**Dokumentacja Projektu**

**Temat Projektu:**  
System HMS – Szpitalny System Zarządzania z Elektroniczną Wymianą Danych  
**Przedmiot:** Studium Przypadku – SQL Server

**Autor: Mikołaj Bacior 137095**

**Spis** **Treści**

[1. Opis projektu 4](#_Toc198300583)

[1.1. Główne funkcje 4](#_Toc198300584)

[2. Instalacja i uruchomienie 5](#_Toc198300585)

[2.1. Wymagane oprogramowanie 5](#_Toc198300586)

[2.2. Kroki instalacji 5](#_Toc198300587)

[2.3. Dane logowania 5](#_Toc198300588)

[3. Struktura 6](#_Toc198300589)

[3.1. Struktura i opis bazy danych 6](#_Toc198300590)

[3.3. Kluczowe relacje i mechanizmy 8](#_Toc198300591)

[3.4. Schemat ERD 8](#_Toc198300592)

[3.5. Opis wybranych plików 9](#_Toc198300593)

[4. Opis fragmentów kodu źródłowego 11](#_Toc198300594)

[4.1. Strona Pacjenta (Patient.php i powiązane) 11](#_Toc198300595)

[Funkcja 1: Ustawianie ciasteczka wybranego pacjenta (cookies) 11](#_Toc198300596)

[Funkcja 2: Umawianie wizyty przez pacjenta (AJAX + PHP) 11](#_Toc198300597)

[4.2. Strona Lekarza 13](#_Toc198300598)

[Funkcja 1: Dynamiczne ładowanie pacjentów wybranego lekarza 13](#_Toc198300599)

[Funkcja 2: Edycja diagnozy wizyty przez lekarza 14](#_Toc198300600)

[4.3. Strona Administratora (Admin.php) 15](#_Toc198300601)

[Funkcja 1: Eksport pacjentów do pliku Excel 15](#_Toc198300602)

[Funkcja 2: Dynamiczne przełączanie widoków (pacjenci, lekarze, oddziały) 17](#_Toc198300603)

[4.4. Strona Farmaceuty (Chemist.php) 18](#_Toc198300604)

[Funkcja 1: Eksport leków do pliku XML 18](#_Toc198300605)

[Funkcja 2: Edycja i zarządzanie lekami 19](#_Toc198300606)

[5. Opis działania poszczególnych ról 21](#_Toc198300607)

[5.1. Administrator 21](#_Toc198300608)

[5.1.1. Wybór widoku 21](#_Toc198300609)

[5.1.3. Zarządzanie lekarzami 24](#_Toc198300610)

[5.1.4. Zarządzanie oddziałami 24](#_Toc198300611)

[5.1.5. Dodatkowe wskazówki 25](#_Toc198300612)

[5.2. Lekarz 25](#_Toc198300613)

[5.2.1. Wybór lekarza i pacjenta 25](#_Toc198300614)

[5.2.2. Przeglądanie i edycja wizyt 25](#_Toc198300615)

[5.2.3. Przeglądanie i edycja badań 26](#_Toc198300616)

[5.2.4. Przeglądanie i wystawianie recept 26](#_Toc198300617)

[5.2.5. Dodatkowe dane statystyczne 26](#_Toc198300618)

[5.2.6. Wskazówki 26](#_Toc198300619)

[5.3. Pacjent 27](#_Toc198300620)

[5.3.1. Wybór specjalizacji i lekarza 27](#_Toc198300621)

[5.3.2. Wybór pacjenta 27](#_Toc198300622)

[5.3.3. Umawianie wizyty 27](#_Toc198300623)

[5.3.4. Przeglądanie swoich wizyt 28](#_Toc198300624)

[5.3.5. Przeglądanie wyników badań 28](#_Toc198300625)

[5.3.6. Przeglądanie recept 28](#_Toc198300626)

[5.3.7. Wskazówki 28](#_Toc198300627)

[5.4. Farmaceuta 29](#_Toc198300628)

[5.4.1. Przeglądanie listy leków 29](#_Toc198300629)

[5.4.2. Edycja danych leku 29](#_Toc198300630)

[5.4.3. Dodawanie nowego leku 29](#_Toc198300631)

[5.4.4. Usuwanie leku 30](#_Toc198300632)

[5.4.5. Eksport leków do pliku XML 30](#_Toc198300633)

[5.4.6. Wskazówki 30](#_Toc198300634)

[6. Podsumowanie 31](#_Toc198300635)

# Opis projektu

Projekt **HMS (Hospital Management System)** to aplikacja webowa umożliwiająca zarządzanie szpitalem, pacjentami, lekarzami, lekami, wizytami oraz elektroniczną wymianę danych (EDI).  
System został stworzony w PHP (backend), JavaScript (frontend), z wykorzystaniem bazy danych MS SQL Server oraz serwera XAMPP.

* 1. Główne funkcje**:**
* Zarządzanie pacjentami, lekarzami, oddziałami, lekami i personelem medycznym
* Umawianie wizyt, przeglądanie historii wizyt i badań
* Wystawianie i obsługa recept
* Automatyczne powiadomienia i obsługa EDI (np. eksport danych do XML/Excel)
* Różne widoki i uprawnienia dla administratora, lekarza, pacjenta i farmaceuty

# Instalacja i uruchomienie

* 1. Wymagane oprogramowanie**:**
* XAMPP (Apache, PHP, SQL Server Driver for PHP)
* Microsoft SQL Server
* Visual Studio Code
* Przeglądarka internetowa
  1. Kroki instalacji**:**
     1. Skopiuj folder projektu do katalogu htdocs XAMPP, (Projekt1).
     2. Zainstaluj sterowniki potrzebne do obsługi sql przez XAMPP (sqlsrv i pdo\_sqlsrv)
  + Link do sterowników [Link](https://learn.microsoft.com/en-us/sql/connect/php/download-drivers-php-sql-server?view=sql-server-ver16#download) 
  + Bazując na pliku SQLSRV\_Readme.htm wybierz wersję sterownika odpowiadającą wersji PHP zainstalowanej na twoim urządzeniu
  + Skopiuj php\_sqlsrv\_[wersja].dll i php\_pdo\_sqlsrv\_[wersja].dll do folderu xampp\php\ext
  + Otwórz plik php.ini z folderu xampp/php i na końcu dodaj 2 linijki kodu:

extension=php\_[wersja].dll

extension=php\_pdo\_sqlsrv\_[wersja].dll

* + 1. Uruchom XAMPP (jeśli XAMPP był uruchomiony przed instalacją sterowników należy go uruchomić ponownie) i włącz Apache.
    2. Utwórz bazę danych:
  + Otwórz SQL Server Management Studio.
  + Wykonaj plik Create+Insert.sql (lub osobno Create.sql i Insert.sql) – utworzy strukturę bazy, użytkowników, role i przykładowe dane.
  + Upewnij się, że użytkownicy i role są poprawnie utworzeni.
    1. Skonfiguruj połączenie w plikach PHP (np. Admin.php, Doctor\_Scripts> Connect.php, Chemist.php, Patient.php) – domyślnie ustawione na localhost\MSSQLSERVER01.
    2. Otwórz przeglądarkę i przejdź pod adres http://localhost/Projekt1/index.html.
  1. Dane logowania**:**
* Administrator: login administrator, hasło admin1
* Lekarz: login Lekarz, hasło Lekarz1
* Pacjent: login Pacjent, hasło Pacjent1
* Farmaceuta: login Farmaceuta, hasło Farmaceuta1

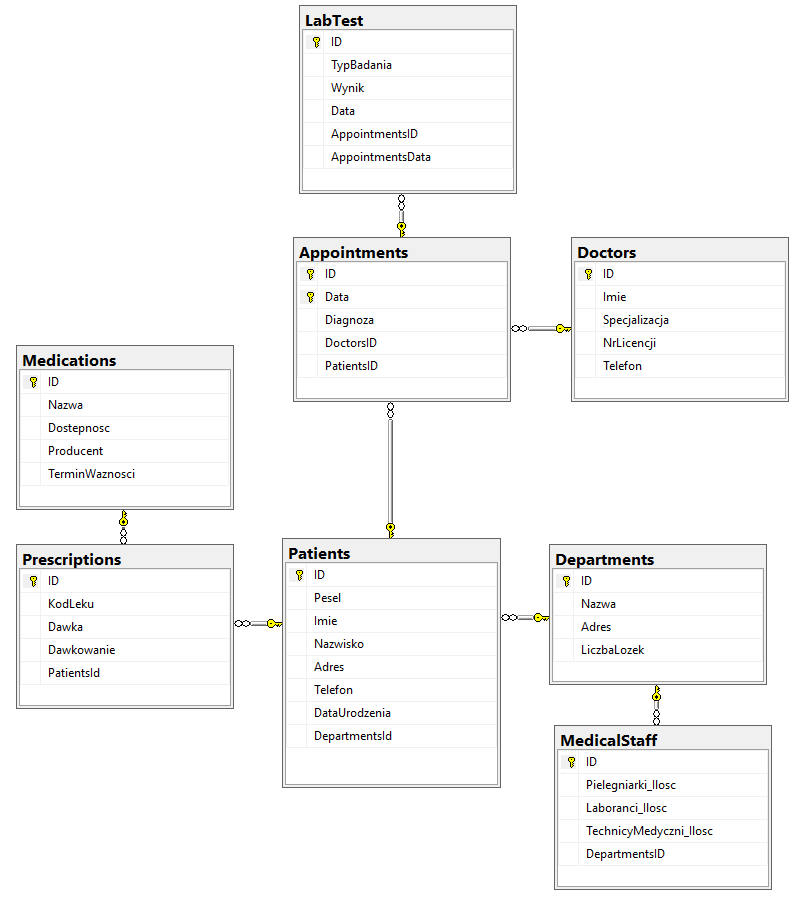
# Struktura

## Struktura i opis bazy danych

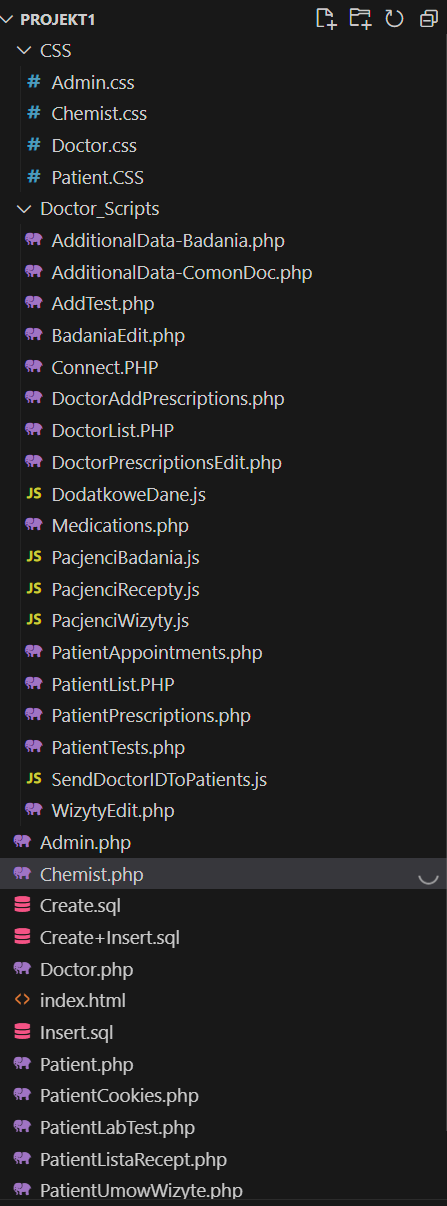
System HMS korzysta z relacyjnej bazy danych Microsoft SQL Server. Struktura bazy została zaprojektowana tak, aby umożliwić przechowywanie i powiązanie wszystkich kluczowych informacji dotyczących pacjentów, lekarzy, oddziałów, leków, wizyt, badań laboratoryjnych oraz personelu medycznego.

* 1. **Główne tabele:**
* **Departments**  
  Przechowuje informacje o oddziałach szpitalnych.
  + ID (int, PK, auto\_increment) – identyfikator oddziału
  + Nazwa (varchar) – nazwa oddziału
  + Adres (varchar) – adres oddziału
  + LiczbaLozek (int) – liczba łóżek
* **Patients**  
  Przechowuje dane pacjentów.
  + ID (int, PK, auto\_increment)
  + Pesel (char(11), unique) – numer PESEL
  + Imie, Nazwisko (varchar) – imię i nazwisko
  + Adres (varchar) – adres zamieszkania
  + Telefon (varchar) – numer telefonu
  + DataUrodzenia (date) – data urodzenia
  + DepartmentsId (int, FK) – powiązanie z oddziałem
* **Doctors**  
  Przechowuje dane lekarzy.
  + ID (int, PK, auto\_increment)
  + Imie (varchar)
  + Specjalizacja (varchar)
  + NrLicencji (varchar, unique) – numer licencji
  + Telefon (varchar)
* **Medications**  
  Przechowuje dane o lekach.
  + ID (int, PK, auto\_increment)
  + Nazwa (varchar)
  + Dostepnosc (int) – liczba dostępnych opakowań
  + Producent (varchar)
  + TerminWaznosci (date)
* **Prescriptions**  
  Przechowuje wystawione recepty.
  + ID (int, PK, auto\_increment)
  + KodLeku (int, FK do Medications)
  + Dawka (varchar)
  + Dawkowanie (varchar)
  + PatientsId (int, FK do Patients)
* **Appointments**  
  Przechowuje wizyty pacjentów.
  + ID (int, PK, auto\_increment)
  + Data (datetime) – data i godzina wizyty
  + Diagnoza (varchar)
  + DoctorsID (int, FK do Doctors)
  + PatientsID (int, FK do Patients)
  + Klucz główny złożony: (ID, Data)
  + Partycjonowanie po dacie wizyty
* **LabTest**  
  Przechowuje wyniki badań laboratoryjnych.
  + ID (int, PK, auto\_increment)
  + TypBadania (varchar)
  + Wynik (varchar)
  + Data (datetime)
  + AppointmentsID (int, FK do Appointments)
  + AppointmentsData (datetime, FK do Appointments)
* **MedicalStaff**  
  Przechowuje informacje o personelu medycznym przypisanym do oddziałów.
  + ID (int, PK, auto\_increment)
  + Pielegniarki\_Ilosc (int)
  + Laboranci\_Ilosc (int)
  + TechnicyMedyczni\_Ilosc (int)
  + DepartmentsID (int, FK do Departments)
  1. Kluczowe relacje i mechanizmy**:**
* **Klucze obce** zapewniają spójność danych między pacjentami, lekarzami, oddziałami, lekami, wizytami i badaniami.
* **Partycjonowanie** tabeli Appointments po dacie umożliwia wydajniejsze zarządzanie dużą liczbą wizyt.
* **Indeksy** na kluczach obcych i często używanych kolumnach przyspieszają zapytania.
* **Funkcje i triggery**:
  + Funkcja GetDepartmentID – pobiera ID oddziału na podstawie nazwy.
  + Funkcja AverageTestPerPatient – zwraca średnią liczbę wizyt na pacjenta.
  + Funkcja MostCommonDoc – zwraca lekarza z największą liczbą wizyt.
  + Trigger LessMedicine – automatycznie zmniejsza dostępność leku po wystawieniu recepty.

## Schemat ERD

****

* 1. Opis wybranych plików**:**
* Admin.php – panel administratora (zarządzanie pacjentami, lekarzami, oddziałami, eksport danych, export danych pacjentów do excela i XM:)
* Doctor.php – panel lekarza (przegląd pacjentów, wizyt, badań, wystawianie recept)
* Patient.php – panel pacjenta (przegląd lekarzy, umawianie wizyt, przegląd badań i recept)
* Chemist.php – panel farmaceuty (zarządzanie lekami, eksport leków do XML)
* Index.html - stanowi stronę startową (główną) aplikacji HMS i pełni funkcję przekierowującą użytkownika do odpowiednich paneli

****

# Opis fragmentów kodu źródłowego

Kod źródłowy podzielony jest na pliki odpowiadające poszczególnym rolom i funkcjom systemu.

## Strona Pacjenta (Patient.php i powiązane)

### Funkcja 1: Ustawianie ciasteczka wybranego pacjenta (cookies)

Opis:  
Po wybraniu pacjenta w interfejsie, jego ID jest zapisywane w ciasteczku, co pozwala na personalizację widoku i operacji dla tego pacjenta.

Kod (PatientCookies.php):

<?php

header('Content-Type: application/json');

$data = json\_decode(file\_get\_contents("php://input"), true);

if (!$data || !isset($data['ID'])) {

    echo json\_encode(["status" => "error", "message" => "Nieprawidłowe dane wejściowe."]);

    exit;

}

$cookieName = "PacjentCookie";

$cookieValue = $data['ID'];

if ($cookieValue === "Basic") {

    echo json\_encode(["status" => "error", "message" => "Nie wybrano poprawnego pacjenta."]);

    exit;

}

setcookie($cookieName, $cookieValue, time() + (86400 \* 30), "/");

echo json\_encode(["status" => "success", "message" => "Ciasteczko zostało ustawione.", "cookieValue" => $cookieValue]);

?>

****

### Funkcja 2: Umawianie wizyty przez pacjenta (AJAX + PHP)

Opis:  
Pacjent może wybrać lekarza, datę i godzinę wizyty, a następnie umówić wizytę. Dane są przesyłane AJAX-em do PHP, który zapisuje wizytę w bazie i uniemożliwia innemu pacjentowi umówić się na wizytę w tym samym czasie.

*Kod (PatientUmowWizyte.php):*

<?php

header('Content-Type: application/json');

$serverName = "localhost\MSSQLSERVER01";

$connectionOptions = array(

    "Database" => "HMS",

    "Uid" => "Pacjent",

    "PWD" => "Pacjent1"

);

$conn = sqlsrv\_connect($serverName, $connectionOptions);

if (!$conn) {

    echo json\_encode(["status" => "error", "message" => "Błąd połączenia z bazą danych."]);

    exit;

}

$data = json\_decode(file\_get\_contents("php://input"), true);

if (!$data || !isset($data['doctorId'], $data['data'], $data['godzina'], $data['pacjentId'])) {

    echo json\_encode(["status" => "error", "message" => "Nieprawidłowe dane wejściowe."]);

    exit;

}

$doctorId = $data['doctorId'];

$patientId = $data['pacjentId'];

$dataWizyty = $data['data'];

$GodzinaWizyty = $data['godzina'];

$dateTimeWizyty = new DateTime("$dataWizyty $GodzinaWizyty");

$sql = "INSERT INTO Appointments (Data, DoctorsID, PatientsID) VALUES (?, ?, ?)";

$params = array($dateTimeWizyty, $doctorId, $patientId);

$stmt = sqlsrv\_query($conn, $sql, $params);

if ($stmt === false) {

    echo json\_encode(["status" => "error", "message" => "Błąd podczas zapisywania wizyty."]);

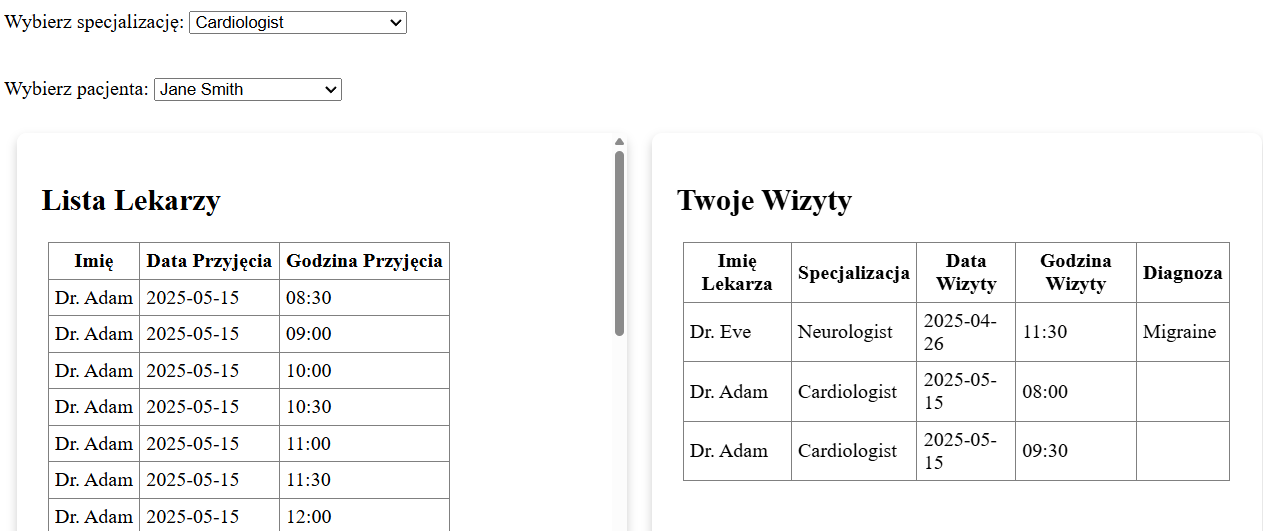
    exit;

}

echo json\_encode(["status" => "success", "message" => "Wizyta została umówiona pomyślnie."]);

sqlsrv\_close($conn);

?>

****

* 1. Strona Lekarza **(Doctor.php i powiązane)**

### Funkcja 1: Dynamiczne ładowanie pacjentów wybranego lekarza

Opis:  
Po wyborze lekarza lista pacjentów jest dynamicznie ładowana AJAX-em, pokazując tylko pacjentów przypisanych do danego lekarza.

*Kod (SendDoctorIDToPatients.js):*

document.getElementById("ListaLekarzy").addEventListener("change", function () {

    var selectedDoctor = this.value;

    const postData = { ID: selectedDoctor };

    var xhr = new XMLHttpRequest();

    document.getElementById("ListaPacjentow").innerHTML = "";

    var option = document.createElement("option");

    option.value = "0";

    option.textContent = "-- Brak Pacjentów --";

    document.getElementById("ListaPacjentow").appendChild(option);

    xhr.open("POST", "Doctor\_Scripts/PatientList.PHP", true);

    xhr.setRequestHeader("Content-Type", "application/json;charset=UTF-8");

    xhr.onload = function () {

        if (xhr.status === 200) {

            try {

                const response = JSON.parse(xhr.responseText);

                if (response.status === "success") {

                    response.data.forEach(function (item) {

                        var option = document.createElement("option");

                        option.value = item.id;

                        option.textContent = item.name;

                        document.getElementById("ListaPacjentow").appendChild(option);

                    });

                }

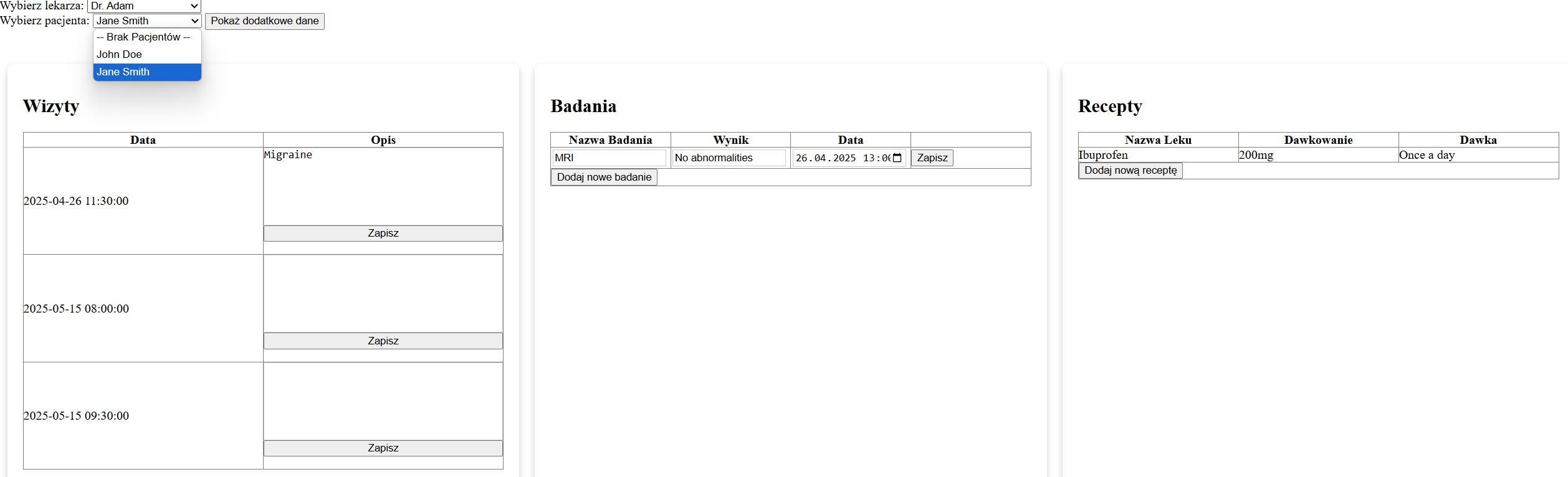
            } catch (e) { }

        }

    };

    xhr.send(JSON.stringify(postData));

});

****

### Funkcja 2: Edycja diagnozy wizyty przez lekarza

Opis:  
Lekarz może edytować diagnozę dla wybranej wizyty pacjenta bez przeładowania strony.

*Kod (fragment PacjenciWizyty.js + WizytyEdit.php):*

// PacjenciWizyty.js (fragment)

document.querySelectorAll(".updateForm").forEach(form => {

    form.addEventListener("submit", function (e) {

        e.preventDefault();

        const formData = new FormData(this);

        fetch('Doctor\_Scripts/WizytyEdit.php', {

            method: 'POST',

            body: formData

        })

            .then(response => response.text())

            .then(data => {

                alert("Dane zostały zaktualizowane pomyślnie!");

            });

    });

});

<?php

// WizytyEdit.php (fragment)

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $Diagnoza = $\_POST['Diagnoza'] ?? null;

    $idWizyty = $\_POST['IDWizyty'] ?? null;

    if ($idWizyty !== null && $Diagnoza !== null) {

        $sql = "UPDATE Appointments SET Diagnoza = ? WHERE ID = ?";

        $params = array($Diagnoza, $idWizyty);

        $stmt = sqlsrv\_query($conn, $sql, $params);

        if ($stmt === false) {

            die(print\_r(sqlsrv\_errors(), true));

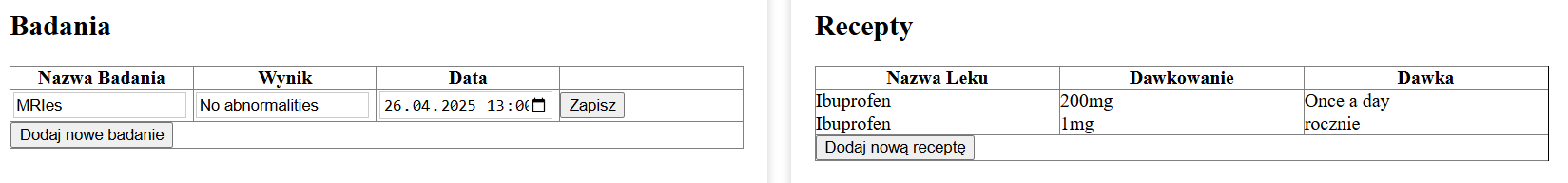
        } else {

            echo "Dane zostały zaktualizowane pomyślnie.";

        }

    }

}

****

## Strona Administratora (Admin.php)

### Funkcja 1: Eksport pacjentów do pliku Excel

Opis:  
Administrator może pobrać listę wszystkich pacjentów w formacie Excel jednym kliknięciem.

*Kod (Admin.php):*

<?php

if (isset($\_POST['download\_excel'])) {

    $serverName = "localhost\\MSSQLSERVER01";

    $connectionOptions = array(

        "Database" => "HMS",

        "Uid" => "administrator",

        "PWD" => "admin1"

    );

    $conn = sqlsrv\_connect($serverName, $connectionOptions);

    $qr = "SELECT Patients.ID, Imie, Nazwisko, Patients.Adres, Telefon, DataUrodzenia, DepartmentsID, Nazwa, Pesel

           FROM [dbo].[Patients]

           LEFT JOIN [dbo].[Departments] ON Patients.DepartmentsID = Departments.ID";

    $result = sqlsrv\_query($conn, $qr);

    header("Content-Type: application/vnd.ms-excel; charset=utf-8");

    header("Content-Disposition: attachment; filename=pacjenci.xls");

    echo "ID\tImie\tNazwisko\tAdres\tTelefon\tDataUrodzenia\tOddzialID\tOddzial\tPesel\n";

    while ($row = sqlsrv\_fetch\_array($result, SQLSRV\_FETCH\_ASSOC)) {

        $row['DataUrodzenia'] = isset($row['DataUrodzenia']) && $row['DataUrodzenia'] instanceof DateTime

            ? $row['DataUrodzenia']->format('Y-m-d')

            : $row['DataUrodzenia'];

        echo $row['ID'] . "\t" . $row['Imie'] . "\t" . $row['Nazwisko'] . "\t" . $row['Adres'] . "\t" .

             $row['Telefon'] . "\t" . $row['DataUrodzenia'] . "\t" . $row['DepartmentsID'] . "\t" .

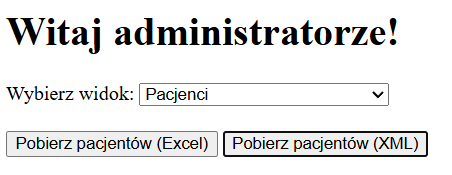
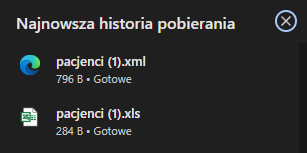
             $row['Nazwa'] . "\t" . $row['Pesel'] . "\n";

    }

    sqlsrv\_close($conn);

    exit;

}

**** ****

### Funkcja 2: Dynamiczne przełączanie widoków (pacjenci, lekarze, oddziały)

Opis:  
Administrator może jednym kliknięciem przełączać się między widokiem pacjentów, lekarzy i oddziałów bez przeładowania strony.

*Kod (Admin.php – fragment JavaScript):*

function switchView() {

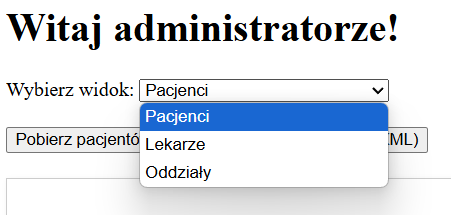
    const view = document.getElementById("viewSelector").value;

    document.getElementById("patientsView").style.display = view === "patients" ? "block" : "none";

    document.getElementById("doctorsView").style.display = view === "doctors" ? "block" : "none";

    document.getElementById("departmentsView").style.display = view === "departments" ? "block" : "none";

}

****

## Strona Farmaceuty (Chemist.php)

### Funkcja 1: Eksport leków do pliku XML

Opis:  
Farmaceuta może pobrać wszystkie dane o lekach w formacie XML.

*Kod (Chemist.php):*

<?php

if (isset($\_GET['download\_xml'])) {

    $serverName = "localhost\\MSSQLSERVER01";

    $connectionOptions = array(

        "Database" => "HMS",

        "Uid" => "Farmaceuta",

        "PWD" => "Farmaceuta1"

    );

    $conn = sqlsrv\_connect($serverName, $connectionOptions);

    $queryallMeds = "SELECT \* FROM Medications";

    $stmtallMeds = sqlsrv\_query($conn, $queryallMeds);

    $medicationsAllData = [];

    while ($row = sqlsrv\_fetch\_array($stmtallMeds, SQLSRV\_FETCH\_ASSOC)) {

        if (isset($row['TerminWaznosci']) && $row['TerminWaznosci'] instanceof DateTime) {

            $row['TerminWaznosci'] = $row['TerminWaznosci']->format('Y-m-d');

        }

        $medicationsAllData[] = $row;

    }

    sqlsrv\_close($conn);

    $xml = new SimpleXMLElement('<Medications/>');

    foreach ($medicationsAllData as $med) {

        $medElem = $xml->addChild('Medication');

        foreach ($med as $key => $value) {

            $medElem->addChild($key, htmlspecialchars($value));

        }

    }

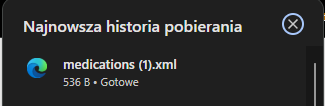
    header('Content-Disposition: attachment; filename="medications.xml"');

    header('Content-Type: application/xml');

    echo $xml->asXML();

    exit;

}

**** ****

### Funkcja 2: Edycja i zarządzanie lekami

Opis:  
Farmaceuta może dodawać, edytować i usuwać leki bezpośrednio z poziomu panelu.

*Kod (Chemist.php – fragment obsługi POST):*

<?php

if ($\_SERVER["REQUEST\_METHOD"] === "POST") {

    $action = $\_POST["action"];

    $id = $\_POST["medicationId"];

    $name = $\_POST["medicationName"];

    $availability = $\_POST["availability"];

    $manufacturer = $\_POST["manufacturer"];

    $expiryDate = $\_POST["expiryDate"];

    if ($action == "update") {

        $sql = "UPDATE [dbo].[Medications]

                SET Nazwa = ?, Dostepnosc = ?, Producent = ?, TerminWaznosci = ?

                WHERE ID = ?";

        $params = array($name, $availability, $manufacturer, $expiryDate, $id);

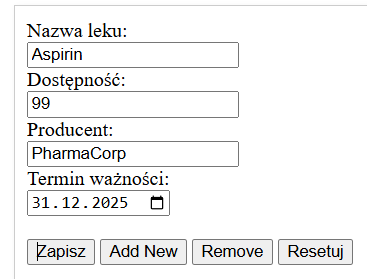
        $stmt = sqlsrv\_query($conn, $sql, $params);

        // ...

    }

    // ...obsługa add i del...

}

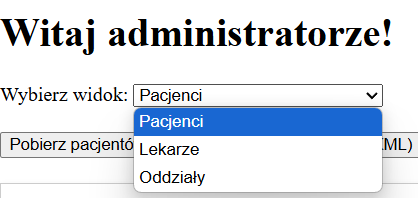
****

# Opis działania poszczególnych ról

## Administrator

### Wybór widoku

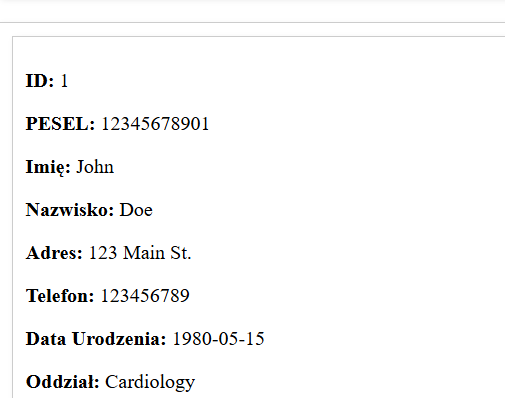
* Na górze strony znajduje się pole **"Wybierz widok"**.
* Możesz przełączać się pomiędzy:
  + Pacjentami
  + Lekarzami
  + Oddziałami

****

* + 1. **Zarządzanie pacjentami**

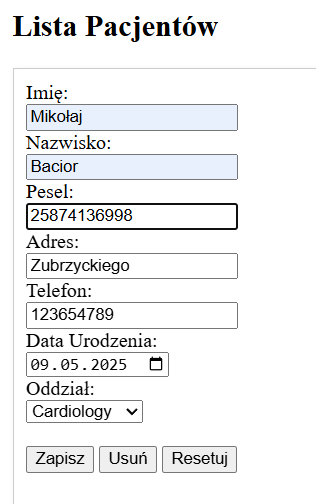
**a) Przeglądanie listy pacjentów**

* Po wybraniu widoku "Pacjenci" zobaczysz listę wszystkich pacjentów.
* Kliknij na wybranego pacjenta, aby załadować jego dane do formularza edycji.

****

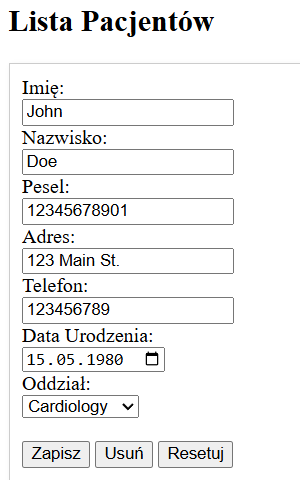
**b) Dodawanie nowego pacjenta**

* Wyczyść formularz (przycisk "Resetuj").
* Wypełnij wszystkie wymagane pola (imię, nazwisko, PESEL, adres, telefon, data urodzenia, oddział).
* Kliknij przycisk **"Zapisz"**.

****

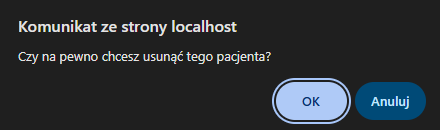
**c) Edycja danych pacjenta**

* Kliknij na pacjenta z listy – jego dane pojawią się w formularzu.
* Wprowadź zmiany.
* Kliknij **"Zapisz"**.



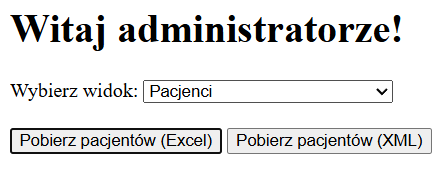
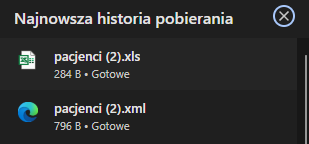
**d) Usuwanie pacjenta**

* Wybierz pacjenta z listy.
* Kliknij **"Usuń"**.
* Potwierdź chęć usunięcia w oknie dialogowym.

****

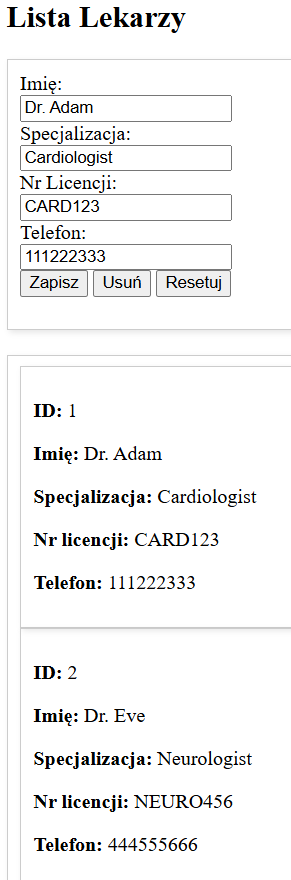
**e) Eksport pacjentów**

* Kliknij **"Pobierz pacjentów (Excel)"** lub **"Pobierz pacjentów (XML)"** aby pobrać listę pacjentów w wybranym formacie.

****

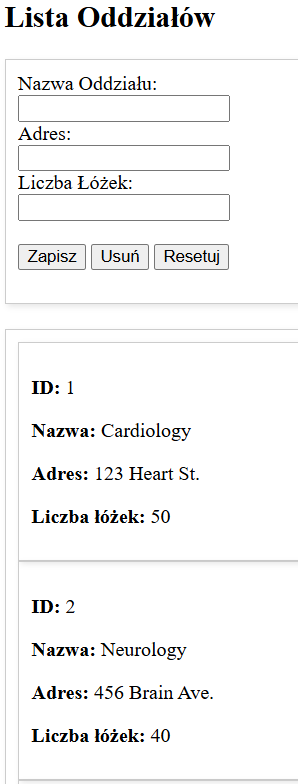
### Zarządzanie lekarzami

* Przełącz widok na "Lekarze".
* Lista lekarzy pojawi się na dole strony.
* Możesz dodawać, edytować i usuwać lekarzy analogicznie jak pacjentów.

****

### Zarządzanie oddziałami

* Przełącz widok na "Oddziały".
* Lista oddziałów pojawi się na dole strony.
* Możesz dodawać, edytować i usuwać oddziały analogicznie jak pacjentów.

****

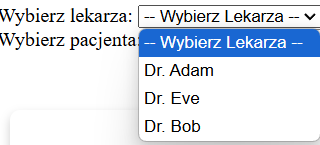
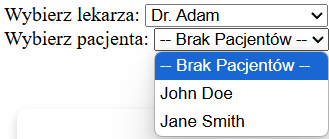
### Dodatkowe wskazówki

* Po każdej operacji (dodanie, edycja, usunięcie) pojawi się komunikat potwierdzający.
* Jeśli wystąpi błąd, sprawdź czy wszystkie pola są poprawnie wypełnione.

## Lekarz

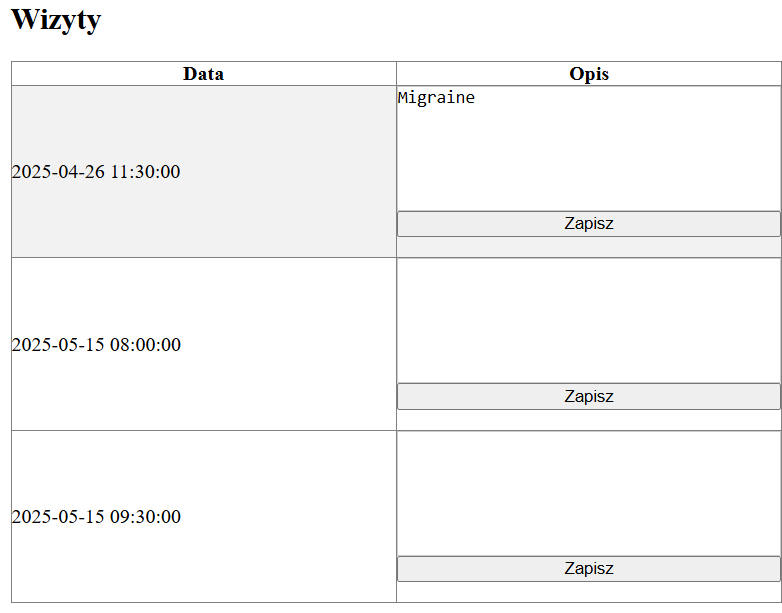
### Wybór lekarza i pacjenta

* Na górze strony wybierz swoje imię z listy **"Wybierz lekarza"**.
* Następnie wybierz pacjenta z listy **"Wybierz pacjenta"** (lista ładuje się dynamicznie po wyborze lekarza).

** **

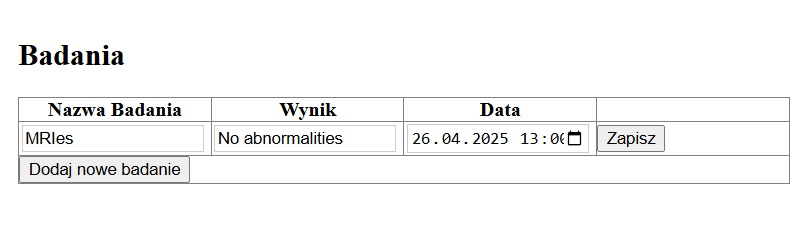
### Przeglądanie i edycja wizyt

* Po wybraniu pacjenta zobaczysz tabelę z jego wizytami.
* Każda wizyta zawiera datę oraz pole do edycji diagnozy.
* Możesz wpisać nową diagnozę i kliknąć **"Zapisz"** – zmiana zostanie zapisana bez przeładowania strony.

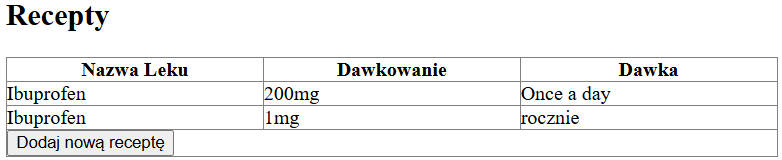
****

### Przeglądanie i edycja badań

* W sekcji **"Badania"** (po wybraniu wizyty) zobaczysz listę badań przypisanych do pacjenta.
* Możesz edytować nazwę, wynik i datę badania, a następnie kliknąć **"Zapisz"**.
* Dodaj nowe badanie klikając **"Dodaj nowe badanie"** i wypełnij formularz w okienku modalnym.

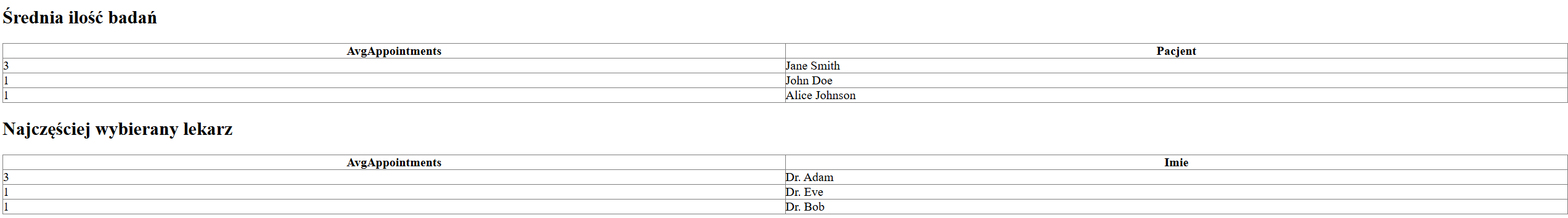
****

* + 1. Przeglądanie i wystawianie recept
* W sekcji **"Recepty"** (po wybraniu wizyty)  znajdziesz listę recept pacjenta.
* Aby dodać nową receptę, kliknij **"Dodaj nową receptę"** – pojawi się okno, w którym wybierasz lek, wpisujesz dawkę i dawkowanie, a następnie klikasz **"Zapisz"**.



### Dodatkowe dane statystyczne

* Kliknij **"Pokaż dodatkowe dane"**, aby zobaczyć:
  + Średnią liczbę badań na pacjenta.
  + Najczęściej wybieranego lekarza.

****

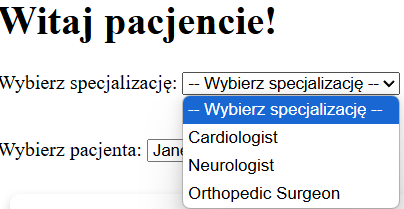
### Wskazówki

* Wszystkie zmiany są zapisywane po kliknięciu odpowiedniego przycisku.
* Jeśli pojawi się komunikat o błędzie, sprawdź czy wszystkie pola są poprawnie

## Pacjent

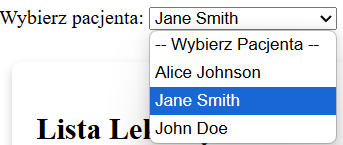
### Wybór specjalizacji i lekarza

* Na górze strony wybierz interesującą Cię **specjalizację** z listy rozwijanej.
* Po wyborze specjalizacji pojawi się lista lekarzy oraz dostępnych terminów wizyt.

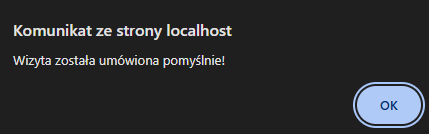
****

### Wybór pacjenta

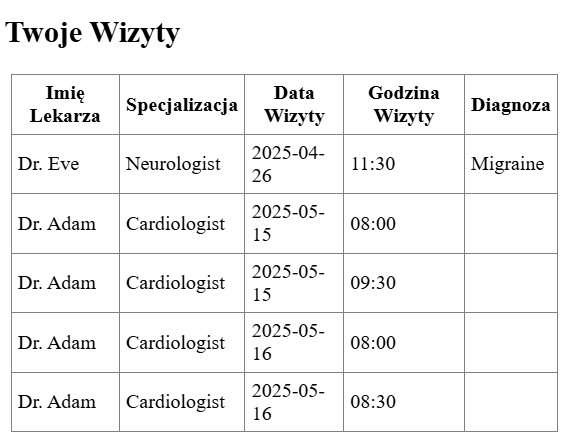
* Wybierz swoje imię i nazwisko z listy **"Wybierz pacjenta"**.
* Po wyborze zostanie ustawione ciasteczko, które pozwoli systemowi rozpoznać Twoje konto.

****

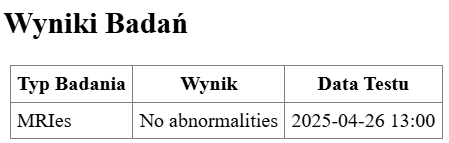
* + 1. Umawianie wizyty
* W tabeli dostępnych terminów kliknij wybrany wiersz (lekarz, data, godzina).
* System automatycznie wyśle żądanie i umówi wizytę na wybrany termin.
* Po sukcesie pojawi się komunikat potwierdzający.

****

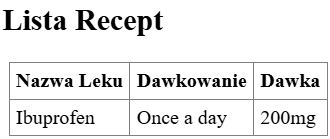
* + 1. Przeglądanie swoich wizyt
* W sekcji **"Twoje Wizyty"** zobaczysz listę wszystkich swoich wizyt.
* Kliknij wybraną wizytę, aby zobaczyć szczegóły badań lub recept.

****

* + 1. Przeglądanie wyników badań
* Po kliknięciu wizyty, w sekcji **"Wyniki Badań"** pojawią się szczegóły badań laboratoryjnych przypisanych do tej wizyty.

****

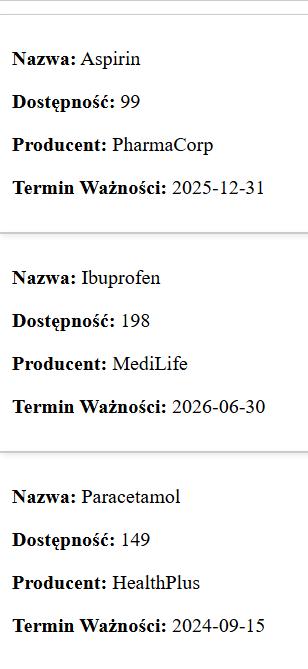
* + 1. Przeglądanie recept
* W sekcji **"Lista Recept"** znajdziesz wszystkie recepty przypisane do Twojego konta.
* Lista jest ładowana automatycznie po wejściu na stronę lub po wybraniu pacjenta.

****

* + 1. Wskazówki
* Jeśli nie widzisz swoich wizyt, upewnij się, że wybrałeś poprawnego pacjenta z listy.
* Po każdej operacji (umówienie wizyty, wybór pacjenta) pojawi się odpowiedni komunikat.

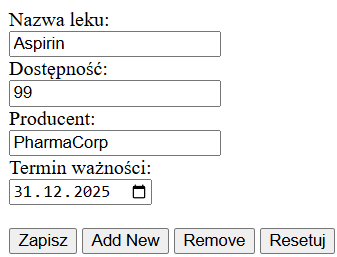
## Farmaceuta

* + 1. Przeglądanie listy leków
* Po wejściu do panelu zobaczysz listę wszystkich leków dostępnych w systemie.
* Każdy lek wyświetla nazwę, dostępność, producenta oraz termin ważności.

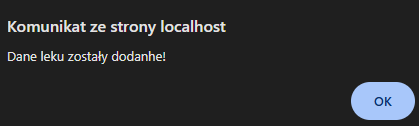
****

### Edycja danych leku

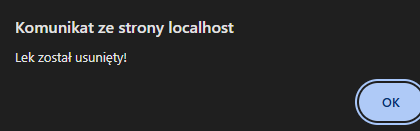
* Kliknij na wybrany lek z listy – jego dane pojawią się w formularzu po prawej stronie.
* Zmień nazwę, dostępność, producenta lub termin ważności.
* Kliknij przycisk **"Zapisz"** – dane zostaną zaktualizowane w bazie.

****

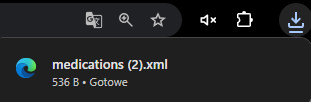
* + 1. Dodawanie nowego leku
* Wyczyść formularz (przycisk **"Resetuj"**).
* Wprowadź dane nowego leku.
* Kliknij przycisk **"Add New"** – lek zostanie dodany do bazy.



* + 1. Usuwanie leku
* Wybierz lek z listy.
* Kliknij przycisk **"Remove"** – lek zostanie usunięty z bazy.

****

* + 1. Eksport leków do pliku XML
* Kliknij przycisk **"Pobierz leki jako XML"** nad listą leków.
* System wygeneruje i pobierze plik medications.xml ze wszystkimi lekami.

****

### Wskazówki

* Po każdej operacji (dodanie, edycja, usunięcie) pojawi się komunikat potwierdzający.
* Jeśli pojawi się błąd, sprawdź czy wszystkie pola są poprawnie wypełnione.

# Podsumowanie

Projekt HMS to rozbudowany system zarządzania szpitalem, umożliwiający obsługę pacjentów, lekarzy, farmaceutów i administratorów.  
System pozwala na zarządzanie danymi, umawianie wizyt, wystawianie recept, przeglądanie badań oraz eksport danych do plików Excel i XML.  
Kod jest modularny, czytelny i łatwy do rozbudowy.  
Możliwe kierunki rozwoju to:

* Dodanie powiadomień e-mail
* Wdrożenie testów jednostkowych
* Rozszerzenie uprawnień i widoków
* Dodanie opcji logowania się do poszczególnych ról
* Utworzenie ekranów logowania dla poszczególnych pacjentów