwisko	Data Waznosci
9	33	Supradyn Energia 30 tabletek powlekanych, LipiForma Plus, kapsułki, 30 szt., Dr Jacobs Tabletki Potasowe 126 tabletek	Dawkować co tydzień	Tomasz	Maciaszczyk	20/02/2022

W panelu recept pacjent ma możliwość zobaczenia swoich recept, jest też możliwość wyświetlenia leków z danej recepty w aptece.

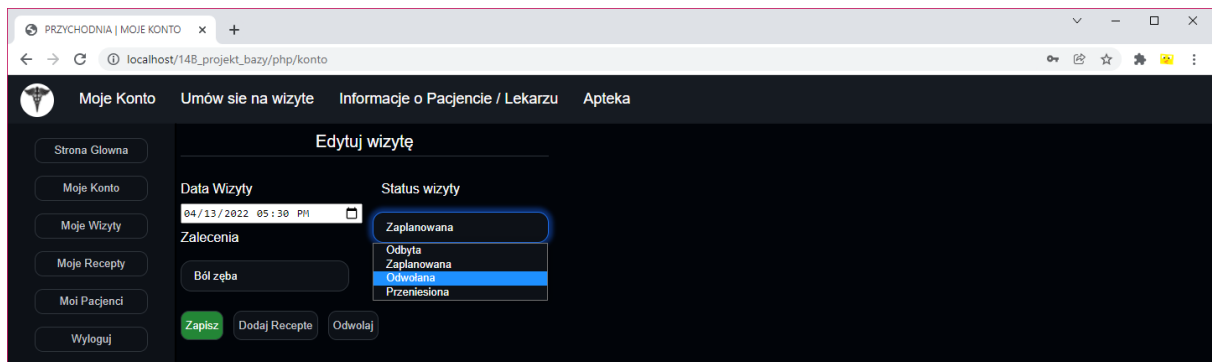
8. Panel lekarza



The screenshot shows a web application interface for a doctor's panel. The top navigation bar includes 'Moje Konto', 'Umów się na wizyte', 'Informacje o Pacjencie / Lekarzu', and 'Apteka'. The left sidebar contains links: 'Strona Główna', 'Moje Konto', 'Moje Wizyty', 'Moje Recepty', 'Moi Pacjenci', and 'Wyloguj'. The main content area displays a table of patients with columns: Numer, Imię, Nazwisko, Data Urodzenia, and Ostatnia Wizyta. Each row has a 'Wiecej' button.

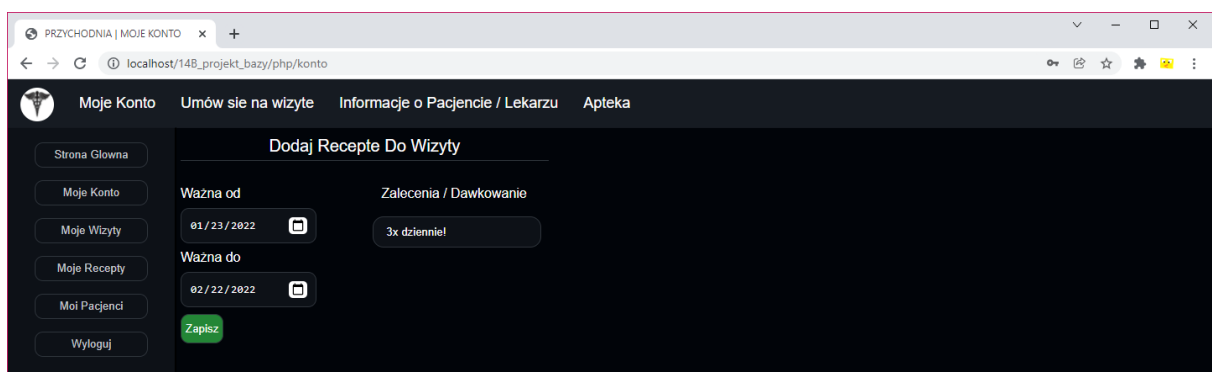
Numer	Imię	Nazwisko	Data Urodzenia	Ostatnia Wizyta
2	Andrzej	Niewulis	09.10.2018	14.08.2021 17:30
3	Łukasz	Żródlowski	19.03.2015	11.08.2021 10:00
5	Bartłomiej	Nowakowski	07.11.1972	25.12.2021 11:30
7	Maciej	Musiak	29.10.2011	19.10.2021 14:30
12	Piotr	Kowalski	29.10.2011	25.10.2021 13:00
15	Katarzyna	Jakubowska	07.12.2020	08.08.2021 16:00
16	Andrzej	Wójcik	21.12.2004	16.12.2021 14:30
17	Barbara	Sadowska	23.08.1962	14.12.2021 11:30
18	Paweł	Kowalczyk	29.07.1990	13.11.2021 10:00
23	Małgorzata	Włodarczyk	22.05.1993	26.11.2021 11:30
26	Tomasz	Szymański	24.12.1945	22.10.2021 10:00
30	Michał	Dąbrowski	05.06.1951	07.10.2021 11:30
32	Jerzy	Kozłowski	30.06.1981	11.12.2021 19:00

W panelu lekarza istnieją analogiczne panele jak w przypadku pacjenta. Lekarz ma możliwość edycji danych swojego konta, wyświetlenie wszystkich wizyt go dotyczących. Oraz recept które wystawił. Dodatkowo lekarz może wyświetlić spis swoich pacjentów. Każdy pacjent który odbędzie choć jedną wizytę (wizyta ze statusem odbyta) zostanie umieszczony w tym spisie.



The screenshot shows the 'Edytuj wizytę' form. It includes fields for 'Data Wizyty' (04/13/2022 05:30 PM) and 'Status wizyty'. The status dropdown menu is open, showing options: 'Zaplanowana', 'Odbita', 'Zaplanowana', 'Odwołana', and 'Przeniesiona'. There are also buttons for 'Zapisz', 'Dodaj Receptę', and 'Odwołaj'.

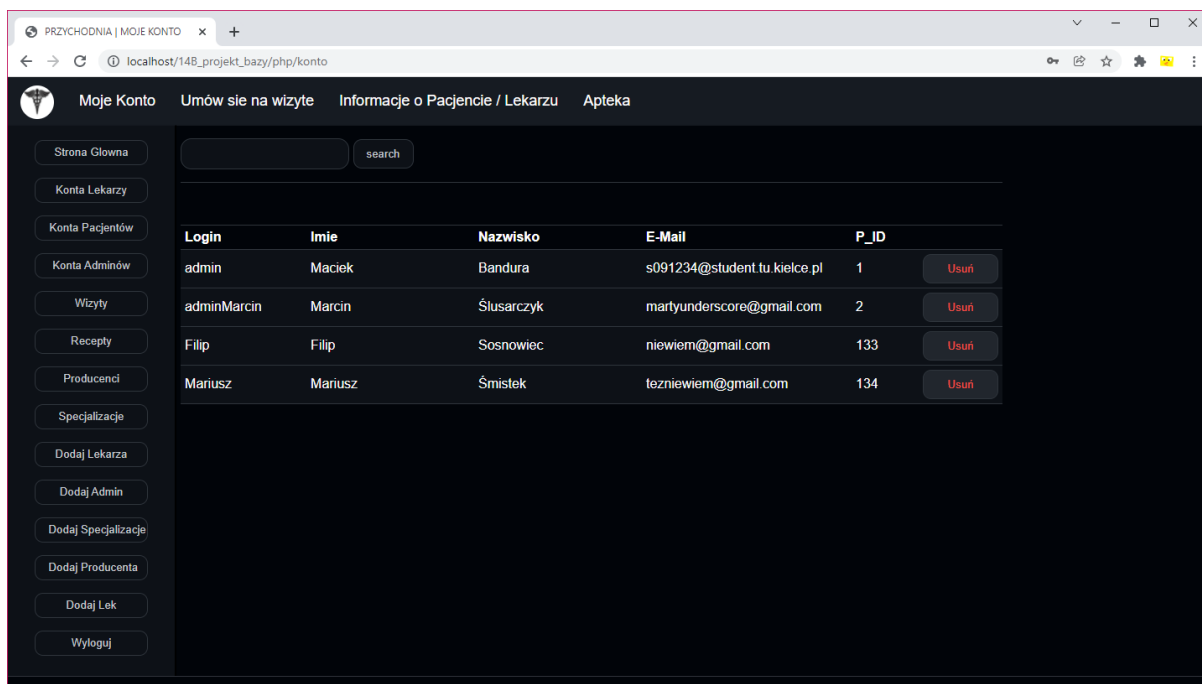
Kolejną ważną funkcjonalnością panelu lekarza jest możliwość edycji wizyty. W tym panelu każdy lekarz może: zmienić status danej wizyty, zmienić datę wizyty(w przypadku np.. przeniesienia), odwołać ją lub przypisać jej nową receptę.



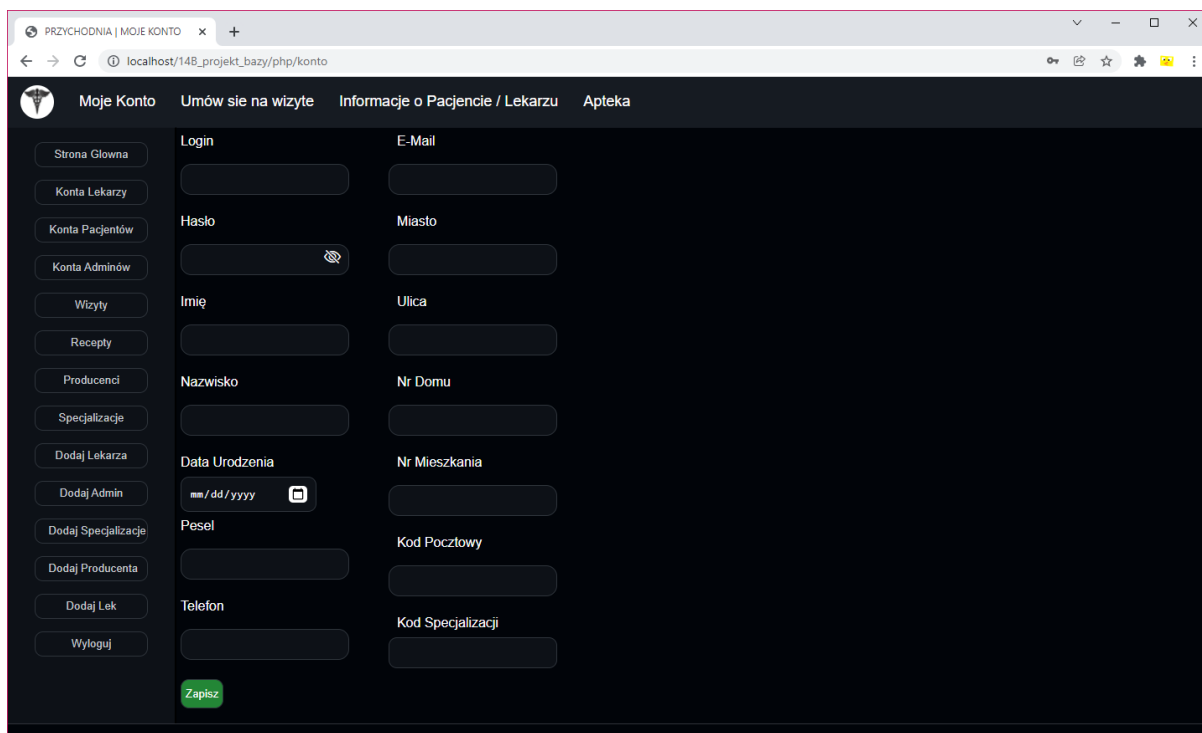
The screenshot shows the 'Dodaj Receptę Do Wizyty' form. It includes fields for 'Ważna od' (01/23/2022) and 'Ważna do' (02/22/2022). There is a 'Zalecenia / Dawkowanie' field with the value '3x dziennie!'. A 'Zapisz' button is visible at the bottom.

Dodając receptę należy ustalić jej właściwości takie jak data wystawienia (domyślnie aktualna data), data ważności (domyślnie aktualna data + 30 dni), Opcjonalnie lekarz może napisać zalecenia dotyczące leków na receptę. Po kliknięciu przycisku **ZAPISZ** lekarz zostanie automatycznie przekierowany do apteki gdzie będzie mógł dodawać leki do tej recepty.

9. Panel administratora



Panel administratora jest najbardziej rozbudowanym modulem naszego klienta. Zawiera on m. in. Listę kont lekarzy, pacjentów i adminów. W jego skład wchodzi również spis wszystkich wizyt, recept, producentów leków i specjalizacji.



Konto administratora ma możliwość dodawania lekarzy oraz dodatkowych administratorów. Po przejściu do panelu Dodaj Lekarza wyświetla się formularz (podobny do tego z rejestracji pacjenta) który zbiera wszystkie dane potrzebne do utworzenia nowego lekarza w naszej bazie.

Nr	Data	Opis	Status	Lekarz	Imie	Nazwisko	Specjalizacja	Pacjent	Imie	Nazwisko	
1	2022-03-11 13:00	Kołatanie serca	Zaplanowana	2	Tomasz	Maciaszczyk	Dentysta	81	Józefa	Mróz	Więcej
2	2021-02-12 13:00	Pacjent odwołał wizytę	Odwolana	18	Julia	Duda	Hematolog	7	Maciej	Musiak	Więcej
3	2021-03-04 13:00	Pacjent zmarł	Odbyła	10	Beata	Mazurek	Pediatra	28	Marek	Woźniak	Więcej
4	2021-02-13 10:00	Problem sam się rozwiązał	Odwolana	10	Beata	Mazurek	Pediatra	7	Maciej	Musiak	Więcej
5	2022-03-21 11:30	Przemęczenie studenckie	Zaplanowana	12	Maciej	Sokolowski	Dermatolog	108	Mikołaj	Pietrzak	Więcej

Spis wizyt zawiera wszelkie informacje odnośnie każdej wizyty.

Nr	Wizyta	Lekarz: Imie	Nazwisko	Pacjent: Imie	Nazwisko	Data Waznosci	Data Wystawienia		
1	3	Beata	Mazurek	28	Marek	Woźniak	2022.01.21	2022.02.20	Usun
2	8	Grzegorz	Nowak	35	Jadwiga	Szczepańska	2022.01.21	2022.02.20	Usun
3	11	Artur	Sadowski	49	Danuta	Wysocka	2022.01.21	2022.02.20	Usun
4	14	Julia	Duda	42	Tadeusz	Krawczyk	2022.01.21	2022.02.20	Usun

W spisie recept administrator ma możliwość usunięcia danej recepty. Wraz z usunięciem instancji recepty z tabeli **Recepty** usuwane też są wpisy w tabeli **Lek_Na_Recepte** dotyczących usuwanej recepty.

Po kliknięciu przycisku więcej administrator ma możliwość edycji niektórych danych wizyty, takich jak: data wizyty, opis wizyty, status wizyty, lekarz obsługujący daną wizytę i pacjent zapisany na daną wizytę. W przypadku wybrania daty wizyty która nie jest dostępna dla wybranego lekarza serwer bazodanowy za pośrednictwem serwera PHP zwróci błąd.

ID	Nazwa	E-Mail	Telefon	
1	USP Zdrowie sp. z o.o.	hurtownia@usp.pl	225436000	Więcej
2	Mabion S.A.	info@mabion.eu	422077890	Więcej
3	Pelion S.A.	hr@pelion.eu	422007468	Więcej

Administrator ma opcję podglądu, dodawania i edycji producentów leków.

Kolejną ważną funkcjonalnością panelu administratora jest możliwość dodania nowych leków do apteki. Odbyna się to poprzez wpisanie numeru producenta leku oraz wszystkich informacji potrzebnych do identyfikacji danego leku.

Kod	Specjalizacja	Opis	Dostępni Lekarze	
1	Kardiolog	Lekarz medycyny zajmujący się chorobami serca i układu krążenia.	2	Więcej
2	Dentysta	Lekarz medycyny zajmujący się chorobami zębów i jamy ustnej	2	Więcej
3	Okulista	Lekarz medycyny zajmujący się badaniem i leczeniem narządu wzroku oraz korygowaniem jego wad	1	Więcej

Administrator ma możliwość podglądu, dodawania i edycji specjalizacji.

Login	Imię	Nazwisko	E-Mail	P_ID	
admin	Maciek	Bandura	s091234@student.tu.kielce.pl	1	Usuń
adminMarcin	Marcin	Ślusarczyk	martyunderscore@gmail.com	2	Usuń
Filip	Filip	Sosnowiec	niewiem@gmail.com	133	Usuń
Mariusz	Mariusz	Śmistek	teznieciem@gmail.com	134	Usuń

W przypadku chęci usunięcia kont administratorów istnieje zabezpieczenie po stronie serwera bazodanowego, mówiące o tym, że nie można usunąć konta administratora numer 1.

14. Kod źródłowy

Kopia zostanie przesłana w załączniku zip natomiast całe repozytorium projektu znajduje się pod adresem:

GIT (github.com): https://github.com/MartyLocke/14B_Projekt_Bazy

15. Problemy napotkane podczas prac nad projektem

Projekt jest bardzo rozbudowany, dlatego też podczas prac nad nim pojawiło się wiele problemów. Jedne były spowodowane brakiem wymaganej wiedzy dotyczącej obsługi baz danych inne zaś zbyt wygórowanymi wymaganiami poszczególnych członków zespołu projektowego. Kilka przykładowych problemów:

- sprawdzanie daty w ograniczeniu **CHECK**. Nie spodziewaliśmy się, że nie można sprawdzać pól typu date w tym ograniczeniu. Jak rozwiązaliśmy ten problem? Za pomocą wyzwalacza. Za pomocą **BEFORE INSERT** na tabeli w której chcieliśmy zastosować to ograniczenie. W wyzwalaczu za pomocą instrukcji **IF** sprawdzającej **:new.Data** decydujemy czy data jest większa/mniejsza od aktualnej i wtedy decydujemy, czy chcemy dodać następujący rekord czy wyświetlić błąd.

- problemem typowo projektowym było podjęcie decyzji dotyczących kluczy obcych i ograniczeń z nich wynikających. Ustawienie parametrów **ON UPDATE** oraz **ON DELETE**.

- dość mocnym problemem było skonfigurowanie bazy, serwera PHP, oraz panelu klienta aby poprawnie wyświetlał polskie znaki. W ostatecznej wersji postawiliśmy na kodowanie UTF-8, a dokładnie: **AL32UTF8**. Takie kodowanie musi być ustawione zarówno na bazie ORACLE, połączeniu PHP z bazą i stronie klienta.

- kolejnym problemem był brak możliwości wywoływania operacji DDL w procedurach i funkcjach, przez to musieliśmy zmienić trochę projekt naszej bazy. W przypadku funkcji bardzo brakowało nam możliwości zwracania całych tabel z funkcji. Chcieliśmy rozbudować instrukcję **SELECT** owijać w funkcję aby w pliku req.php. Niestety ale nie udało nam się tego zrobić i postanowiliśmy, że zamiast funkcji skorzystamy z widoków.

- największym problemem podczas prac nad tym projektem była jego złożoność. Projekt ten kosztował nas bardzo wiele czasu, sporo wyrzeczeń i wysiłku. Nowo powstałe pomysły musiały być przesuwane na dalszy plan z powodu braku czasu i zbliżającej się sesji.

16. Wnioski

Projekt nauczył nas bardzo wiele z zakresu projektowania i zarządzania systemami bazodanowymi. Problemy napotkane podczas prac nad projektem nauczyły nas kolektywnego rozwiązywania przeciwności stawianych podczas pracy. Wspólny czas spędzony na pracy nad projektem pozwolił usprawnić umiejętności z zakresu współpracy i dyskusji nad złożonymi problemami. Ten miesiąc spędzony przy pracy nad projektem poszerzył nasze horyzonty oraz wiedzę zarówno z przedmiotu Bazy Danych 1 jak i w poznawaniu i serwisów internetowych. Połączenie bazy danych z językiem PHP znacząco poszerza jego możliwości. Zespołowa praca nad projektem tego rozmiaru, znacząco upraszcza się dzięki narzędziu do kontroli wersji jakim jest GIT(Repozytorium hostowane na serwerach Github). Z perspektywy czasu gdybyśmy dziś mieli wykonywać podobny projekt podeszli byśmy do tego w inny sposób. W bardziej precyzyjny sposób określilibyśmy schemat i strukturę bazy dzięki czemu nie byłyby konieczne późniejsze poprawki. Projekt można uznać za zakończony nie mniej jednak istnieją możliwości dalszej rozbudowy. Pojawił się pomysł systemu wymiany wiadomości pomiędzy lekarzem i pacjentem. Dodanie takich funkcjonalności jak: obsługa skierowań czy większa funkcjonalność producentów leków. Zaprojektowanie bardziej rozbudowanej strony głównej.