

Przychodnia Lekarska

Autorzy: Mateusz Osiński (projekt wykonany samodzielnie)

Baza danych

Baza danych składa się z trzech tabel

- lekarze,
- pacjenci,
- wizyty,

Gdzie tabela `wizyty` jest tabelą pośredniczącą między tabelami `lekarze` i `pacjenci` (potrzebna do relacji many-to-many). Diagram:

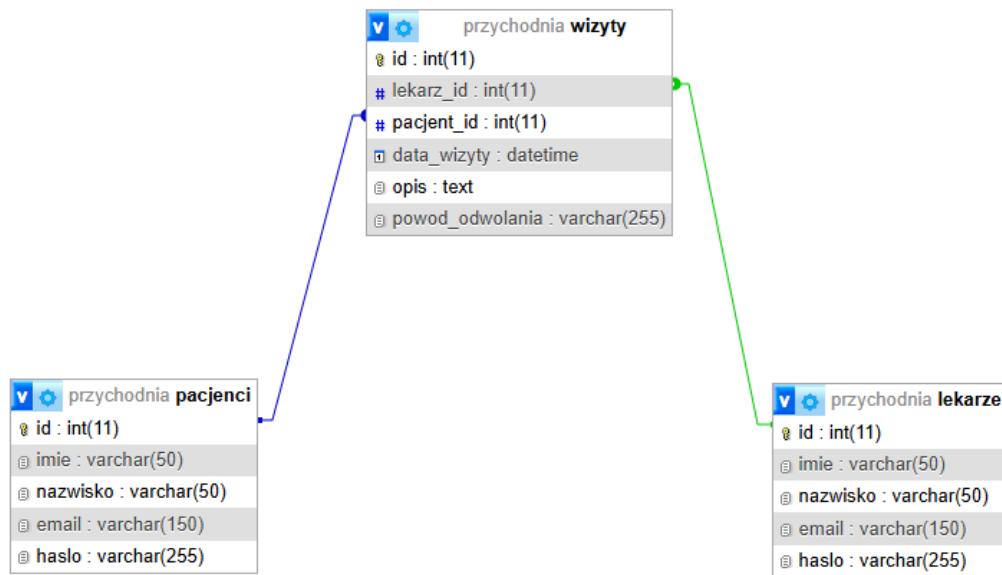


Tabela `lekarze`:

Tabela służąca do przechowywania danych lekarza. Składa się z takich pól jak imię (varchar), nazwisko (varchar), email (varchar) i hasło (varchar).

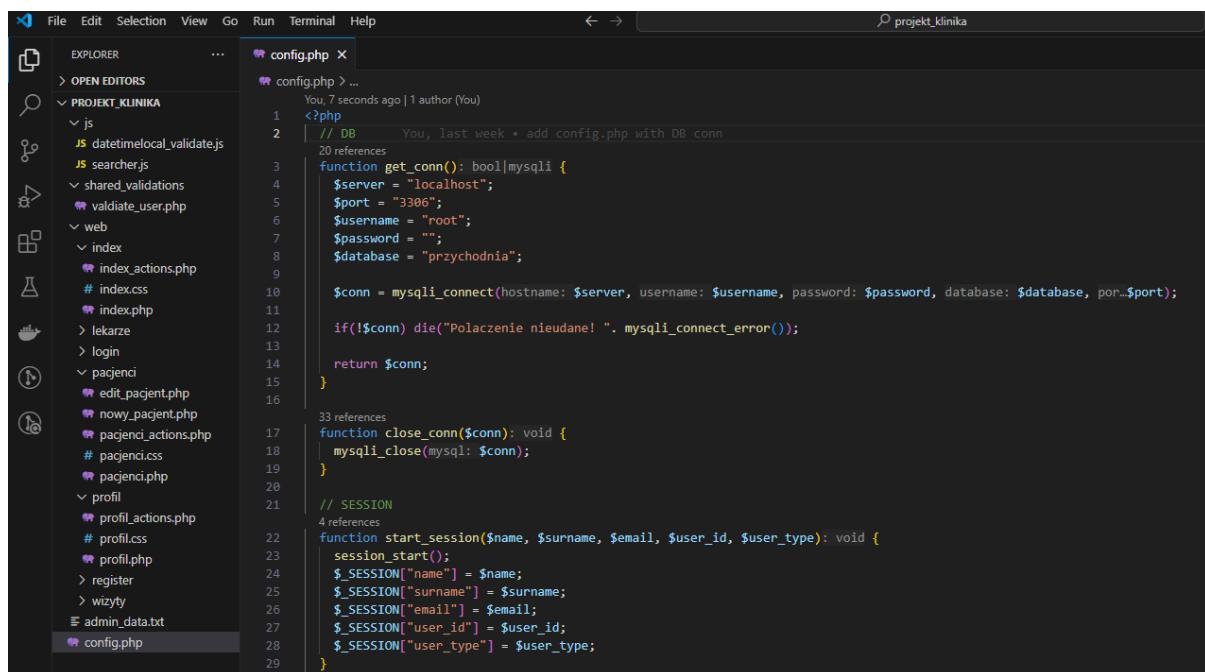
Tabela `pacjenci`:

Tabela służąca do przechowywania danych pacjenta. Składa się z takich pól jak imię (varchar), nazwisko (varchar), email (varchar) i hasło (varchar).

Tabela `wizyty`:

Tabela służąca do przechowywania danych odnośnie wizyty (tabela pośrednicząca w relacji many-to-many). Składa się z takich pól jak lekarz_id (id lekarza), pacjent_id (id pacjenta), data_wizyty (datetime), opis_wizyty (text) oraz powo_odwołania (varchar).

Połączenie z bazą jest zdefiniowane w pliku `config.php` w funkcji `get_conn`:



```
File Edit Selection View Go Run Terminal Help 🔍 projekt_klinika

EXPLORER OPEN EDITORS
PROJEKT_KLINIKA
  js
    datetimelocal_validate.js
    searcher.js
  shared_validations
    validate_user.php
  web
    index
      index_actions.php
      index.css
      index.php
    lekarze
    login
    pacjenci
      edit_pacjent.php
      nowy_pacjent.php
      pacjenci_actions.php
      pacjenci.css
      pacjenci.php
    profil
      profil_actions.php
      profil.css
      profil.php
  register
  wizyty
  admin_data.txt
  config.php

config.php X
config.php > ...
You, 7 seconds ago | 1 author (You)
1 <?php
2 // DB
3 function get_conn(): bool|mysqli {
4   $server = "localhost";
5   $port = "3306";
6   $username = "root";
7   $password = "";
8   $database = "przychodnia";
9
10 $conn = mysqli_connect(hostname: $server, username: $username, password: $password, database: $database, port: $port);
11
12 if(!$conn) die("Połączenie nieudane! ". mysqli_connect_error());
13
14 return $conn;
15 }
16
17 function close_conn($conn): void {
18   mysqli_close(mysql: $conn);
19 }
20
21 // SESSION
22 function start_session($name, $surname, $email, $user_id, $user_type): void {
23   session_start();
24   $_SESSION["name"] = $name;
25   $_SESSION["surname"] = $surname;
26   $_SESSION["email"] = $email;
27   $_SESSION["user_id"] = $user_id;
28   $_SESSION["user_type"] = $user_type;
29 }
```

Pliki

W aplikacji znajduje się jeden plik konfiguracyjny CSV, o nazwie `admin_data.txt`, którego zawartość jest następująca:

admin@przychodnia.com;password12345;mateusz;osinski

Gdzie:

admin@przychodnia.com jest emailem administratora,
password12345 jest hasłem administratora,
mateusz to imię administratora,
osinski to nazwisko administratora.

Plik ten jest używany w dwóch momentach:

1. Logowanie administratora do systemu (system sprawdza, czy podane hasło jest takie samo jak zapisane w pliku),
2. Przy zmianie hasła w profilu administratora.

Podsumowanie zarządzania danymi:

Dane administratora są przechowywane w pliku (plik ala konfiguracyjny),

Dane lekarzy, pacjentów i wizyt są przechowywane w bazie danych.

Przykładowe dane do logowania:

Pacjent: email: ram@gmail.com, hasło: password12345

Lekarz: email: pat@przychodnia.com, hasło: password12345

Administrator: email: admin@przychodnia.com, hasło: password12345

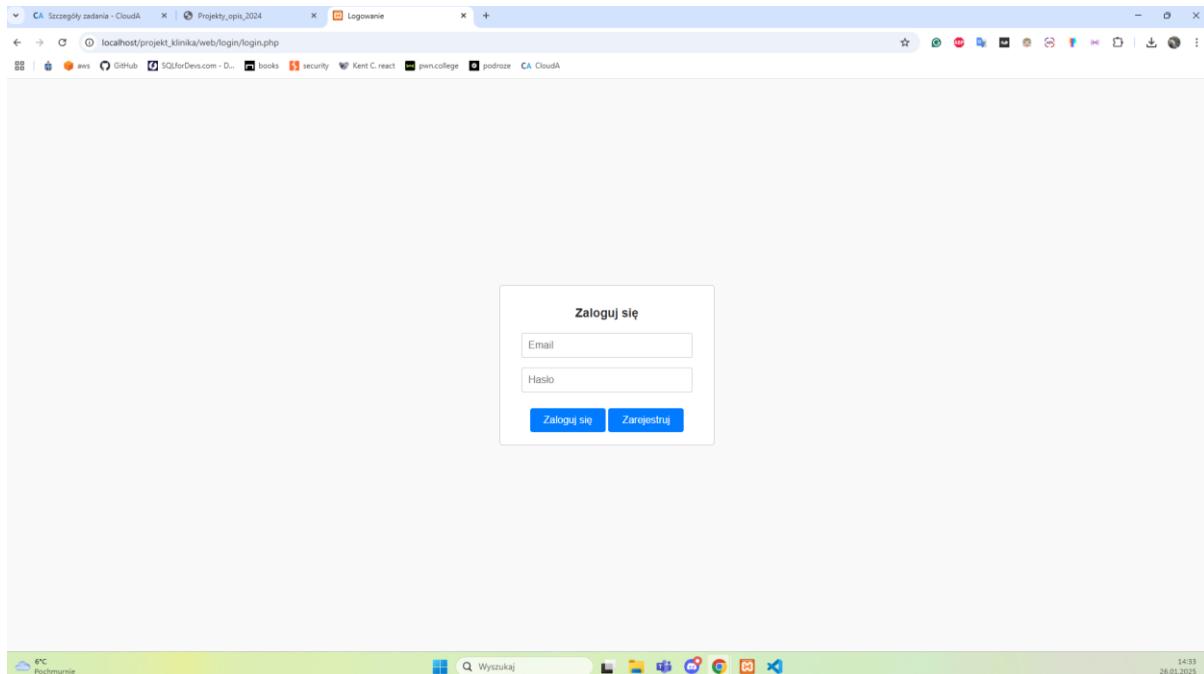
Domena przychodnia.com jest zarezerwowana dla lekarzy i administratora.

Rejestrowanie jest tylko i wyłącznie dla pacjentów. Lekarze są tworzeni przez administratora. Wszyscy mogą zalogować się przez tą samą stronę logowania (wszystko opisane poniżej za pomocą screen-shotów).

Strony i poruszanie się po aplikacji

Login page:

Punktem startowym aplikacji jest `login` page, który w moim przypadku znajduje się pod adresem http://localhost/projekt_klinika/web/login/login.php



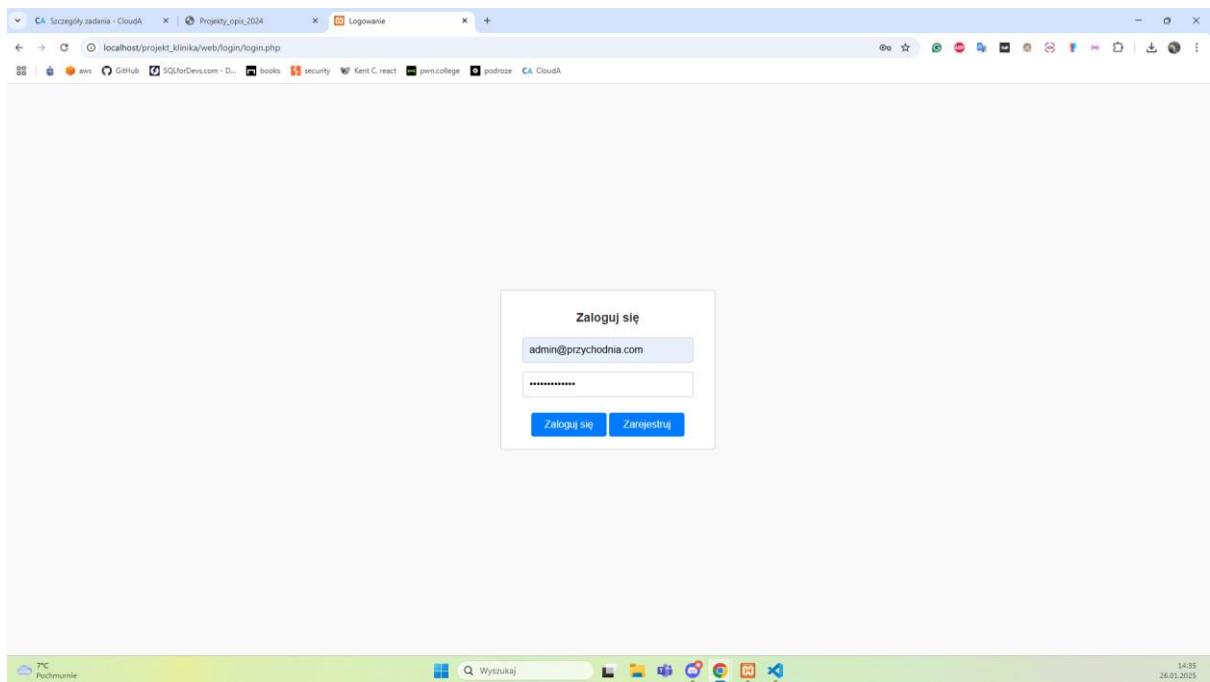
W tym miejscu podajemy dane do zalogowania - jako administrator, pacjent lub lekarz.

Przykładowane logowania:

Jako administrator:

Email: administrator@przychodnia.com

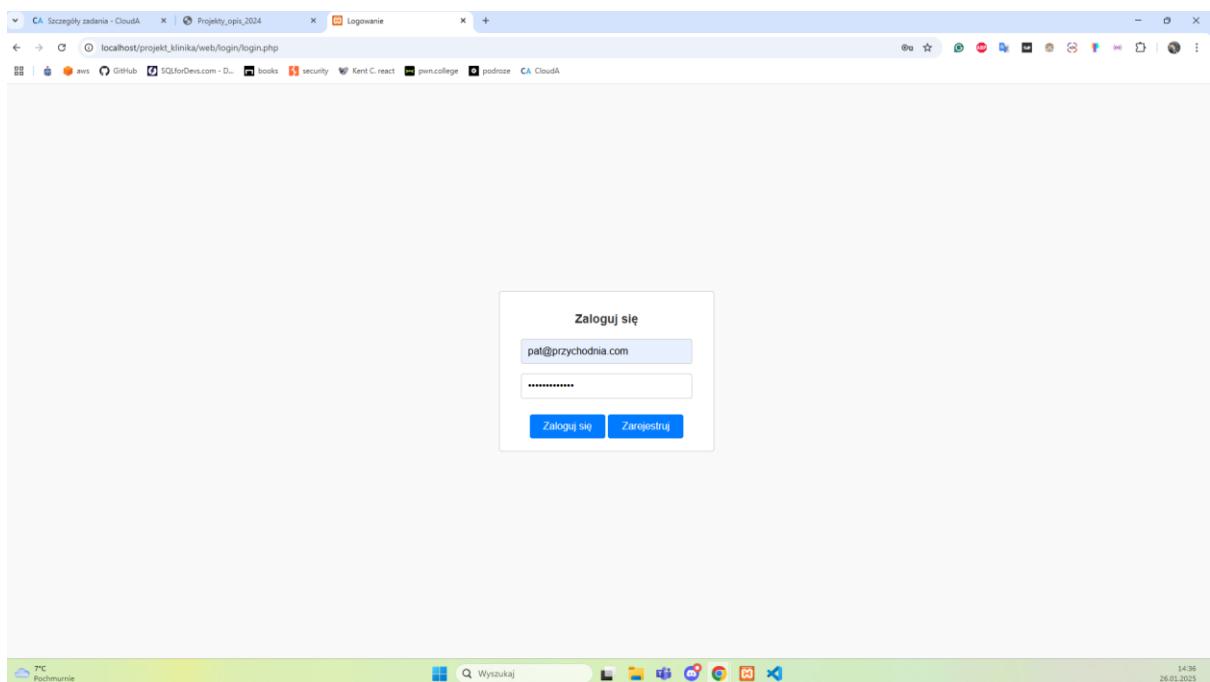
Hasło: password12345 (chyba, że zostało zmienione w pliku `admin_data.txt`)



Jako lekarz:

Email: pat@przychodnia.com

Hasło: password12345

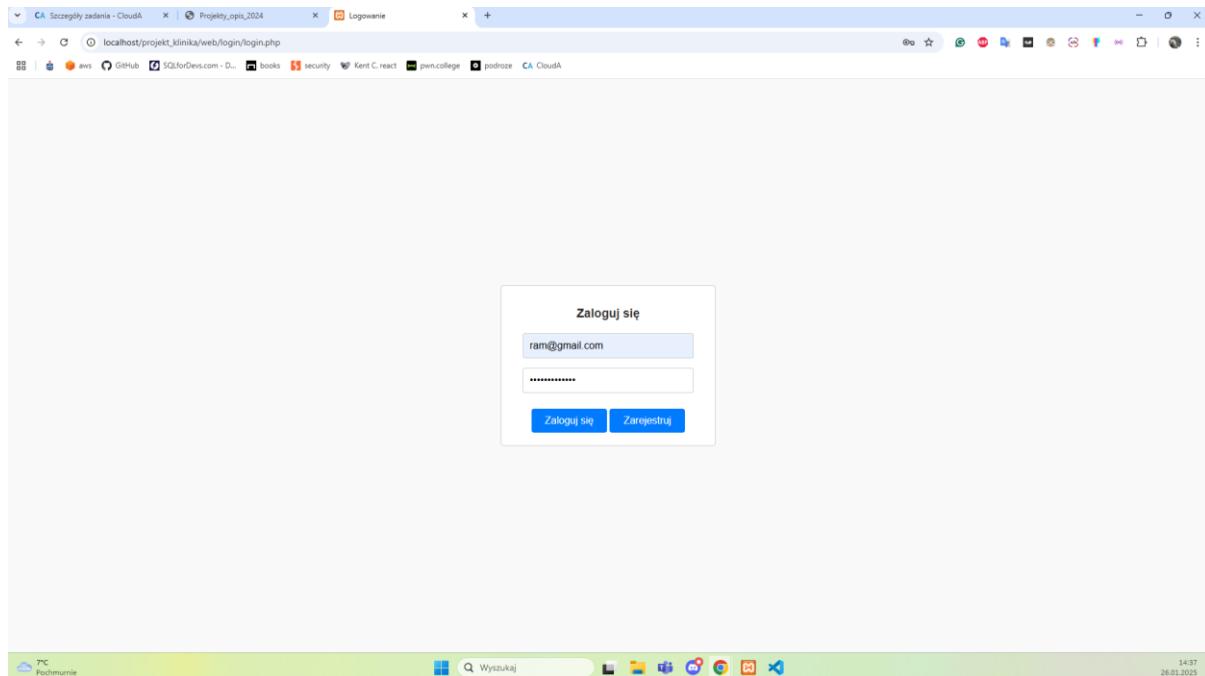


Wszyscy lekarze mają domenę przychodnia.com w mailu!

Jako pacjent:

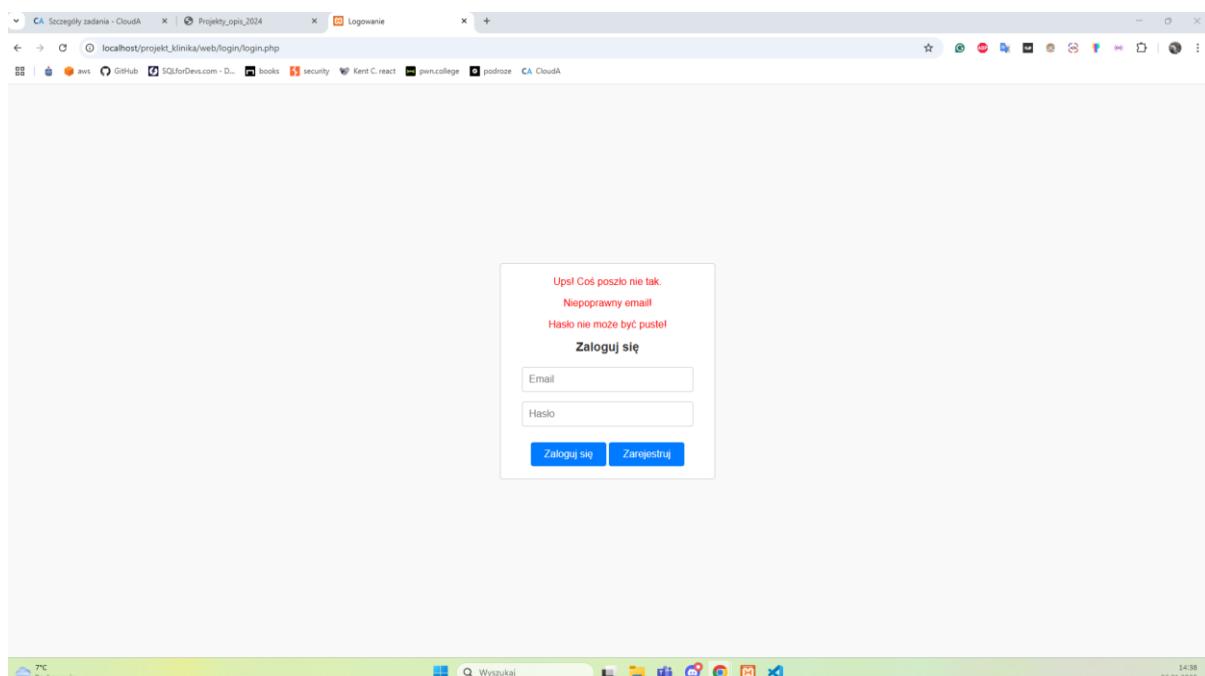
Email: ram@gmail.com

Hasło: password12345

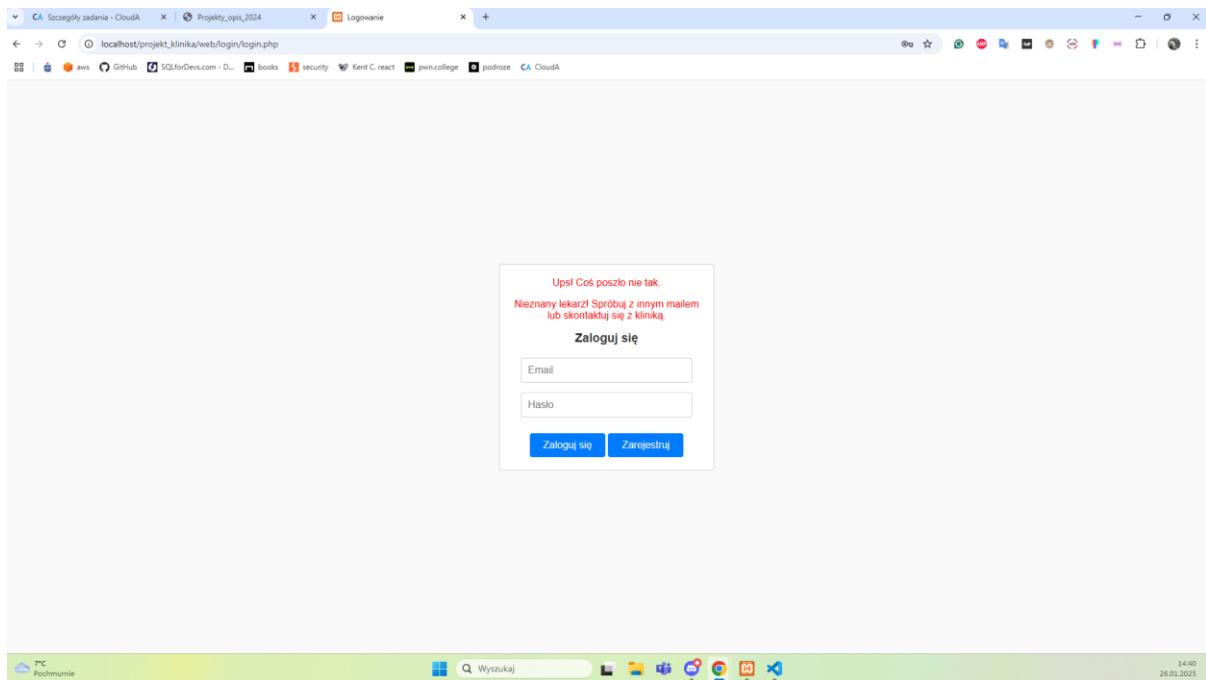


Przykładowe błędy logowania:

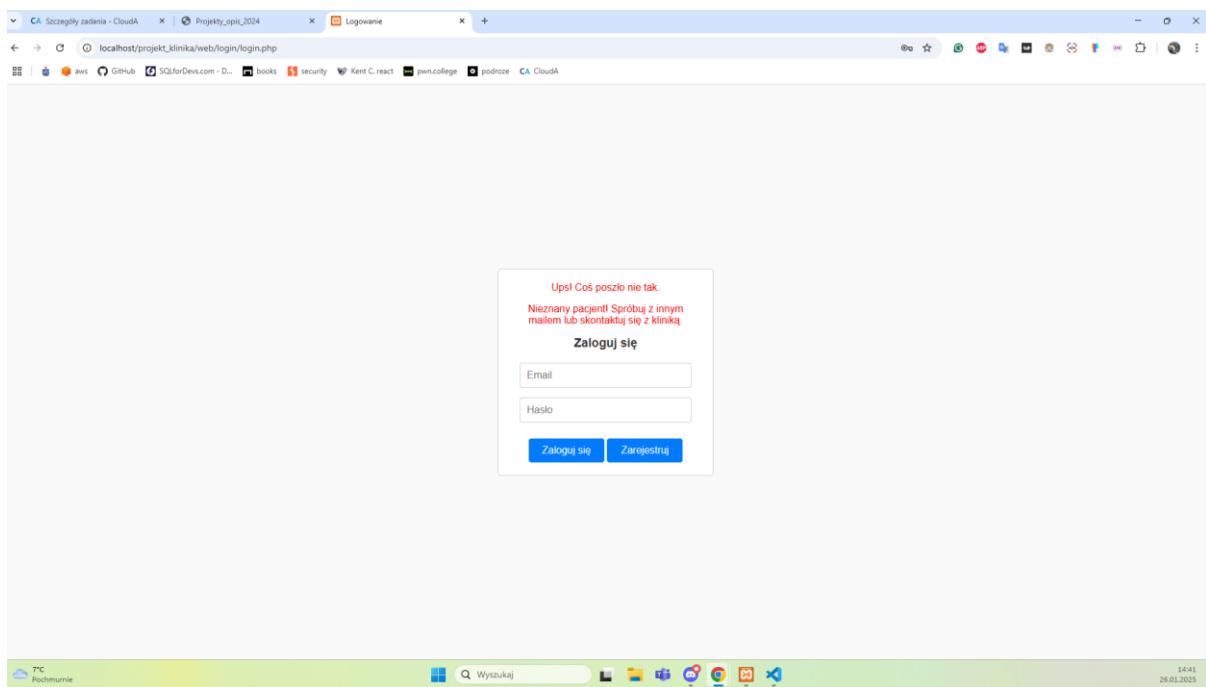
Gdy użytkownik nie poda danych:



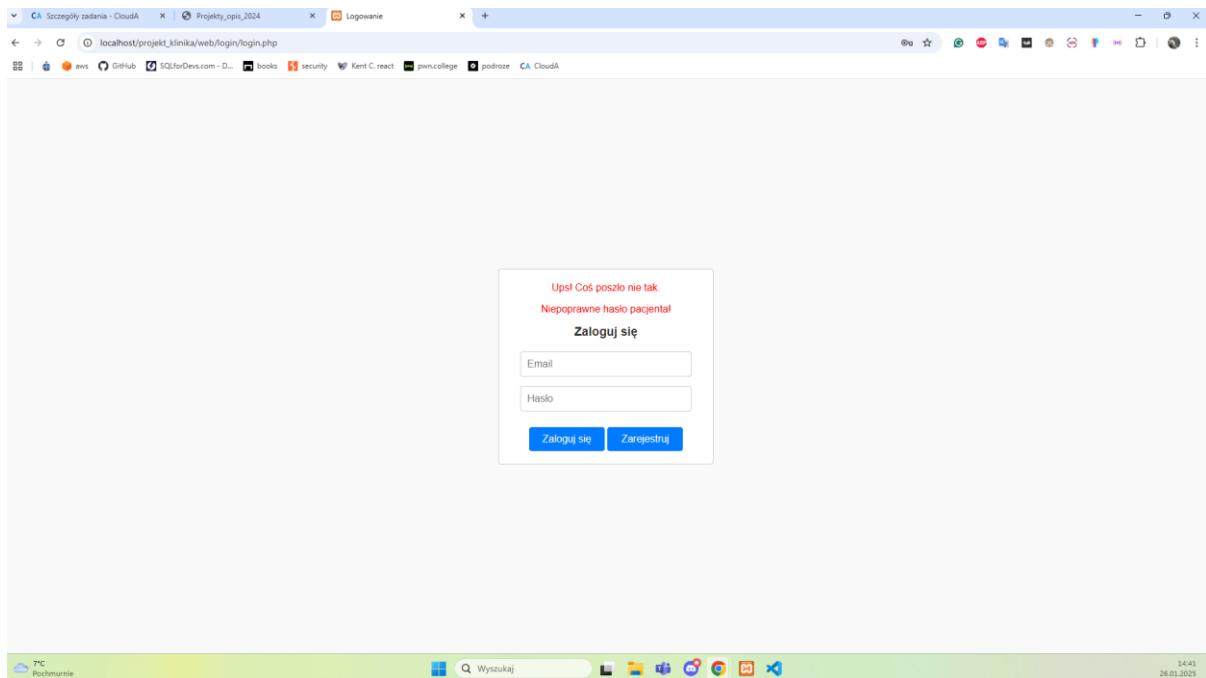
Dla nieistniejącego lekarza (email: nieistnieje@przychodnia.com)



Dla nieistniejącego pacjenta (email: nieistnieje@gmail.com)



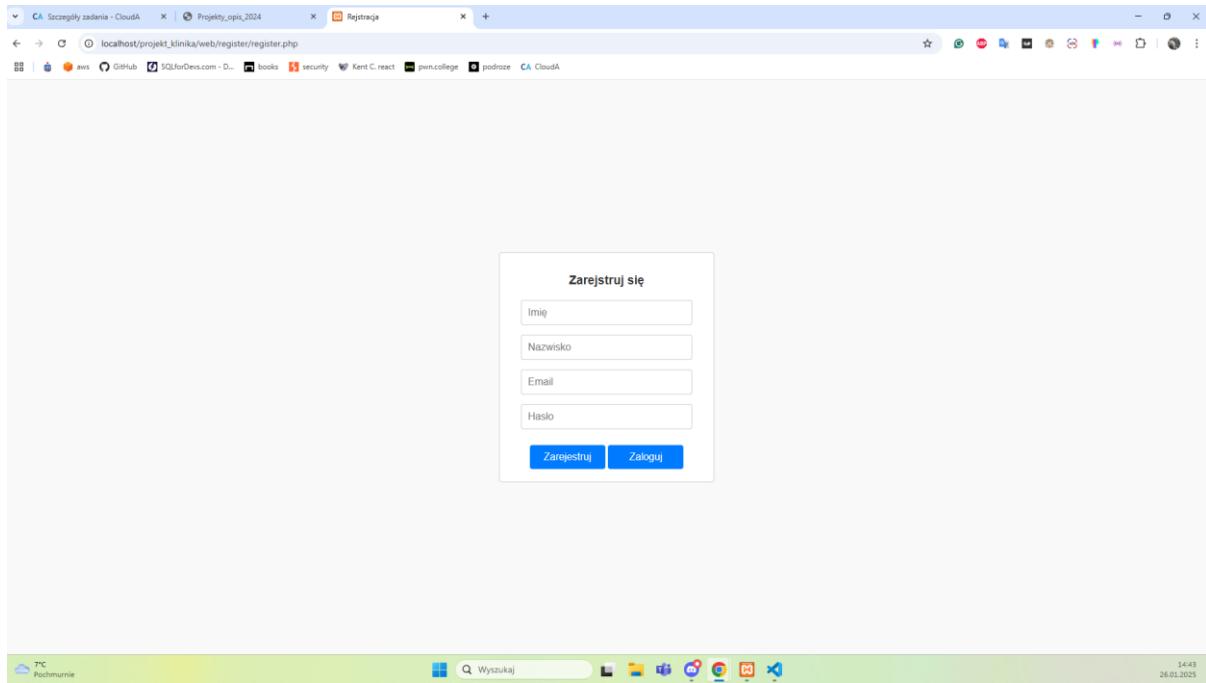
Dla niepoprawnego hasła pacjenta:



Register page

Ze strony logowania możemy przejść na stronę rejestracji.

Co ważne - ta strona służy tylko i wyłącznie dla pacjentów! Lekarze dodawni są przez administratora w jego panelu.



Przykładowe błędy rejestracji:

Nie podanie żadnych danych:

The screenshot shows a registration form titled "Zarejestruj się". The form fields are empty, and several validation errors are displayed in red text:

- Ups! Coś poszło nie tak.
- Imię nie może być puste
- Nazwisko nie może być puste
- Niepoprawny adres email
- Niepoprawne hasło. Powinno składać się z min 5 znaków.

Below the form are two buttons: "Zarejestruj" and "Zaloguj".

Próba założenia maila na domenę przychodni:

The screenshot shows a registration form titled "Zarejestruj się". The "Email" field contains an invalid email address, resulting in a validation error in red text:

- Niepoprawny adres email

Below the form are two buttons: "Zarejestruj" and "Zaloguj".

(specjalnie nie ma za dużo kontekstu do errora, żeby użytkownik nie znam maila przez którego logują się lekarze i administrator).

Przy udanym logowaniu/rejestracji użytkownik jest przekierowany na index pagea, a jego dane zapisywane są w sesji, np.

```
| reference
function login_user($email, $password): void {
    $conn = get_conn();
    $user_query = "SELECT * FROM pacjenci WHERE email = '$email';";
    $user_result = mysqli_query(mysql: $conn, query: $user_query);

    if(mysqli_num_rows(result: $user_result) === 1) {
        $user = mysqli_fetch_assoc(result: $user_result);
        close_conn(conn: $conn);

        if(password_verify(password: $password, hash: $user["haslo"])) {
            start_session(name: $user["imie"], surname: $user["nazwisko"], email: $email, user_id: $user['id'], user_type: "pacjent");
            header(header: "Location: ../index/index.php");
            exit;
        } else {
            global $errors;
            array_push(array: &$errors, values: "Niepoprawne hasło pacjenta!");
        }
    } else {
        global $errors;
        close_conn(conn: $conn);
        array_push(array: &$errors, values: "Nieznany pacjent! Spróbuj z innym mailiem lub skontaktuj się z kliniką.");
    }
}
```

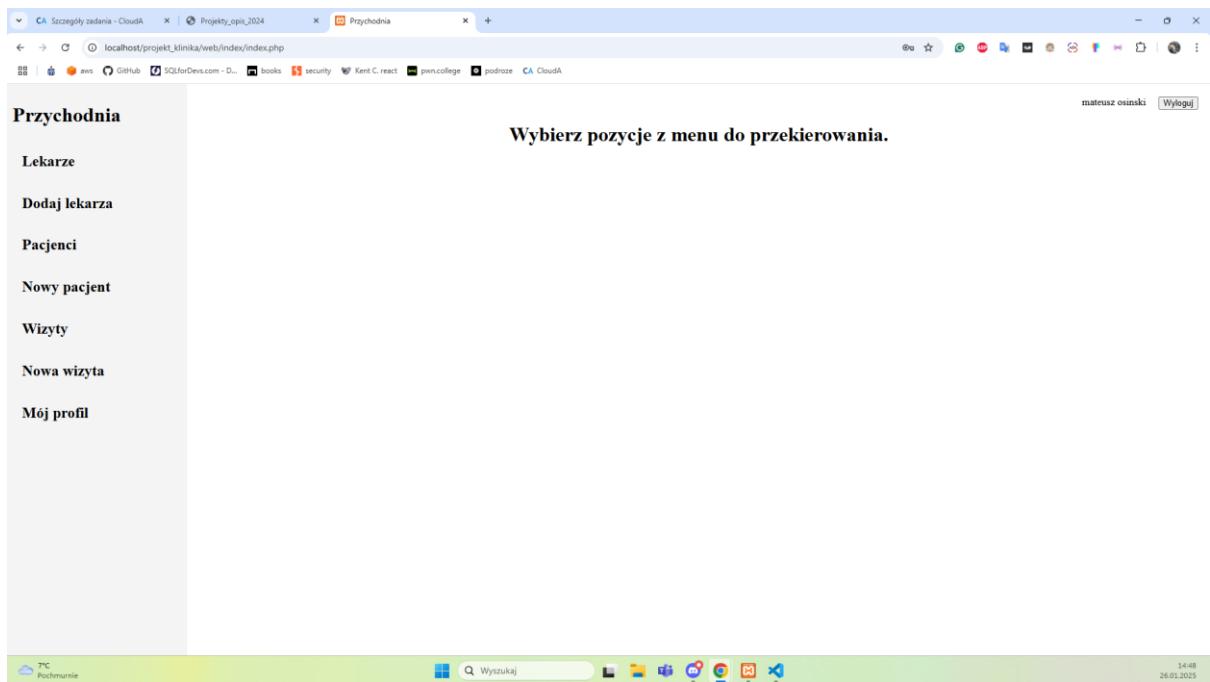
Funckja `start_session` jest zdefiniowana w pliku `config.php`:

```
You, 18 hours ago | 1 author (You)
1 ?php You, last week • add config.php with DB conn
2 // DB
3 20 references
4 function get_conn(): bool|mysqli {
5     $server = "localhost";
6     $port = "3307";
7     $username = "root";
8     $password = "";
9     $database = "przychodnia";
10
11     $conn = mysqli_connect(hostname: $server, username: $username, password: $password, database: $database, port:$port);
12
13     if(!$conn) die("Polaczenie nieudane! ". mysqli_connect_error());
14
15     return $conn;
16 }
17
18 33 references
19 function close_conn($conn): void {
20     mysqli_close(mysqli: $conn);
21 }
22
23 // SESSION
24 4 references
25 function start_session($name, $surname, $email, $user_id, $user_type): void {
26     session_start();
27     $_SESSION["name"] = $name;
28     $_SESSION["surname"] = $surname;
29     $_SESSION["email"] = $email;
30     $_SESSION["user_id"] = $user_id;
31     $_SESSION["user_type"] = $user_type;
32 }
33
34 1 reference
35 function stop_session(): void {
36     session_start();
37     session_unset();
38     session_destroy();
39 }
40
41 19 references
42 function get_session_property($key): mixed {
43     if(session_status() !== PHP_SESSION_ACTIVE) session_start();
44     return $_SESSION[$key];
45 }
46 ?>
```

Który jest również odpowiedzialny za dostarczanie innych funkcjonalności, tj. łączenie z bazą danych, czytanie wartości z sesji, czy stopowaniu sessji (przy wylogowaniu użytkownika). Ten plik dostarcza nam po prostu generyczne funkcje, dzięki czemu nie musimy powtarzać kodu w aplikacji. (np. pisać wszędzie start_session(), żeby zfetchować wartości z sessji).

Index page

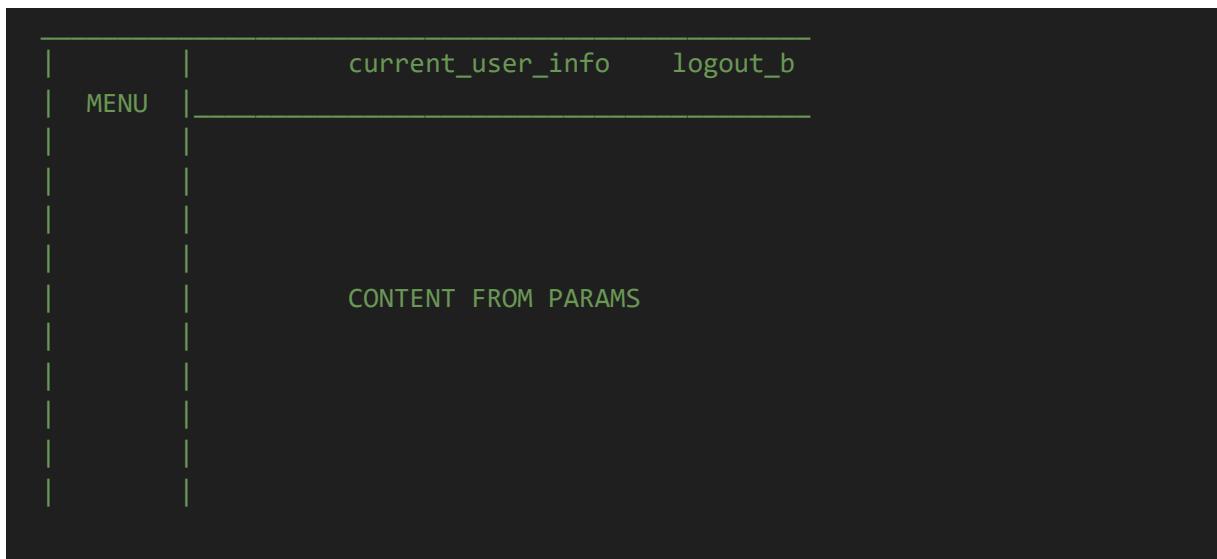
Przejdę na główną stronę jako admin (zaloguję się przy pomocy admin@przychodnia.com) - później pokażę widoki lekarza i pacjenta.



Po poprawnym zalogowaniu zostajemy przekierowani na index page, który znajduje się pod adresem http://localhost/projekt_klinika/web/index/index.php.

Index page jest głównym “mózgiem” całej aplikacji, ponieważ działa on jak layout.

Tak wygląda layout:



Menu i górny navbar są statyczne, natomiast środek jest dynamiczny.

Wygląda to w ten sposób w kodzie:

```

index.php
7 <html>
8 <head>
13   <link rel="stylesheet" href="index.css">
14   <link rel="stylesheet" href="./lekarze/lekarze.css">
15   <link rel="stylesheet" href="./pacjenci/pacjenci.css">
16   <link rel="stylesheet" href="./wizyty/wizyta.css">
17   <link rel="stylesheet" href="./profil/profil.css">
18   <script src="../../js/searcher.js">/script>
19   <script src="../../js/datetimelocal_validate.js">/script>
20 </head>
21 <body>
22   <div class="index_page">
23     <!-- MENU -->
24     <div class="index_menu">
25       <h1>Przychodnia</h1>
26       <div class="index_menu_items">
27         <?php echo get_menu_items() ?>
28       </div>
29     </div>
30   <!-- CONTENT -->
31   <div class="index_content">
32     <div class="index_logout">
33       <span class="index_fullname"><?php echo get_full_name(); ?></span>
34       <form method="POST" action="index_actions.php">
35         <button type="submit" name="log_out">Wyloguj</button>
36       </form>
37     </div>
38   </div>
39   <?php render_content_from_params() ?>
40   </div>
41   You, 5 days ago * define layout(index) page (vip for menu)
42 </div>
43 </body>
44 </html>

```

```

index_actions.php
89
90
91
92
93
94
95
96
97
98
99
100 Thanks to this approach, we have the same layout on each sub-page.
101 (This makes "index.php" a root file when user is logged in :))
102 */
103 function render_content_from_params(): mixed {
104   $page = $_GET["page"] ?? '';
105
106   if ($page === "wizyty") return require("../wizyty/wizyta.php");
107   if ($page === "nowa_wizyta") return require("../wizyty/nova_wizyta.php");
108   if ($page === "edolaj_wizyte") return require("../wizyty/obnajaj_wizyte.php");
109   if ($page === "edit_wizyte") return require("../wizyty/edit_wizyta.php");
110   if ($page === "lekarze") return require("../lekarze/lekarze.php");
111   if ($page === "nowy_lekarz") return require("../lekarze/novy_lekarz.php");
112   if ($page === "edit_lekarz") return require("../lekarze/edit_lekarz.php");
113   if ($page === "pacjenci") return require("../pacjenci/pacjenci.php");
114   if ($page === "nowy_pacjent") return require("../pacjenci/novy_pacjent.php");
115   if ($page === "edit_pacjent") return require("../pacjenci/edit_pacjent.php");
116   if ($page === "profil") return require("../profil/profil.php");
117
118   echo <>INDEX_PAGE
119   <h1>Wybierz pozycje z menu do przekierowania.</h1>
120
121 }
122

```

Funkcja `render_content_from_params` sprawdza czy w URL znajduje się parametr `page`, i dalej na jego podstawie dynamicznie renderuje dany HTML z danego pliku lub wyświetla po prostu napis, żeby użytkownik wybrał opcję z menu. (taki mini router).

Parametr `page` jest dodawany do URLa w momencie, gdy użytkownik wybierze jakąś opcję z menu. Tak wygląda generowanie menu:

```

1 reference
function get_admin_menu_items(): string {
    return <<<ADMIN_MENU_ITEMS
        <h2><a href="?page=lekarze">Lekarze</a></h2>
        <h2><a href="?page=nowy_lekarz">Dodaj lekarza</a></h2>
        <h2><a href="?page=pacjenci">Pacjenci</a></h2>
        <h2><a href="?page=nowy_pacjent">Nowy pacjent</a></h2>
        <h2><a href="?page=wizyty">Wizyty</a></h2>
        <h2><a href="?page=nowa_wizyta">Nowa wizyta</a></h2>
        <h2><a href="?page=profil">Mój profil</a></h2>
    ADMIN_MENU_ITEMS;
}

1 reference
function get_pacjent_menu_items(): string {
    return <<<PACJENT_MENU_ITEMS
        <h2><a href="?page=wizyty">Wizyty</a></h2>
        <h2><a href="?page=nowa_wizyta">Zarejestruj wizytę</a></h2>
        <h2><a href="?page=profil">Mój profil</a></h2>
    PACJENT_MENU_ITEMS;
}

1 reference
function get_lekarz_menu_items(): string {
    return <<<LEKARZ_MENU_ITEMS
        <h2><a href="?page=pacjenci">Pacjenci</a></h2>
        <h2><a href="?page=wizyty">Wizyty</a></h2>
        <h2><a href="?page=profil">Mój profil</a></h2>
    LEKARZ_MENU_ITEMS;
}

```

Za każdym razem jak użytkownik wybierze opcję z menu, to do URLa zostanie dopisany parametr `page` , który później jest używany w funkcji `render_content_from_params()` .

Co również ważne, to menu jest dynamicznie, tzn. Zależne od aktualnie zalogowanego użytkownika:

```

File Edit Selection View Go Run Terminal Help
index.php x
web > index > index.php
1 index.php
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

<html>
<head>
</head>
<body>
<div class="index_page">
<!-- MENU -->
<div class="index_menu">
<h1>Przychodnia</h1>
<div class="index_menu_items">
| <?php echo get_menu_items() ?> You, 2 days ago + define menus
</div>
<!-- CONTENT -->
<div class="index_content">
<div class="index_logout">
<span class="index_fullname">><?php echo get_full_name(); ?></span>
<form method="POST" action="index_actions.php">
| <button type="submit" name="log_out">Wyloguj</button>
</form>
<div>
| <?php render_content_from_params() ?>
</div>
</div>
</div>
</body>
</html>

```

```

index_actions.php x
web > index > index_actions.php > render_content_from_params()
1 reference
function get_menu_items(): string {
    $user_type = get_session_property(key: "user_type");

    if ($user_type === "admin") return get_admin_menu_items();
    if ($user_type === "pacjent") return get_pacjent_menu_items();
    if ($user_type === "lekarz") return get_lekarz_menu_items();
}

1 reference
function get_admin_menu_items(): string {
    return <><>ADMIN_MENU_ITEMS
    <h2><a href="?page=lekarze">Dodaj lekarza</a></h2>
    <h2><a href="?page=novy_lekarz">Nowy lekarz</a></h2>
    <h2><a href="?page=pacjenci">Pacjenci</a></h2>
    <h2><a href="?page=novy_pacjent">Nowy pacjent</a></h2>
    <h2><a href="?page=wizyty">Wizyty</a></h2>
    <h2><a href="?page=nova_wizyta">Nowa wizyta</a></h2>
    <h2><a href="?page=profil">Mój profil</a></h2>
ADMIN_MENU_ITEMS;
}

1 reference
function get_pacjent_menu_items(): string {
    return <><>PACJENT_MENU_ITEMS
    <h2><a href="?page=wizyty">Wizyty</a></h2>
    <h2><a href="?page=nova_wizyta">Zarejestruj wizytę</a></h2>
    <h2><a href="?page=profil">Mój profil</a></h2>
PACJENT_MENU_ITEMS;
}

1 reference
function get_lekarz_menu_items(): string {
    return <><>LEKARZ_MENU_ITEMS
    <h2><a href="?page=pacjenci">Pacjenci</a></h2>
    <h2><a href="?page=wizyty">Wizyty</a></h2>
    <h2><a href="?page=profil">Mój profil</a></h2>
LEKARZ_MENU_ITEMS;
}

/*

```

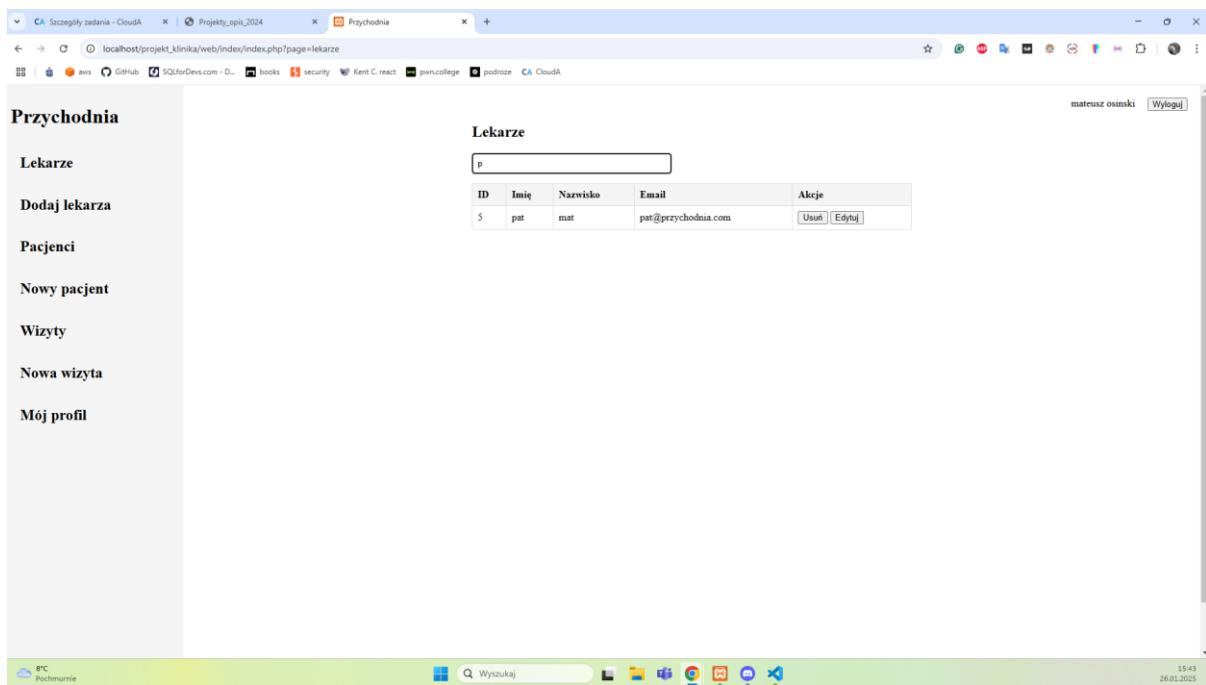
Przejdzmy do kolejnych widoków.

Lekarze

Znajduje się tutaj lista wszystkich lekarzy w aplikacji.

ID	Imię	Nazwisko	Email	Akcje
3	mat	pat	mat@przychodnia.com	Usuń Edytuj
5	pat	mat	pat@przychodnia.com	Usuń Edytuj

Możemy wyszukiwać danego lekarza z listy za pomocą inputa, podając imię i nazwisko:



Do filtrowania używam kodu JavaScriptowego:

```
JS searcher.js < ...
js > JS searcher.js > ...
      You, yesterday | 1 author (You)
1  /*
2  Universal app searcher.
3  Usage:
4  Define searcher in html:
5
6  <div class="search">
7  | <input type="text" class="search_input" data-table="lekarze_tabela" data-tr-data="lekaczDane" placeholder="Wpisz dane lekarza" />
8  </div>
9
10 Where:
11 - `data-table` is name of table to be filtered,
12 - `data-tr-data` is name of dataset attr settled in the `tr` of `tbody` attr.
13 */
14 document.addEventListener("DOMContentLoaded", () => {
15   const searchInputs = document.querySelectorAll(".search_input");
16
17   searchInputs.forEach((input) => {
18     const trData = input.dataset.trData;
19     const tableClass = input.dataset.table;
20     const table = document.querySelector(`.${tableClass}`);
21     const rows = table.querySelectorAll("tbody tr");
22
23     input.addEventListener("input", (event) => {
24       const filter = event.target.value.toLowerCase();
25
26       rows.forEach((row) => {
27         const rowData = row.dataset[trData].toLowerCase();
28         row.style.display = rowData.includes(filter) ? "" : "none";
29       })
30     })
31   });
32 });


```

Działa on w takim sposób, że za każdym załadowaniem DOM'u (Document Object Model) szuka on wszystkich input'ów z klasą `search_input`. Następnie, przechodzi po liście wszystkich inputów i:

1. pobiera `trData` - jest to nazwa atrybutu dataset, który jest ustawiony na wierszu do filtrowania – w nim jest sprecyzowana fraza do filtrowania

```
JS searcher.js | lekarze.php X
web > lekarze > lekarze.php
You, yesterday | 1 author (you)
1  <?php require("lekarze_actions.php") ?>
2
3  <div class="lekarze_content">
4    <h2>Lekarze</h2>
5    <div class="search">
6      <input type="text" class="search_input" data-table="lekarze_tabela" data-tr-data="lekarzDane" placeholder="Wpisz dane lekarza" />
7    </div>
8    <table class="lekarze_tabela">
9      <thead>
10     <tr>
11       <th>ID</th>
12       <th>Imię</th>
13       <th>Nazwisko</th>
14       <th>Email</th>
15       <th>Akcje</th>
16     </tr>
17   </thead>
18   <tbody>
19     <?php get_lekarze() ?>
```



```
lekarze_actions.php X
web > lekarze > lekarze_actions.php > ...
1 reference
54  function get_lekarze(): void {
55    $conn = get_conn();
56    $query = "SELECT * FROM lekarze;";
57    $result = mysqli_query(mysqli: $conn, query: $query);
58
59    while ($lekarz = mysqli_fetch_row(result: $result)) {
60      echo <<<WYPISZ_LEKARZA
61        <tr data-lekarz-dane="$lekarz[0] $lekarz[1]">
62          <td>$lekarz[0]</td>
63          <td>$lekarz[1]</td>
64          <td>$lekarz[2]</td>
65          <td>$lekarz[3]</td>
66          <td>
67            <form method="POST">
68              <button type="submit" name="delete" value="$lekarz[0]">Usuń</button>
69              <button type="submit" name="edit" value="$lekarz[0]">Edytuj</button>
70            </form>
71          </td>
72        </tr>
73      WYPISZ_LEKARZA;
74    }
75
76    close_conn(conn: $conn);
77  }
```

W tym przypadku jest to imię i nazwisko lekarza

2. Pobiera `tableClass` - jest to nazwa tabeli, w której będzie dokonywać filtrowania

```

JS searcher.js | lekarze.php x
web > lekarze > lekarze.php
    You, yesterday | 1 author (You)
1   <?php require("lekarze_actions.php") ?>
2
3   <div class="lekarze_content">
4     <h2>Lekarze</h2>
5     <div class="search">
6       <input type="text" class="search_input" data-table="lekarze_tabela" data-tr-data="lekaczDane" placeholder="Wpisz dane lekarza" />
7     </div>
8     <table class="lekarze_tabela">
9       <thead>
10      <tr>
11        <th>ID</th>
12        <th>Imię</th>
13        <th>Nazwisko</th>
14        <th>Email</th>
15        <th>Akcje</th>
16      </tr>
17    </thead>
18    <tbody>
19      <tr> You, 2 days ago + show all lekarze

```

W tym przypadku nazwą jest "lekarze_tabela"

3. Pobiera `table` - pobiera z DOMu tabelę (`table class="lekarze_table` w naszym przypadku)
4. Pobiera wszystkie wiersze zdefiniowane w `<tbody>` tabeli. Dzięki temu mamy dostęp do tej zawartości!

```

function get_lekarze(): void {
    $conn = get_conn();
    $query = "SELECT * FROM lekarze;";
    $result = mysqli_query(mysql: $conn, query: $query);

    while ($lekarz = mysqli_fetch_row(result: $result)) {
        echo <<<WYPISZ_LEKARZA
        <tr data-lekarz-dane="$lekarz[0] $lekarz[1]">
            <td>$lekarz[0]</td>
            <td>$lekarz[1]</td>
            <td>$lekarz[2]</td>
            <td>$lekarz[3]</td>
            <td>
                <form method="POST">
                    <button type="submit" name="delete" value="$lekarz[0]">Usuń</button>
                    <button type="submit" name="edit" value="$lekarz[0]">Edytuj</button>
                </form>
            </td>
        </tr>
    WYPISZ_LEKARZA;
}

```

5. Następnie dodajemy listenera do naszego input'a

```

JS searcher.js | lekarze.php x
web > lekarze > lekarze.php
    You, yesterday | 1 author (You)
1   <?php require("lekarze_actions.php") ?>
2
3   <div class="lekarze_content">
4     <h2>Lekarze</h2>
5     <div class="search">
6       <input type="text" class="search_input" data-table="lekarze_tabela" data-tr-data="lekaczDane" placeholder="Wpisz dane lekarza" />
7     </div>
8     <table class="lekarze_tabela">
9       <thead>
10      <tr>
11        <th>ID</th>
12        <th>Imię</th>
13        <th>Nazwisko</th>
14        <th>Email</th>

```

który będzie odpalać dalej zdefiniowaną funkcję, gdy zostanie zrobiona zmiana w inpucie (np. Dodanie frazy),

- Następnie przechodzimy przez wszystkie wiersze i sprawdzamy, czy posiadają one podaną w inpucie zawartość - jeżeli tak, to nic nie robimy na stylu. W przeciwnym przypadku nadajemy elementowi `<tr>` styl `display: none` , przez co nie jest on już widoczny w tabeli.

Mamy również akcje, takie jak usuwanie lekarza czy edycja.

Strona edycji lekarza

Przychodnia

Lekarze

Dodaj lekarza

Pacjenci

Nowy pacjent

Wizyty

Nowa wizyta

Mój profil

mateusz osimski Wyloguj

Edytuj lekarza

Imię
pat

Nazwisko
mat

Edytuj

I po zmianie imienia:

The screenshot shows a web application interface for managing a clinic. On the left, there is a sidebar with links: 'Przychodnia' (selected), 'Lekarze', 'Dodaj lekarza', 'Pacjenci', 'Nowy pacjent', 'Wizyty', 'Nowa wizyta', and 'Mój profil'. The main content area is titled 'Lekarze' and contains a table with two rows of doctor data:

ID	Imię	Nazwisko	Email	Akcje
3	mat	pat	mat@przychodnia.com	[Usuń] [Edytuj]
5	mateusz	mat	pat@przychodnia.com	[Usuń] [Edytuj]

At the bottom right of the main area, there are buttons for 'mateusz osimski' and 'Wyloguj'. The status bar at the bottom shows '15:59 26.01.2023'.

Dodawanie lekarza

The screenshot shows the same web application interface, but the 'Nowy lekarz' (New Doctor) section is active. It contains four input fields: 'Imię' (Name), 'Nazwisko' (Surname), 'email' (Email), and 'Hasło' (Password). Below these fields is a 'Stwórz' (Create) button. The rest of the sidebar and main content area are identical to the previous screenshot.

Mamy tutaj walidację na imieniu, nazwisko, haśle czy emailu:

Przychodnia

Lekarze

Dodaj lekarza

Pacjenci

Nowy pacjent

Wizyty

Nowa wizyta

Mój profil

Nowy lekarz

Imię
Kamil

Nazwisko
Osinski

email
mat@przychodnia.com

Hasło

Stwórz

mateusz osinski [Wyloguj]

16:00
26.01.2023

Po wprowadzeniu poprawnych danych:

Przychodnia

Lekarze

Dodaj lekarza

Pacjenci

Nowy pacjent

Wizyty

Nowa wizyta

Mój profil

Lekarze

Wpisz dane lekarza				
ID	Imię	Nazwisko	Email	Akcje
3	mat	pat	mat@przychodnia.com	[Usuń] [Edytuj]
5	mateusz	mat	pat@przychodnia.com	[Usuń] [Edytuj]
6	Kamil	Osinski	kamil.osinski@przychodnia.com	[Usuń] [Edytuj]

mateusz osinski [Wyloguj]

16:01
26.01.2023

Pacjenci

Na stronie pacjentów mamy takie same możliwości jak na stronie lekarzy.

Screenshot of a web application interface titled "Przychodnia". The main content area displays a table of patients (Pacjenci) with columns: ID, Imię, Nazwisko, Email, and Akcje (User, Edytuj). The table contains 6 rows of data. On the left sidebar, there are links for Lekarze, Dodaj lekarza, Pacjenci, Nowy pacjent, Wizyty, Nowa wizyta, and Mój profil.

ID	Imię	Nazwisko	Email	Akcje
18	mat2	pat	mat@gmail.com	[User] [Edytuj]
20	taw	wat	taw@gmail.com	[User] [Edytuj]
21	kam	osi	lat@gmail.com	[User] [Edytuj]
25	ram	pamsiam	ram@gmail.com	[User] [Edytuj]
26	tam	pat	ram123@gmail.com	[User] [Edytuj]

Edycja pacjenta:

Screenshot of the same web application interface, showing the "Edytuj pacjenta" (Edit patient) form. The form fields are: Imię (tam) and Nazwisko (pat). There is also an "Edytuj" (Edit) button.

The screenshot shows a web application window titled "Przychodnia". On the left, there is a sidebar with links: "Lekarze", "Dodaj lekarza", "Pacjenci", "Nowy pacjent", "Wizyty", "Nowa wizyta", and "Mój profil". The main content area is titled "Pacjenci" and contains a table with patient data:

ID	Imię	Nazwisko	Email	Akcje
18	mat2	pat	mat@gmail.com	[Usuń] [Edytuj]
20	taw	wat	taw@gmail.com	[Usuń] [Edytuj]
21	kam	osi	lat@gmail.com	[Usuń] [Edytuj]
25	ram	pamsiam	ram@gmail.com	[Usuń] [Edytuj]
26	tamsiam	pat	ram123@gmail.com	[Usuń] [Edytuj]

The status bar at the bottom shows the weather as "8°C Pochmurnie" and the date/time as "16:23 26.01.2025".

Po usunięciu:

The screenshot shows the same web application window as before, but the patient with ID 18 (mat2, pat) has been removed from the list. The table now contains the following data:

ID	Imię	Nazwisko	Email	Akcje
20	taw	wat	taw@gmail.com	[Usuń] [Edytuj]
21	kam	osi	lat@gmail.com	[Usuń] [Edytuj]
25	ram	pamsiam	ram@gmail.com	[Usuń] [Edytuj]
26	tamsiam	pat	ram123@gmail.com	[Usuń] [Edytuj]

The status bar at the bottom shows the weather as "8°C Pochmurnie" and the date/time as "16:23 26.01.2025".

Nowy pacjent

Administrator ma możliwość także dodawania pacjentów z poziomu swojego panelu:

The screenshot shows a web browser window with a sidebar menu on the left. The sidebar contains links: 'Przychodnia', 'Lekarze', 'Dodaj lekarza', 'Pacjenci', 'Nowy pacjent' (which is highlighted in green), 'Wizyty', 'Nowa wizyta', and 'Mój profil'. The main content area is titled 'Nowy pacjent' and contains four input fields: 'Imię' (Name) with value 'nowy', 'Nazwisko' (Surname) with value 'pacjent', 'email' with value 'nowy@gm', and 'Hasło' (Password). A 'Stwórz' (Create) button is at the bottom of the form. The browser's address bar shows the URL: `localhost/projekt_klinika/web/index/index.php?page=newy_pacjent`. The top right corner shows the user 'mateusz osimski' and a 'Wyloguj' (Logout) button. The bottom status bar shows the date and time: 26.01.2023 16:24.

Walidacja:

This screenshot shows the same web application interface as the previous one, but with validation errors displayed. The errors are: 'Up! Coś poszło nie tak.' (An error occurred), 'Niepoprawny adres email.' (Incorrect email address), and 'Niepoprawne hasło. Powinno składać się z min 5 znaków.' (Incorrect password. It should consist of at least 5 characters). The rest of the interface and sidebar are identical to the first screenshot.

Po dodaniu:

Pacjenci

ID	Imię	Nazwisko	Email	Akcje
18	mat2	pat	mat@gmail.com	[Usuń] [Edytuj]
20	taw	wat	taw@gmail.com	[Usuń] [Edytuj]
21	kam	osi	lat@gmail.com	[Usuń] [Edytuj]
25	ram	pamsiam	ram@gmail.com	[Usuń] [Edytuj]
27	nowy	pacjent	nowy@gmail.com	[Usuń] [Edytuj]

Wizyty

Na stronie wizyty znajduje się lista wizyt z lekarzem i pacjentem, którzy biorą udział w danej wizycie

Wizyty

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
1	mat pat	mat2 pat	2024-01-25 00:00:00	Problemy z nogą	Tak, powód: Pacjent nie da rady przyjść	[Edytuj]
2	mateusz mat	mat2 pat	2024-01-25 00:00:00	Wizyta kontrolna	Tak, powód: Lekarz się rochowował.	[Edytuj]
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problemy z gardłem	Tak, powód: nie ma żadnego problemu z gardłem	[Edytuj]
4	mat pat	mat2 pat	2025-02-25 00:00:00	Ból gardła - jeszcze bardziej boli	Nie	[Odwołaj] [Edytuj]
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą	Tak, powód: Już nie boli	[Edytuj]
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarcycą	Nie	[Odwołaj] [Edytuj]
7	mateusz mat	ram pamsiam	2025-01-30 00:00:00	Ból po wyrwaniu śemki	Nie	[Odwołaj] [Edytuj]
8	mat pat	mat2 pat	2025-01-12 13:30:00	Wizyta kontrolna po chorobie	Nie	[Odwołaj] [Edytuj]
9	mat pat	mat2 pat	2025-01-31 13:30:00	Ból w klatce piersiowej	Nie	[Odwołaj] [Edytuj]

Wizyta może zostać odwołana (przez administratora, lekarza lub pacjenta):

Trzeba podać powód odwołania wizyty

The screenshot shows a web browser window with the URL localhost/projekt_klinika/web/index.php?page=odwołaj_wizyte&wizyta_id=9. The page title is 'Przychodnia'. On the left, there's a sidebar with links: 'Lekarze', 'Dodaj lekarza', 'Pacjenci', 'Nowy pacjent', 'Wizyty', 'Nowa wizyta', and 'Mój profil'. The main content area is titled 'Odwołaj wizytę' and contains a form with a text input field containing 'Dzisiaj mam kolokwium i nie dam rady przyjść' and a button labeled 'Odwołaj'. At the bottom right of the screen, there's a status bar showing 'mateusz osimski Wyloguj' and the date '26.01.2025'.

Po odwołaniu zostajemy przekierowani na listę wizyt

The screenshot shows a web browser window with the same URL as the previous screenshot. The page title is 'Przychodnia'. The sidebar and main content area are identical to the first screenshot, but the main content now displays a table of appointments. The table has columns: 'ID', 'Lekarz', 'Pacjent', 'Data wizyty', 'Opis', 'Czy odwołana?', and 'Akcje'. There are 9 rows of data. The 'Akcje' column for each row contains two buttons: 'Edytuj' and 'Odwołaj'. The status bar at the bottom right shows 'mateusz osimski Wyloguj' and the date '26.01.2025'.

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
1	mat pat	mat2 pat	2024-01-25 00:00:00	Problemy z nogą	Tak, powód: Pacjent nie da rady przyjść	Edytuj
2	mateusz mat	mat2 pat	2024-01-25 00:00:00	Wizyta kontrolna	Tak, powód: Lekarz się rozchorował.	Edytuj
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problem z gardłem	Tak, powód: nie ma żadnego problemu z gardłem	Edytuj
4	mat pat	mat2 pat	2025-02-25 00:00:00	Ból gardła - jeszcze bardziej boli	Nie	Odwołaj Edytuj
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą	Tak, powód: Już nie boli	Edytuj
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	Odwołaj Edytuj
7	mateusz mat	ram pamsiam	2025-01-30 00:00:00	Ból po wyrwaniu śenki	Nie	Odwołaj Edytuj
8	mat pat	mat2 pat	2025-01-12 13:30:00	Wizyta kontrolna po chorobie	Nie	Odwołaj Edytuj
9	mat pat	mat2 pat	2025-01-31 13:30:00	Ból w klatce piersiowej	Tak, powód: Dzisiaj mam kolokwium i nie dam rady przyjść	Edytuj

Daną wizytę można również edytować (może to też zrobić pacjent)

Szczegóły zadania - CloudA

localhost/projekt_klinika/web/index/index.php?page=edit_wizyta&wizyta_id=8

Przychodnia

Lekarze
Dodaj lekarza
Pacjenci
Nowy pacjent
Wizyty
Nowa wizyta
Mój profil

Edytuj wizyty

Lekarz	mat pat
Pacjent	mat2 pat
Opis	Wizyta kontrolna po chorobie
Edytuj	

8°C Pochmurnie 16:34 26.01.2025

Szczegóły zadania - CloudA

localhost/projekt_klinika/web/index/index.php?page=wizyty

Przychodnia

Lekarze
Dodaj lekarza
Pacjenci
Nowy pacjent
Wizyty
Nowa wizyta
Mój profil

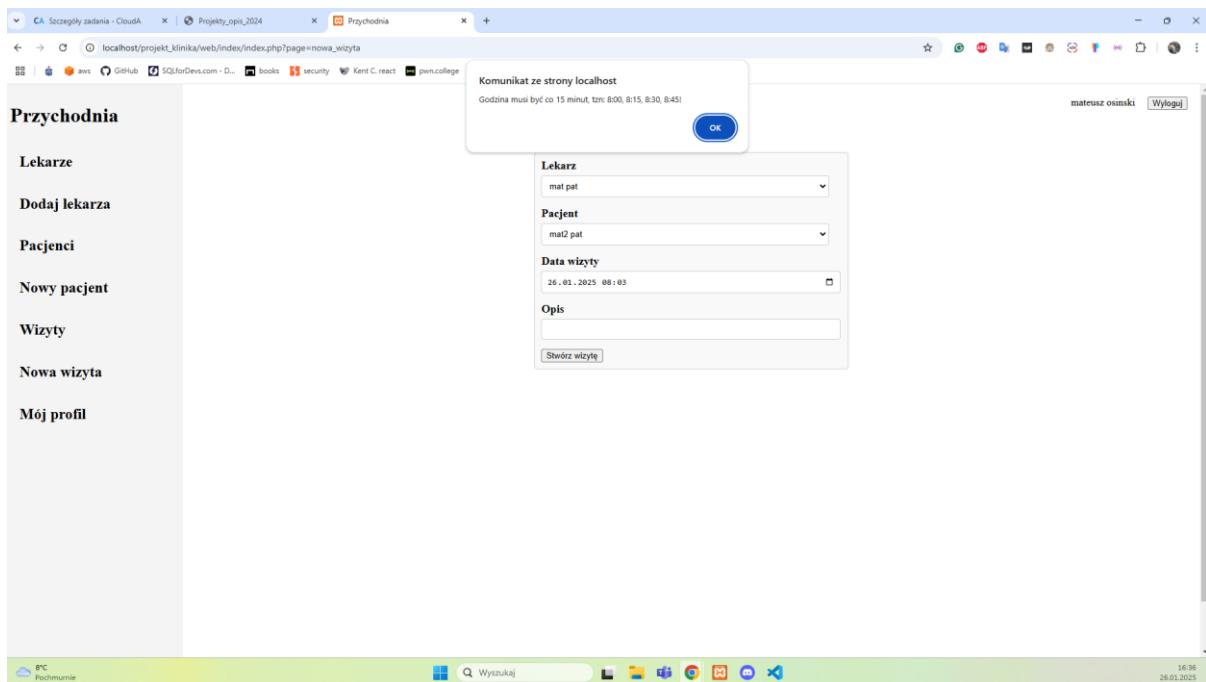
Wizyty

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
1	mat pat	mat2 pat	2024-01-25 00:00:00	Problemy z nogą	Tak, powód: Pacjent nie da rady przyjść	Edytuj
2	mateusz mat	mat2 pat	2024-01-25 00:00:00	Wizyta kontrolna	Tak, powód: Lekarz się rozchorował.	Edytuj
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problem z gardłem	Tak, powód: nie ma żadnego problemu z gardłem	Edytuj
4	mat pat	mat2 pat	2025-02-25 00:00:00	Ból gardła - jeszcze bardziej boli	Nie	Odwólać Edytuj
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą	Tak, powód: Już nie boli	Edytuj
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	Odwólać Edytuj
7	mateusz mat	ram pamsiam	2025-01-30 00:00:00	Ból po wyrwaniu śemki	Nie	Odwólać Edytuj
8	mat pat	mat2 pat	2025-01-12 13:30:00	Wizyta kontrolna po chorobie - to jest moja edytowana wizyta	Nie	Odwólać Edytuj
9	mat pat	mat2 pat	2025-01-31 13:30:00	Ból w klatce piersiowej	Tak, powód: Dzisiaj mam kolokwium i nie dam rady przyjść	Edytuj

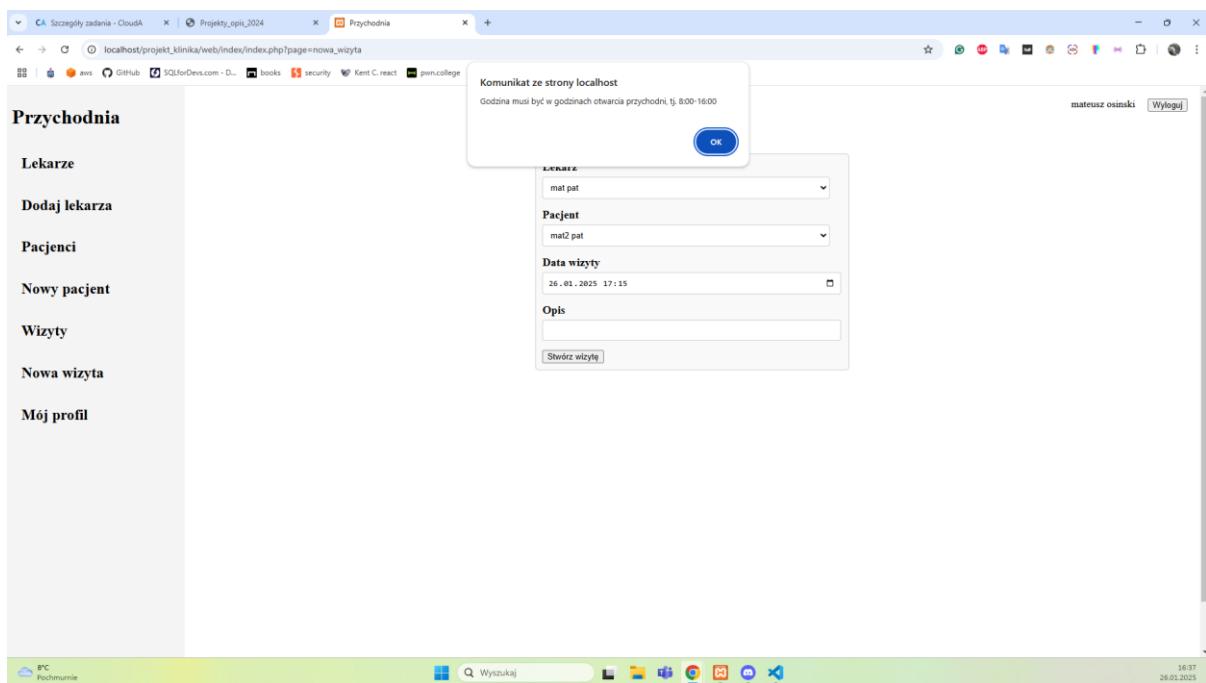
8°C Pochmurnie 16:35 26.01.2025

Nowa wizyta

Administrator (tak samo jak pacjent) może również tworzyć wizyty. Co ważne, każda wizyta musi rozpoczynać się o godzinie x:00, x:15, x: 30, lub x:45 (np. 8:45 czy 8:30. Godzina 8:20 już nie jest poprawna, ponieważ wizyty trwają 15 minut).

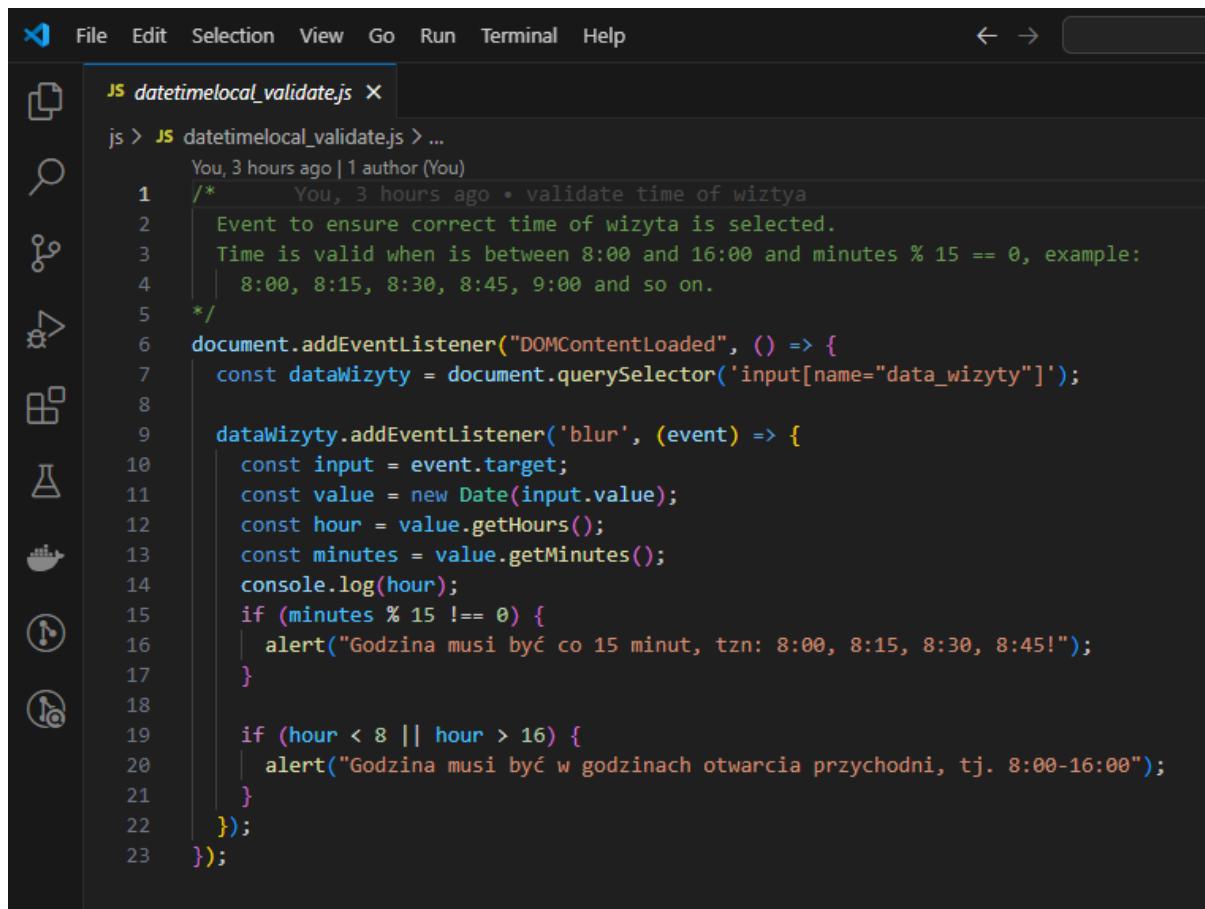


Analogicznie, godzina wizyty musi być z zakresu 8-16 (godziny otwarcia przychodni)



Nie można tworzyć wizyt z przeszłości, tylko dzisiaj i przyszłość.

Za alerty odpowiada dany skrypt JavaScriptowy:



```
JS datetimelocal_validate.js X
js > JS datetimelocal_validate.js > ...
You, 3 hours ago | 1 author (You)
1  /*      You, 3 hours ago • validate time of wizyta
2   Event to ensure correct time of wizyta is selected.
3   Time is valid when is between 8:00 and 16:00 and minutes % 15 == 0, example:
4   | 8:00, 8:15, 8:30, 8:45, 9:00 and so on.
5 */
6  document.addEventListener("DOMContentLoaded", () => {
7    const dataWizyty = document.querySelector('input[name="data_wizyty"]');
8
9    dataWizyty.addEventListener('blur', (event) => {
10      const input = event.target;
11      const value = new Date(input.value);
12      const hour = value.getHours();
13      const minutes = value.getMinutes();
14      console.log(hour);
15      if (minutes % 15 !== 0) {
16        alert("Godzina musi być co 15 minut, tzn: 8:00, 8:15, 8:30, 8:45!");
17      }
18
19      if (hour < 8 || hour > 16) {
20        alert("Godzina musi być w godzinach otwarcia przychodni, tj. 8:00-16:00");
21      }
22    });
23  });

```

Sprawdzana on, czy liczba minut modulo 15 jest równe 0 (innymi słowy, wielokrotność liczby 15) oraz czy godzina mieści się w przedziale 8-16.

Po utworzeniu wizyty następuje przekierowanie na listę wizyt:

Szczegóły zadania - CloudA | Projekty_otp_2024 | Przychodnia

localhost/projekt_klinika/web/index.php?page=wizyty

mateusz osinski Wyloguj

Przychodnia

Wizyty

Wpisz opis wizyty

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
1	mat pat	mat2 pat	2024-01-25 00:00:00	Problemy z nogą	Tak, powód: Pacjent nie da rady przyjść	Edytuj
4	mat pat	mat2 pat	2025-02-25 00:00:00	Ból gardła - jeszcze bardziej bola	Nie	Odwołaj Edytuj
8	mat pat	mat2 pat	2025-01-12 13:30:00	Wizyta kontrolna po chorobie - to jest moja edytowana wizyta	Nie	Odwołaj Edytuj
9	mat pat	mat2 pat	2025-01-31 13:30:00	Ból w klatce piersiowej	Tak, powód: Dzisiaj mam kolokwium i nie dam rady przyjść	Edytuj
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą	Tak, powód: Już nie boli	Edytuj
10	mat pat	nowy pacjent	2025-01-30 14:15:00	Nowa wizyta na 14:15	Nie	Odwołaj Edytuj
2	mateusz mat	mat2 pat	2024-01-25 00:00:00	Wizyta kontrolna	Tak, powód: Lekarz się rozchorował	Edytuj
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problem z gardłem	Tak, powód: nie ma żadnego problemu z gardłem	Edytuj
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	Odwołaj Edytuj
7	mateusz mat	ram pamsiam	2025-01-30 00:00:00	Ból po wyryaniu ślemki	Nie	Odwołaj Edytuj

8°C Pochmurnie 16:39 26.01.2025

Mój profil

Szczegóły zadania - CloudA | Projekty_otp_2024 | Przychodnia

localhost/projekt_klinika/web/index.php?page=profil

mateusz osinski Wyloguj

Przychodnia

Profil użytkownika

Imię: mateusz
Nazwisko: osinski
Email: admin@przychodnia.com
Typ użytkownika: admin

Nowe hasło

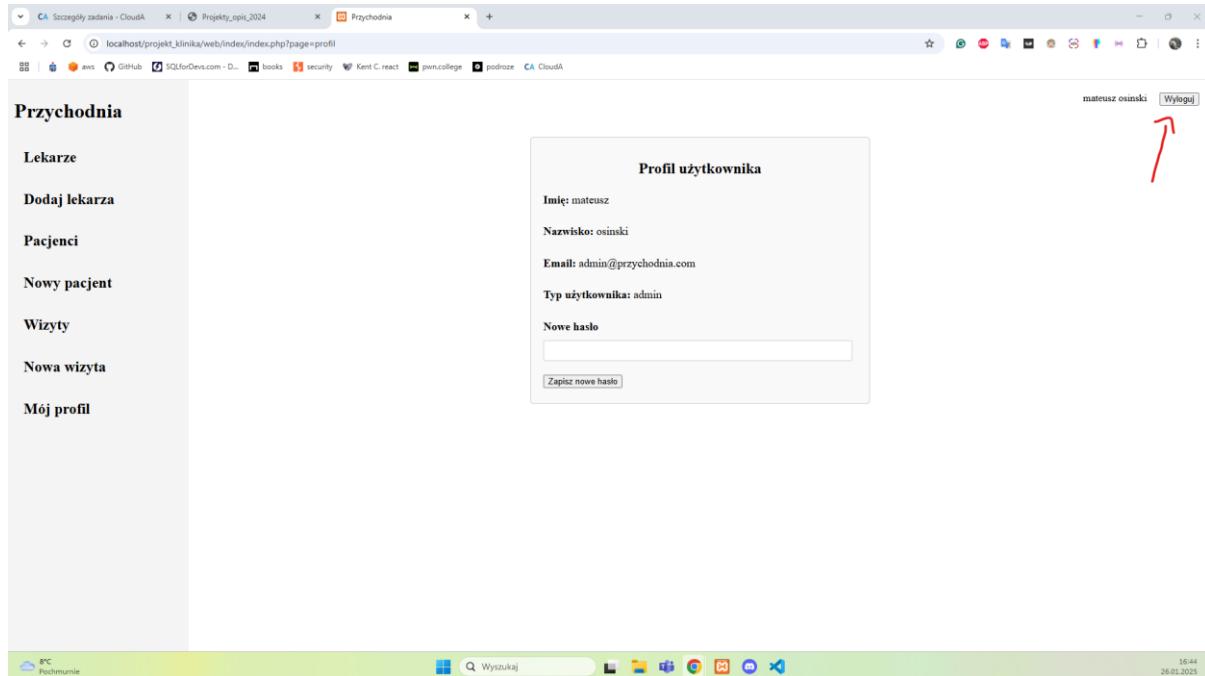
[Zapisz nowe hasło](#)

8°C Pochmurnie 16:40 26.01.2025

W przypadku administratora mamy możliwość również zmienienia hasła (które zostanie zapisane do pliku `admin_data.txt`).

Wyloguj

Aby wylogować się z konta, trzeba naciągnąć przycisk `Wyloguj` , w prawnym górnym narożniku.



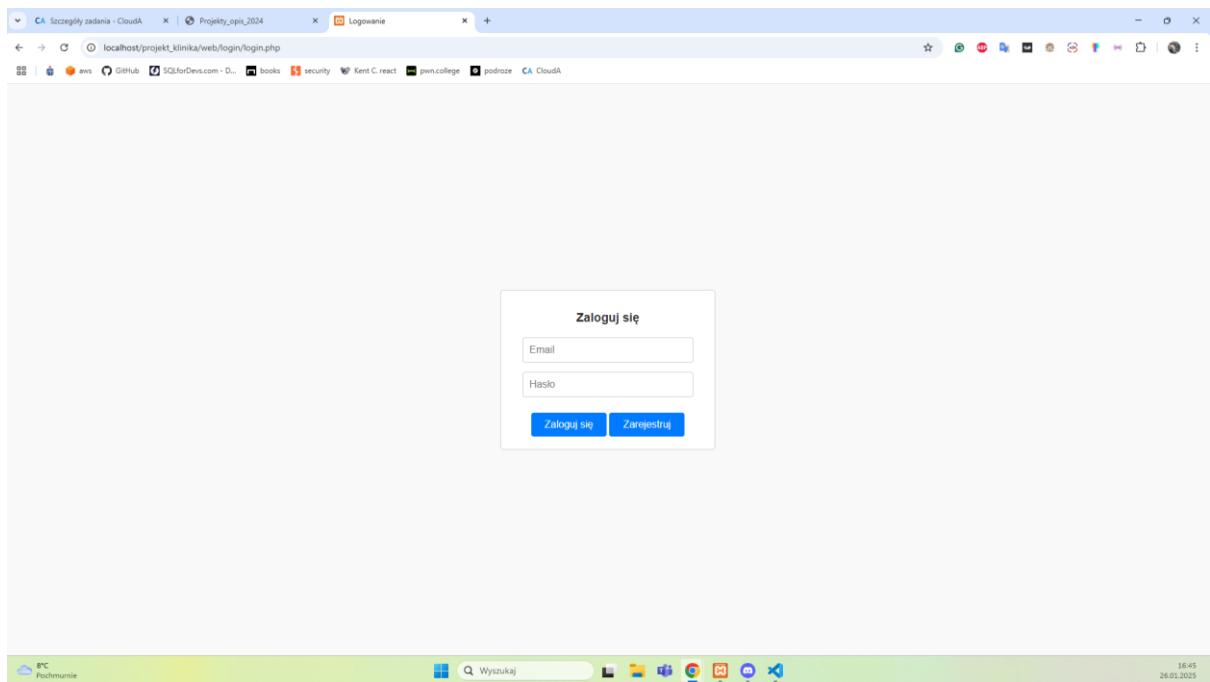
Podczas wylogowania czyszczona jest sesja

The code editor displays three files:

- index.php:** Contains the HTML structure of the application, including a menu and a logout button labeled "wyloguj".
- index_actions.php:** Contains PHP code that handles the logout process. It checks if the "log_out" button was pressed and then calls the "log_out" function.
- config.php:** Contains functions for managing sessions, including "start_session" which initializes session variables like \$SESSION["surname"], \$SESSION["email"], \$SESSION["user_id"], and \$SESSION["user_type"], and "stop_session" which ends the session.

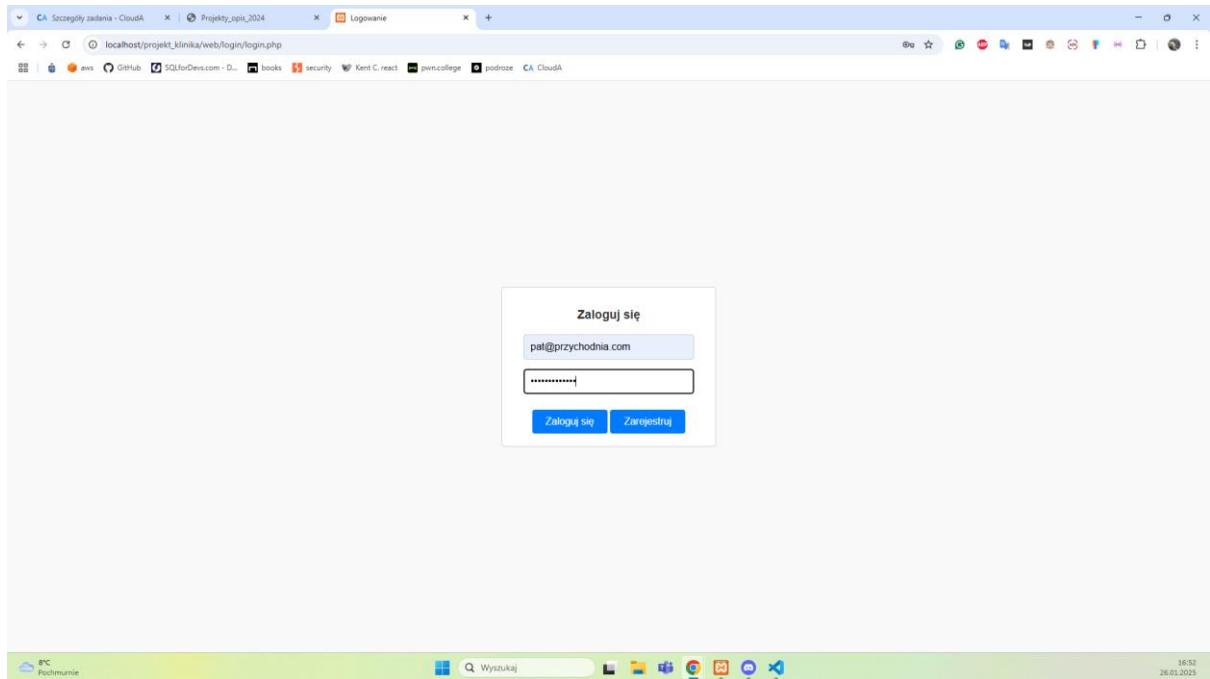
Red annotations highlight the "wyloguj" button in index.php, the "log_out()" function call in index_actions.php, and the "start_session()" function in config.php.

a końcu zostajemy przekierowani na stronę logowania.

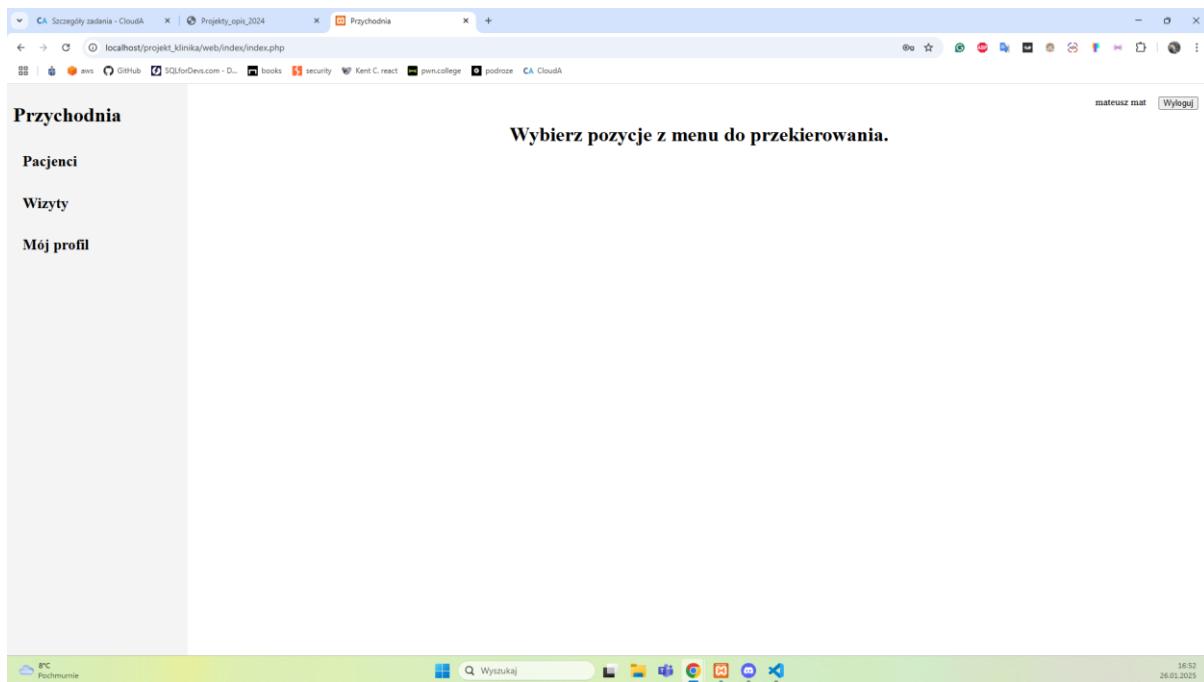


Widok lekarza

Zaloguję się teraz jako lekarz (email: pat@przychodnia.com, hasło: password12345)

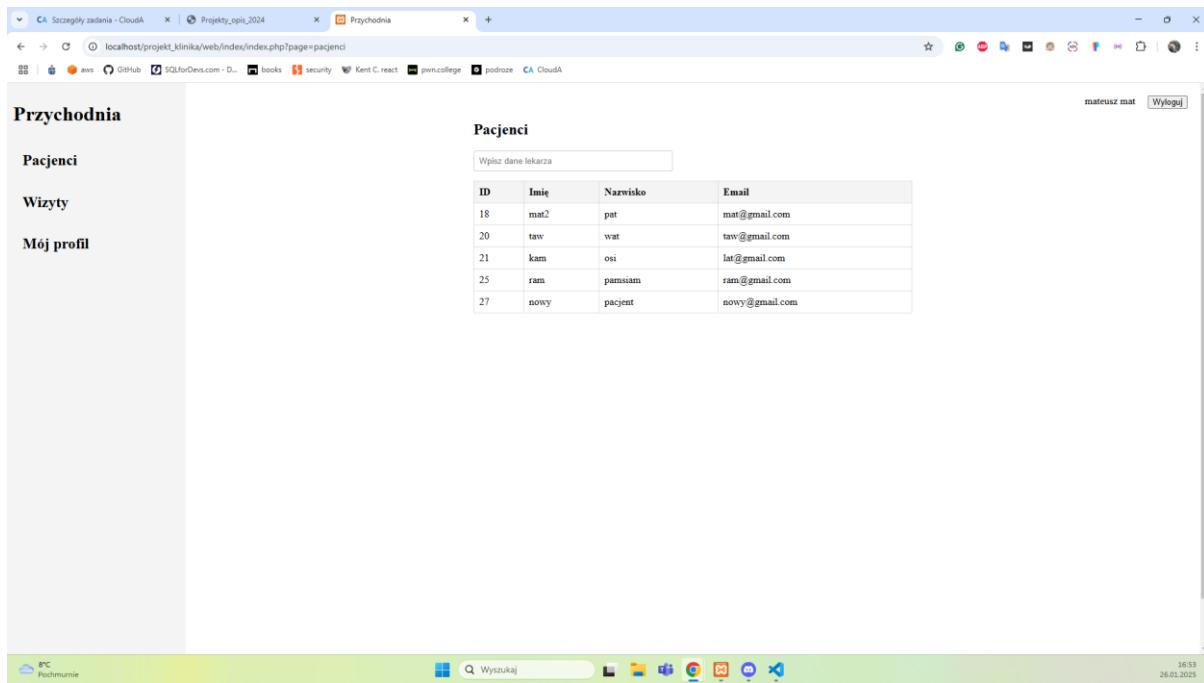


Tak wygląda index page po zalogowaniu:



Pacjenci

Strona pacjenci listuje wszystkich pacjentów w aplikacji, z możliwością wyszukiwania (taki sam mechanizm jak opisany wyżej).



Lekarz w przeciwieństwie do administratora nie ma możliwości edytowania czy usuwania pacjentów.

Wizyty

Na tej stronie lekarz widzi swoje wszystkie wizyty

The screenshot shows a web browser window with a URL starting with 'localhost/projekt_klinika/web/index.php?page=wizyty'. The page title is 'Przychodnia'. On the left, there's a sidebar with links for 'Pacjenci' (Patients), 'Wizyty' (Visits), and 'Mój profil' (My Profile). The main content area is titled 'Wizyty' and contains a table with the following data:

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
2	mateusz mat	mat2 pat	2024-01-25 00:00:00	Wizyta kontrolna	Tak, powód: Lekarz się rozchorował.	<button>Edytuj</button>
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problem z gardłem	Tak, powód: nie ma zadnego problemu z gardłem	<button>Edytuj</button>
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	<button>Odwóla</button> <button>Edytuj</button>
7	mateusz mat	ram pamsiam	2025-01-30 00:00:00	Ból po wyrwaniu 8emki	Nie	<button>Odwóla</button> <button>Edytuj</button>

Jest to zrobione za pomocą konkatenacji zapytań po stronie serwera

The image shows two side-by-side code editors. The left editor contains 'wizyty.php' with the following code:

```
1 //php require('wizyty_actions.php'); You, 2 days ago + define dynamic line numbers
2 <?php
3 <?php yesterday | file (You)
4 <?>
5 <?>
6 <?>
7 <?>
8 <?>
9 <?>
10 <?>
11 <?>
12 <?>
13 <?>
14 <?>
15 <?>
16 <?>
17 <?>
18 <?>
19 <?>
20 <?>
21 <?>
22 <?>
23 <?>
24 <?>
```

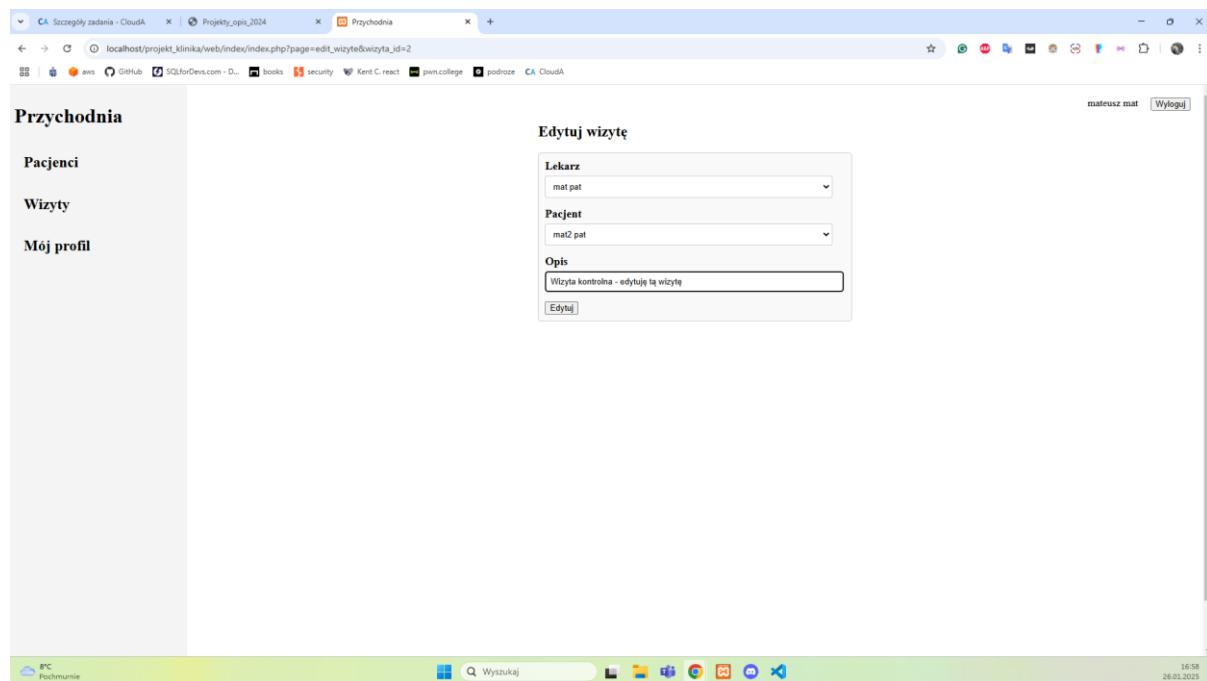
The right editor contains 'wizyty_actions.php' with the following code:

```
1 <?>
2 <?>
3 <?>
4 <?>
5 <?>
6 <?>
7 <?>
8 <?>
9 <?>
10 <?>
11 <?>
12 <?>
13 <?>
14 <?>
15 <?>
16 <?>
17 <?>
18 <?>
19 <?>
20 <?>
21 <?>
22 <?>
23 <?>
24 <?>
```

Jeżeli aktualnym użytkownikiem jest lekarz lub pacjent, to wtedy do głównego zapytania dodajemy ` WHERE lekarze.id = lekarz_id` lub ` WHERE pacjenci.id = pacjent_id` , dzięki czemu dostajemy tylko wizyty danego użytkownika. (administrator widzi wszystkie wizyty).

Analogicznie jak w przypadku administratora, lekarz może edytować wizytę lub ją odwołać

Edycja wizyty:



Jako, że został zmieniony lekarz, nie widzimy tej wizyty więcej w naszej liście

Przychodnia

Wizyty

Wpisz opis wizyty

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
3	mateusz mat	kam osi	2025-01-30 00:00:00	Problem z gardlem	Tak, powód: nie ma zadnego problemu z gardlem	Edytuj Odezwój
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	Odezwój Edytuj
7	mateusz mat	mat2 pat	2025-01-30 00:00:00	Ból po wyrwaniu Semki - tak	Nie	Odezwój Edytuj

Mój profil

Strona z informacjami odnośnie lekarza:

Przychodnia

Pacjenci

Wizyty

Mój profil

Profil użytkownika

Imię: mateusz

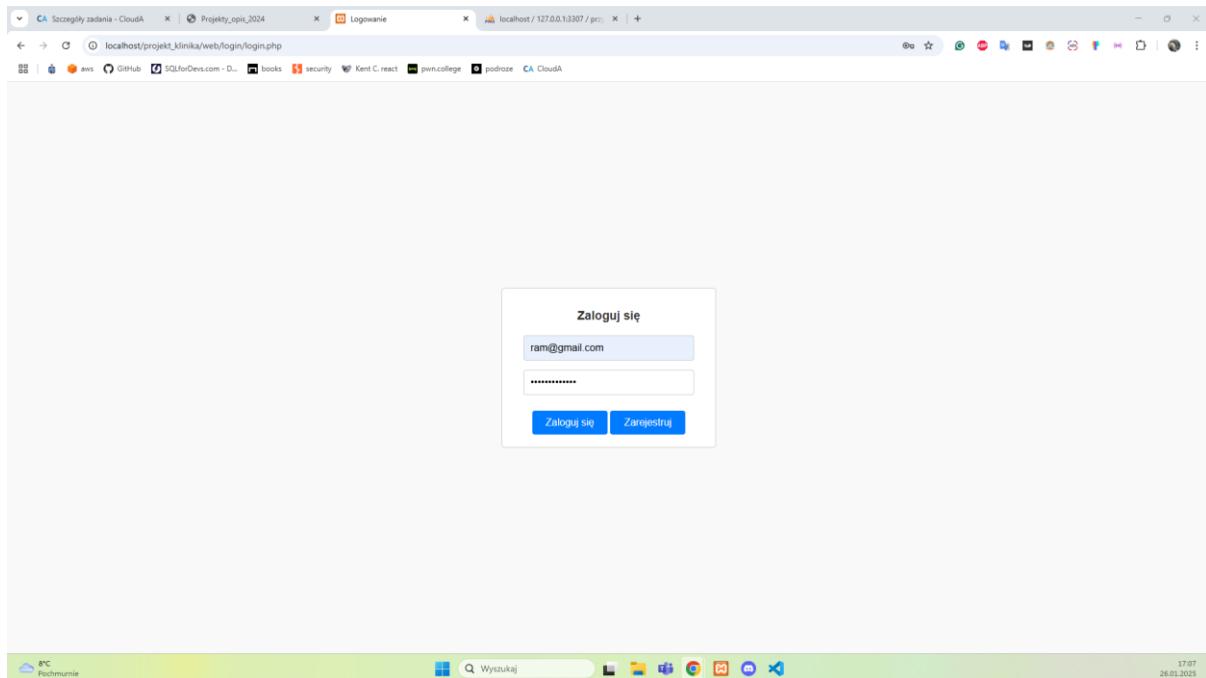
Nazwisko: mat

Email: pat@przychodnia.com

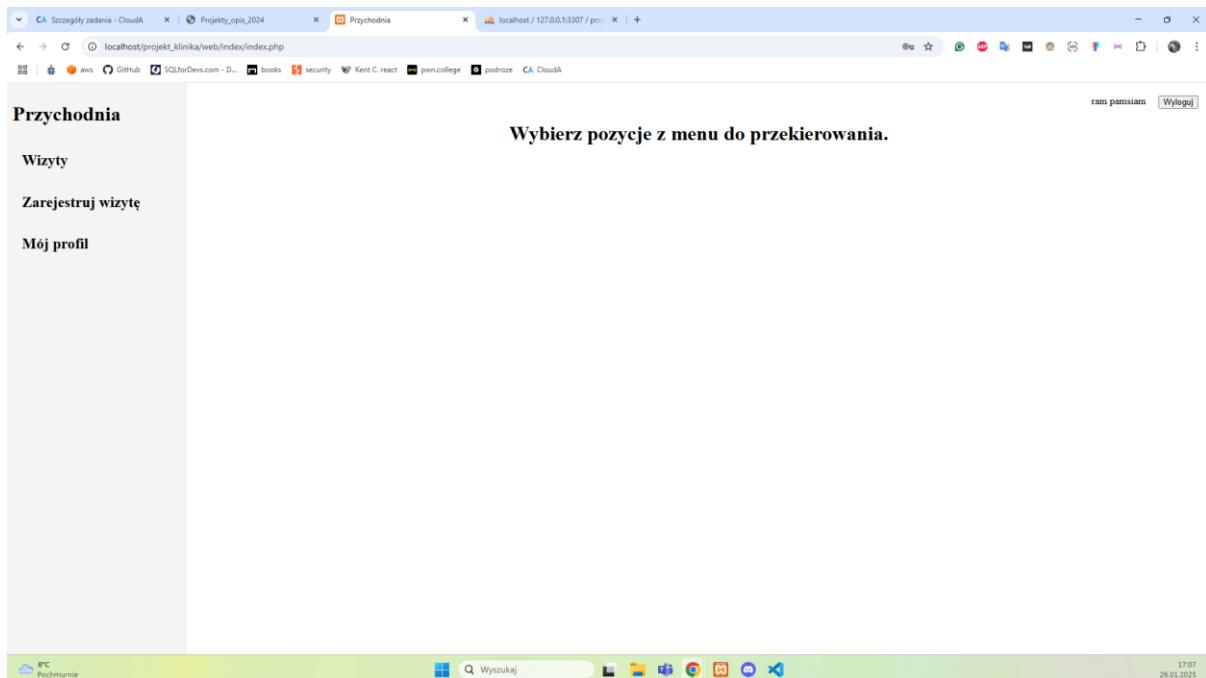
Typ użytkownika: lekarz

Widok pacjenta

Zalogujmy się teraz jako pacjent (email: ram@gmail.com, hasło: password12345)



Widok po zalogowaniu:



Wizyty

Jest to lista wizyt danego pacjenta (każdy pacjent widzi tylko swoje wizyty). Analogicznie jak lekarz i administrator, pacjent może edytować i odwołać wizytę

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problem z nogą	Tak, powód: Już nie boli	<button>Edytuj</button> <button>Odwołaj</button>
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problem z tarczycą	Nie	<button>Odwołaj</button> <button>Edytuj</button>

Różnica jest natomiast w edycji:

Lekarz
mat pat

Opis
Problem z nogą

Edytuj

Pacjent nie może już edytować pacjenta w danej wizycie. Może tylko zmienić lekarza oraz opis:

The screenshot shows a web browser window with three tabs open: 'Szczegółowy zadanie - CloudA', 'Projekty_zpin_2024', and 'Przychodnia'. The 'Przychodnia' tab is active, displaying a table of appointments. The table has columns: ID, Lekarz, Pacjent, Data wizyty, Opis, Czy odwołana?, and Akcje. There are two rows of data:

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą - zmieniłem opis tej wizyty właśnie	Tak, powód: Jut nie boli	Edytuj
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczycą	Nie	Odwołaj Edytuj

Nowa wizyta

Pacjent może zarejestrować się na nową wizytę, wybierając lekarza, datę oraz opis.

Na tej stronie jest taka sama walidacja jak wyżej opisana na widoku administratora.

CA Szczegóły zadania - CloudA x Projekty_opis_2024 x Przychodnia x localhost / 127.0.0.1:3307 / pr... +

localhost/projekt_klinika/web/index/index.php?page=nowa_wizyta

aws GitHub SQLforDev.com - D... books security Kent C. react pwn.college podroze CA CloudA

ram pamsiam Wyloguj

Przychodnia

Nowa wizyta

Wizyty

Zarejestruj wizytę

Mój profil

Lekarz
mat pat

Data wizyty
26-01-2025 08:00

Opis
Nowa wizyta na godzinę 8:00

Stwórz wizytę

8°C Pochmurnie

Wyszukaj

17:11 26.01.2025

Przychodnia

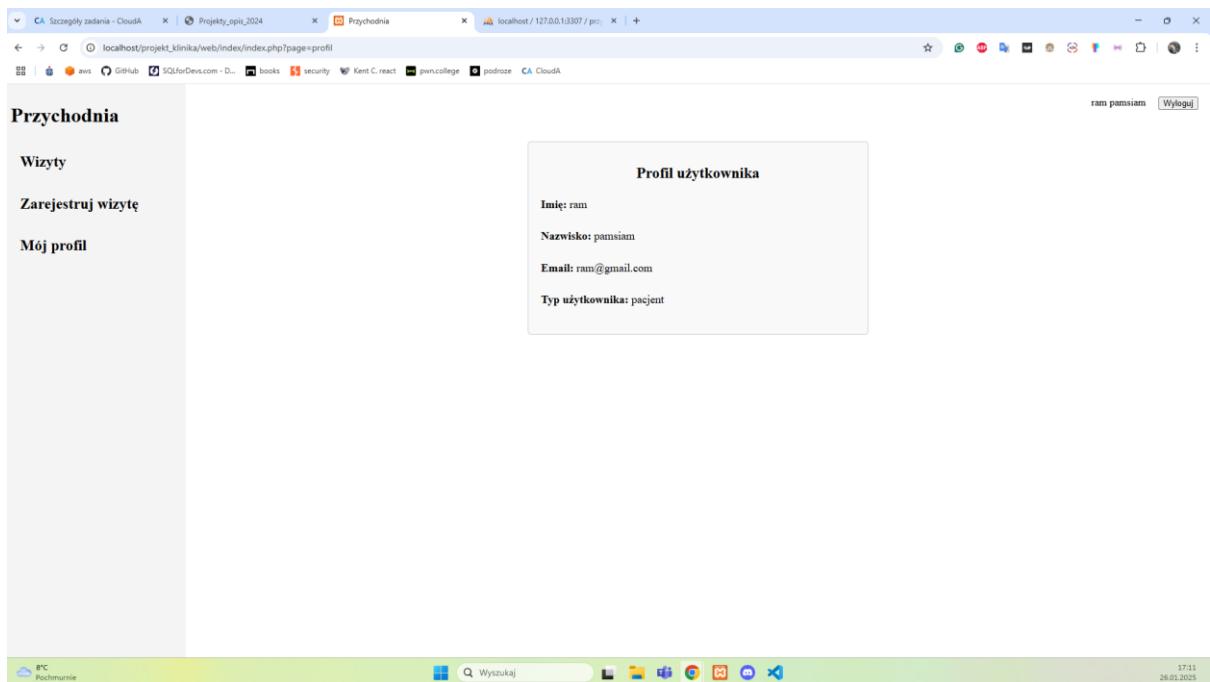
Wizyty

Wpisz opis wizyty

ID	Lekarz	Pacjent	Data wizyty	Opis	Czy odwołana?	Akcje
5	mat pat	ram pamsiam	2025-01-31 00:00:00	Problemy z nogą - zmieniłem opis tej wizyty właśnie	Tak, powód: Już nie boli	Edytuj
11	mat pat	ram pamsiam	2025-01-26 08:00:00	Nowa wizyta na godzinę 8:00	Nie	Odwółaj Edytuj
6	mateusz mat	ram pamsiam	2025-01-24 00:00:00	Problemy z tarczącą	Nie	Odwółaj Edytuj

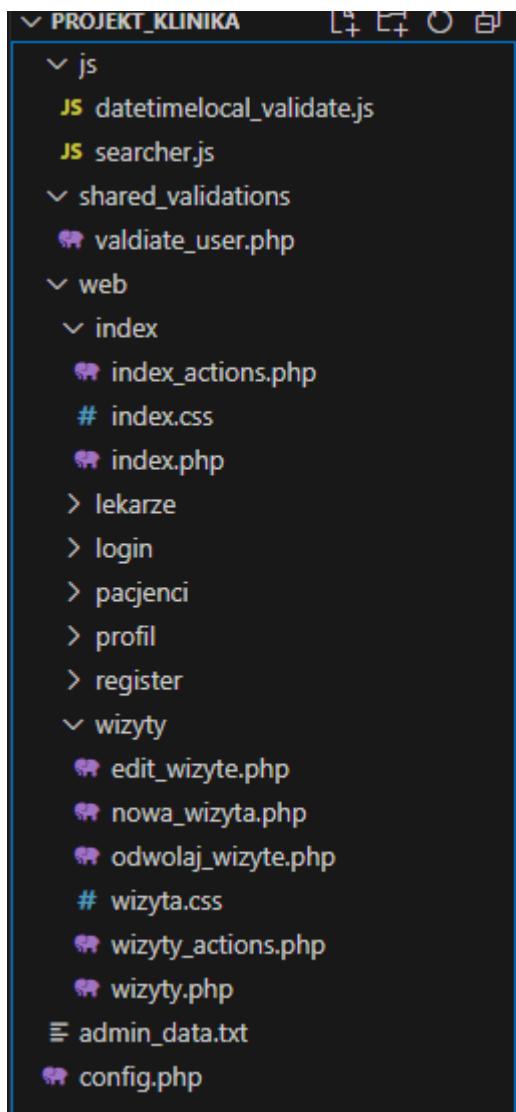
Mój profil

Strona z podstawowymi informacjami na temat użytkownika



Struktura projektu (kod)

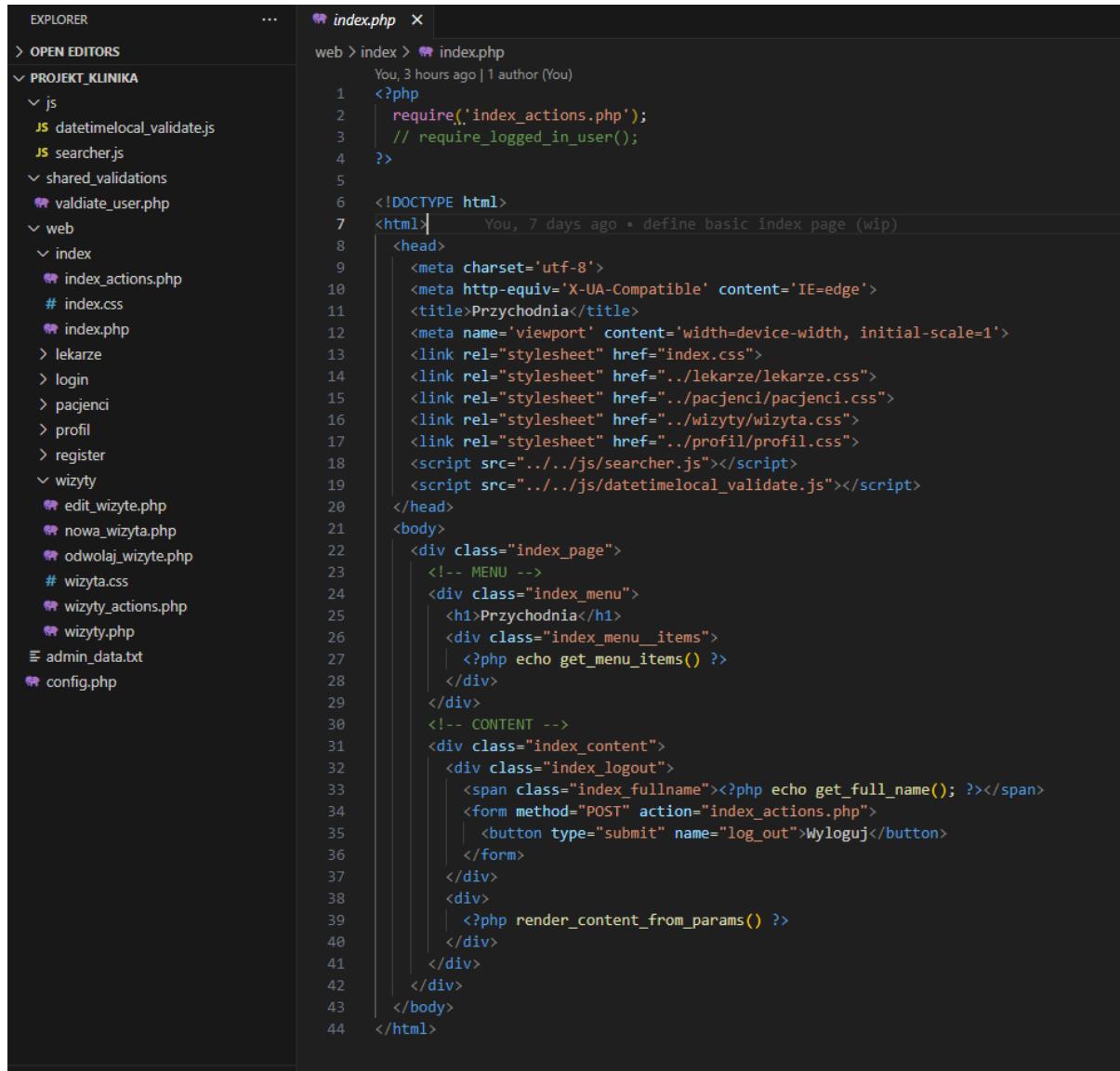
Kod został podzielony na trzy foldery + dwa pliki w głównej lokalizacji.



Krótki opis folderów:

- `js` - znajdują się skrypty JavaScriptowe,
- `shared_validations` – znajduje się walidacja, która jest używana w paru miejscach,
- `web` – w tym miejscu są zdefiniowane widoki (jako podfoldery). Każdy widok posiada swój główny plik (np. wizyty.php), plik css (np. wizyta.css) oraz plik ze zdefiniowanymi funkcjami (np. wizyty_actions.php). Następnie mamy dodatkowe pliki, np. do edytowania czy tworzenia nowego zasobu. Co ważne, funkcje napisane w PHP, style CSS czy kod HTML w tym podfolderach są odpowiedzialne tylko za funkcjonalności danego folderu (innymi słowy, w `wizyty_actions.php` nie ma funkcjonalności związanej z logowaniem, i tak dalej).

Wszystkie style CSS oraz kod JavaScriptowy (zdefiniowany w folderze `js/`) są ładowane w `index.php`, czyli w layoucie.



```
index.php
You, 3 hours ago | 1 author (You)
<?php
require('index_actions.php');
// require_logged_in_user();

<!DOCTYPE html>
<html> You, 7 days ago * define basic index page (wip)
<head>
<meta charset='utf-8'>
<meta http-equiv='X-UA-Compatible' content='IE=edge'>
<title>Przychodnia</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<link rel="stylesheet" href="index.css">
<link rel="stylesheet" href="../lekarze/lekarze.css">
<link rel="stylesheet" href="../pacjenci/pacjenci.css">
<link rel="stylesheet" href="../wizyty/wizyta.css">
<link rel="stylesheet" href="../profil/profil.css">
<script src="../../js/searcher.js"></script>
<script src="../../js/datetimelocal_validate.js"></script>
</head>
<body>
<div class="index_page">
<!-- MENU -->
<div class="index_menu">
<h1>Przychodnia</h1>
<div class="index_menu__items">
<?php echo get_menu_items(); ?>
</div>
</div>
<!-- CONTENT -->
<div class="index_content">
<div class="index_logout">
<span class="index_fullname"><?php echo get_full_name(); ?></span>
<form method="POST" action="index_actions.php">
<button type="submit" name="log_out">Wyloguj</button>
</form>
</div>
<div>
<?php render_content_from_params() ?>
</div>
</div>
</div>
</body>
</html>
```