

Dokumentacja Projektu Filip Dziedzic

Zarządzanie Zmianami pracowników

Łączenie z baza

Kod

```
<?php
function db_connect() {
    $serwer = mysqli_connect("localhost", "root", "")
        or exit("Nie można połączyć się z serwerem bazy danych");

    mysqli_select_db($serwer, "zarzadzanie_harmonogramem")
        or exit("Nie można połączyć się z bazą zarzadzanie_harmonogramem");
    return $serwer;
}
?>
```

`$serwer = mysqli_connect("localhost", "root", "")` - jest to połączenie do lokalnego serwera bazy danych .

`mysqli_select_db($serwer, "zarzadzanie_harmonogramem")` - wybranie odpowiedniej bazy danych

Logowanie

Logowanie

Nazwa użytkownika

Hasło

Kod widoku

```
<?php
session_start();
$error = isset($_SESSION['error']) ? $_SESSION['error'] : '';
unset($_SESSION['error']);
?>

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Logowanie</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style_index.css">
</head>
<body>
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-6">
                <h2 class="text-center mt-5">Logowanie</h2>
                <form method="post" action="../../PHP_Logic/login_logic.php">
                    <div class="mb-3">
                        <label for="username" class="form-label">Nazwa użytkownika</label>
                        <input type="text" class="form-control" id="username" name="username" required>
                    </div>
                    <div class="mb-3">
                        <label for="password" class="form-label">Hasło</label>
                        <input type="password" class="form-control" id="password" name="password" required>
                    </div>
                    <?php if ($error)
                    {
                        echo ' <div class="alert alert-danger" role="alert">';
                        echo $error;
                        echo "<br/>";
                    }

                    <button type="submit" class="btn btn-primary w-100">Zaloguj się</button>
                </form>
            </div>
        </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

session_start() - start sesji lub jej dnowienie

\$error = isset(\$_SESSION['error']) ? \$_SESSION['error'] : ''; -

sprawdzanie czy istnieje zmienna error, jeśli tak to już ustawia powiązanie do backendu

unset(\$_SESSION['error']); - resetowanie zmiennej aby komunikat nie pokazywał się cały czas

Kod logiki działania

w tym kodzie sprawdzamy czy użytkownik istnieje w bazie oraz czy podane jego dane są zgodne z bazą danych.

```
<?php
require('database_connection.php');
session_start();
//sprawdza czy pola są wypełnione w formularzu
if (isset($_POST['username']) && isset($_POST['password'])) {
    $login = $_POST['username'];
    $password = $_POST['password'];

    $conn = db_connect(); // łączenie
    $query = "SELECT id, haslo FROM uzytkownicy WHERE login = '$login'"; // zapytanie do bazy danych
    $result = mysqli_query($conn, $query); // wykonanie zapytania
    //sprawdza czy wynik zapytania istnieje
    if ($result && mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result); // pobiera wynik jako pierwszy wiersz tablicy asocjacyjnej

        if ($password === $row['haslo']) {
            $_SESSION['user_id'] = $row['id']; // ustawienie id sesji
            $_SESSION['login'] = $login; // ustawienie loginu sesji
            header('Location: ../sites/index.php'); // po poprawnym zalogowaniu przekierowuje do strony index.php
            exit();
        } else {
            $_SESSION['error'] = "Nieprawidłowy login lub hasło."; // error który był użyty w wioku jako bład
        }
    } else {
        $_SESSION['error'] = "Nieprawidłowy login lub hasło."; // error który był użyty w wioku jako bład
    }

    mysqli_close($conn); // zamknięcie połączenia
    header('Location: ../sites/login.php'); // powrót do strony logowania
    exit();
}
?>
```

Strona główna aplikacji

Menu

Harmonogram Grup

Aktualne zastępstwa

Nadgodziny

Dodaj zastępstwo

Dodaj

Nieprzypisane zastępstwo

Zalogowany jako: Filip Dziedzic

UstawieniaWyloguj

Wybierz datę początkową:09.02.2025

Pokaż

Grupa	Niedziela 2025-02-09	Poniedziałek 2025-02-10	Wtorek 2025-02-11	Środa 2025-02-12	Czwartek 2025-02-13	Piątek 2025-02-14	Sobota 2025-02-15
Grupa 1		Anna Nowak 08:00:00 - 16:00:00	Ala Kot 06:30:00 - 08:00:00 Anna Nowak 08:00:00 - 16:00:00				
Grupa 2		Piotr Teściak 08:00:00 - 16:00:00	Piotr Teściak 08:00:00 - 16:00:00	Tomasz Lipa 15:30:00 - 20:30:00 Ala Kot 15:29:00 - 19:30:00 Zastępstwo za: Tomasz Lipa			

W tej części wyświetlony zostaje aktualne zastępstwa oraz harmongram
Kod widoku

```
<?php
require('../PHP_Logic/sidebar_logic.php'); // logika działania sidebar
require('../PHP_Logic/index_logic.php'); // logika działania index
if (!isset($_SESSION['user_id'])) {
    echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
    exit();
} // warunek który sprawdza czy użytkownik jest zalogowany

?>

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Harmonogram Grup</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" />
    <link rel="stylesheet" href="style_index.css" />
</head>
<body>
<div class="sidebar d-flex flex-column">
    <h4 class="text-center">Menu</h4>
    <a href="index.php">Harmonogram Grup</a>
    <?php if(isAdmin()) // z sidebar sprawdza role
    |     echo '<a href="AktualneZastepstwa.php">Aktualne zastępstwa</a>';
    ?>
    <a href="Nadgodziny.php">Nadgodziny</a>
    <a href="dodaj_zastepstwo.php">Dodaj zastępstwo</a>
    <?php
    if([isAdmin()])
    | {
    |     echo "<a href='dodaj.php'>Dodaj</a>";
    | }
    ?>
    <div class="accordion mt-3" id="notificationAccordion">
        <?php
        if(!isAdmin())
        | {
        |     displaySubstitutionsAccept(); // wyswietla zastepstwa do akceptacji
        | }
        ?>
    </div>
</div>
</body>
</html>
```

```

        <?php
        if(isAdmin())
        {
            displaySubstitutionsPending(); // wyswietla zastepstwa oczekujace
        }
        ?>
    </div>
    <div class="user-info mt-auto">
        <p>Zalogowany jako: <?php WhoAmI() // wyswietla jako kot jestesmy zalogowani?></p>
        <div class="links">
            <a href="ustawienia.php">Ustawienia</a>
            <form method="post" action="../../PHP_Logik/logout.php" style="display:inline;">
                <button type="submit" class="btn btn-link">Wyloguj</button>
            </form>
        </div>
    </div>
</div>
<div class="content">
    <h2 class="text-center">Harmonogram Grup</h2>
    <form method="get" action="index.php" class="mb-4">
        <label for="startDate">Wybierz datę początkową:</label>
        <input type="date" id="startDate" name="startDate" value="<?php echo $startDate; // utrzymuje wybrana date ?>" />
        <button type="submit" class="btn btn-primary">Pokaż</button>
    </form>
    <div class="table-responsive">
        <?php displaySchedule($startDate); // wyswietla harmonogram statujac od podaniej daty - ?>
    </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

funkcja isAdmin sprawdza role zalogowanego użytkownika. WhoAmI sprawdza danego zalogowanego użytkownika aby wyświetlić info na ten temat. displaySchedule wyświetla harmonogram.

index_logic logika działania strony głównej

```

<?php
require_once('../PHP_Logik/database_connection.php');

function getSchedule($startDate = null) { // funkcja pobierająca harmonogram
    $conn = db_connect(); // łączenie
    if ($startDate === null) { // warunek sprawdzający czy została podana data
        $startDate = date('Y-m-d'); // ustawienie daty w odpowiedni schemat
    }
    $query = "SELECT DISTINCT g.nazwa AS grupa, u.imie, u.nazwisko, h.dzien, h.godzina_od, h.godzina_do
    FROM harmonogram h
    JOIN uzytkownicy u ON h.id_pracownika = u.id
    JOIN pracownik_grupa pg ON u.id = pg.id_pracownika
    JOIN grupy g ON pg.id_grupy = g.id
    ORDER BY g.nazwa, FIELD(h.dzien, 'Poniedziałek', 'Wtorek', 'Środa', 'Czwartek', 'Piątek', 'Sobota', 'Niedziela'), h.godzina_od";
    // zapytanie do bazy

    $result = mysqli_query($conn, $query);
    $schedule = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $schedule[] = $row;
        }
    }
    mysqli_close($conn);
    return $schedule; // zwraca tablice wierszy wyników
}

function getSubstitutions() { // funkcja pobiera zastępstwa
    $conn = db_connect();
    $query = "SELECT DISTINCT z.id, z.data_zastepstwa AS data, z.godzina_od, z.godzina_do, g.nazwa AS grupa,
    u1.imie AS imie_potrzebujacego, u1.nazwisko AS nazwisko_potrzebujacego,
    u2.imie AS imie_zastepujacego, u2.nazwisko AS nazwisko_zastepujacego
    FROM zastepstwa z
    JOIN uzytkownicy u1 ON z.id_pracownika_proszacego = u1.id
    LEFT JOIN uzytkownicy u2 ON z.id_pracownika_zastepujacego = u2.id
    LEFT JOIN pracownik_grupa pg ON u1.id = pg.id_pracownika
    LEFT JOIN grupy g ON pg.id_grupy = g.id
    WHERE z.status = 'zatwierdzone'";
    $result = mysqli_query($conn, $query);
    $substitutions = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $substitutions[] = $row;
        }
    }
    mysqli_close($conn);
    return $substitutions;
}

```

```

$query = "SELECT DISTINCT g.nazwa AS grupa, u.imie, u.nazwisko,
h.dzien, h.godzina_od, h.godzina_do
    FROM harmonogram h
    JOIN uzytkownicy u ON h.id_pracownika = u.id
    JOIN pracownik_grupa pg ON u.id = pg.id_pracownika
    JOIN grupy g ON pg.id_grupy = g.id
    ORDER BY g.nazwa, FIELD(h.dzien, 'Poniedziałek',
'Wtorek', 'Środa', 'Czwartek', 'Piątek', 'Sobota', 'Niedziela'),
h.godzina_od";

```

opis zapytania

- Distinct złoży do kasowania duplikujących się rekordów
- Field sortuje wyniki w odpowiedniej kolejności dla pola dzien

```
function displaySchedule($startDate = null) {
    $schedule = getSchedule($startDate); // pobiera harmonogram od konkretnej daty
    $substitutions = getSubstitutions(); // pobiera zastępstwa
    $daysOfWeek = ['Poniedziałek', 'Wtorek', 'Środa', 'Czwartek', 'Piątek', 'Sobota', 'Niedziela'];
    $groupedSchedule = []; // tablica grupująca harmonogram
    $groupedSubstitutions = []; // tablica grupująca zastępstwa

    foreach ($schedule as $entry) { // grupuje harmonogram po grupie i dniu zapisuje entry do tablicy tak aby były grupowane po grupie i dniu
        $groupedSchedule[$entry['grupa']][$entry['dzien']] = $entry;
    }

    foreach ($substitutions as $entry) {
        $groupedSubstitutions[$entry['grupa']][$entry['data']] = $entry;
    }

    echo '<table class="table schedule-table mt-4">';
    <thead>
    <tr>
    <th>Grupa</th>;
    $currentDate = strtotime($startDate); // zamieniam na unix dla łatwiejszej operacji
    for ($i = 0; $i < 7; $i++) {
        $dayOfWeek = $daysOfWeek[date('N', $currentDate) - 1]; // pobieranie dnia tygodnia
        echo '<th>' . $dayOfWeek . '<br>' . date('Y-m-d', $currentDate) . '</th>'; // zmienia unix na date
        $currentDate = strtotime('+1 day', $currentDate); // opcja do iteracji po kolejnych dniach
    }
}
```

Funkcja displaySchedule jest odpowiedzialna za wyświetlanie widoku harmonogramu oraz zastępstw. Foreach grupuje harmonogram po grupie i dniu i zapisuje do tablic tak że A [poniedziałek [imię nazwisko itd]]. strtotime zmienia datę na zawartość liczby formatu unix aby móc bez problemu przechodzić pomiędzy dniami następnie for pozwala ustalić jaki to dzień tygodnia.

```
<tbody>; // generacja wierszy tabeli dla harmonogramu i zastępstw
foreach ($groupedSchedule as $group => $days) {
    echo '<tr>';
    echo '<td>' . $group . '</td>';
    $currentDate = strtotime($startDate);
    for ($i = 0; $i < 7; $i++) {
        $dayOfWeek = $daysOfWeek[date('N', $currentDate) - 1];
        $currentDateStr = date('Y-m-d', $currentDate);
        echo '<td>';
        if (isset($days[$dayOfWeek])) { // sprawdzenie czy dane są dla tego dnia
            $uniqueEntries = []; // tablica unikalnych wartości
            foreach ($days[$dayOfWeek] as $entry) { // iteracja dla po wszystkich wydarzeniach dla danego dnia tygodnia
                $entryKey = $entry['imie'] . $entry['nazwisko'] . $entry['godzina_od'] . $entry['godzina_do']; // unikalny zapis
                if (in_array($entryKey, $uniqueEntries)) { // sprawdzenie czy wartość już jest w uniqueEntries
                    $uniqueEntries[] = $entryKey; // zapisanie żeby zapamiętać że już ten wpis był
                    echo '<div class="user-box">';
                    echo '<strong>' . $entry['imie'] . ' ' . $entry['nazwisko'] . '</strong><br />' . $entry['godzina_od'] . ' - ' . $entry['godzina_do'];
                    echo '</div>';
                }
            }
        }
        if (isset($groupedSubstitutions[$group][$currentDateStr])) { // sprawdza czy w tablicy są odpowiednio czyli dla danej grupy i dnia są zastępstwa
            foreach ($groupedSubstitutions[$group][$currentDateStr] as $entry) {
                echo '<div class="user-box subs" @var mixed $entry["data"] as $entry["data"]>';
                echo '<strong>' . $entry['imie_zastepujacego'] . ' ' . $entry['nazwisko_zastepujacego'] . '</strong><br />' . $entry['godzina_od'] . ' - ' . $entry['godzina_do'];
                echo '<br /><em>Zastępstwo za: ' . $entry['imie_potrzebujacego'] . ' ' . $entry['nazwisko_potrzebujacego'] . '</em>';
            }
        }
        echo '</td>';
        $currentDate = strtotime('+1 day', $currentDate);
    }
    echo '</tr>';
}
echo '</tbody> </table>';
}

$startDate = isset($_GET['startDate']) ? $_GET['startDate'] : date('Y-m-d'); // obsługa daty początkowej
?>
```

generacja wierszy tabeli dla harmonogramu i zastępstw. funkcja wypisuje grupę następnie iteruje po dniach tygodnia żeby ustalić w jakim dniu się znajdujemy następnie sprawdza czy istnieją jakieś zajęcia w tym dniu dalej tworzy tablice do przechowywania unikalnych wartości do zapisania że już zostało to wyświetlone następnie robi iterację po wszystkich zajęciach od tego dnia zapisuje rekord z wartościami i jeśli klucz nie znajduje się w tablicy unikalnych wartości przypisuje i wyświetla wartości. dla zastępstw sprawdza tablice czy są tam odpowiednie wartości i for'em wypisuje jeśli to prawda potem dodajemy jeden dzień aby przejść po kolejnym dniu tygodnia. startDate to kontrola do ustalenia przez użytkownika startowej daty.

Aktualne zastępstwa widok

<div>Menu</div> <div>Harmonogram Grup</div> <div>Aktualne zastępstwa</div> <div>Nadgodziny</div> <div>Dodaj zastępstwo</div> <div>Dodaj</div> <div>Nieprzypisane zastępstwo ⓘ ⌵</div>	<div>Aktualne Zastępstwa</div> <table><tr><th>Data</th><th>Grupa</th><th>Osoba potrzebująca zastępstwa</th><th>Osoba zastępująca</th></tr><tr><td>2025-02-12</td><td>Grupa 2</td><td><div>Tomasz Lipa 15:29:00 - 19:30:00</div></td><td><div>Ala Kot 15:29:00 - 19:30:00</div></td></tr></table>	Data	Grupa	Osoba potrzebująca zastępstwa	Osoba zastępująca	2025-02-12	Grupa 2	<div>Tomasz Lipa 15:29:00 - 19:30:00</div>	<div>Ala Kot 15:29:00 - 19:30:00</div>
Data	Grupa	Osoba potrzebująca zastępstwa	Osoba zastępująca						
2025-02-12	Grupa 2	<div>Tomasz Lipa 15:29:00 - 19:30:00</div>	<div>Ala Kot 15:29:00 - 19:30:00</div>						
Zalogowany jako: Filip Dziedzic Ustawienia Wyloguj									

Kod widoku nie różni się bardzo od indexu a więc podaje tylko różnice

```
<?php
require('../PHP_Logik/sidebar_logic.php');
require('../PHP_Logik/AktualneZastepstwa_logic.php');
if (!isset($_SESSION['user_id'])) {
    echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
    exit();
}
```

inne wymagane pliki

oraz w content inna funkcja do wyświetlania

```
<tbody>
|   <?php displaySubstitutions(); ?>
</tbody>
```

Kod Aplikacji

```
function getSubstitutions() { // pobieranie wartosci z bazy danych dotyczace zastepstw
    $conn = db_connect();
    $query = "SELECT DISTINCT z.id, z.data_zastepstwa AS data, z.godzina_od, z.godzina_do, g.nazwa AS grupa,
        u1.imie AS imie_potrzebujacego, u1.nazwisko AS nazwisko_potrzebujacego,
        u2.imie AS imie_zastepujacego, u2.nazwisko AS nazwisko_zastepujacego
    FROM zastepstwa z
    JOIN uzytkownicy u1 ON z.id_pracownika_proszacego = u1.id
    LEFT JOIN uzytkownicy u2 ON z.id_pracownika_zastepujacego = u2.id
    LEFT JOIN pracownik_grupa pg ON u1.id = pg.id_pracownika
    LEFT JOIN grupy g ON pg.id_grupy = g.id
    WHERE z.status = 'zatwierdzone'";
    $result = mysqli_query($conn, $query);
    $substitutions = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $substitutions[] = $row;
        }
    }
    mysqli_close($conn);
    return $substitutions;
}
```

funkcja odpowiedzialna za pobieranie z bazy danych aktualnych zastępstw które za o statusie zatwierdzone .


```
function displaySubstitutions() { //wyswietlanie zastepstw
    $substitutions = getSubstitutions();
    $uniqueSubstitutions = [];
    foreach ($substitutions as $substitution) { //zapewniwie o niepowtarzaniu sie zastepstw
        $uniqueKey = $substitution['id'];
        if (!isset($uniqueSubstitutions[$uniqueKey])) {
            $uniqueSubstitutions[$uniqueKey] = $substitution;
        }
    }
    foreach ($uniqueSubstitutions as $substitution) {
        echo '<tr> <td>' . $substitution['data'] . '</td>';
        echo '<td>' . $substitution['grupa'] . '</td> <td>';
        echo '<div class="user-box">';
        echo '<strong>' . $substitution['imie_potrzebujacego'] . ' '
            . $substitution['nazwisko_potrzebujacego'] . '</strong><br />' . $substitution['godzina_od'] . ' - ' . $substitution['godzina_do'];
        echo '</div> </td><td>';
        echo '<div class="user-box">';
        echo '<strong>' . $substitution['imie_zastepujacego'] . ' '
            . $substitution['nazwisko_zastepujacego'] . '</strong><br />' . $substitution['godzina_od'] . ' - ' . $substitution['godzina_do'];
        echo '</div> </td> </tr>';
    }
}
?>
```

wyświetlanie z uwzględnianiem unikalności.

Nadgodziny widok

Menu

- Harmonogram Grup
- Aktualne zastępstwa
- Nadgodziny
- Dodaj zastępstwo
- Dodaj
- Nieprzydzielone zastępstwo

Nadgodziny

Wyszukaj użytkownika: Od: Do:

Ala Kot
Całkowite nadgodziny: 4

Data	Liczba godzin
2025-02-12	4

Zalogowany jako: Filip Dziedzic

[Ustawienia](#) [Wyloguj](#)

Kod w widoku różniący się od index

```
<div id="overtimeContainer" class="row">
|   <?php displayOvertimeCards($searchTerm, $startDate, $endDate); ?>
</div>
```

```
1  <?php
2  require('../PHP_Logik/sidebar_logic.php');
3  require('../PHP_Logik/nadgodziny_logic.php');
4  if (!isset($_SESSION['user_id'])) {
5      echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
6      exit();
7  }
8
9  ?>
```

Kod logiki

```

<?php
require_once('database_connection.php');

function getOvertimeData($searchTerm = '', $startDate = '', $endDate = '') { // pobieranie danych o nadgodzinach (wyszukiwanie)
    $conn = db_connect(); // laczenie
    $user_id = $_SESSION['user_id'];
    $isAdmin = $_SESSION['rola'] === 'admin';

    $query = "SELECT u.id, CONCAT(u.imie, ' ', u.nazwisko) AS full_name, SUM(n.liczba_godzin) AS total_hours
    FROM nadgodziny n
    JOIN uzytkownicy u ON n.id_pracownika = u.id"; // baza zapytania
    $conditions = []; // tablica warunkow
    if (!empty($searchTerm)) { // jesli nie puste dodaje do tablicy
        $conditions[] = "CONCAT(u.imie, ' ', u.nazwisko) LIKE '%$searchTerm%'";
    }
    if (!empty($startDate)) {
        $conditions[] = "n.data >= '$startDate'";
    }
    if (!empty($endDate)) {
        $conditions[] = "n.data <= '$endDate'";
    }
    if (!$isAdmin) {
        $conditions[] = "n.id_pracownika = '$user_id'";
    }
    if (!empty($conditions)) {
        $query .= " WHERE " . implode(' AND ', $conditions); // łącznie warunków implode łączy wartosc w tablicy w jedna calosc
    }
    $query .= " GROUP BY u.id";
    $result = mysqli_query($conn, $query);
    $overtimeData = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $overtimeData[] = $row;
        }
    }
    mysqli_close($conn);
    return $overtimeData;
}

```

W tej funkcji pobieramy dane dotyczące wyszukiwanego użytkownika tzn jest ddo wyświetlana konkretnego użytkownika. za pomocą if sprawdzam czy jakiś warunek został podany jeśli nie dodaje go do tablic jeśli tak dodaje następnie łączę wszystkie warunki do zapytania i wysyłam.

```

function getOvertimeDetails($userId, $startDate = '', $endDate = '') { // pobieranie ilosci godzin
    $conn = db_connect(); // laczenie
    $query = "SELECT data, liczba_godzin FROM nadgodziny WHERE id_pracownika = '$userId'"; // zapytanie
    if (!empty($startDate)) {
        $query .= " AND data >= '$startDate'";
    }
    if (!empty($endDate)) {
        $query .= " AND data <= '$endDate'";
    }
    $result = mysqli_query($conn, $query);
    $overtimeDetails = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $overtimeDetails[] = $row;
        }
    }
    mysqli_close($conn);
    return $overtimeDetails;
}

```

Funkcja która pobiera datę oraz liczbę godzin od wybranego użytkownika po id.

```
function displayOvertimeCards($searchTerm = '', $startDate = '', $endDate = '') { // wyswietlanie kart z info
    $overtimeData = getOvertimeData($searchTerm, $startDate, $endDate);
    if (!empty($overtimeData)) {
        foreach ($overtimeData as $user) { // informacje o userze
            echo '<div class="col-md-4 mb-4 user-overtime" data-username="' . strtolower($user['full_name']) . '">';
            <div class="card">
            <div class="card-body">;
            echo '<h5 class="card-title">' . $user['full_name'] . '</h5>';
            echo '<p class="card-text">Całkowite nadgodziny: ' . $user['total_hours'] . '</p>';
            echo '<div class="table-responsive"> <table class="table schedule-table mt-4">
            <thead><tr>
            <th>Data</th>
            <th>Liczba godzin</th>
            </tr>
            </thead> <tbody>';
            $overtimeDetails = getOvertimeDetails($user['id'], $startDate, $endDate);
            foreach ($overtimeDetails as $detail) { //info o nadgodzinach
                echo '<tr> <td>' . $detail['data'] . '</td>';
                echo '<td>' . $detail['liczba_godzin'] . '</td>';
                echo '</tr>';
            }
            echo '</tbody>
            </table>
            </div></div></div></div>';
        }
    } else {
        echo '<p class="text-center">Brak danych o nadgodzinach.</p>';
    }
}
```

Wyświetlanie.

```
$searchTerm = isset($_GET['searchUser']) ? $_GET['searchUser'] : '';
$startDate = isset($_GET['startDate']) ? $_GET['startDate'] : '';
$endDate = isset($_GET['endDate']) ? $_GET['endDate'] : '';
```

sprawdzenie czy pola są ustawione.

Dodaj Zastępstwo

Menu

- Harmonogram Grup
- Aktualne zastępstwa
- Nadgodziny
- Dodaj zastępstwo
- Dodaj
- Nieprzypisane zastępstwo

Zalogowany jako: Filip Dziedzic
Ustawienia [Wyloguj](#)

Dodaj zastępstwo

Data:

Godzina Od:

Godzina Do:

Dodaj Zastępstwo

Różnie w kodzie od index

```

1 <?php
2 require('../PHP_Logik/sidebar_logic.php');
3 require('../PHP_Logik/dodaj_zastepstwo_logic.php');
4 if (!isset($_SESSION['user_id'])) {
5     echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
6     exit();
7 }
8 ?>
9 <!DOCTYPE html>

```

```

<div class="content">
    <div class="form-container">
        <h2 class="text-center">Dodaj zastępstwo</h2>
        <form method="POST" action="../PHP_Logik/dodaj_zastepstwo_logic.php">
            <div class="form-group">
                <label for="date">Data:</label>
                <input type="date" id="date" name="date" class="form-control" required>
            </div>
            <div class="form-group">
                <label for="start_time">Godzina Od:</label>
                <input type="time" id="start_time" name="start_time" class="form-control" required>
            </div>
            <div class="form-group">
                <label for="end_time">Godzina Do:</label>
                <input type="time" id="end_time" name="end_time" class="form-control" required>
            </div>
            <button type="submit" class="btn btn-primary">Dodaj Zastępstwo</button>
        </form>
    </div>
</div>
</div>

```

Kod

```

<?php
if (session_status() == PHP_SESSION_NONE) { //sprawdzenie czy sesja działa
    session_start();
}
require_once('../PHP_Logik/database_connection.php');

if (isset($_POST['date']) && isset($_POST['start_time']) && isset($_POST['end_time'])) {
    $date = $_POST['date'];
    $startTime = $_POST['start_time'];
    $endTime = $_POST['end_time'];
    $requestingEmployeeId = $_SESSION['user_id'];

    $conn = db_connect();

    $query = "INSERT INTO zastepstwa (data_zastepstwa, godzina_od, godzina_do, id_pracownika_proszacego, status)
        VALUES ('$date', '$startTime', '$endTime', '$requestingEmployeeId', 'oczekujące')";

    if (mysqli_query($conn, $query)) {
        header("Location: ../sites/dodaj_zastepstwo.php");
        exit();
    } else {
        echo "Błąd: " . mysqli_error($conn);
    }

    mysqli_close($conn);
}
?>

```

prosty formularz który dodaj do bazy danych odpowiednie wartości

Widok Dodaj

Menu

Harmonogram Grup

Aktualne zastępstwa

Nadgodziny

Dodaj zastępstwo

Dodaj

Nieprzypisane zastępstwo

Zalogowany jako: Filip Dziedzic

UstawieniaWyloguj

Dodaj Pracownika

Imię

Nazwisko

Login

Filip

Hasło

Grupy

☐ Grupa 1

☐ Grupa 2

Dodaj Pracownika

Dodaj Godziny Pracy

Pracownik

Anna Nowak

Dzień Tygodnia

Poniedziałek

Grupa

Dodaj Grupę

Nazwa Grupy

Dodaj Grupę

Zarządzaj Pracownikiem

Pracownik

Anna Nowak

Usuń Pracownika

różnice od index

```
<?php
require('../PHP_Logik/sidebar_logik.php');
require('../PHP_Logik/dodaj_logik.php');

if (!isset($_SESSION['user_id'])) {
    echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
    exit();
}

$groups = getGroups();

?>
```

\$groups pobiera z funkcji informacje na temat grup z bazy danych i zostaje do nich przypisane. W content znajdują się formularze z odpowiednimi funkcjami.

```
<?php
    wrtieGroups($groups);
?>
```

wypisuje grupy

```
<?php getUsers(); ?>
```

pobiera użytkowników

Logika działania

```

<?php
require_once('database_connection.php');

function getGroups() { // pobieranie grup
    $conn = db_connect();
    $query = "SELECT id, nazwa FROM grupy";
    $result = mysqli_query($conn, $query);
    $groups = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $groups[] = $row;
        }
    }
    mysqli_close($conn);
    return $groups;
}

function writeGroups($groups) { // wypisanie
    foreach ($groups as $group) {
        echo '<div class="form-check">';
        echo '<input class="form-check-input" type="checkbox" name="groups[]" value="" . $group['id'] . "" id="group' . $group['id'] . "">';
        echo '<label class="form-check-label" for="group' . $group['id'] . "">' . $group['nazwa'] . '</label>';
        echo '</div>';
    }
}

//dodawanie pracownika
if (isset($_POST['firstName']) && isset($_POST['lastName']) && isset($_POST['login']) && isset($_POST['password'])) { // sprawdzanie czy dane są wpisane
    $firstName = $_POST['firstName'];
    $lastName = $_POST['lastName'];
    $login = $_POST['login'];
    $password = $_POST['password'];
    $groups = isset($_POST['groups']) ? $_POST['groups'] : []; // sprawdza czy istnieje grupy

    $conn = db_connect();

    // Sprawdzenie, czy login już istnieje
    $query = "SELECT id FROM uzytkownicy WHERE login = '$login'";
    $result = mysqli_query($conn, $query);

    if ($result && mysqli_num_rows($result) == 0) {
        $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
        $query = "INSERT INTO uzytkownicy (imie, nazwisko, login, haslo, rola) VALUES ('$firstName', '$lastName', '$login', '$hashedPassword', 'pracownik')";
        if (mysqli_query($conn, $query)) {
            $userId = mysqli_insert_id($conn);

            // Dodanie grup do pracownika
            foreach ($groups as $groupId) {
                $query = "INSERT INTO pracownik_grupa (id_pracownika, id_grupy) VALUES ('$userId', '$groupId')";
                mysqli_query($conn, $query);
            }
        }
    }

    //java script
    echo "<script>alert('Pracownik został pomyślnie dodany.');

```

```

if (isset($_POST['groupName'])) { //dodanie grup
    $groupName = $_POST['groupName'];

    $conn = db_connect();

    $query = "SELECT id FROM grupy WHERE nazwa = '$groupName'";
    $result = mysqli_query($conn, $query);

    if ($result && mysqli_num_rows($result) == 0) {

        $query = "INSERT INTO grupy (nazwa) VALUES ('$groupName')";
        if (mysqli_query($conn, $query)) {
            echo "<script>alert('Grupa została pomyślnie dodana.');

```

Zarządzanie zastępstwami widok

Menu

Harmonogram Grup

Aktualne zastępstwa

Nadgodziny

Dodaj zastępstwo

Dodaj

Nieprzypisane zastępstwo

Zalogowany jako: Filip Dziedzic

Ustawienia

Wyloguj

Nieprzypisane zastępstwa

Data	Godzina Od	Godzina Do	Potrzebujący	Mozliwe Zastępstwa
2025-02-04	08:30:00	16:00:00	Anna Nowak	<div><div><input type="radio"/> Ala Kot</div><div><input type="radio"/> Tomasz Lipa</div><div>Wyślij</div><div>Odrzuć</div></div>
2025-02-12	20:20:00	21:20:00	Filip Dziedzic	<div><div><input type="radio"/> Anna Nowak</div><div><input type="radio"/> Piotr Teściak</div><div><input type="radio"/> Ala Kot</div><div>Wyślij</div><div>Odrzuć</div></div>

Kod Widoku

```
<?php
require('../PHP_Logik/sidebar_logic.php');
require('../PHP_Logik/ZaradzajZastepstwami_logic.php');
if (!isset($_SESSION['user_id'])) {
    echo "Proszę się <a href='login.php'>zalogować</a>, aby uzyskać dostęp do tej strony.";
    exit();
}
?>

<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Harmonogram Grup</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" />
    <link rel="stylesheet" href="style_index.css" />
</head>
<body>
<div class="sidebar d-flex flex-column">...
</div>
<div class="content">
<div class="table-responsive">
<h2 class="text-center">Nieprzypisane zastępstwa</h2>
    <?php displayPendingSubstitutions(); ?>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

za pomocą require ustawienie potrzebnych plików
displayPendingSubstitutions wyświetla oczekuje zastępstwa
Kod działania


```

<?php
require_once('../PHP_Logik/database_connection.php');

function getPendingSubstitutions() { // pobiera z bazy danych zastępstwa
    $conn = db_connect();
    $query = "SELECT DISTINCT z.id, z.data_zastepstwa AS data, z.godzina_od, z.godzina_do, g.nazwa AS grupa,
        u1.id AS id_pracownika_proszacego, u1.imie AS imie_potrzebujacego, u1.nazwisko AS nazwisko_potrzebujacego,
        u2.imie AS imie_zastepujacego, u2.nazwisko AS nazwisko_zastepujacego
    FROM zastepstwa z
    JOIN uzytkownicy u1 ON z.id_pracownika_proszacego = u1.id
    LEFT JOIN uzytkownicy u2 ON z.id_pracownika_zastepujacego = u2.id
    LEFT JOIN pracownik_grupa pg ON u1.id = pg.id_pracownika
    LEFT JOIN grupy g ON pg.id_grupy = g.id
    WHERE z.status = 'oczekujące' ";
    $result = mysqli_query($conn, $query);
    $pendingSubstitutions = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $pendingSubstitutions[] = $row;
        }
    }
    mysqli_close($conn);
    return $pendingSubstitutions;
}

```

getPendingSubstitutions pobiera aktualne zastępstwa z statusem oczekujące

```

function getAvailableEmployees($date, $startTime, $endTime, $requestingEmployeeId) { // pokazuje liste pracowników ktorzy moga zastapic
    $conn = db_connect();

    $daysOfWeek = [ // tablica do dni tygodnia
        'Monday' => 'Poniedziałek',
        'Tuesday' => 'Wtorek',
        'Wednesday' => 'Środa',
        'Thursday' => 'Czwartek',
        'Friday' => 'Piątek',
        'Saturday' => 'Sobota',
        'Sunday' => 'Niedziela'
    ];
    $dayOfWeek = $daysOfWeek[date('l', strtotime($date))]; // przypisanie odpowiedniej wartosci do dnia tygodnia zwraca pelna nazwe
    // $query = "SELECT u.id, u.imie, u.nazwisko
    // FROM uzytkownicy u
    // WHERE u.id != $requestingEmployeeId
    // AND u.id NOT IN (
    //     SELECT h.id_pracownika
    //     FROM harmonogram h
    //     WHERE h.dzien = '$dayOfWeek'
    //     AND NOT ('$endTime' <= h.godzina_od OR '$startTime' >= h.godzina_do)
    // ) AND rola = 'pracownik'";
    $query = "SELECT u.id, u.imie, u.nazwisko
    FROM uzytkownicy u
    WHERE u.id NOT IN (
        SELECT h.id_pracownika
        FROM harmonogram h
        WHERE h.dzien = '$dayOfWeek'
        AND NOT ('$endTime' <= h.godzina_od OR '$startTime' >= h.godzina_do)
    ) AND rola = 'pracownik'"; // zapytanie

    $result = mysqli_query($conn, $query);
    $availableEmployees = [];
}

```

Następnie w funkcji getAvailableEmployees pobieramy pracowników którzy mogą pracować w ten dzień. tablica zmienia tworzona po to aby zmienić z angielskich nazw na polskie ponieważ w bazie danych są polskie nazwy. następnie pobierany jest dzień tygodnia za pomocą z daty i zmieniony na polski. po czym wykonywane jest zapytanie

```

$result = mysqli_query($conn, $query);
$availableEmployees = [];

if ($result && mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $availableEmployees[] = $row;
    }
}

mysqli_close($conn);
return $availableEmployees;
}

```

```

function displayPendingSubstitutions() {
    $pendingSubstitutions = getPendingSubstitutions();
    echo '<form method="POST" action="ZarządzajZastępstwami.php">
    <table class="table table-striped">
    <thead>
    <tr><th>Data</th>
    <th>Godzina Od</th>
    <th>Godzina Do</th>
    <th>Potrzebujący</th>
    <th>Możliwe Zastępstwa</th></tr>
    </thead> <tbody>';
    foreach ($pendingSubstitutions as $substitution) { // iteracja po wszystkich potrzebujących
        echo '<tr> <td>' . $substitution['data'] . '</td>';
        echo '<td>' . $substitution['godzina_od'] . '</td>';
        echo '<td>' . $substitution['godzina_do'] . '</td>';
        echo '<td>' . $substitution['imie_potrzebujacego'] . ' ' . $substitution['nazwisko_potrzebujacego'] . '</td> <td>';
        //załadowanie dostępnych pracowników
        $availableEmployees = getAvailableEmployees($substitution['data'], $substitution['godzina_od'], $substitution['godzina_do'], $substitution['id_pracownika_proszacego']);
        if (!empty($availableEmployees)) { // sprawdzenie czy ktos jest
            foreach ($availableEmployees as $employee) {
                echo '<div class="form-check">
                <input class="form-check-input" type="radio" name="selected_employee[' . $substitution['id'] . ']"
                value="' . $employee['id'] . '" id="employee_' . $employee['id'] . '">';
                echo '<label class="form-check-label" for="employee_' . $employee['id'] . '">' . $employee['imie'] . '
                ' . $employee['nazwisko'] . '</label> </div>';
            }
        } else {
            echo 'Brak dostępnych pracowników';
        }
        echo '<button type="submit" name="action" value="accept" class="btn btn-primary">Wyślij</button>
        <button type="submit" name="action" value="reject" class="btn btn-danger">Odrzuć</button> </td> </tr>';
    }
    echo '</tbody>
    </table></form>';
}

```

wyświetlanie rekordów za pomocą foreach przeszukujemy tablice potrzebujących.
następnie. do zmiennej zapisujemy dostępnych pracowników po danych z substitution.

```

function ChangeStatus() {
    $conn = db_connect();
    if (isset($_POST['selected_employee']) && $_POST['action'] === 'accept') { // sprawdź czy wybrany i akcept
        foreach ($_POST['selected_employee'] as $substitutionId => $employeeId) { // iteracja po tablicy
            // zmiana statusu i ustawienie pracownika zastepujacego
            $query = "UPDATE zastepstwa SET id_pracownika_zastepujacego = $employeeId, status = 'DoAkceptacji' WHERE id = $substitutionId";
            if (!mysqli_query($conn, $query)) {
                echo "Error: " . mysqli_error($conn);
            }
        }
    } elseif ($_POST['action'] === 'reject') { // jeśli odrzucone zmiana statusu na odrzucone
        foreach ($_POST['selected_employee'] as $substitutionId => $employeeId) {
            $query = "UPDATE zastepstwa SET status = 'odrzucone' WHERE id = $substitutionId";
            if (!mysqli_query($conn, $query)) {
                echo "Error: " . mysqli_error($conn);
            }
        }
    }
    mysqli_close($conn);
    header('Location: ZarządzajZastępstwami.php');
    exit();
}

if (isset($_POST['selected_employee'])) {
    ChangeStatus();
}

```

Zmiana status zastępstwa po kliknięciu przycisku .

Widok Ustawienia

Menu

Harmonogram Grup

Aktualne zastępstwa

Nadgodziny

Dodaj zastępstwo

Dodaj

Nieprzypisane zastępstwo

Zalogowany jako: Filip Dziedzic

Ustawienia

Wyloguj

Ustawienia

Obecny Login

admin

Nowy Login

Obecne Hasło

Nowe Hasło

Zmień Login i Hasło

Kod widoku

```
<form method="post" action="../../PHP_Logik/ustawienia_logic.php" >
  <div class="mb-3">
    <label for="currentLogin" class="form-label">Obecny Login</label>
    <input type="text" class="form-control" id="currentLogin" name="currentLogin" value="<?php echo $_SESSION['login']; ?>" required />
  </div>
  <div class="mb-3">
    <label for="newLogin" class="form-label">Nowy Login</label>
    <input type="text" class="form-control" id="newLogin" name="newLogin" required />
  </div>
  <div class="mb-3">
    <label for="currentPassword" class="form-label">Obecne Hasło</label>
    <input type="password" class="form-control" id="currentPassword" name="currentPassword" required />
  </div>
  <div class="mb-3">
    <label for="newPassword" class="form-label">Nowe Hasło</label>
    <input type="password" class="form-control" id="newPassword" name="newPassword" required />
  </div>
  <button type="submit" class="btn btn-primary">Zmień Login i Hasło</button>
</form>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

value jest ustawione tak żeby od razu po wejściu wpisywane był login użytkownika.

Kod działania

```
<?php
session_start();
require('database_connection.php');

if (isset($_POST['newLogin']) && isset($_POST['currentPassword']) && isset($_POST['newPassword'])) {
    $newLogin = $_POST['newLogin'];
    $currentPassword = $_POST['currentPassword'];
    $newPassword = $_POST['newPassword'];
    $user_id = $_SESSION['user_id'];
    $currentLogin = $_SESSION['login'];

    // Sprawdzenie, czy nowe hasło spełnia wymagania
    if (strlen($newPassword) < 8 || !preg_match('/[0-9]/', $newPassword)) {
        echo "<script>alert('Nowe hasło musi mieć co najmniej 8 znaków i zawierać co najmniej jedną cyfrę.');
```

Ta część kodu jest odpowiedzialna za sprawdzenie czy pola są ustawione następnie sprawdzenie walidacji za pomocą strlen sprawdzamy długość hasła oraz preg_match czy posiada numery odpowiedni pattern musi być użyty

```

$conn = db_connect();

// Sprawdzenie obecnego loginu i hasła
$query = "SELECT haslo FROM uzytkownicy WHERE id = '$user_id' AND login = '$currentLogin'";
$result = mysqli_query($conn, $query);

if ($result && mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_assoc($result);
}

```

sprawdzenie obecnego loginu i hasła czy istnieją i czy się zgadzają.

```

if (password_verify($currentPassword, $row['haslo']) || $currentPassword === $row['haslo']) {
    // Sprawdzenie, czy nowy login już istnieje
    $query = "SELECT id FROM uzytkownicy WHERE login = '$newLogin'";
    $result = mysqli_query($conn, $query);

    if ($result && mysqli_num_rows($result) == 0) {
        // Aktualizacja loginu i hasła
        $hashedPassword = password_hash($newPassword, PASSWORD_DEFAULT);
        $query = "UPDATE uzytkownicy SET login = '$newLogin', haslo = '$hashedPassword' WHERE id = '$user_id'";
        if (mysqli_query($conn, $query)) {
            $_SESSION['login'] = $newLogin;
            echo "<script>alert('Login i hasło zostały pomyślnie zmienione.');

```

sprawdzenie czy podane hasło jest poprawne za pomocą password_verify sprawdzamy hashowane hasła. Następnie po przejściu kroków zmienimy podane hasło na hashowane za pomocą password_hash. Po udanej zmianie danych zmienia jest obecna login w sesji.

Wylogowanie

[Wyloguj](#)

```

<form method="post" action="../../PHP_Logic/logout.php" style="display:inline;">
    <button type="submit" class="btn btn-link">Wyloguj</button>
</form>

```

kod działania

```
<?php
session_start();
session_destroy();
header('Location: ../sites/login.php');
exit();
?>
```

podtrzymuje sesje a następnie ja kasuje i przenosi do logowania.

SideBar

Menu

Harmonogram Grup

Aktualne zastępstwa

Nadgodziny

Dodaj zastępstwo

Dodaj

Nieprzypisane zastępstwo 2



Zalogowany jako: Filip Dziedzic

Ustawienia

[Wyloguj](#)

Kod widoku dołączony do każdego widoku

```

<div class="sidebar d-flex flex-column">
    <h4 class="text-center">Menu</h4>
    <a href="index.php">Harmonogram Grup</a>
    <?php if(isAdmin()) // z sidebar sprawdza role
    |     echo ' <a href="AktualneZastepstwa.php">Aktualne zastępowstwa</a>';
    |
    ?>
    <a href="Nadgodziny.php">Nadgodziny</a>
    <a href="dodaj_zastepstwo.php">Dodaj zastępowstwo</a>
    <?php
    if(isAdmin())
    {
    |     echo " <a href='dodaj.php'>Dodaj</a>";
    |
    }
    ?>
    <div class="accordion mt-3" id="notificationAccordion">
    |     <?php
    |     if(!isAdmin())
    |     {
    |         displaySubstitutionsAccept(); // wyswietla zastepstwa do akceptacji
    |     }
    |
    |     <?php
    |     if(isAdmin())
    |     {
    |         displaySubstitutionsPending(); // wyswietla zastepstwa oczekujace
    |     }
    |
    |     ?>
    |
    </div>
    <div class="user-info mt-auto">
    |     <p>Zalogowany jako: <?php WhoAmI() // wyswietla jako kot jestesmy zalogowani?></p>
    |     <div class="links">
    |         <a href="ustawienia.php">Ustawienia</a>
    |         <form method="post" action=" ../PHP_Logic/logout.php" style="display:inline;">
    |             <button type="submit" class="btn btn-link">Wyloguj</button>
    |         </form>
    |     </div>
    </div>
</div>

```

Kod działania

```

<?php
session_start();
require_once('database_connection.php');
//obsługa zatwierdzania i odrzucania zastępstwa przez użytkownika
if (isset($_POST['substitution_id']) && isset($_POST['action'])) { // obsługa potwierdzania lub odrzucenia zastępstwa z poziomu użytkownika
    $conn = db_connect(); // łączenie
    $substitution_id = $_POST['substitution_id'];
    $action = $_POST['action'];
    $user_id = $_SESSION['user_id'];

    if ($action === 'accept') {
        $status = 'zatwierdzone'; // ustawienie zmiennej na zatwierdzone
        // zapytanie zmienia status i dodaje pracownika do tego zastępstwa
        $query = "UPDATE zastepstwa SET status = '$status', id_pracownika_zastepujacego = '$user_id' WHERE id = $substitution_id";
        if (mysqli_query($conn, $query)) {
            $substitutionQuery = "SELECT data_zastepstwa, godzina_od, godzina_do FROM zastepstwa WHERE id = $substitution_id"; // data i godzina zastępstwa
            $substitutionResult = mysqli_query($conn, $substitutionQuery);
            if ($substitutionResult && mysqli_num_rows($substitutionResult) > 0) {
                $substitution = mysqli_fetch_assoc($substitutionResult);
                $date = $substitution['data_zastepstwa'];
                $startTime = $substitution['godzina_od'];
                $endTime = $substitution['godzina_do'];
                $hours = (strtotime($endTime) - strtotime($startTime)) / 3600; // liczba godzin

                $overtimeQuery = "INSERT INTO nadgodziny (id_pracownika, data, liczba_godzin) VALUES ('$user_id', '$date', '$hours')"; // dodanie do nadgodzin dla pracownika
                mysqli_query($conn, $overtimeQuery);
            }
        }
    } elseif ($action === 'reject') {
        $status = 'oczekujące';
        $query = "UPDATE zastepstwa SET status = '$status', id_pracownika_zastepujacego = NULL WHERE id = $substitution_id"; // zmiana na oczekujące
        mysqli_query($conn, $query);
    } else { // obsługa dodatkowych opcji
        $status = 'oczekujące';
        $query = "UPDATE zastepstwa SET status = '$status' WHERE id = $substitution_id";
        mysqli_query($conn, $query);
    }

    mysqli_close($conn);

    header('Location: ../sites/index.php'); // przekierowanie na główną
    exit();
}

```

ten segment odpowiada za akceptację bądź odrzucenie przez użytkownika zastępstwa. za pomocą isset sprawdzamy czy istnieją dane. następnie pobierane są dane do zmiennych. po wybraniu accept zostanie zmienione zostanie zastępstwo a dokładnie status oraz id pracownika zastępującego oraz zastępstwo zostanie naliczone w nadgodzinach. po odrzuceniu zostanie ustawione na oczekujące aby znów admin musiał jej przydzielić.

```

function isAdmin() { // sprawdzenie typu sesji
    checkRole();
    return isset($_SESSION['rola']) && $_SESSION['rola'] === 'admin';
}

```

sprawdza czy pole sesji jest ustawione i czy jest nim admin sprawdź czy obecnie jest zalogowany admin

```

function checkRole() { // sprawdza rolę
    $conn = db_connect();
    $user_id = $_SESSION['user_id'];
    $query = "SELECT rola FROM uzytkownicy WHERE id = '$user_id'"; // sprawdzenie roli użytkownika
    $result = mysqli_query($conn, $query);

    if ($result && mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result);
        $_SESSION['rola'] = $row['rola'];
        mysqli_close($conn);
        return $row['rola'] === 'admin';
    }

    mysqli_close($conn);
}

```

sprawdza rolę po sesyjnym id użytkownika.


```

function WhoAmI() { // sprawdzenie na kim jesesmy zalogowani
    $conn = db_connect();
    $user_id = $_SESSION['user_id'];

    $query = "SELECT imie, nazwisko FROM uzytkownicy WHERE id = '$user_id'";
    $result = mysqli_query($conn, $query);

    if ($result && mysqli_num_rows($result) > 0) {
        $row = mysqli_fetch_assoc($result);
        $user_name = $row['imie'] . ' ' . $row['nazwisko'];
        echo $user_name;
    } else {
        echo "Nie znaleziono użytkownika.";
    }

    mysqli_close($conn);
}

```

Sprawdza na kim jest zalogowany wyświetla na dole strony imię i nazwisko.

```

function getSubstitutionsAccept() { // pobieranie zastępców dla konkretnych pracowników
    $conn = db_connect();
    $user_id = $_SESSION['user_id'];
    $query = "SELECT DISTINCT z.id, z.data_zastepstwa AS data, z.godzina_od, z.godzina_do, g.nazwa AS grupa,
        u1.imie AS imie_potrzebujacego, u1.nazwisko AS nazwisko_potrzebujacego,
        u2.imie AS imie_zastepujacego, u2.nazwisko AS nazwisko_zastepujacego
    FROM zastepstwa z
    JOIN uzytkownicy u1 ON z.id_pracownika_proszacego = u1.id
    LEFT JOIN uzytkownicy u2 ON z.id_pracownika_zastepujacego = u2.id
    LEFT JOIN pracownik_grupa pg ON u1.id = pg.id_pracownika
    LEFT JOIN grupy g ON pg.id_grupy = g.id
    WHERE z.status = 'DoAkceptacji' AND z.id_pracownika_zastepujacego = '$user_id'";
    $result = mysqli_query($conn, $query);
    $substitutions = [];
    if ($result && mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $substitutions[] = $row;
        }
    }
    mysqli_close($conn);
    return $substitutions;
}

```

ta funkcja pobiera wszystkie zastępstwa ze statusem do akceptacji.


```

function displaySubstitutionsPending() { // wyświetlanie zastępstw dla admina
    $substitutions = getSubstitutionsPending();
    echo ' <div class="accordion-item">
        <h2 class="accordion-header" id="headingUnassignedSubstitution">
            <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseUnassignedSubsti
                Nieprzypisane zastępstwo
                <span class="badge bg-primary rounded-circle ms-2">'. countSubstitutionsByPending();
            echo '</span>
        </button>
        </h2>
        <div id="collapseUnassignedSubstitution" class="accordion-collapse collapse" aria-labelledby="headingUnassignedSubstitution" dat
            ';
            echo " <a href='ZarządzajZastępstwami.php' class='btn btn-primary mb-2'>Przypisz</a>" ;
            echo '
        <div class="accordion-body">';
        foreach ($substitutions as $entry) {
            echo '<p>' . $entry['data'] . ' od ' . $entry['godzina_od'] . ' do ' . $entry['godzina_do'] . '</p>';
            echo '<hr />';
        }
    echo ' </div>
        </div>
    </div>';
}
?>

```

Wyświetlanie w side bar odpowiednich danych dla odpowiedniej roli.