

Dokumentacja bazy danych: „Domowe wydatki”

Wykonał: Marek Kurańda

Wstęp

Każdy z nas doświadczył zapewne, że potrzebuje kontrolować wydatki swoje oraz swoich najbliższych. Wielokrotnie zapewne każdy myślał o tym, jak zaoszczędzić chociaż pewną sumę swojego zarobku.

Dzięki tej aplikacji będziesz mógł kontrolować na co idą twoje wydatki. Będziesz również świadomy o wydatkach swoich najbliższych, przez co pomożesz zaoszczędzić im pewną sumę pieniędzy.

Pozwala ona na dodawanie użytkowników, zatem nawet członkowie rodziny, dodani jako Osoby, mogą założyć sobie konto i przeglądać swoje wydatki. Jest to bardzo ciekawa opcja, ponieważ nie każdy z członków musi mieć konto.

Dlatego też, ci którzy nie boją się wydawać pieniędzy, powinni poobserwować swoje bilanse, aby uświadomić sobie, że moje jednak można i warto coś oszczędzić... 😊

Parę słów...

Od początku tworzenia projektu starałem się, aby był on rozszerzalny, skalowalny, wolny od błędów.

Najlepszy dowodem na to jest system formularzy i raportów. W łatwy sposób wystarczy dołączyć kolejne obiekty do plików .json, a aplikacja będzie zdolna wyświetlić kolejne formularze, pod warunkiem, że zostaną uwzględnione w pliku index.php oraz w pliku form.php/raport.php.

W tym projekcie mogę pochwalić się, czego nie lubię czynić, wspomnianym systemem formularzy i raportów, algorytmem generowania kolorów, systemem palet, ustawieniami, dynamicznym zmienianiu kolorów strony, skalowalnością projektu, systemem paginacji, automatycznym pobieraniem wszystkich danych z formularzy, automatycznym generowaniem JSON'ów, oddzieleniem HTML'a, CSS'a oraz JS'a od siebie (umieszczenie ich we własnych plikach – nie mieszanie technologii), wykorzystaniem AJAX'a, systemem walidacji, systemem sprawdzania, czy nazwy użytkowników się nie powtarzają oraz systemem logowania.

Mam nadzieję, że projekt wzbudzi ciekawość. Myślę, że nie opisałem wszystkich cech, a drobniejsze szczegóły pozostały w samym kodzie, który (tak myślę) będzie najlepszym dowodem na to.

Sposób działania

Aby uruchomić aplikację, należy wprowadzić poprawne dane do pliku:
`./engine/connector.php`

Następnie należy uruchomić program XAMPP, oraz wpisać lokalizację aplikacji w folderze htdocs. Aplikacja zawiera plik `.htaccess`, przez co jej adres może się zmienić. Mianowicie doda na początek frazę: `/main`

Strona składa się z trzech części:

- nagłówek (ang. header)
- zawartość, część główna (ang. main)
- stopka (ang. footer)

Nagłówek jest dynamiczny. Niektóre przyciski się pojawiają, niektóre nie. Również widać komunikat powitalny dla użytkownika.

Strona również posiada swoje ramy, które zostały nazwane: „web-border”. Na górnym elemencie znajduje się po lewej stronie informacja o czasie i strefie czasowej serwera. Kolejno znajdują się informacje o kliencie.

Po prawej stronie można zauważyć symbol zębatki. Klikając na ten malutki przycisk, użytkownik zostanie przeniesiony na podstronę ustawień.

Aplikacja zawiera kilka podstron, są to:

`/main` – podstrona strony głównej

`/report` – podstrona raportów

`/form` – podstrona formularzy

`/settings` – podstrona ustawień

Niektóre podstrony zawierają dodatkowe informacje w swoim nagłówku, które np. pozwalają modyfikować dane.

Podstrona: /settings

Warto poświęcić chwilę uwagi dla tej podstrony. Szczególnie dlatego, ponieważ pozwala ona dokonać w pewnym sensie personalizacji. Otóż, użytkownik, niezależnie czy jest zalogowany, może zmienić paletę kolorów strony, dobrać czcionkę, a nawet jej kolor!

Oczywiście podczas zmieniania ustawień, strona dynamicznie się do nich dostosowuje, jednak odświeżenie strony porzuci niezapisane ustawienia. Pozwala to użytkownikowi na dynamiczne dobranie kolorów.

Aby zapisać ustawienia, trzeba kliknąć przycisk poniżej wszystkich ustawień.

Server time: 15:30:02, Europe/Berlin | Client time: 15:30:02, Europe/Warsaw

Witaj, Bix!

Strona Główna Formularze Raporty Administracja

Opcje

Paleta: Paleta 4

Czcionka

Rodzaj czcionki: Arial

Kolor czcionki: #F4F440

Zapisz!

Autor: Marek Kurańda

Podstrona: /main



Na załączonym obrazku możemy zauważyć wiele elementów, szczególnie w nagłówku.

Poniższy obrazek tłumaczy za co odpowiedzialne są poszczególne przyciski:



1. Ustawienia strony (w dalszej części więcej)
2. Raporty (w zależności, czy jest ktoś zalogowany i z jakimi uprawnieniami, takie wyświetlane są raporty)
3. Formularze (w zależności, czy jest ktoś zalogowany i z jakimi uprawnieniami, takie wyświetlane są formularze)
4. Przycisk przenoszący do „Strony Głównej”
5. Przycisk przenoszący do rejestracji
6. Przycisk przenoszący do logowania
7. Odnośniki do twórców plików .svg, z których strona korzysta.

Server time: 13:30:13, Europe/Berlin Client time: 13:30:13, Europe/Warsaw

1 Witaj, użytkowniku! [Zaloguj się](#) lub [Zarejestruj się](#) 2

Strona Główna	Formularze	Raporty
	Osoby	Osoby
	Sklepy	Sklepy
	Kategorie	Kategorie
	Wydatki	Wydatki

Strona główna

O stronie

Jeśli potrzebujesz kontrolować swoje wydatki jest to idealne narzędzie stworzone wprost dla ciebie, które zapewni ci pełną swobodę. Przekonasz się, że jest to idealne narzędzie!

Dzięki tej aplikacji możesz zarządzać swoimi wydatkami oraz wydatkami swoich najbliższych. Umożliwia łatwą kontrolę. System został dopracowany w najmniejszych szczegółach.

Dzięki systemowi użytkowników możesz sterować wszystkim ogólnie oraz dbać o to, by ktoś niepowołany mógł nie co namieszać, przez co mógłby się zrobić mały bałagan.

3 Twórcy

Główny autor: Marek Kurańda
oraz pozostali:

Icons made by Pixel perfect from www.flaticon.com
Icons made by Icongeek26 from www.flaticon.com
Icons made by Freepik from www.flaticon.com
Icons made by Pixel perfect from www.flaticon.com
Icons made by Kiranshastry from www.flaticon.com
Icons made by bqlqn from www.flaticon.com

Milej zabawy! :)

1. Czas serwerowy pobierany z bazy danych przy ładowaniu strony, wykorzystując odpowiednią do tego procedurę, której opis znajduje się w dalszej części.
2. Czas po stronie klienta. Użyty został JavaScript do pobrania podanych informacji. Oba czasy po sekundzie są aktualizowane, lecz w taki sposób, że dodawana jest tylko jedna sekunda. Pierwsze czas jest tłumaczony na timestamp, a następnie zwiększany o jeden. W ostatniej fazie jest wyświetlany.
3. Informacja o twórcy.

Logowanie

Server time: 16:00:37, Europe/Berlin | Client time: 16:00:37, Europe/Warsaw

Witaj, użytkowniku! [Zaloguj się](#) lub [Zarejestruj](#)

Strona Główna Formularze Raporty

Logowanie

Podaj dane użytkownika:

Nazwa użytkownika:

Hasło:

Autor: Marek Kurańda

Należy wprowadzić dane logowania dowolnego użytkownika. Najwyżej stojącym w hierarchii użytkownikiem jest „Admin”, którego prawa są niezbywalne. To znaczy, że nikt nie może ich odebrać. Hasło do użytkownika „Admin” jest takie samo jak jego nazwa, tzn.: „Admin”.

Niżej w hierarchii stoją pozostali użytkownicy. Nawet gdyby ktoś z nich nabył najwyższe uprawnienie, tj. „Administracja”, to nie może „obalić” Admina.

Oczywiście jeżeli użytkownik wprowadzi niepoprawne dane, lub takie dane, które nie dopasuje nigdzie aplikacja, wyrzuci błąd, a jego treść wypisze poniżej.

Logowanie

Podaj dane użytkownika:

Nazwa użytkownika:

Hasło:

Wystąpił błąd przy logowaniu!

Po zalogowaniu (w zależności na jakie konto) pojawią się nowe możliwości w pasku nawigacyjnym, między innymi:

- Zobacz uprawnienia
- Zobacz zasady
- Zobacz logi
- Wyloguj

 Witaj, Kadf!	Strona Główna	Formularze	Raporty	Administracja
		Osoby Sklepy Kategorie Wydatki	Osoby Sklepy Kategorie Wydatki	Zobacz uprawnienia Zobacz zasady Zobacz logi Wyloguj

Takie przyciski otrzymuje każdy użytkownik. Jednakże, użytkownik z większą liczbą uprawnień może doczekać się nawet takich przycisków:

 Witaj, Admin!	Strona Główna	Formularze	Raporty	Administracja
		Osoby Sklepy Kategorie Wydatki Uprawnienia	Osoby Sklepy Kategorie Wydatki	Użytkownicy Zobacz uprawnienia Zobacz zasady Zobacz logi Wyloguj

Logi są zapisami. Jest to dziennik aplikacji. Zapisywane są tam wszystkie działania na bazie danych przez użytkowników.

W podstronie uprawnień można zobaczyć, kto posiada jakie uprawnienia. Oczywiście, jeżeli użytkownik ma możliwość odbierania uprawnień, może to tam uczynić.

Podstrona z zasadami umożliwia wgląd w zasady, a konkretnie jakie istnieją, wraz z ich opisem. Jednakże, nawet Admin nie może usunąć ani jednej zasady. Jest to stała, której nikt nie może i nie powinien zmieniać.

Aby dowiedzieć się, jacy istnieją użytkownicy, należy kliknąć właśnie w owy przycisk pod przyciskiem „Administracja”. Zostanie wyświetlony raport o użytkownikach.

Przycisk „Formularze” dostarcza nam pakiet dostępnych formularzy. Zaś przycisk „Raporty” pozwala przeglądać dane. Również, jeżeli zalogowany użytkownik posiada odpowiednie uprawnienia, może wtenczas usuwać lub modyfikować rekord.

Aplikacja została ubrana system paginacji, który pozwala dzielić tabelkę na strony wyników.

Tabela przedstawiająca uprawnienia:

Nazwa	Opis
Administracja	Zarządzanie wszystkim, pełna kontrola
Dodawanie	Możliwość dodawania nowych rekordów
Usuwanie	Możliwość usuwania rekordów/uprawnień
Modyfikacja	Możliwość zarządzania danymi (UPDATE)
Wyświetlanie	Możliwość wyświetlania danych (SELECT)

Server time: 15:33:24, Europe/Berlin | Client time: 15:33:24, Europe/Warsaw

Witaj, Admin!

Strona Główna Formularze Raporty Administracja

Formularz tabeli: Wydatki

Tabela - Wydatki:

Kategoria:	Utrzymanie czystości	▼
Osoba:	Mikołaj Kotowicz	▼
Sklep:	U Romka	▼
Data:	rrrr-mm-dd	📅
Kwota:	0.01	
Opis:		

Dodaj! Wyczyść

Autor: Marek Kurańda

Pusty formularz dla tabeli **Wydatki**.

Server time: 15:38:06, Europe/Berlin | Client time: 15:38:06, Europe/Warsaw

Witaj, Admin!

Strona Główna Formularze Raporty Administracja

Formularz tabeli: Wydatki

Tabela - Wydatki:

Kategoria:	Utrzymanie czystości	▼
Osoba:	Mikołaj Kotowicz	▼
Sklep:	U Romka	▼
Data:	2021-01-06	📅
Kwota:	0.01	
Opis:		

Dodaj! Wyczyść

Autor: Marek Kurańda

Błędne wprowadzanie danych, wyrzuca komunikat w formie alertu, a następnie przesuwa kursor w miejsce wystąpienia błędu.

Server time: 15:39:31, Europe/Berlin | Client time: 15:39:31, Europe/Warsaw

Witaj, Admin! Strona Główna Formularze Raporty Administracja

Formularz tabeli: Wydatki

Tabela - Wydatki:

Kategoria:

Osoba:

Sklep:

Data:

Kwota:

Opis:

Dodaj! Wyczyść

Rekord został dodany poprawnie!



Autor: Marek Kurańda

Jeżeli dane zostaną poprawnie wprowadzone, wtedy wyskoczy komunikat poniżej formularza, wszystkie dane zostaną wyczyszczone, a kursor uzyska fokus nad pierwszym elementem formularza.

Server time: 15:41:17, Europe/Berlin | Client time: 15:41:17, Europe/Warsaw

Witaj, Admin! Strona Główna Formularze Raporty Administracja

Raport tabeli: Osoby

Nr	Imię	Nazwisko	Działania
11	Dionizy	Kazanowski	 
12	Ksenia	Kazanowski	 
13	Adam	Adamowicz	 
27	Marcus	Antonius	 
28	Wesoly	Roman	 

Ilość rekordów: | 25 | 50 | 100 Strona: 1, 2

Autor: Marek Kurańda

Oto raport dla tabeli osoby. Została wybrana druga podstrona. Aplikacja domyślnie ma zaznaczone paginowanie na 10 rekordów, jednakże łatwo to można zmienić pod napisem: „Ilość rekordów”.

Server time: 15:41:17, Europe/Berlin | Client time: 15:41:17, Europe/Warsaw

Witaj, Admin! Strona Główna Formularze Raporty Administracja

Raport tabeli: Osoby

Nr	Imię	Nazwisko	Działania
11	Dionizy	Kazanowski	<div>3</div> <div>4</div>

Numerzy odpowiednio:

1. Zmiana ilości rekordów na stronę
2. Nawigowanie pomiędzy stronami (w przypadku wielu stron, wyświetlane są pierwsza i ostatnia, a pomiędzy nimi tylko trzy).
3. Modyfikacja rekordu (wyświetlane, gdy użytkownik posiada takie uprawnienie)
4. Usuwanie rekordu (wyświetlane, gdy użytkownik posiada takie uprawnienie)

Server time: 15:51:15, Europe/Berlin | Client time: 15:51:15, Europe/Warsaw

Witaj, Admin! Strona Główna Formularze Raporty Administracja

Raport tabeli: Uprawnienia

Nazwa użytkownika	Nazwa zasady	Działania
Admin	Administracja	Nie masz żadnych uprawnień.
Admin	Dodawanie	Nie masz żadnych uprawnień.
Admin	Usuwanie	Nie masz żadnych uprawnień.
Admin	Modyfikacja	Nie masz żadnych uprawnień.
Admin	Wyświetlanie	Nie masz żadnych uprawnień.
Kadf	Dodawanie	
Kadf	Modyfikacja	
Kadf	Wyświetlanie	
Lads	Wyświetlanie	

Ilość rekordów: 10 | 25 | 50 | 100 Strona: 1

Autor: Marek Kurańda

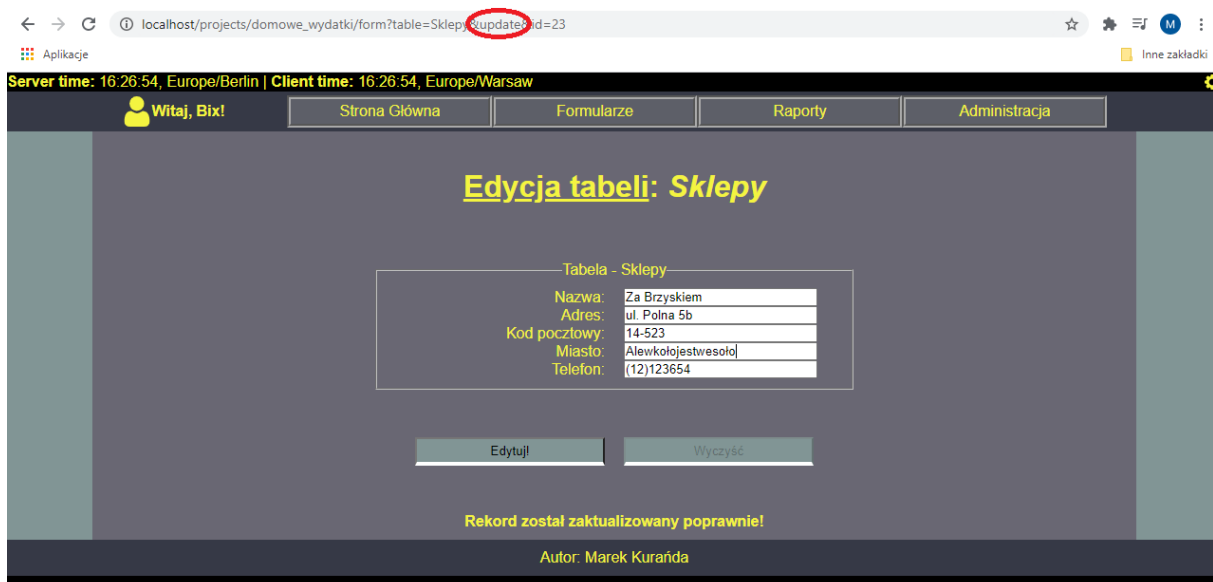
Jak widać na załączonym powyżej obrazku, pomimo, że zalogowany użytkownik to Admin, który posiada wszystkie uprawnienia, nie może usunąć, ani odebrać swoich uprawnień.



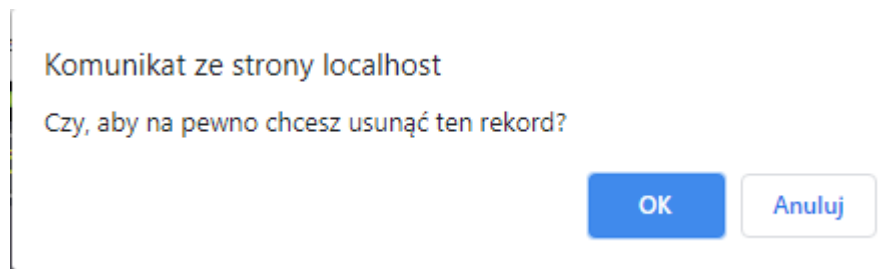
Z kolei ten obrazek przedstawia użytkownika, który posiada wszystkie uprawnienia oprócz usuwania rekordów. Użytkownik posiadający uprawnienie „Administrowanie”, może usuwać uprawnienia, jednakże nie może ich dodawać.



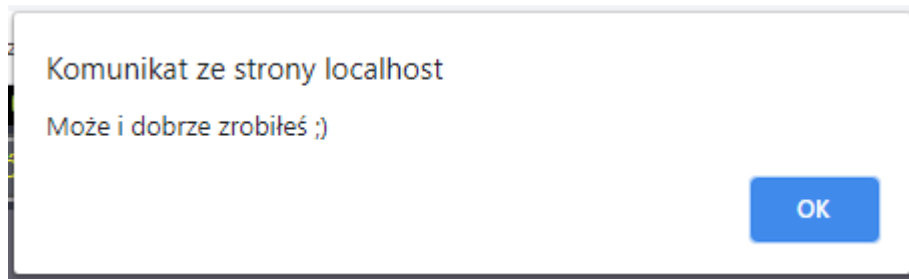
Kolejny obrazek pokazuje pierwszą stronę dla uprawnień i pokazuje, że zarówno Adminowi, jak samemu sobie, nie można odebrać uprawnień.



Aktualizowanie rekordu.



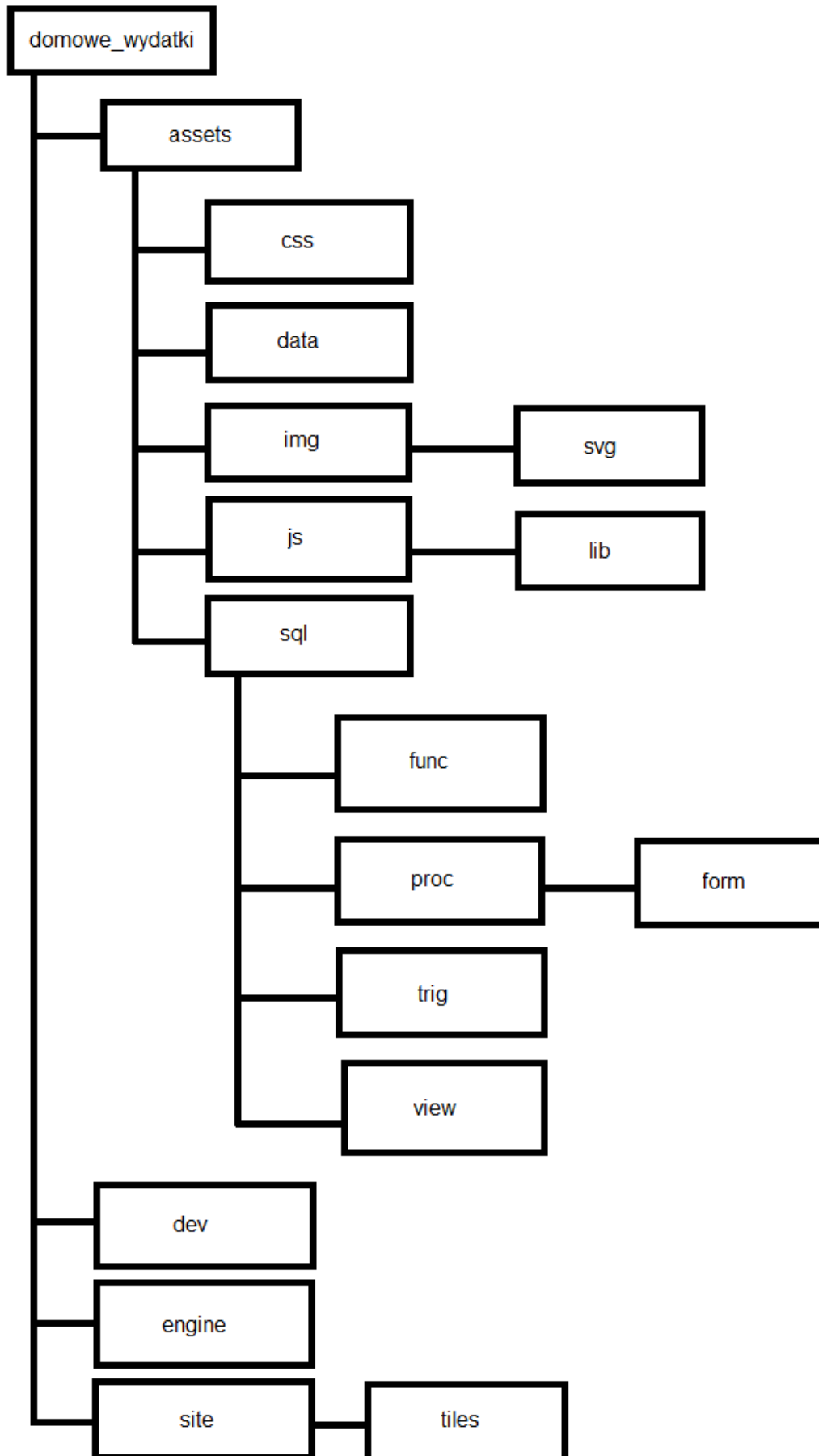
Przy usuwaniu rekordów otrzymujemy zapytanie.



Gdy zrezygnujemy, otrzymujemy odpowiedź.

Opis od strony programisty

Struktura katalogów



Folder	Zawartość
domowe_wydatki	Główny folder projektu
assets	Nadrzędny folder dla plików związanych z wyglądem i działaniem strony
css	Wszystkie pliki .css
data	Miejsce dla plików .json, które zawierają różne dane
img	Nadrzędny folder dla zdjęć oraz obrazków
svg	Folder, gdzie znajdują się pliki z grafiką wektorową
js	Folder dla skryptów napisanych w języku JavaScript
lib	Folder dla bibliotek napisanych w języku JavaScript
sql	Folder dla plików o rozszerzeniu .sql
func	Miejsce dla plików z tworzeniem funkcji użytych w projekcie
proc	Miejsce dla plików z tworzeniem procedur użytych w projekcie
form	Miejsce dla plików z tworzeniem procedur, które zostały zastosowane w formularzach (dodawanie nowych rekordów)
trig	Miejsce dla plików z tworzeniem triggerów, użytych w projekcie
view	Miejsce dla plików z tworzeniem widoków, użytych w projekcie
dev	„Folder developer’a”, gdzie znajduje się plik z dziennikiem oraz z pomysłami, które autor zamierzał wdrożyć
engine	Znajduje się plik, który łączy się z bazą danych
site	Znajdują się pliki podstron
tiles	Znajdują się mniejsze komponenty

Budowa pliku index.php:

```
<?php include_once('engine/connector.php'); ?>
<?php include_once('engine/manager.php'); ?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <meta name="author" content="Marek Kurańda" />
        <meta name="description" content="Domowe Wydatki - dobry sposób na kontrolę wydatków!" />
        <meta name="creation-date" content="2020-12-18" />

        <title>Domowe Wydatki</title>
        <link rel="icon" href="assets/img/house-rental.png">

        <link rel="stylesheet" href="assets/css/basic.css" />
        <link rel="stylesheet" href="assets/css/header.css" />
        <link rel="stylesheet" href="assets/css/footer.css" />
        <?php
            echo '<link rel="stylesheet" href="assets/css/' . $_GET['sub'] . '.css' />';
        ?>

        <script src="assets/js/lib/jquery-3.5.1.min.js"></script>
        <script src="assets/js/lib/jquery.cookie.js"></script>
        <script src="assets/js/manager.js"></script>
        <script src="assets/js/header.js"></script>
    </head>

    <body class="__col3">
        <div class="web-border __col0 flex-between">
            <div class="flex-center">
                <div class="flex-center sec-server-time">
                    <strong>Server time:</strong>&nbsp;<?php include_once('site/server_time.php'); ?></p>
                </div>
                &nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;
                <div class="flex-center sec-client-time">
                    <strong>Client time:</strong>&nbsp;<div class="client-time"></div>&nbsp;<div class="client-timezone"></div>
                </div>
            </div>
        </div>
    </body>
</html>
```

```

        <div class="flex-center">
            <a href="settings" class="flex-
center"><?php include('assets/img/svg/settings.svg'); ?></a>
        </div>
    </div>
    <!-- Header -->
    <?php include_once('site/header.php'); ?>
    <!-- Center -->
    <?php
        switch ($_GET['sub']) {
            case 'main':
            case 'report':
            case 'form':
            case 'settings':
            case 'reg':
                include_once('site/' . $_GET['sub'] . '.php');
                break;

            // Error 404
            default:
                include_once('site/error404.php');
                break;
        }
    ?>
    <!-- Footer -->
    <?php include_once('site/footer.php'); ?>
    <div class="web-border __col0"></div>
</body>
</html>

<?php include_once('engine/disconnector.php'); ?>

```

Podłączone pliki connector.php i disconnector.php zarządzają połączeniem. Na początku jest tworzone połączenie, a gdy wczyta się cała strona, to połączenie jest zamykane.

Plik manager.php

```
<?php
/*
    Plik odpowiada za zarządzanie stroną podczas jej ładowania.
    Chodzi o zarządzanie URL'em.
    Również czyści bufor.
*/

// Init
ob_start();
ob_clean();
session_start();

// Podprogram sprawdza, czy istnieje zmienna 'sub'
if (!isset($_GET['sub'])) $_GET['sub'] = 'main';

// Pliki cookies
if (!isset($_COOKIE['id_palette'])) setcookie('id_palette', '5', time() +
60*60*24, '/');
if (!isset($_COOKIE['id_font-color'])) setcookie('id_font-
color', '#FFFFFF', time() + 60*60*24, '/');
?>
```

Tworzy również pliki cookie, gdy nie istnieją (tworzy ustawienia domyślne) oraz czyści wcześniej bufor oraz rozpoczyna sesję.

Gdy nie jest ustawiona żadna podstrona, wtedy ustawia jako główną „main”.

Budowa pliku header.php

```
<?php $logged = false; ?>
<?php
    // If someone has a icons
    function hasRule ($con, $rule) {
        $i = (isset($_COOKIE['id_user']) ? $_COOKIE['id_user'] : -1);
        $q = "SELECT dbo.func_check_user_rule('" . $i . "', dbo.func_get_Zasad
y_Nr('" . $rule . "'))";
        $r = sqlsrv_query($con, $q);
        return sqlsrv_fetch_array($r);
    }
?>
<header class="__col1">
    <div class="center flex-around">
        <section class="sec-user flex-center">
            <div class="sec-user-dignity flex-center">
                <svg viewBox="-
42 0 512 512.002" xmlns="http://www.w3.org/2000/svg">
                    <path d="m210.351562 246.632812c33.882813 0 63.222657-
12.152343 87.195313-36.128906 23.972656-23.972656 36.125-53.304687 36.125-
87.191406 0-33.875-12.152344-63.210938-36.128906-87.191406-23.976563-23.96875-
53.3125-36.121094-87.191407-36.121094-33.886718 0-63.21875 12.152344-
87.191406 36.125s-36.128906 53.308594-
36.128906 87.1875c0 33.886719 12.15625 63.222656 36.132812 87.195312 23.976563
23.96875 53.3125 36.125 87.1875 36.125zm0 0"></path><path d="m426.128906 393.
703125c-.691406-9.976563-2.089844-20.859375-4.148437-32.351563-2.078125-
11.578124-4.753907-22.523437-7.957031-32.527343-3.308594-10.339844-7.808594-
20.550781-13.371094-30.335938-5.773438-10.15625-12.554688-19-20.164063-
26.277343-7.957031-7.613282-17.699219-13.734376-28.964843-18.199219-11.226563-
4.441407-23.667969-6.691407-36.976563-6.691407-5.226563 0-10.28125 2.144532-
20.042969 8.5-6.007812 3.917969-13.035156 8.449219-20.878906 13.460938-
6.707031 4.273438-15.792969 8.277344-27.015625 11.902344-10.949219 3.542968-
22.066406 5.339844-33.039063 5.339844-10.972656 0-22.085937-1.796876-
33.046874-5.339844-11.210938-3.621094-20.296876-7.625-26.996094-11.898438-
7.769532-4.964844-14.800782-9.496094-20.898438-13.46875-9.75-6.355468-
14.808594-8.5-20.035156-8.5-13.3125 0-25.75 2.253906-36.972656 6.699219-
11.257813 4.457031-21.003906 10.578125-28.96875 18.199219-7.605469 7.28125-
14.390625 16.121094-20.15625 26.273437-5.558594 9.785157-10.058594 19.992188-
13.371094 30.339844-3.199219 10.003906-5.875 20.945313-7.953125 32.523437-
2.058594 11.476563-3.457031 22.363282-4.148437 32.363282-.679688 9.796875-
1.023438 19.964844-
1.023438 30.234375 0 26.726562 8.496094 48.363281 25.25 64.320312 16.546875 15
.746094 38.441406 23.734375 65.066406 23.734375h246.53125c26.625 0 48.511719-
7.984375 65.0625-23.734375 16.757813-15.945312 25.253906-37.585937 25.253906-
64.324219-.003906-10.316406-.351562-20.492187-1.035156-30.242187zm0 0"></path>
                </svg>
                <strong>
            <?php
```

```

        if (!isset($_COOKIE['id_user']) || $_COOKIE['id_user']
== -1) echo 'Witaj, użytkowniku!';
        else {
            // Otrzymaj imię użytkownika
            $res_name = sqlsrv_query($con, "SELECT [Nazwa użyt
kownika] FROM func_get_Uzytkownicy_Nazwa(" . $_COOKIE['id_user'] . ")");
            $row_name = sqlsrv_fetch_array($res_name);

            echo 'Witaj, ' . $row_name['Nazwa użytkownika'] .
'!';

            $logged = true;
        }
    ?>
</strong>
</div>
<div class="sec-user-nav">
    <?php if (!isset($_COOKIE['id_user']) || $_COOKIE['id_user'] =
= -1) { ?>
        <a href="form?table=Uzytkownicy&log"><strong>Zaloguj się!<
/strong></a> lub <a href="form?table=Uzytkownicy&register"><strong>Zarejestruj
!</strong></a>
        <?php } ?>
    </div>
</section>
<nav class="flex-center">
    <div><p>Strona Główna</p></div>
    <?php if (hasRule($con, 'Dodawanie')) { ?>
    <div>
        <p>Formularze</p>
        <ul class="hidden">
            <a href="form?table=Osoby"><li>Osoby</li></a>
            <a href="form?table=Sklepy"><li>Sklepy</li></a>
            <a href="form?table=Kategorie"><li>Kategorie</li></a>
            <a href="form?table=Wydutki"><li>Wydutki</li></a>
            <?php if ($_COOKIE['id_user'] == 0) { ?><a href="form?tabl
e=Uprawnienia"><li>Uprawnienia</li></a><?php } ?>
        </ul>
    </div>
    <?php } ?>
    <?php if (hasRule($con, 'Wyświetlanie')) { ?>
    <div>
        <p>Raporty</p>
        <ul class="hidden">
            <a href="report?table=Osoby"><li>Osoby</li></a>
            <a href="report?table=Sklepy"><li>Sklepy</li></a>
            <a href="report?table=Kategorie"><li>Kategorie</li></a>
            <a href="report?table=Wydutki"><li>Wydutki</li></a>
        </ul>
    </div>
    <?php } ?>

```

```
<?php if ($logged) { ?>
    <div>
        <p>Administracja</p>
        <ul class="hidden">
            <?php if ($_COOKIE['id_user'] == 0) { ?><a href="report?table=Użytkownicy"><li>Użytkownicy</li></a><?php } ?>
            <?php if (hasRule($con, 'Wyświetlanie')) { ?>
                <a href="report?table=Uprawnienia"><li>Zobacz uprawnienia</li></a>
                <a href="report?table=Zasady"><li>Zobacz zasady</li></a>
                <a href="report?table=Logi"><li>Zobacz logi</li></a>
            <?php } ?>
            <a href="form?table=Użytkownicy&logout"><li>Wyloguj</li></a>
        </ul>
    </div>
    <?php } ?>
</nav>
</div>
</header>
```


Plik footer.php

```
<footer class="__col1">
  <div class="flex-center">
    Autor: Marek Kurańda
  </div>
</footer>
```

Plik error404.php

```
<main>
  <h1>Error!</h1>
  <p>Podstrona <i><?php echo $_GET['sub']; ?></i> nie istnieje! :(</p>
  <br>
  <p>Powrót do <a href="main">strony głównej...</a></p>
</main>
```

Plik form.php

```
<?php
    if (!isset($_GET['table']) or empty($_GET['table']) or
        ($_GET['table'] != 'Osoby' and $_GET['table'] != 'Kategorie' and $_GET
['table'] != 'Sklepy' and $_GET['table'] != 'Wydatki'
        and $_GET['table'] != 'Użytkownicy' and $_GET['table'] != 'Uprawni
enia') or ($_COOKIE['id_user'] != -
1 && !hasRule($con, "Administracja")[0])) header('location: main');
    if ((!isset($_COOKIE['id_user']) || $_COOKIE['id_user'] == -
1) && $_GET['table'] != 'Użytkownicy') header('location: form?table=Użytkownic
y&log');
?>

<main class="center __col2 <?php if (isset($_GET['register'])) echo '__reg__';
?>>">
    <div class="flex-center">
        <div>
            <h1 class="flex-
center"><p>Formularz tabeli</p>:&nbsp;  <strong><?php echo $_GET['table']; ?></s
trong></h1>

            <form action="" method="POST">
                <section class="form-datas">
                    <fieldset><legend>Tabela - <?php echo $_GET['table']; ?></
legend></fieldset>
                </section>
                <section class="form-bottom flex-center">
                    <button class="button-
normal __col3" name="b_add" title="Dodaj rekord">Dodaj!</button>
                    <button class="button-
normal __col3" name="b_clear" type="reset" title="Wyczyść formularz">Wyczyść</
button>
                </section>
            </form>
        </div>
    </div>

    <section class="flex-center form-info">
        <p></p>
        <div class="hidden"><?php include_once('tiles/get-URL.php'); ?></div>
    </section>

</main>
<script src="assets/js/form.js"></script>
<?php if ($_GET['table'] == 'Uprawnienia') { ?><script src="assets/js/rules.js
"></script><?php } ?>
```

```

<?php if ( isset($_GET['update']) && isset($_GET['id']) ) { ?><script src="assets/js/update.js"></script><?php } ?>
<?php if ($_GET['table'] == 'Użytkownicy' && isset($_GET['log'])) { ?><script src="assets/js/login.js"></script><?php } ?>
<?php if ($_GET['table'] == 'Użytkownicy' && isset($_GET['logout'])) {
    $res = sqlsrv_query($con, "SELECT [Nazwa użytkownika] FROM [dbo].func_get_Użytkownicy_Nazwa('" . $_COOKIE['id_user'] . "')");
    $nam = sqlsrv_fetch_array($res)['Nazwa użytkownika'];

    $query = "EXEC proc_add_to_Logi @nr_użytkownika='" . $_COOKIE['id_user'] . "', @data='" . date('Y-m-d H:i:s') . "', @opis='Użytkownik " . $nam . " wylogował się'";
    sqlsrv_query($con, $query);

    setcookie('id_user', -1, time() + 24*60*60, '/');
    header('location: form?table=Użytkownicy&log');
} ?>

```

Plik report.php

```
<?php $error = false; ?>
<main class="center __col2">
    <div class="flex-center">
        <?php
            if (!isset($_GET['table']) || (
                $_GET['table'] != 'Osoby' &&
                $_GET['table'] != 'Sklepy' &&
                $_GET['table'] != 'Kategorie' &&
                $_GET['table'] != 'Wydatki' &&
                $_GET['table'] != 'Logi' &&
                $_GET['table'] != 'Zasady' &&
                $_GET['table'] != 'Uprawnienia' &&
                $_GET['table'] != 'Użytkownicy')
            ) {
                ?>
                <div class="element-error flex-
center"><strong>Ups... Coś poszło nie tak...</strong></div>
                <?php
                    $error = true;
            }
            // If all is okay
            else {
                if (!isset($_COOKIE['id_user']) || $_COOKIE['id_user'] == -
1) header('location: form?table=Użytkownicy&log');
                ?>
                <div>
                    <h1 class="flex-
center"><p>Raport tabeli</p>:&nbsp;<strong><?php echo $_GET['table']; ?></stro
ng></h1>
                    <?php

                    if (!isset($_GET['page']) || !isset($_GET['count']) || $_GET['page
'] < 1) {
                        $_GET['page'] = 1;
                        $_GET['count'] = 10;
                        header('location: http://'.$_SERVER['SERVER_NAME'].'$_SERVER['P
HP_SELF'].'../report?table='.$_GET['table'].'&page='.$_GET['page'].'&count='.
$_GET['count'].'');
                    }

                    // Display the rest
                    $que = "EXECUTE proc_paging @table='view_select_all_'.$_GET['table
'].'", @offset='".($_GET['page']-
1)*$_GET['count'].'", @count='". $_GET['count']."'";
                    $res = sqlsrv_query($con, $que);
```



```

                if (gettype($row[$i]) == 'object') echo '<td>'
.date_format($row[$i], 'Y-m-d').'</td>';
                else if (isset($row['Kwota']) && $row[$i] == $
row['Kwota']) echo '<td>'.number_format(round($row[$i], 2), 2, '.', '').'</td>
';

                else if (isset($row['Opis']) && $row[$i] == $r
ow['Opis']) echo '<td><div class="cut-
string" style="width: 322px">'.$row[$i].'</div></td>';
                else echo '<td>'.$row[$i].'</td>';
            }
        ?>
        <td class="flex-center">
            <div>
                <?php /*if (($_GET['table'] == 'Uprawnienia')
&& ($_GET['table'] != 'Użytkownicy' || $row[0] != 0)) {
                    include('assets/img/svg/selection.svg');
                    include('assets/img/svg/pen.svg');
                    include('assets/img/svg/delete.svg');
                } else*/ if (($i1 != 1 && $i2 != 1 && $i3 != 1
&& $ia != 1) || ($_GET['table'] == 'Użytkownicy' && $row[0] == 0)) echo '<spa
n>Nie masz żadnych uprawnień.</span>'; else { ?>
                    <span title="Przeglądaj"><?php if ($i1 == 1) {
include('assets/img/svg/selection.svg'); } ?></span>
                    <span title="Edytuj"><?php if ($i2 == 1) { ech
o '<a href="form?table='.$_GET['table'].'&update&id='.$row[0].'">'; include('a
ssets/img/svg/pen.svg'); echo '</a>'; } ?></span>
                    <?php if ($_GET['table'] == 'Uprawnienia' && (
$row[0] == 0) || ($row[0] == $_COOKIE['id_user']) || $ia != 1)) echo '<span>N
ie masz żadnych uprawnień.</span>'; else { ?><span title="Usuń"><?php if ($ia
== 1 || $i3 == 1) include('assets/img/svg/trash.svg'); ?></span><?php } ?>
                    <?php } ?>
                </div>
            </td>
        </tr>
    <?php
    }
    echo '</table>';
    echo '</div>';

    echo '</div>';

    //if ($j == 0) header('location: main');
}
?>
</div>

<?php if (!$error) { ?>
<div class="__panel-report-container flex-center">
    <div>
        <div class="__panel-report-text">Ilość rekordów:</div>
        <div>

```

```

        <section class="__panel-report __col3 flex-center">
            <?php
                $counts = array(10, 25, 50, 100);
                for ($i = 0; $i < count($counts); $i++) {
                    if ($counts[$i] == $_GET['count']) { echo '<div>'.
$_GET['count'].'</div>'; }
                    else {
                        ?><div><a href="report?table=<?php echo $_GET[
'table']?>&page=1&count=<?php echo $counts[$i]; ?>"><?php echo $counts[$i]; ?>
</a></div><?php
                            }
                    }
                }
            ?>
        </section>
    </div>
</div>
<div class="flex-center">
    <i>Strona:&nbsp;</i>
    <?php
        // $res = sqlsrv_query($con, 'SELECT [dbo].func_get_table_count
('".$_GET['table'].'"', '".$_GET['count'].'')');
        $res = sqlsrv_query($con, "EXEC proc_get_table_count @table='"
. $_GET['table'] . "'");
        $row = sqlsrv_fetch_array($res);
        $lines = $row[0];
        if ($row[0] < $_GET['count']) $row[0] = 1;
        else $row[0] = (int) ($row[0]/$_GET['count'])+1;
        if ($lines % $_GET['count'] == 0) $row[0]--;
        $row[0]++;

        if ($row[0] > 5) {
            // First
            if ($_GET['page'] == 1) {
                ?>
                <div>1,&nbsp;</div>
                <?php } else { ?>
                <a href="report?table=<?php echo $_GET['table']?>&page
=1&count=<?php echo $_GET['count']; ?>"><div>1</div></a>,&nbsp;  
                <?php }
                if ($_GET['page'] > 4) echo '...&nbsp;  ';

                // Rest
                $j = $_GET['page'] + 2;
                for ($i = $_GET['page']-3; $i < $j; $i++) {
                    if ($i > 0 && $i < $row[0]-2) {
                        if ($i == $_GET['page']-
1) echo '<div>'.($i+1).'</div>';
                        else echo '<a href="report?table='.$_GET['table'].
'&page='.( $i+1).'&count='.$_GET['count'].'">div>'.($i+1).'</div></a>';
                        if ($i < $row[0] - 1) echo ',&nbsp;  ';
                    }
                }
            }
        }
    }

```



```

    }

    // Last
    if ($_GET['page'] - $row[0]+1 < -3) echo '...&nbsp;';
    if ($_GET['page'] == $row[0]-1) {
        ?>
        <div><?php echo $row[0]-1; ?></div>
        <?php } else { ?>
        <a href="report?table=<?php echo $_GET['table']; ?>&page=<?p
hp echo $row[0]-
1; ?>&count=<?php echo $_GET['count']; ?>"><div><?php echo $row[0]-
1; ?></div></a>

        <?php }
    }
    else {
        for ($i = 0; $i < $row[0]; $i++) {
            if ($i+1 == $_GET['page']) echo '<div>'.($i+1).</div>
';
            else echo '<a href="report?table='.$_GET['table'].'&pa
ge='.$_GET['page'].'&count='.$_GET['count'].'"><div>'.($i+1).</div></a>';

            if ($i < $row[0]-1) echo ',&nbsp;';

        }
    }
    ?>
</div>
</div>
<?php } ?>
</main>

```

Plik insert_new_record.php

```
<?php
    include_once('../../engine/connector.php');

    $res = null;
    switch ($_POST['table']) {
        case 'Osoby':
            $res = sqlsrv_query($con, 'EXEC proc_add_to_' . $_POST['table'] . ' @imię=' . $_POST['form']['Imię'] . ', @nazwisko=' . $_POST['form']['Nazwisko']);
            break;

        case 'Kategorie':
            $res = sqlsrv_query($con, 'EXEC proc_add_to_' . $_POST['table'] . ' @nazwa_kategorii=' . $_POST['form']['Nazwa kategorii']);
            break;

        case 'Sklepy':
            $res = sqlsrv_query($con, "EXEC proc_add_to_" . $_POST['table'] . " @nazwa='" . $_POST['form']['Nazwa'] . "', @adres='" . $_POST['form']['Adres'] . "', @kod_pocztowy='" . $_POST['form']['Kod pocztowy'] . "', @miasto='" . $_POST['form']['Miasto'] . "', @telefon='" . $_POST['form']['Telefon'] . "'");
            break;

        case 'Wydatki':
            $res = sqlsrv_query($con, "EXEC proc_add_to_" . $_POST['table'] . " '" . $_POST['form']['Kategoria'] . "', '" . $_POST['form']['Osoba'] . "', '" . $_POST['form']['Sklep'] . "', '" . $_POST['form']['Data'] . "', '" . $_POST['form']['Kwota'] . "', '" . $_POST['form']['Opis'] . '");
            break;

        case 'Użytkownicy':
            $res = sqlsrv_query($con, 'EXEC proc_add_to_' . $_POST['table'] . ' @nazwa=' . $_POST['form']['Nazwa użytkownika'] . ', @haslo=' . $_POST['form']['Hasło']);
            break;

        case 'Uprawnienia':
            $res = sqlsrv_query($con, 'EXEC proc_add_to_' . $_POST['table'] . ' ' . $_POST['form']['Użytkownik'] . ', ' . $_POST['form']['Zasada']);
            break;
    }

    // Add log
    $res = sqlsrv_query($con, "SELECT [Nazwa użytkownika] FROM [dbo].func_get_Użytkownicy_Nazwa('" . $_POST['user']['id'] . "')");
    $nam = sqlsrv_fetch_array($res)['Nazwa użytkownika'];
```

```
$query = "EXEC proc_add_to_Logi @nr_użytkownika='" . $_POST['user']['id']
. "', @data='" . date('Y-m-
d H:i:s') . "', @opis='Użytkownik " . $nam . " dodał nowy rekord do tabeli: "
. $_POST['table'] . "'";
sqlsrv_query($con, $query);

$err = (sqlsrv_errors()) ? true : false;

echo json_encode(
    array(
        'table' => $_POST['table'],
        'form' => $_POST['form'],
        'error' => $err,
        'errs' => sqlsrv_errors()
    )
);
?>
```

Plik update_record.php

```
<?php
    include_once('../../engine/connector.php');

    $res = null;
    switch ($_POST['table']) {
        case 'Osoby':
            $res = sqlsrv_query($con, "EXEC proc_update_to_".$_POST['table']."
            @id='".$_POST['id']."', @imie='".$_POST['form']['Imię']."', @nazwisko='".$_POST['form']['Nazwisko'].""");
            break;

            case 'Kategorie':
                $res = sqlsrv_query($con, 'EXEC proc_add_to_".$_POST['table'].' @nazwa_kategorii='.$_POST['form']['Nazwa Kategorii']);
                break;

            case 'Sklepy':
                $res = sqlsrv_query($con, "EXEC proc_add_to_".$_POST['table']." @nazwa='".$_POST['form']['Nazwa']."', @adres='".$_POST['form']['Adres']."', @kod_pocztowy='".$_POST['form']['Kod pocztowy']."', @miasto='".$_POST['form']['Miasto']."', @telefon='".$_POST['form']['Telefon'].""");
                break;

            case 'Wydatki':
                $res = sqlsrv_query($con, "EXEC proc_add_to_".$_POST['table']." '".$_POST['form']['Kategoria']."', '".$_POST['form']['Osoba']."', '".$_POST['form']['Sklep']."', '".$_POST['form']['Data']."', '".$_POST['form']['Kwota']."', '".$_POST['form']['Opis']);
                break;

            case 'Użytkownicy':
                $res = sqlsrv_query($con, 'EXEC proc_add_to_".$_POST['table'].' @nazwa='.$_POST['form']['Nazwa użytkownika'].' @haslo='.$_POST['form']['Hasło']);
                break;

            case 'Uprawnienia':
                $res = sqlsrv_query($con, 'EXEC proc_add_to_".$_POST['table'].' '.$_POST['form']['Użytkownik'].' '.$_POST['form']['Zasada']);
                break;
    }

    // Add log
    $res = sqlsrv_query($con, "SELECT [Nazwa użytkownika] FROM [dbo].func_get_Użytkownicy_Nazwa('".$_POST['user']['id']. "')");
    $nam = sqlsrv_fetch_array($res)['Nazwa użytkownika'];
```

```
$query = "EXEC proc_add_to_Logi @nr_użytkownika='" . $_POST['user']['id']
. "', @data='" . date('Y-m-
d H:i:s') . "', @opis='Użytkownik " . $nam . " zaktualizował rekord w tabeli:
" . $_POST['table'] . "'";
sqlsrv_query($con, $query);

$err = (sqlsrv_errors()) ? true : false;

echo json_encode(
    array(
        'table' => $_POST['table'],
        'form' => $_POST['form'],
        'error' => $err
    )
);
?>
```

Plik delete_record.php

```
<?php
    include_once('../../engine/connector.php');

    $x = "";
    if (isset($_POST['id2']) && $_POST['id2'] != 'undefined') $x = ", @id2='".
$_POST['id2']."'";
    $res = sqlsrv_query($con, "EXEC proc_delete_row @table='".$_POST['table'].
"', @id='".$_POST['id']."'".$x);

    // Add log
    $res = sqlsrv_query($con, "SELECT * FROM [dbo].[func_get_Użytkownicy_Nazwa]('" . $_COOKIE['id_user'] . "')");
    $nam = sqlsrv_fetch_array($res)['Nazwa użytkownika'];

    $query = "EXEC proc_add_to_Logi @nr_użytkownika='" . $_COOKIE['id_user']
. "', @data='" . date('Y-m-
d H:i:s') . "', @opis='Użytkownik " . $nam . " usunął rekord w tabeli: " . $_P
OST['table'] . "'";
    sqlsrv_query($con, $query);

    $err = (sqlsrv_errors()) ? true : false;

    echo json_encode(
        array(
            'error' => $err
        )
    );
?>
```

Plik manager.js

```
var id_palette      = 5,
    id_font_family = 'Times New Roman',
    id_font_color   = '#FFFFFF'; // previous: #444F0F, #494F09
var date_client     = getTime(new Date().getHours() + ':' + new Date().getMinutes() + ':' + new Date().getSeconds()),
    date_server     = null;

/*
    Gdy dokument się załaduje,
    jest gotowy do działania.
*/
$(document).ready(() => {
    // Ustaw ustawienia
    if (!$.cookie('id_palette')) { $.cookie('id_palette', id_palette, { expires: 9999, path: '/' }); }
    if (!$.cookie('id_font-family')) { $.cookie('id_font-family', id_font_family, { expires: 9999, path: '/' }); } else { $('body').css('font-family', $.cookie('id_font-family')); }
    if (!$.cookie('id_font-color')) { $.cookie('id_font-color', id_font_color, { expires: 9999, path: '/' }); } else { $('body').css('color', $.cookie('id_font-color')); }

    if (!$.cookie('id_user')) { $.cookie('id_user', -1, { expires: 9999, path: '/' }); }

    // Zarządzaj czasem
    date_server = getTime($('.server-time').text());
    timeUpdate(new Date(date_client), new Date(date_server));
    setInterval(clock, 1000);
    $('.client-timezone').text(Intl.DateTimeFormat().resolvedOptions().timeZone);

    // Dodaj metodę usuwania
    $(document).on('click', 'span[title="Usuń"]', clickDelete);

    // Pobierz kolory z palety
    var pal = new Palette($.cookie('id_palette'));

    // Stwórz style palety
    createPaletteStyle(pal);

    // Zarządzaj kolorami linków
    let bodyHex = $('.__col0').css('background-color');
    $('a').css('color', bodyHex);
    $('nav a').css('color', 'green');
});
```

```

class Palette {
    constructor(id) {
        let pal_obj = null;
        $.ajax({
            dataType: 'json',
            url: 'assets/data/colors.json',
            async: false,
            data: (data) => { console.log(data); },
            success: (data) => { print('Success! File "colors.json" was loaded
!'); pal_obj = data; },
            error: (xhr, ajaxOptions, thrownError) => { console.log(thrownError); }
        ));

        // Kolory palety
        this.cols = pal_obj.palettes[id];
        print('The number ' + id + ' palette was loaded!');
    }

    // Zwróć tablicę kolorów z palety
    getPalette() {
        return this.cols;
    }

    // Zwróć konkretny kolor z konkretnej palety
    getColor(col) {
        return this.cols[col];
    }
}

/* pal - tablica kolorów palety */
function createPaletteStyle (pal) {
    // Utwórz dynamiczne style
    let styles = '';
    for (i = 0; i < pal.cols.length; i++) {
        styles += '.__col' + i + ' { background-
color: #' + pal.cols[i].hex + ' }';
    }

    // Dla svg
    styles += 'svg { fill: ' + $.cookie('id_font-color') + ' }';

    // Dodaj
    $('head').append('<style>' + styles + '</style>');
}

```



```

// Programowa wiadomość
function print (msg, obj) {
    console.log('%cSystem: ' + '%c' + msg, 'color: red;', 'color: blue;', (obj
) ? obj : "");
}

// function css (element, property) {
//     return window.getComputedStyle(document.querySelectorAll(element), null
).getPropertyValue(eval(property));
// }

// alert(css('nav p', 'font-style'));

function getTime (date) {
    return new Date('2020-12-31 ' + date).getTime();
}

function clock() {
    date_client += 1000;
    date_server += 1000;

    let dc = new Date(date_client);
    let ds = new Date(date_server);

    timeUpdate(dc, ds);
}

function timeUpdate (dc, ds) {
    $('.client-time').text(
        ((dc.getHours() < 10) ? '0' + dc.getHours() : dc.getHours()) + ':' +
        ((dc.getMinutes() < 10) ? '0' + dc.getMinutes() : dc.getMinutes()) + '
:' +
        ((dc.getSeconds() < 10) ? '0' + dc.getSeconds() : dc.getSeconds()));
    $('.server-time').text(
        ((ds.getHours() < 10) ? '0' + ds.getHours() : ds.getHours()) + ':' +
        ((ds.getMinutes() < 10) ? '0' + ds.getMinutes() : ds.getMinutes()) + '
:' +
        ((ds.getSeconds() < 10) ? '0' + ds.getSeconds() : ds.getSeconds()));
}

function clickDelete () {
    if (confirm("Czy, aby na pewno chcesz usunąć ten rekord?")) {
        let parent_obj = $(this).closest('tr');
        let parent_val = parent_obj.attr('value');

        let table = $('h1 > strong').text();
    }
}

```

```

        let d = JSON.parse('{ "table": "' + table + '", "id": "' + parent_val.
split('-')[0] + '", "id2": "' + parent_val.split('-')[1] + '" }');

        // Usuń rekord
        $.ajax({
            url: 'site/tiles/delete_record.php',
            method: 'POST',
            dataType: 'json',
            data: d,
            async: false,
            success: (data) => { print('Success'); console.log('Received data:
', data); location.reload(); },
            error: (x, y, z) => { console.log('Error:', x, y, z); }
        })
        .done(() => { print('Done!'); });
    }
    else alert("Może i dobrze zrobiłeś ;)");
}

```

Strona również posiada „wbudowane style”. Mam na myśli to, że gdy zostanie załadowana paleta, do strony dopisywane jest 5 stylów – tyle ile jest kolorów.

Są to klasy, które wyglądają w ten sposób: „__col0”, „__col1” itd. do „__col4”. Tworzone są dynamicznie, to znaczy, że po odświeżeniu strony, skrypt „manager.js” dopisuje style do strony.

A oto kod, który generuje style:

```

/* pal - tablica kolorów palety */
function createPaletteStyle (pal) {
    // Utwórz dynamiczne style
    let styles = '';
    for (i = 0; i < pal.cols.length; i++) {
        styles += '.__col' + i + ' { background-
color: #' + pal.cols[i].hex + ' }';
    }

    // Dla svg
    styles += 'svg { fill: ' + $.cookie('id_font-color') + ' }';

    // Dodaj
    $('head').append('<style>' + styles + '</style>');
}

```

Jak widać, pliki svg mają taki sam kolor wypełnienia co czcionka!

Plik login.js

```
$(document).ready(loginReady);

function loginReady () {
    // Cancell that event
    $(document).off('click', 'form button[name="b_add"]', clickAdd);
    // Add new event
    $(document).on('click', 'form button[name="b_add"]', logIn);

    // Change title of page
    $('h1').html('Logowanie');
    $('legend').html('Podaj dane użytkownika');
    $('button[name=b_add]').html('Zaloguj!');

    // Remove one option
    $('select#Osoby').parent().parent().remove();
}

function logIn () {
    let login    = $('input[name="Nazwa użytkownika"]').val();
    let password = $('input[name="Hasło"]').val();

    let logged = false;
    let id = -1;

    $.ajax({
        url: 'site/tiles/login.php',
        method: 'POST',
        data: { "Nazwa": login, "Hasło": password },
        dataType: 'json',
        async: false,
        success: (data) => { console.log(data); logged = data.logged; if (!data.logged) $('.form-info > p').html('Wystąpił błąd przy logowaniu!'); $('input[type=password]').val(''); },
        error: (x, y, z) => { print('Error!', x, y, z); $('.form-info > p').html('Wystąpił błąd!'); }
    }).fail( function(xhr, textStatus, errorThrown) {
        console.log(xhr.responseText);
    });

    if (logged) {
        // Return Nr
        $.ajax({
            url: 'site/tiles/get_Użytkownicy_Nr.php',
            method: 'POST',
            data: { "Nazwa": login },
            dataType: 'json',
```

```
        async: false,
        success: (data) => { console.log(data); id = data.Nr; },
        error: (x, y, z) => { print('Error!', x, y, z); }
    }).fail( function(xhr, textStatus, errorThrown) {
        console.log(xhr.responseText);
    });

    $.cookie('id_user', id, { expires: 1, path: '/' });
    window.location.href = window.location.origin + window.location.pathname.substring(0, 25) + 'main';
}

return false;
}
```

Plik form.js

```
$(document).ready(formReady);

// json form.json file
var json = null,
    curr_table = null,
    datas = null;

function formReady () {
    // Add buttons events
    $(document).on('click', 'form button[name="b_add"]', clickAdd);

    // Get url datas
    datas = JSON.parse($('.form-info > div.hidden').text());

    // Getting json file
    $.ajax({
        data: 'json',
        url: 'assets/data/form.json',
        async: false,
        success: (data) => { json = data; },
        error: (x, y, z) => { print(x + ", " + y + ", " + z); }
    });

    // Adding all tags to form
    json[datas.table].elements.forEach(element => {
        // Attributes
        let attrs = '';
        let keys = Object.keys(element);
        let values = Object.values(element);

        for (let i = 1; i < keys.length; i++) {
            attrs += ' ' + keys[i] + '=' + values[i] + ' ';
        }

        let html_tag = '';
        if (element.tag !== 'select' && element.tag !== 'textarea') html_tag = '
<' + element.tag + attrs + ' />';
        else html_tag = '<' + element.tag + attrs + '></' + element.tag + '>';

        $('main form section.form-datas > fieldset').append('<div class="flex-
center"></div>');
        $('main form section.form-datas > fieldset > div:last-
child').append('<div class="flex-
right"><label for="' + element.name + '">' + element.name + ':</label></div>')
;
        $('main form section.form-datas > fieldset > div:last-
child').append('<div class="flex-left">' + html_tag + '</div>');
```

```

});

// Adding all options to elements
json[datas.table].select.forEach(element => {
    if (datas.table !== 'Uprawnienia') {
        let select = $('select[id="' + element + '"]');

        // Printing
        console.log(element, select);

        // Adding options
        $.ajax({
            url: 'site/tiles/get_all_options.php',
            method: 'POST',
            data: { "table": element, "reg": datas.register },
            dataType: 'html',
            async: false,
            success: (data) => { print('Success'); $('select#' + element).
html(data); },
            error: (x, y, z) => { console.log('Error:', x, y, z); }
        })
        .done(() => { print('Done!'); });
    }
});

// Assignment
curr_table = json[datas.table].elements;
}

function clickAdd () {
    let inputs    = document.forms[0].querySelectorAll('input');
    let selects   = document.forms[0].querySelectorAll('select');
    let textareas = document.forms[0].querySelectorAll('textarea');
    let result = true;

    // Inputs
    for (let i = 0; i < inputs.length; i++) {
        let regex = new RegExp(inputs[i].pattern);
        let value = inputs[i].value;

        let errorMsg = null;

        // Jeśli wystąpił problem
        if (!regex.test(value)) {
            errorMsg = 'Wprowadzono niepoprawną wartość. Wartość wymagana w fo
rmacie: ' + inputs[i].getAttribute('error');
        }
        else if (value == '') {
            errorMsg = 'Pole "' + inputs[i].name + '" nie może być puste!';
        }
    }
}

```

```

        else if (inputs[i].type === 'date' && (new Date(new Date().getFullYear(), new Date().getMonth(), new Date().getDate()).getTime()) > (new Date(inputs[i].value).getTime())) {
            errorMsg = 'Pole "' + inputs[i].name + '" może zawierać dzisiejszą datę lub późniejszą!';
        }

        if (errorMsg !== null) {
            print('Input "' + inputs[i].name + '" incorrect!');
            $(inputs[i]).addClass('__wrong').removeClass('__correct');
            inputs[i].focus();
            result = false;
            alert(errorMsg);
            $('form-info > p').html(errorMsg);
            break;
        }
        else $(inputs[i]).addClass('__correct').removeClass('__wrong');

        // if (!regex.test(value) || value === '' || (inputs[i].type === 'date' && (new Date(new Date().getFullYear(), new Date().getMonth(), new Date().getDate()).getTime()) > (new Date(inputs[i].value).getTime())) ) {
        //     print('Input "' + inputs[i].name + '" incorrect!');
        //     inputs[i].focus();
        //     $(inputs[i]).addClass('__wrong').removeClass('__correct');
        //     result = false;
        //     alert('Wprowadzone dane są niepoprawne!');
        //     break;
        // }
        // else $(inputs[i]).addClass('__correct').removeClass('__wrong');
    }

    $(selects).each((i, el) => {
        $(el).addClass('__correct').removeClass('__wrong');
    });

    // Textareas
    if (result)
        $(textareas).attr('required', true).each((i, el) => {
            if (el.value === '') {
                print('Textarea "' + el.name + '" incorrect!');
                el.focus();
                $(el).addClass('__wrong').removeClass('__correct');
                result = false;
                alert('Pole "' + el.name + '" nie może być puste!');
            }
            else $(el).addClass('__correct').removeClass('__wrong');
        });

    if (result) {
        // Creating basic JSON with basic structure
        let js = JSON.parse($('section.form-info > div').text());
    }

```

```

// JSON with form
let form_json = '{}';
if (inputs.length != 0) {
    inputs.forEach((el, i, array) => {
        form_json += '"' + el.name + '": ' + el.value + ' ';
        if (!(i === array.length - 1)) form_json += ', ';
    });
}
if (selects.length != 0) {
    if (inputs.length != 0) form_json += ', '; // end of inputs
    selects.forEach((el, i, array) => {
        form_json += '"' + el.name + '": ' + $(el).children(':selected').attr('id') + ' ';
        if (!(i === array.length - 1)) form_json += ', ';
    });
}
if (textareas.length != 0) {
    if (inputs.length != 0 || selects.length != 0) form_json += ', '; //
// end of inputs/selects
    textareas.forEach((el, i, array) => {
        form_json += '"' + el.name + '": ' + el.value + ' ';
        if (!(i === array.length - 1)) form_json += ', ';
    });
}

form_json += '>';

// Completing
console.log(form_json);
js.form = JSON.parse(form_json);
js.user = JSON.parse('{ "id": ' + $.cookie('id_user') + ' }');
console.log('Given JSON:', js);

let samename = false;
if (datas.register == true) {
    $.ajax({
        url: 'site/tiles/get_Użytkownicy_Ile.php',
        method: 'POST',
        data: { "Nazwa": $('input[name="Nazwa użytkownika"]').val() },
        dataType: 'json',
        async: false,
        success: (data) => { print('Count:', data.count); if (data.count != null) samename = true; },
        error: (x, y, z) => { console.log('Error:', x, y, z); }
    })
    .done(() => {
        console.log('Counted...');
        if (samename) $('form-info > p').html('Użytkownik o podanej nazwie istnieje już w bazie!');
    });
}

```



```

        if (samename == false || samename == undefined) {
            // Dodaj rekord
            $.ajax({
                url: 'site/tiles/insert_new_record.php',
                method: 'POST',
                data: js,
                dataType: 'json',
                async: false,
                success: (data) => { print('Success'); console.log('Received d
ata:', data); alert('Rekord został dodany poprawnie!'); $('<form-
info > p').html('Rekord został dodany poprawnie!'); $('<form-
info').addClass('form-info-
notempty'); $('<__correct').removeClass('__correct');
                },
                error: (x, y, z) => { console.log('Error:', x, y, z); }
            })
            .done(() => { print('Done!'); document.forms[0].reset(); document.
forms[0].elements[1].focus(); if (datas.table == 'Uprawnienia') $('<select[name
="Zasada"]').html(''); });
        }
    }
    return false;
}

function getRegExp (expression) {
    return '/' + expression + '/';
}

```

Plik update.js

```
$(document).ready(updateReady);

function updateReady () {
    // Remove button
    $(document).off('click', 'form button[name="b_add"]', clickAdd);
    $(document).on('click', 'form button[name="b_add"]', clickUpdate);
    $('form button[name="b_clear"]').attr('disabled', 'disabled');

    $('h1 > p').text('Edycja tabeli');
    $('button[name="b_add"]').html('Edytuj!');

    let json = JSON.parse($('.form-info > div.hidden').text());
    let json_req = null;

    // Load all fields
    $.ajax({
        url: 'site/tiles/select_record.php',
        method: 'POST',
        data: json,
        dataType: 'html',
        async: false,
        success: (data) => { print('Success'); json_req = JSON.parse(data); json_req.data.Kwota = '0' + json_req.data.Kwota; },
        error: (x, y, z) => { console.log('Error:', x, y, z); }
    })
    .done(() => { print('Done!'); });

    // If all is good
    if (json_req != null) {
        // If data is null
        if (json_req.data == null) $('.main > div.flex-center > div').html('Uppss... Coś poszło nie tak :/');
        else {
            Object.keys(json_req.data).forEach((el, id, arr) => {
                if ($('#input[name="' + el + '"]').length != 0) {
                    if (typeof json_req.data[el] == 'object') $('#input[name="' + el + '"]').val(json_req.data[el].date.split(' ')[0]);
                    else $('#input[name="' + el + '"]').val(json_req.data[el]);
                } else
                if ($('#select[name="' + el + '"]').length != 0) {
                    $('#select[name="' + el + '"]').val(json_req.data[el]);
                } else
                if ($('#textarea[name="' + el + '"]').length != 0) {
                    $('#textarea[name="' + el + '"]').val(json_req.data[el]);
                }
            });
        }
    }
}
```

```

    }
}

function clickUpdate () {
    let inputs    = document.forms[0].querySelectorAll('input');
    let selects   = document.forms[0].querySelectorAll('select');
    let textareas = document.forms[0].querySelectorAll('textarea');
    let result = true;

    // Inputs
    for (let i = 0; i < inputs.length; i++) {
        let regex = new RegExp(inputs[i].pattern);
        let value = inputs[i].value;

        // Jeśli wystąpił problem
        if (!regex.test(value) || value === '' || (inputs[i].type === 'date' &
& (new Date(new Date().getFullYear(), new Date().getMonth(), new Date().getDate()).getTime()) > (new Date(inputs[i].value).getTime()) )) {
            print('Input "' + inputs[i].name + '" incorrect!');
            inputs[i].focus();
            $(inputs[i]).addClass('__wrong').removeClass('__correct');
            result = false;
            alert('Wprowadzone dane są niepoprawne!');
            break;
        }
        else $(inputs[i]).addClass('__correct').removeClass('__wrong');
    }

    $(selects).each((i, el) => {
        $(el).addClass('__correct').removeClass('__wrong');
    });

    // Textareas
    if (result)
    $(textareas).attr('required', true).each((i, el) => {
        if (el.value === '') {
            print('Textarea "' + el.name + '" incorrect!');
            el.focus();
            $(el).addClass('__wrong').removeClass('__correct')
            result = false;
            alert('Wprowadzone dane są niepoprawne!');
        }
        else $(el).addClass('__correct').removeClass('__wrong');
    });

    if (result) {
        // Creating basic JSON with basic structure
        let js = JSON.parse($('section.form-info > div').text());

        // JSON with form
        let form_json = '{}';
    }
}

```

```

        if (inputs.length != 0) {
            inputs.forEach((el, i, array) => {
                form_json += '"' + el.name + '": ' + el.value + '";';
                if (!(i === array.length - 1)) form_json += ', ';
            });
        }
        if (selects.length != 0) {
            if (inputs.length != 0) form_json += ', '; // end of inputs
            selects.forEach((el, i, array) => {
                form_json += '"' + el.name + '": ' + $(el).children(':selected').attr('id') + '";';
                if (!(i === array.length - 1)) form_json += ', ';
            });
        }
        if (textareas.length != 0) {
            if (inputs.length != 0 || selects.length != 0) form_json += ', '; // end of inputs/selects
            textareas.forEach((el, i, array) => {
                form_json += '"' + el.name + '": ' + el.value + '";';
                if (!(i === array.length - 1)) form_json += ', ';
            });
        }

        form_json += '>';

        // Completing
        js.form = JSON.parse(form_json);
        js.user = JSON.parse('{ "id": ' + $.cookie('id_user') + ' }');
        console.log('Given JSON:', js);

        // Update subquery
        // let que = "";
        // Object.keys(js.form).forEach((el, id, arr) => {
        //     que += "[" + el + "] = " + Object.values(js.form)[id] + '";';
        //     if (arr.length-1 != id) que += ", ";
        // });
        // js.subquery = que;

        // Dodaj rekord
        $.ajax({
            url: 'site/tiles/update_record.php',
            method: 'POST',
            data: js,
            dataType: 'json',
            async: false,
            success: (data) => { print('Success'); console.log('Received data: ', data); $('#form-info > p').html('Rekord został zaktualizowany poprawnie!'); $('#form-info').addClass('form-info-notempty'); $('#__correct').removeClass('__correct');
            },

```

```
        error: (x, y, z) => { console.log('Error:', x, y, z); }
    })
    .done(() => { print('Done!'); });
}

return false;
}
```

Plik settings.js

Jednakże, na szczególną uwagę zasługują ten plik:

```
$(document).ready(formReady);

// json form.json file
var json = null,
    pals = null;

function formReady () {
    // Add buttons events
    $(document).on('click', 'button[name="b_save"]', clickSave);

    // Getting json file
    $.ajax({
        data: 'json',
        url: 'assets/data/colors.json',
        async: false,
        success: (data) => { json = data; pals = data; },
        error: (x, y, z) => { print(x + ", " + y + ", " + z); }
    });

    // Adding all palettes
    json.palettes.forEach((el, i, arr) => {
        $('main fieldset#__main-settings select#__main-settings-list').append('<option id="col' + i + '" value="" + i + '">Paleta ' + (i + 1) + '</option>');
    });

    // -----

    // Getting json file
    $.ajax({
        data: 'json',
        url: 'assets/data/settings.json',
        async: false,
        success: (data) => { json = data; },
        error: (x, y, z) => { print(x + ", " + y + ", " + z); }
    });

    // Adding all font families
    json.settings['font-family'].forEach((el, i, arr) => {
        $('main fieldset#__main-settings select#__main-settings-font-family').append('<option id="" + i + '" style="font-family: ' + el + '" value="" + el + '">' + el + '</option>');
    });
}
```

```

// -----
// -----

// json.settings['font-color'].forEach((el, i, arr) => {
//     $('main fieldset#__main-settings select#__main-settings-font-
color').append('<option id="' + i + '" style="color: ' + el + '" value="' + el
+ '>' + el + '</option>');
// });
// let num = 0; // 16777215 = 0xFFFFFFFF
// // 65536
// for (let n = num; n < (0xFFFFFFFF / 65536) + 1; n++) {
//     let hex = (n * 65536).toString(16);

//     // Jeżeli potrzeba zer...
//     while (hex.length < 6) {
//         hex = '0' + hex;
//     }

//     console.log(hex);

//     $('main fieldset#__main-settings select#__main-settings-font-
color').append('<option id="' + n + '" style="color: ' + ('#' + hex) + '" valu
e="' + ('#' + hex) + '>' + ('#' + hex) + '</option>');
// }

// Adding all font colors
for (let n = 0; n < 4096; n+=15) {
    let bin = n.toString(4);
    // Jeżeli potrzeba zer...
    while (bin.length < 6) {
        bin = '0' + bin;
    }

    let hex = bin;
    hex = hex.replaceAll('0', '0');
    hex = hex.replaceAll('1', '4');
    hex = hex.replaceAll('2', '9');
    hex = hex.replaceAll('3', 'F');

    console.log(bin, hex);

    $('main fieldset#__main-settings select#__main-settings-font-
color').append('<option id="' + n + '" style="color: ' + ('#' + hex) + '" valu
e="' + ('#' + hex) + '>' + ('#' + hex) + '</option>');
}
$('main fieldset#__main-settings select#__main-settings-font-
color').append('<option style="color: #FFFFFF" value="#FFFFFF">#FFFFFF</option
>');

```

```

// -----
// -----

// Select settings
$('#select#__main-settings-
list option[value="' + $.cookie('id_palette') + '"').attr('selected', 'select
ed');
$('#select#__main-settings-font-family option[value="' + $.cookie('id_font-
family') + '"').attr('selected', 'selected');
$('#select#__main-settings-font-color option[value="' + $.cookie('id_font-
color') + '"').attr('selected', 'selected');

// Add events
$('#select#__main-settings-list').change(function () {
    $('#head style').remove();

    let el = $(this).children('option:selected').val();
    createPaletteStyle(new Palette(el));
});
$('#select#__main-settings-font-family').change(function () {
    let el = $(this).children('option:selected').val();
    $('body').css('font-family', el);
});
$('#select#__main-settings-font-color').change(function () {
    let el = $(this).children('option:selected').val();
    $('body').css('color', el);
});
}

function clickSave () {
    let id_palette = $('#select#__main-settings-
list').children('option:selected').val();
    let id_font_family = $('#select#__main-settings-font-
family').children('option:selected').val();
    let id_font_color = $('#select#__main-settings-font-
color').children('option:selected').val();

    // console.log(
    //     id_palette,
    //     id_font_family,
    //     id_font_color
    // );

    $.cookie('id_palette', id_palette, { expires: 9999, path: '/' });
    $.cookie('id_font-family', id_font_family, { expires: 9999, path: '/' });
    $.cookie('id_font-color', id_font_color, { expires: 9999, path: '/' });

    location.reload();
}

```


Szczególnie należy zwrócić uwagę na ten fragment:

```
// Adding all font colors
for (let n = 0; n < 4096; n+=15) {
  let bin = n.toString(4);
  // Jeżeli potrzeba zer...
  while (bin.length < 6) {
    bin = '0' + bin;
  }

  let hex = bin;
  hex = hex.replaceAll('0', '0');
  hex = hex.replaceAll('1', '4');
  hex = hex.replaceAll('2', '9');
  hex = hex.replaceAll('3', 'F');

  console.log(bin, hex);

  $('main fieldset#__main-settings select#__main-settings-font-color').append('<option id="' + n + '" style="color: ' + ('#' + hex) + '" value="' + ('#' + hex) + '">' + ('#' + hex) + '</option>');
}
```

Ten kod dodaje wszystkie kombinacje kolorów w podstronie ustawień. Można go bardziej zuniwersalizować.

Zanim udało mi się napisać ten fragment kodu, musiałem poradzić sobie z problemem, który doskwierał od kilku lat. Chciałem wygenerować paletę kolorów.

Rozwiązanie przyszło oczywiście od strony matematyki. Zastosowałem wariacje z powtórzeniami z działu kombinatoryki. Otóż swój problem przedstawiłem w następujący sposób:

Liczba heksadecymalna jest złożona z hasha (tzw. płotka) oraz 6-ciu cyfr. Skupmy się na tych cyfrach. Dla każdej cyfry może przypisać aż 16 możliwości (0-9 oraz A-F). Zatem, tworzy nam się wariacja sześćcio-elementowa zbioru szesnastoelementowego (16^6). W sposób matematyczny można przedstawić: $(_, _, _, _, _, _)$, dla każdego elementu 16 możliwości. Nie będę kalkulować jaka to liczba, lecz właśnie wynik tego to wszystkie kombinacje kolorów.

Moim celem na początku było napisanie takiego algorytmu, który wygeneruje 256 kolorów, ponieważ wszystkie te kolory umieszczałem w 'select'ie, a byłoby to niekomfortowe wstawiać więcej opcji (przewijanie całych stron kolorów).

Kolejnym etapem mojego rozumowania było zmniejszenie liczby możliwości do dwóch, a mianowicie: 2^6 , co daje nam 64 kolory. Ustaliłem, że 0 będzie równe 7, a 1 będzie równe F (zatem będą to jasne kolory).

Jednakże było to zbyt mało, zatem zwiększyłem ilość możliwości do 3, czyli otrzymałem 3^6 , a to jest równe 729 kolorom. Jednakże, jest to już liczba zbyt duża na wstawienie tylu opcji. Uznałem, że nie ma sensu wstawiać tyle kolorów.

Co ciekawe, okazywało się, że kolory są bardzo podobne. Zmieniłem swój tok rozumowania, gdy jako liczbę możliwości ustaliłem 4 (4^6), czyli otrzymałem 4096 kombinacji kolorów. Oczywiście, była to zbyt duża liczba, by wstawić je do selecta.

Wtedy to wpadłem na pomysł, jak można ominąć podobne kolory! Postanowiłem, że pętla będzie wyświetlać co 16 kolor, zatem liczba kolorów z 4096 zmniejszyła się do 256. Nota bene, były to kolory różniące się od siebie. W dodatku, były to odcienie czerwone, zielone, żółte, niebieskie, fioletowe, różowe oraz inne.

Tak oto udało mi się opracować sposób na generowanie kolorów. Kwestie uniwersalizacji pozostawię dla chętnych.

Oczywiście nie uważam, że owy sposób jest najlepszy, ponieważ dostrzegam w nim kilka wad. Jednakże, bardzo mnie ucieszył i utwierdził, jak ważna jest matematyka w programowaniu.

W przypadku zmieniania kolorów lub palet, gdy jeszcze nie zapisaliśmy zmian, to po zmianie jakiejś opcji uruchamia się poniższy kod, który usuwa element „style” ze strony, a następnie dodaje nową paletę, dzięki czemu widoczny jest efekt zmiany strony, przy nieodświeżaniu jej:

```
// Add events
$('select#__main-settings-list').change(function () {
    $('head style').remove();

    let el = $(this).children('option:selected').val();
    createPaletteStyle(new Palette(el));
});
$('select#__main-settings-font-family').change(function () {
    let el = $(this).children('option:selected').val();
    $('body').css('font-family', el);
});
$('select#__main-settings-font-color').change(function () {
    let el = $(this).children('option:selected').val();
    $('body').css('color', el);
});
```

Folder data

Wspomniany na początku folder 'data' zasługuje na swój mały podrozdział. Znajdują się w nim pliki '.json', które zawierają informacje o polach formularza, paletach kolorów, ikonkach akcji oraz inne.

actions.js

```
{
  "Osoby": {
    "pen": true,
    "delete": true,
    "selection": false
  },
  "Sklepy": {
    "pen": true,
    "delete": true,
    "selection": false
  },
  "Kategorie": {
    "pen": true,
    "delete": true,
    "selection": false
  },
  "Wydatki": {
    "pen": true,
    "delete": true,
    "selection": false
  },
  "Logi": {
    "pen": false,
    "delete": true,
    "selection": false
  },
  "Zasady": {
    "pen": false,
    "delete": false,
    "selection": false
  },
  "Uprawnienia": {
    "pen": false,
    "delete": true,
    "selection": false
  },
  "Użytkownicy": {
    "pen": false,
    "delete": true,
    "selection": false
  }
}
```

Każde pole tego obiektu to osobna tabelka. Każde z tych pól zawiera trzy pola: „pen”, „delete”, „selection”. Wszystkie one dotyczą raportów. Pierwsze pole odpowiedzialne jest za pozwolenie na edycję rekordu. Drugie pole zezwala na wyświetlenie ikonki, która po

kliknięciu usuwa wybrany rekord. Ostatnie pole odpowiada za wyświetlenie ikonki do wyświetlenia szczegółów rekordu.

Oczywiście, jeżeli użytkownik nie będzie posiadał żadnego ze wspomnianych uprawnień, wtedy nie zostanie wyświetlona poszczególna ikonka.

colors.json

```
{
  "palettes": [
    [
      {"hex": "3E1BF9"},
      {"hex": "6C18DB"},
      {"hex": "B527F2"},
      {"hex": "D618DB"},
      {"hex": "F91B95"}
    ],
    [
      {"hex": "51D663"},
      {"hex": "7FED66"},
      {"hex": "5EF79E"},
      {"hex": "98D651"},
      {"hex": "E5F75E"}
    ],
    [
      {"hex": "5B45DE"},
      {"hex": "4E63FA"},
      {"hex": "9958F5"},
      {"hex": "AE45DE"},
      {"hex": "F44EFA"}
    ],
    [
      {"hex": "4C4C4C"},
      {"hex": "636363"},
      {"hex": "85C7F2"},
      {"hex": "DBDBDB"},
      {"hex": "D1D1D1"}
    ],
    [
      {"hex": "FF6B35"},
      {"hex": "F7C59F"},
      {"hex": "EFEFD0"},
      {"hex": "004E89"},
      {"hex": "1A659E"}
    ],
    [
      {"hex": "1F363D"},
      {"hex": "40798C"},
      {"hex": "70A9A1"},
      {"hex": "9EC1A3"},
      {"hex": "CFE0C3"}
    ],
    [
      {"hex": "000000"},
      {"hex": "363946"},

```

```
        {"hex": "696773"},
        {"hex": "819595"},
        {"hex": "B1B6A6"}
    ],
    "source": "https://coolors.co/",
    "created": "2020-12-18",
    "modified": "2021-02-02",
    "author": "Marek Kurańda"
}
```

Jest to plik odpowiedzialny za palety. Zawiera 4 palety, choć można dodać również własne. Należy pamiętać, że każda paleta zawiera pięć kolorów, a kolory są ustawione od najciemniejszych do najjaśniejszych.

fields.json

```
{
  "Osoby": [
    "Nr",
    "Imię",
    "Nazwisko"
  ],
  "Sklepy": [
    "Nr",
    "Nazwa",
    "Adres",
    "Kod pocztowy",
    "Miasto",
    "Telefon"
  ],
  "Kategorie": [
    "Nr",
    "Nazwa Kategorii"
  ],
  "Wydatki": [
    "Nr",
    "Kategoria",
    "Osoba",
    "Sklep",
    "Data",
    "Kwota",
    "Opis"
  ],
  "Logi": [
    "Nr",
    "Użytkownik",
    "Data",
    "Opis"
  ],
  "Zasady": [
    "Nr",
    "Nazwa",
    "Opis"
  ],
  "Uprawnienia": [
    "Nazwa użytkownika",
    "Nazwa zasady"
  ],
  "Użytkownicy": [
    "Nr",
    "Nazwa użytkownika",
    "Osoba"
  ]
}
```


}

Plik odpowiedzialny jest za nazwy pól w raportach – to właśnie z tego pliku pobiera sobie aplikacja nazwy pól.

form.json

```
{
  "Osoby": {
    "elements": [
      {
        "tag": "input", "type": "text", "name": "Imię", "value": "", "pattern": "^[A-ZĘÓĄŚŁŻŻĆŃ][a-zA-Zęóąśłżżćń]{1,29}$", "error": "Pierwsza duża litera, a drugi znak dowolny. Minimalna długość 2 znaki, maksymalna 30 znaków. Znaki polskie dozwolone."},
      {
        "tag": "input", "type": "text", "name": "Nazwisko", "value": "", "pattern": "^[A-ZĘÓĄŚŁŻŻĆŃ][a-zA-Zęóąśłżżćń]{1,29}$", "error": "Pierwsza duża litera, a drugi znak dowolny. Minimalna długość 2 znaki, maksymalna 30 znaków. Znaki polskie dozwolone."}
    ],
    "select": []
  },
  "Kategorie": {
    "elements": [
      {
        "tag": "input", "type": "text", "name": "Nazwa kategorii", "value": "", "pattern": "^[A-ZĘÓĄŚŁŻŻĆŃ]{1}{1,48}$", "error": "Pierwsza litera duża. Pozostałe 48 znaków dowolne."}
    ],
    "select": []
  },
  "Sklepy": {
    "elements": [
      {
        "tag": "input", "type": "text", "name": "Nazwa", "value": "", "pattern": "^.{3,50}$", "error": "Minimalna długość 3 znaki, maksymalna 50 znaków."},
      {
        "tag": "input", "type": "text", "name": "Adres", "value": "", "pattern": "^.{3,50}$", "error": "Minimalna długość 3 znaki, maksymalna 50 znaków."},
      {
        "tag": "input", "type": "text", "name": "Kod pocztowy", "value": "", "pattern": "^[0-9]{2}-[0-9]{3}$", "placeholder": "00-000", "error": "Wymagany wzór: NN-NNN, gdzie N to liczba od 0 do 9."},
      {
        "tag": "input", "type": "text", "name": "Miasto", "value": "", "pattern": "^[A-ZĘÓĄŚŁŻŻĆŃ][a-zA-Zęóąśłżżćń]{1,49}$", "error": "Pierwsza litera duża. Minimalna długość 2 znaki, maksymalna 50."},
      {
        "tag": "input", "type": "text", "name": "Telefon", "value": "", "pattern": "^[0-9]{1}[0-9]{2}[0-9]{6}$", "placeholder": "(00)000000", "error": "Wymagany wzór: (NN)NNNNNN, gdzie N to dowolna cyfra od 0 do 9."}
    ],
    "select": []
  },
  "Wydatki": {
    "elements": [
```

```

        {"tag": "select", "name": "Kategoria", "value": "", "id": "Kategoria"},
        {"tag": "select", "name": "Osoba", "value": "", "id": "Osoby"},
        {"tag": "select", "name": "Sklep", "value": "", "id": "Sklepy"},
        {"tag": "input", "type": "date", "name": "Data", "value": "", "required": ""},
        {"tag": "input", "type": "number", "step": "0.01", "name": "Kwota", "value": "0.01", "min": "0.01", "max": "999999.99"},
        {"tag": "textarea", "name": "Opis", "value": "", "cols": "20", "rows": "4", "style": "resize: none;", "pattern": "^.{1,79}$", "required": "", "error": "Minimalna długość 1 znak, maksymalna 79 znaków"}
    ],
    "select": [
        "Kategorie",
        "Osoby",
        "Sklepy"
    ]
},
"Użytkownicy": {
    "elements": [
        {"tag": "input", "type": "text", "name": "Nazwa użytkownika", "value": "", "pattern": "^[a-zA-Z']{3,30}$", "error": "Minimalna długość 3 znaki, maksymalna 30 znaków. Bez polskich znaków - czysty łaciński alfabet wyłącznie!"},
        {"tag": "select", "name": "Osoba", "value": "", "id": "Osoby"},
        {"tag": "input", "type": "password", "name": "Hasło", "value": "", "pattern": "^[a-zA-Z0-9!@#${8,32}$", "error": "Dozwolone znaki: małe i duże litery alfabetu łacińskiego, !, @, #, $. Minimalna długość 8 znaków, maksymalna 32 znaki."}
    ],
    "select": [
        "Osoby"
    ]
},
"Uprawnienia": {
    "elements": [
        {"tag": "select", "name": "Użytkownik", "value": "", "id": "Użytkownicy"},
        {"tag": "select", "name": "Zasada", "value": "", "id": "Zasady_Niepasujące"}
    ],
    "select": [
        "Użytkownicy",
        "Zasady"
    ]
}
}

```

Ten plik zawiera wszystkie elementy formularzy. Dodając kolejne obiekty do tego pola, spowoduje pojawienie się tych pól w danym formularzy.

settings.json

Jest to plik odpowiedzialny za ustawienia. Zawiera trzy pola: „settings”, „author” oraz „creation-date”. Pole settings zawiera nazwy czcionek oraz kolory, które już są bezużyteczne z powodu algorytmu.

Aby dodać nowe czcionki wystarczy dopisać nowe pole w tablicy „font-family”.

```
{
  "settings": {
    "font-family": [
      "Arial",
      "Calibri",
      "Century",
      "Consolas",
      "Comic Sans MS",
      "Constantia",
      "Cooper Black",
      "Times New Roman"
    ],
    "font-color": [
      "#000000",
      "#777777",
      "#FFFFFF",
      "#770000",
      "#007700",
      "#000077",
      "#FF0000",
      "#00FF00",
      "#0000FF"
    ]
  },
  "author": "Marek Kurańda",
  "creation-date": "2021-01-14"
}
```

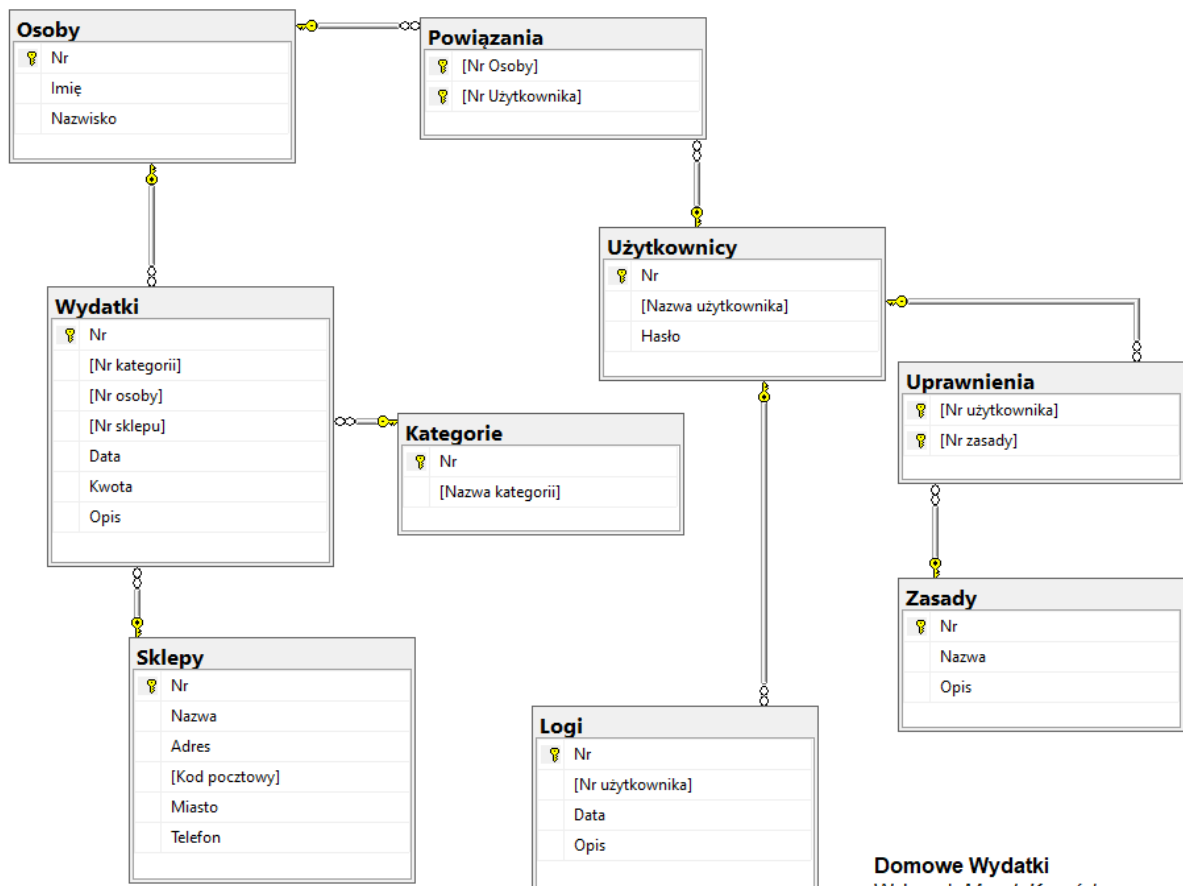
Pliki cookie

W projekcie korzystam z biblioteki „jquery-cookies”, która upraszcza tworzenie ciasteczek za pomocą JavaScript. Poniżej zostały wypisane wszystkie pliki, które są możliwe do utworzenia i wykorzystania:

Nazwa	Wykorzystanie oraz wartość
id_font-color	Przechowuje heksadecymalny kolor czcionki
id_font-family	Przechowuje nazwę czcionki
id_palette	Przechowuje numer palety. Jest to numer elementu w tablicy „palettes” z pliku „colors.json”
id_user	Domyślnie -1 (gość). Przechowuje numer identyfikacyjny zalogowanego użytkownika

Dzięki tym plikom strona posiada kolor, paletę oraz wie kto jest zalogowany, o ile jest.

Schemat bazy danych




Tabelki

Kategorie

Wszystkie kategorie dotyczące wydatków

[Nr] – numer identyfikacyjny, klucz główny - int

[Nazwa kategorii] – słowna nazwa - nvarchar(50)

DESKTOP-I1PANLB.D...ki - dbo.Kategorie			
	Column Name	Data Type	Allow Nulls
	Nr	int	<input type="checkbox"/>
	[Nazwa kategorii]	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Logi

Zawiera wszystkie komunikaty, wygenerowane przez użytkowników, np. dodanie rekordu, usunięcie użytkownika, dodanie uprawnień, zaktualizowanie rekordu itp.

[Nr] - numer identyfikacyjny, klucz główny - int

[Nr użytkownika] – klucz obcy do tabeli użytkownicy. Jest to numer użytkownika, wykonywującego daną akcję.

[Data] – data akcji - smalldatetime

[Opis] – opis wykonywanej akcji- varchar(100)

DESKTOP-I1PANLB.D...ydatki - dbo.Logi			
	Column Name	Data Type	Allow Nulls
▶	Nr	int	<input type="checkbox"/>
	[Nr użytkownika]	int	<input checked="" type="checkbox"/>
	Data	smalldatetime	<input checked="" type="checkbox"/>
	Opis	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


Osoby

Zawiera listę wszystkich osób. Osoby są wykorzystywane w tabeli „Wydatki”, aby przypisać jakiś wydatek.

[Nr] - numer identyfikacyjny, klucz główny - int

[Imię] – słowne imię - nvarchar(30)

[Nazwisko] – słowne nazwisko - nvarchar(30)

DESKTOP-I1PANLB.D...datki - dbo.Osoby			
	Column Name	Data Type	Allow Nulls
	Nr	int	<input type="checkbox"/>
	Imię	nvarchar(30)	<input checked="" type="checkbox"/>
	Nazwisko	nvarchar(30)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Powiązania

Zawiera listę przypisanych osób do użytkowników. Każdy użytkownik jest tworzony na podstawie danej osoby. Jeden użytkownik odpowiada jednej osobie.

[Nr Osoby] - klucz główny - int

[Nr Użytkownika] - klucz główny – int

DESKTOP-I1PANLB.D...i - dbo.Powiazania			
	Column Name	Data Type	Allow Nulls
▶	[Nr Osoby]	int	<input type="checkbox"/>
▶	[Nr Użytkownika]	int	<input type="checkbox"/>
			<input type="checkbox"/>

Sklepy

Zawiera listę wszystkich sklepów.

[Nr] - numer identyfikacyjny, klucz główny - int

[Nazwa] – słowna nazwa sklepu, np. „U Romka” - nvarchar(50)

[Adres] – adres sklepu, np. „ul. Słoneczna 12” - nvarchar(50)

[Kod pocztowy] – kod pocztowy, zgodnie z maską: „nn-nnn” - nvarchar(6)

[Miasto] – nazwa miasta, gdzie ulokowany jest dany sklep, np. „Kraków” - nvarchar(50)

[Telefon] – nr telefonu do sklepu, zgodnie z maską: „(nn)nnnnnn” - nvarchar(10)

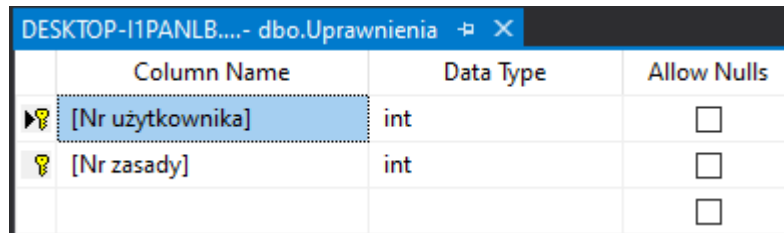
DESKTOP-I1PANLB.D...datki - dbo.Sklepy			
	Column Name	Data Type	Allow Nulls
PK	Nr	int	<input type="checkbox"/>
	Nazwa	nvarchar(50)	<input checked="" type="checkbox"/>
	Adres	nvarchar(50)	<input checked="" type="checkbox"/>
	[Kod pocztowy]	nvarchar(6)	<input checked="" type="checkbox"/>
	Miasto	nvarchar(50)	<input checked="" type="checkbox"/>
	Telefon	nvarchar(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Uprawnienia

Zawiera listę użytkowników wraz przypisanymi do nich prawami, np. admin oraz możliwość modyfikacji danych.

[Nr użytkownika] - klucz główny - int

[Nr zasady] - klucz główny – int



DESKTOP-I1PANLB....- dbo.Uprawnienia			
	Column Name	Data Type	Allow Nulls
▶ 🔑	[Nr użytkownika]	int	<input type="checkbox"/>
🔑	[Nr zasady]	int	<input type="checkbox"/>
			<input type="checkbox"/>

Użytkownicy

Zawiera listę wszystkich użytkowników, wraz z ich danymi.

[Nr] - numer identyfikacyjny, klucz główny - int

[Nazwa użytkownika] – słowna nazwa użytkownika - varchar(30)

[Hasło] – zaszyfrowane hasło użytkownika - varchar(30)

DESKTOP-I1PANLB....- dbo.Użytkownicy			
	Column Name	Data Type	Allow Nulls
▶	Nr	int	<input type="checkbox"/>
	[Nazwa użytkownika]	varchar(30)	<input checked="" type="checkbox"/>
	Hasło	varchar(32)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Wydatki

Lista wszystkich wydatków. Jest to najbogatsza tabela spośród wszystkich.

[Nr] - numer identyfikacyjny, klucz główny - int

[Nr kategorii] – klucz obcy do tabeli Kategorie - int


[Nr osoby] – klucz obcy do tabeli Osoby - int

[Nr sklepu] – klucz obcy do tabeli Sklepy - int

[Data] – data wydatku - datetime

[Kwota] – ilość pieniężna wydatku - money

[Opis] – dodatkowy opis nt. wydatku - nvarchar(80)

DESKTOP-I1PANLB.D...atki - dbo.Wydatki			
	Column Name	Data Type	Allow Nulls
	Nr	int	<input type="checkbox"/>
	[Nr kategorii]	int	<input checked="" type="checkbox"/>
	[Nr osoby]	int	<input checked="" type="checkbox"/>
	[Nr sklepu]	int	<input checked="" type="checkbox"/>
	Data	datetime	<input checked="" type="checkbox"/>
	Kwota	money	<input checked="" type="checkbox"/>
	Opis	nvarchar(80)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


Zasady

Lista wszystkich zasad dla użytkowników, np. „Administracja”, „Modyfikacja”, itp.

[Nr] - numer identyfikacyjny, klucz główny - int

[Nazwa] – słowna nazwa zasady - varchar(30)

[Opis] – opis danej zasady - varchar(100)

DESKTOP-I1PANLB.D...atki - dbo.Zasady			
	Column Name	Data Type	Allow Nulls
	Nr	int	<input type="checkbox"/>
	Nazwa	varchar(30)	<input checked="" type="checkbox"/>
	Opis	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Widoki

view_select_all_Kategorie

Wyświetla wszystkie kategorie dotyczące wydatków.

[Nr] - numer identyfikacyjny, klucz główny

[Nazwa kategorii] – słowna nazwa

DESKTOP-I1PANLB.D...lect_all_Kategorie

Kategorie

☐ * (All Columns)
☒ **Nr**
☒ Nazwa kategorii

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
►	Nr		Kategorie	<input checked="" type="checkbox"/>						
	[Nazwa kateg...		Kategorie	<input checked="" type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						

SELECT Nr, [Nazwa kategorii]
FROM dbo.Kategorie

view_select_all_Logi

Wyświetla wszystkie logi – akcje użytkowników.

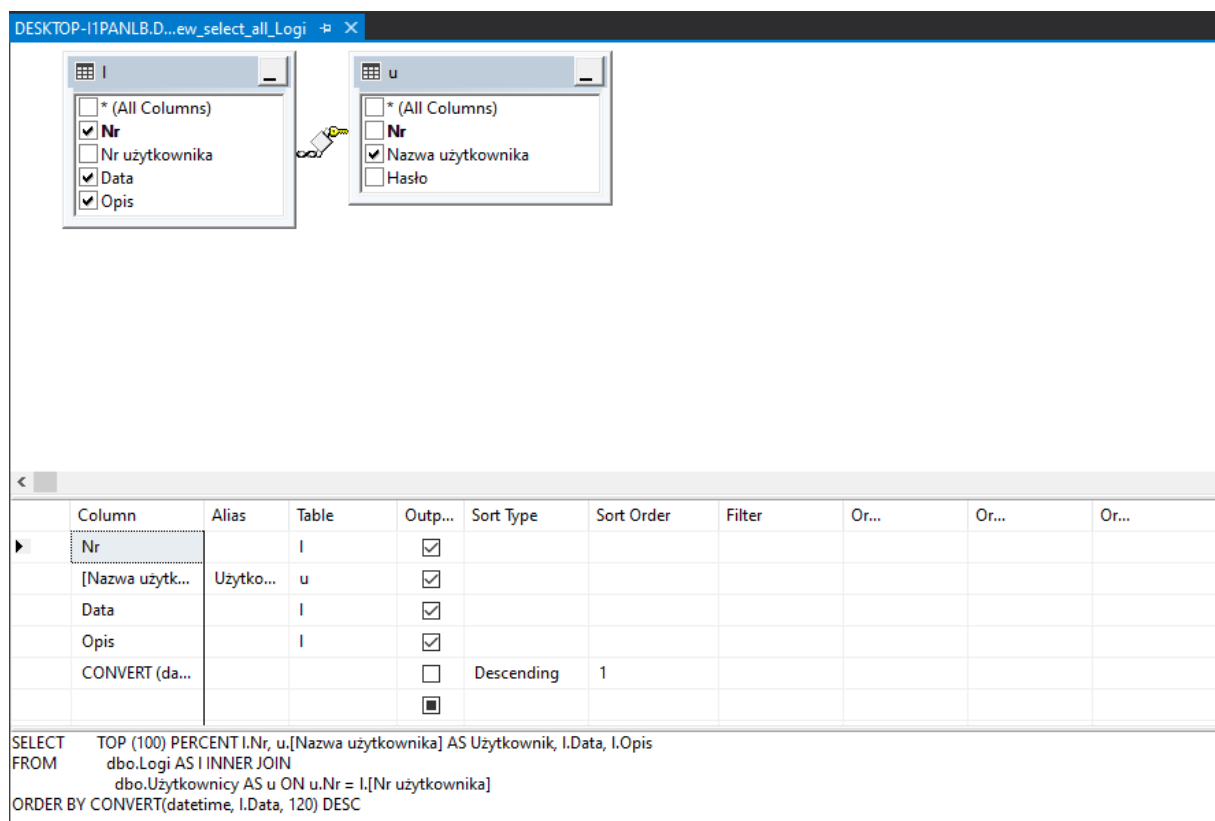
[Nr] - numer identyfikacyjny, klucz główny

[Nazwa użytkownika] – nazwa użytkownika wykonywującego daną akcję

[Data] – data akcji

[Opis] - opis wykonywanej akcji

DESKTOP-I1PANLB.D...ew_select_all_Logi



The screenshot displays the SQL Server Enterprise Designer interface. At the top, the title bar reads 'DESKTOP-I1PANLB.D...ew_select_all_Logi'. Below it, two table schemas are shown: 'l' and 'u'. Table 'l' has columns: 'Nr' (checked), 'Nr użytkownika' (unchecked), 'Data' (checked), and 'Opis' (checked). Table 'u' has columns: 'Nr' (unchecked), 'Nazwa użytkownika' (checked), and 'Hasło' (unchecked). A relationship line connects the 'Nr' column of table 'l' to the 'Nr' column of table 'u'. Below the schemas, a query window shows the following SQL query:

```
SELECT TOP (100) PERCENT I.Nr, u.[Nazwa użytkownika] AS Uzytkownik, I.Data, I.Opis
FROM      dbo.Logi AS I INNER JOIN
         dbo.Uzytkownicy AS u ON u.Nr = I.[Nr użytkownika]
ORDER BY CONVERT(datetime, I.Data, 120) DESC
```

view_select_all_Nousers

Zwraca wszystkie osoby, które nie mają przypisanego żadnego konta.

[Nr] - numer identyfikacyjny, klucz główny

[Imię] – słowne imię Osoby

[Nazwisko] – słownie nazwisko Osoby

DESKTOP-11PANLB.D...select_all_Nousers

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▶	Nr		o	<input checked="" type="checkbox"/>						
	Imię		o	<input checked="" type="checkbox"/>						
	Nazwisko		o	<input checked="" type="checkbox"/>						
	Nr		u	<input type="checkbox"/>			IS NULL			
				<input checked="" type="checkbox"/>						
				<input checked="" type="checkbox"/>						

```
SELECT o.Nr, o.Imię, o.Nazwisko
FROM   dbo.view_select_all_Osoby AS o LEFT OUTER JOIN
       dbo.view_select_all_Użytkownicy AS u ON o.Nr = u.Nr
WHERE  (u.Nr IS NULL)
```

view_select_all_Osoby

Zwraca wszystkie osoby.

[Nr] - numer identyfikacyjny, klucz główny

[Imię] – słowne imię

[Nazwisko] – słownie nazwisko

DESKTOP-I1PANLB.D..._select_all_Osoby

Osoby

- ☐ * (All Columns)
- ☒ Nr
- ☒ Imię
- ☒ Nazwisko

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▶	Nr		Osoby	<input checked="" type="checkbox"/>						
	Imię		Osoby	<input checked="" type="checkbox"/>						
	Nazwisko		Osoby	<input checked="" type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						

SELECT Nr, Imię, Nazwisko
FROM dbo.Osoby

view_select_all_Sklepy

Zwraca wszystkie sklepy.

[Nr] - numer identyfikacyjny, klucz główny

[Nazwa] – słowna nazwa sklepu, np. „U Romka”

[Adres] – adres sklepu, np. „ul. Słoneczna 12”

[Kod pocztowy] – kod pocztowy, zgodnie z maską: „nn-nnn”

[Miasto] – nazwa miasta, gdzie ulokowany jest dany sklep, np. „Kraków”

[Telefon] – nr telefonu do sklepu, zgodnie z maską: „(nn)nnnnnn”

DESKTOP-I1PANLB.D..._select_all_Sklepy

Sklepy

- ☐ * (All Columns)
- ☒ Nr
- ☒ Nazwa
- ☒ Adres
- ☒ Kod pocztowy

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
►	Nr		Sklepy	<input checked="" type="checkbox"/>						
	Nazwa		Sklepy	<input checked="" type="checkbox"/>						
	Adres		Sklepy	<input checked="" type="checkbox"/>						
	[Kod pocztow...		Sklepy	<input checked="" type="checkbox"/>						
	Miasto		Sklepy	<input checked="" type="checkbox"/>						
	Telefon		Sklepy	<input checked="" type="checkbox"/>						

SELECT Nr, Nazwa, Adres, [Kod pocztowy], Miasto, Telefon
FROM dbo.Sklepy

view_select_all_Uprawnienia

Zwraca listę uprawnień, które posiada każdy użytkownik.

[Nr użytkownika] - klucz główny

[Nr zasady] - klucz główny

DESKTOP-I1PANLB.D...t_all_Uprawnienia

up

☐ * (All Columns)

☒ Nr użytkownika

☒ Nr zasady

z

☐ * (All Columns)

☒ Nr

☒ Nazwa

☐ Opis

uz

☐ * (All Columns)

☒ Nr

☒ Nazwa użytkownika

☐ Hasło

SELECT

uz.Nr AS [Nr użytkownika], z.Nr AS [Nr zasady], uz.[Nazwa użytkownika], z.Nazwa AS [Nazwa zasady]

FROM

dbo.Uprawnienia AS up INNER JOIN

dbo.Użytkownicy AS uz ON uz.Nr = up.[Nr użytkownika] INNER JOIN

dbo.Zasady AS z ON z.Nr = up.[Nr zasady]

view_select_all_Uzytkownicy

Zwraca wszystkich użytkowników wraz z ich odpowiednikami w tabeli Osoby.

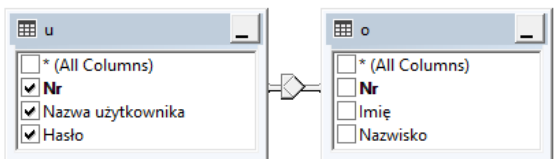
[Nr] - numer identyfikacyjny, klucz główny

[Nazwa użytkownika] – słowna nazwa użytkownika

[Osoba] – imię i nazwisko osoby. Połączone za pomocą "INNER JOIN".

[Hasło] – zaszyfrowane hasło użytkownika

DESKTOP-11PANLB.D...t_all_Uzytkownicy



	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▶	Nr		u	<input checked="" type="checkbox"/>						
	[Nazwa użyt...		u	<input checked="" type="checkbox"/>						
	{ fn CONCAT(...	Osoba		<input checked="" type="checkbox"/>						
	Hasło		u	<input checked="" type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						

```
SELECT u.Nr, u.[Nazwa użytkownika], { fn CONCAT(o.Imię, ' ', o.Nazwisko) } AS Osoba, u.Hasło
FROM dbo.Uzytkownicy AS u LEFT OUTER JOIN
      dbo.Osoby AS o ON o.Nr = u.Nr
```

view_select_all_Wydatki

Wyświetlanie wszystkich wydatków.

[Nr] - numer identyfikacyjny, klucz główny

[Nazwa kategorii] – słowna nazwa kategorii

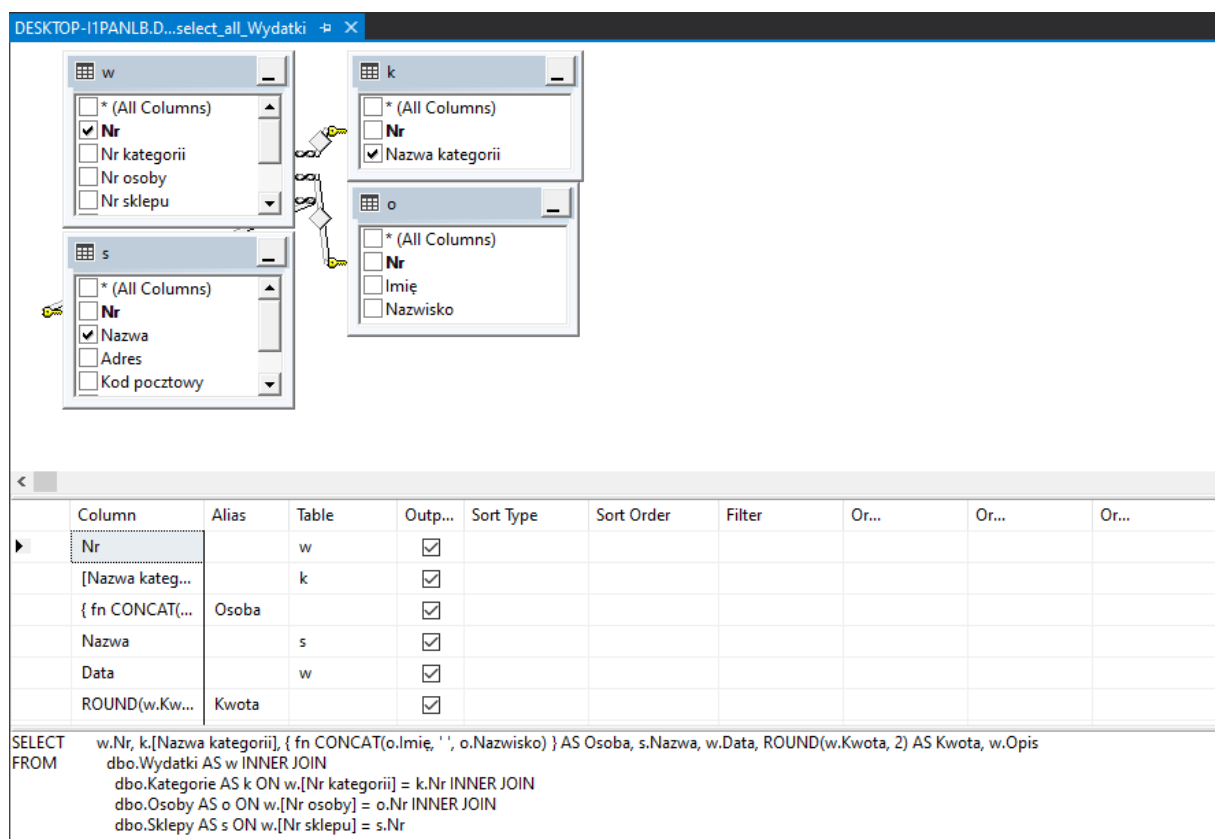
[Osoba] – imię i nazwisko osoby

[Nazwa] – nazwa sklepu

[Data] – data wydatku

[Kwota] – wydana ilość pieniędzy

[Opis] – opis wydatku, czego dotyczył



DESKTOP-11PANLB.D...select_all_Wydatki

Table Design Grid:

Table	Field	Selected
w	Nr	Yes
w	Nr kategorii	No
w	Nr osoby	No
w	Nr sklepu	No
k	Nr	No
k	Nazwa kategorii	Yes
o	Nr	No
o	Imię	No
o	Nazwisko	No
s	Nr	No
s	Nazwa	Yes
s	Adres	No
s	Kod pocztowy	No

Query Design Grid:

Column	Alias	Table	Output To	Sort Type	Sort Order	Filter	Or...	Or...	Or...
Nr		w	Yes						
[Nazwa kateg...		k	Yes						
{ fn CONCAT(...	Osoba		Yes						
Nazwa		s	Yes						
Data		w	Yes						
ROUND(w.Kw...	Kwota		Yes						

SQL View:

```
SELECT w.Nr, k.[Nazwa kategorii], { fn CONCAT(o.Imię, ' ', o.Nazwisko) } AS Osoba, s.Nazwa, w.Data, ROUND(w.Kwota, 2) AS Kwota, w.Opis
FROM
  dbo.Wydatki AS w INNER JOIN
    dbo.Kategorie AS k ON w.[Nr kategorii] = k.Nr INNER JOIN
      dbo.Osoby AS o ON w.[Nr osoby] = o.Nr INNER JOIN
        dbo.Sklepy AS s ON w.[Nr sklepu] = s.Nr
```

view_select_all_Zasady

Wyświetla wszystkie zasady dostępne dla użytkowników.

[Nr] - numer identyfikacyjny, klucz główny

[Nazwa] – nazwa zasady

[Opis] – opis zasady

DESKTOP-I1PANLB.D..._select_all_Zasady

Zasady

☐ * (All Columns)

☒ **Nr**

☒ Nazwa

☒ Opis

	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▶	Nr		Zasady	<input checked="" type="checkbox"/>						
	Nazwa		Zasady	<input checked="" type="checkbox"/>						
	Opis		Zasady	<input checked="" type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						
				<input type="checkbox"/>						

SELECT Nr, Nazwa, Opis
FROM dbo.Zasady

Procedury

proc_add_to_Kategorie

Dodaje nowy rekord do tabeli „Kategorie”.

@nazwa_kategorii – nvarchar(30)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:    <2021-01-13>
-- Description:    <Adding new category>
-- =====
CREATE PROCEDURE proc_add_to_Kategorie
    @nazwa_kategorii nvarchar(30) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [dbo].[Kategorie] ([Nazwa kategorii]) VALUES (@nazwa_kategorii);
END
GO
```

proc_add_to_Logi

Dodaje nowy rekord do tabeli „Logi”.

@nr_użytkownika - int

@data - smalldatetime

@opis – nvarchar(100)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:     <2021-01-19>
-- Description:     <Adding new log>
-- =====
CREATE PROCEDURE proc_add_to_Logi
    @nr_użytkownika int = NULL,
    @data            smalldatetime = NULL,
    @opis            nvarchar(100) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT
        INTO [dbo].[Logi] ([Nr użytkownika], [Data], [Opis])
        VALUES (@nr_użytkownika, @data, @opis);
END
GO
```

proc_add_to_Osoby

Dodaje nowy rekord do tabeli „Osoby”

@imie nvarchar(30)

@nazwisko nvarchar(30)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:    <2020-12-29>
-- Description:    <Adding new person>
-- =====
CREATE PROCEDURE proc_add_to_Osoby
    @imie      nvarchar(30) = NULL,
    @nazwisko  nvarchar(30) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [dbo].[Osoby] (Imię, Nazwisko) VALUES (@imie, @nazwisko);
END
GO
```

proc_add_to_Sklepy

Dodaje nowy rekord do tabeli „Sklepy”

@nazwa nvarchar(50)

@adres nvarchar(50)

@kod_pocztowy nvarchar(6)

@miasto nvarchar(50)

@telefon nvarchar(10)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:     <2021-01-13>
-- Description:     <Adding new shop>
-- =====
CREATE PROCEDURE proc_add_to_Sklepy
    @nazwa          nvarchar(50) = NULL,
    @adres          nvarchar(50) = NULL,
    @kod_pocztowy   nvarchar(6)   = NULL,
    @miasto         nvarchar(50) = NULL,
    @telefon        nvarchar(10) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT
        INTO [dbo].[Sklepy] (Nazwa, Adres, [Kod pocztowy], Miasto, Telefon)
        VALUES (@nazwa, @adres, @kod_pocztowy, @miasto, @telefon);
END
GO
```

proc_add_to_Uprawnienia

Dodaje nowy rekord do tabeli „Uprawnienia”

@id_u int

@id_z int

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:    <2021-01-23>
-- Description:    <Adding new rule for user>
-- =====
CREATE PROCEDURE proc_add_to_Uprawnienia
    @id_u int          = NULL,
    @id_z int          = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT
        INTO [dbo].[Uprawnienia] ([Nr użytkownika], [Nr zasady])
        VALUES (@id_u, @id_z);
END
GO
```

proc_add_to_Użytkownicy

Dodaje nowy rekord do tabeli „Użytkownicy”

@nazwa nvarchar(30)

@haslo nvarchar(32)

@nr_osoby int

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-16
-- Description:  Adding new user
-- =====
CREATE PROCEDURE proc_add_to_Użytkownicy
    @nazwa          nvarchar(30) = NULL,
    @haslo          nvarchar(32) = NULL,
    @nr_osoby       int          = NULL
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [Użytkownicy] ([Nazwa użytkownika], [Hasło]) VALUES (@nazwa, @haslo)

    DECLARE @id AS INT;
    SET @id = (SELECT MAX(Nr) FROM view_select_all_Użytkownicy)

    INSERT INTO [Powiązania] ([Nr Osoby], [Nr Użytkownika]) VALUES (@id, @id);
END
```

Dodatkowo, procedura pobiera numer identyfikacyjny utworzonego użytkownika. Następnie, dodaje powiązanie z osobą, ponieważ numer osoby jest równy numeru użytkownika. Kolejno, procedura dodaje zasadę „Wyświetlania danych” dla nowo utworzonego użytkownika.

proc_add_to_Wydatki

Dodaje nowy rekord do tabeli „Wydatki”

@nr_kategorii INT

@nr_osoby INT

@nr_sklepu INT

@data datetime

@kwota varchar(20)

@opis nvarchar(80)

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
-- =====  
-- Author:          <Marek Kurańda>  
-- Create date:    <2021-01-25>  
-- Description:    <Add Wydatki>  
-- =====
```

```
CREATE PROCEDURE proc_add_to_Wydatki
```

```
    @nr_kategorii INT,
```

```
    @nr_osoby INT,
```

```
    @nr_sklepu INT,
```

```
    @data datetime,
```

```
    @kwota varchar(20),
```

```
    @opis nvarchar(80)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO [dbo].[Wydatki]
```

```
        ([Nr kategorii], [Nr osoby], [Nr sklepu], [Data], [Kwota], [Opis])
```

```
    VALUES
```

```
        (@nr_kategorii, @nr_osoby, @nr_sklepu, @data, CAST(@kwota AS
```

```
decimal(18, 2)), @opis);
```

```
END
```

```
GO
```

proc_delete_row

Usuwa jeden rekord z podanej tabelki.

@table nvarchar(50)

@id nvarchar(30)

@id2 nvarchar(30)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:     <2021-01-13>
-- Edited date:     <2021-01-23>
-- Description:     <Deleting one row>
-- =====
CREATE PROCEDURE [dbo].[proc_delete_row]
    @table nvarchar(50) = NULL,
    @id     nvarchar(30) = NULL,
    @id2    nvarchar(30) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @SqlStatment AS NVARCHAR(1000);

    IF (LEN(@id2) <> 0)
        SET @SqlStatment = 'DELETE FROM [dbo].[Uprawnienia] WHERE [Nr
użytkownika] = ' + @id + ' AND [Nr zasady] = ' + @id2;
    ELSE
        SET @SqlStatment = 'DELETE FROM [dbo].[' + @table + '] WHERE [Nr] = ' +
@id;
    EXECUTE sp_executesql @SqlStatment;
END
```

Jest to jedyna procedura na usuwanie rekordu. Otóż, jej magia tkwi w ciele procedury. Tam też, tworzona jest zmienna **SqlStatement**, które zawiera gotowe zapytanie w zależności w jakiej tabelce chcemy usunąć rekord.

Również znajduje się tam instrukcja warunkowa. W zależności od tego, czy parametr @id2 jest pusty, czy nie, instrukcja wybiera konstrukcję zapytania. Musiałem rozdzielić to na dwa przypadki, ponieważ w tabeli **Uprawnienia** nie występuje pole [Nr]. Dlatego też, trzeba było wydzielić osobny warunek.

Problem pojawił się przy tworzeniu instrukcji warunkowej. Problem był z warunkiem, kiedy porównywałem zmienną do **null**. Jednak, do dziś nie wiem dlaczego były z tym problemy. Zatem, zmieniłem warunek w sposób taki, że liczę długość napisu i jeśli równy jest długości **null**, wtedy wybiera pierwszą opcję.

proc_get_table_count

Zwraca ilość rekordów dla danej tabelki.

@table NVARCHAR(50)

```
CREATE PROCEDURE proc_get_table_count
@table NVARCHAR(50)
AS
BEGIN
    DECLARE @c AS INT;
    DECLARE @SqlStatment AS NVARCHAR(1000)

    SET @SqlStatment = 'SELECT COUNT(*) FROM ' + @table;

    EXECUTE sp_executesql @SqlStatment
END
```

Działa dla każdej tabelki.

proc_paging

Procedura, która paginuje tabelkę.

@table VARCHAR(50)

@offset VARCHAR(10)

@count VARCHAR(10)

```
CREATE PROCEDURE proc_paging
@table VARCHAR(50),
@offset VARCHAR(10),
@count VARCHAR(10)
AS
BEGIN
    DECLARE @query NVARCHAR(MAX);
    IF (@table = 'view_select_all_Logi')
        SET @query = 'SELECT * FROM [' + @table + '] ORDER BY CONVERT(datetime,
[Data], 120) DESC OFFSET ' + @offset + ' ROWS FETCH NEXT ' + @count + ' ROWS ONLY;';
    ELSE IF (@table != 'view_select_all_Uprawnienia')
        SET @query = 'SELECT * FROM [' + @table + '] ORDER BY [Nr] ASC OFFSET ' +
@offset + ' ROWS FETCH NEXT ' + @count + ' ROWS ONLY;';
    ELSE
        SET @query = 'SELECT * FROM [' + @table + '] ORDER BY [Nazwa użytkownika]
ASC OFFSET ' + @offset + ' ROWS FETCH NEXT ' + @count + ' ROWS ONLY;';
    EXEC (@query);
END
```

Procedura zawiera instrukcję warunkową ze względu na tabelki „Logi” i „Uprawnienia”, ponieważ w tabelce „Logi” występuje sortowanie malejące, a w tabeli „Uprawnienia” brak pola [Nr], za pomocą którego paginuję w pozostałych tabelkach. Stąd właśnie pomysł i idea na rozdzielenie na trzy przypadki. Ta metoda usuwa problem z nierozpoznanym polem dla tabeli Uprawnienia oraz pozwala sortować na różne sposoby.

Procedura również stosuje sortowanie wraz ze słowem kluczowym „offset”, które przesuwa rekordy wynikowe. Dlatego też, możemy wyświetlić, np. 10 rekordów z przesunięciem np. 30 rekordów.

proc_select_datetime

Zwraca datę i czas serwera.

Data i czas są konwertowane na system: **nnnn-nn-nnTnn:nn:nn**

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Marek Kurańda
-- Create date: 2021-01-14
-- Description:      Returns server datetime
-- =====
CREATE PROCEDURE proc_select_datetime
AS
BEGIN
    SET NOCOUNT ON;

    SELECT    CONCAT(
                CONVERT(DATE, GETDATE()),
                'T',
                CONVERT(TIME, GETDATE())
            );
END
GO
```

proc_select_record

Wybiera rekord o określonym id z określonej tabeli.

@table NVARCHAR(50)

@id NVARCHAR(10)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-22
-- Description:  Select one record
-- =====
CREATE PROCEDURE proc_select_record
    @table NVARCHAR(50),
    @id NVARCHAR(10)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @SqlStatment AS NVARCHAR(1000)

    SET @SqlStatment = 'SELECT * FROM view_select_all_' + @table + ' WHERE [Nr] = '
+ @id;

    EXECUTE sp_executesql @SqlStatment;
END
GO
```

Jest to uniwersalna procedura, ponieważ wystarczy podać nazwę tabeli o numer identyfikacyjny rekordu z określonej wcześniej tabeli, a procedura zwróci cały rekord!

proc_select_time

Procedura, która korzysta z funkcji **SYSDATETIME()** i właśnie wynik funkcji zwraca w selekcje.

```
CREATE PROCEDURE proc_select_time  
AS  
    SELECT SYSDATETIME();
```

proc_update_to_Kategorie

Procedura aktualizuje jeden rekord w tabeli Kategorie.

@id INT

@nazwa_kategorii varchar(50)

```
USE [Domowe_Wydatki]
GO
/***** Object:  StoredProcedure [dbo].[proc_update_row]      Script Date: 2021-01-23
20:24:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:    <2021-01-26>
-- Description:     <Updating Kategorie>
-- =====
CREATE PROCEDURE [dbo].[proc_update_to_Kategorie]
    @id INT,
    @nazwa_kategorii varchar(50) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE [dbo].[Kategorie] SET [Nazwa kategorii] = @nazwa_kategorii WHERE [Nr] =
    @id;
END
```

proc_update_to_Osoby

Procedura, która aktualizuje jeden rekord w tabeli Osoby.

@id INT

@imie varchar(30)

@nazwisko varchar(30)

```
USE [Domowe_Wydatki]
GO
/***** Object:  StoredProcedure [dbo].[proc_update_row]    Script Date: 2021-01-23
20:24:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:    <2021-01-23>
-- Description:    <Updating Osoby>
-- =====
ALTER PROCEDURE [dbo].[proc_update_to_Osoby]
    @id INT,
    @imie varchar(30) = NULL,
    @nazwisko varchar(30) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE [dbo].[Osoby] SET [Imię] = @imie, [Nazwisko] = @nazwisko WHERE [Nr] =
    @id;
END
```

proc_update_to_Sklepy

Aktualizuje jeden rekord w tabeli Sklepy

[Nr] - numer identyfikacyjny, klucz główny - int

[Nazwa] – słowna nazwa sklepu, np. „U Romka” - nvarchar(50)

[Adres] – adres sklepu, np. „ul. Słoneczna 12” - nvarchar(50)

[Kod pocztowy] – kod pocztowy, zgodnie z maską: „nn-nnn” - nvarchar(6)

[Miasto] – nazwa miasta, gdzie ulokowany jest dany sklep, np. „Kraków” - nvarchar(50)

[Telefon] – nr telefonu do sklepu, zgodnie z maską: „(nn)nnnnnn” - nvarchar(10)

```
USE [Domowe_Wydatki]
GO
/***** Object:  StoredProcedure [dbo].[proc_update_row]      Script Date: 2021-01-23
20:24:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Marek Kurańda>
-- Create date:     <2021-01-26>
-- Description:     <Updating Sklepy>
-- =====
CREATE PROCEDURE [dbo].[proc_update_to_Sklepy]
    @id INT,
    @nazwa varchar(50) = NULL,
    @adres varchar(50) = NULL,
    @kod_pocztowy nvarchar(6) = NULL,
    @miasto nvarchar(50) = NULL,
    @telefon nvarchar(10) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE [dbo].[Sklepy]
    SET [Nazwa] = @nazwa,
        [Adres] = @adres,
        [Kod pocztowy] = @kod_pocztowy,
        [Miasto] = @miasto,
        [Telefon] = @telefon
    WHERE [Nr] = @id;
END
```


Funkcje tabelaryczne

func_get_all_Kategorie

Funkcja, która zwraca wszystkie rekordy z tabeli Kategorie.

```
CREATE FUNCTION func_get_all_Kategorie ()  
RETURNS TABLE  
AS  
    RETURN SELECT [Nr], [Nazwa Kategorii] FROM view_select_all_Kategorie;
```

func_get_all_Nousers

Funkcja, która zwraca wszystkie osoby, nieposiadające swojego konta. Zwraca numer, a także imię i nazwisko jako jedno pole.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-23
-- Description:  Returns all nousers
-- =====
CREATE FUNCTION func_get_all_Nousers ()
RETURNS TABLE
AS
    RETURN
    SELECT [Nr],
           CONCAT([Imię], ' ', [Nazwisko]) AS [Osoba]
    FROM   [view_select_all_Nousers];
```

func_get_all_Osoby

Funkcja, która zwraca wszystkie rekordy z tabeli Osoby.

```
CREATE FUNCTION func_get_all_Osoby ()  
RETURNS TABLE  
AS  
    RETURN SELECT [Nr], CONCAT([Imię], ' ', [Nazwisko]) AS [Osoba] FROM  
view_select_all_Osoby;
```

func_get_all_Sklepy

Funkcja, która zwraca wszystkie rekordy z tabeli Sklepy.

```
CREATE FUNCTION func_get_all_Sklepy ()  
RETURNS TABLE  
AS  
    RETURN SELECT [Nr], [Nazwa] FROM view_select_all_Sklepy;
```

func_get_all_Użytkownicy

Funkcja, która zwraca wszystkie rekordy z tabeli Użytkownicy.

```
CREATE FUNCTION func_get_all_Użytkownicy ()  
RETURNS TABLE  
AS  
    RETURN SELECT * FROM view_select_all_Użytkownicy;
```

func_get_all_Zasady

Funkcja, która zwraca wszystkie rekordy z tabeli Zasady.

```
CREATE FUNCTION func_get_all_Zasady ()  
RETURNS TABLE  
AS  
    RETURN SELECT * FROM view_select_all_Zasady;
```

func_get_all_Zasady_Niepasujące

Funkcja, która wszystkie niepasujące zasady dla danego użytkownika (@id INT).

Problemem było znaleźć wynik, który pragnąłem uzyskać. Rozwiązanie, które przyszło mi do głowy, opiera się na pomyśle:

1. Iloczyn kartezjański dla tabeli „Użytkowników” oraz dla tabeli „Zasady”.
2. Z iloczynu wybieram tylko kombinacje dla użytkownika o podanym „id”.
3. Następnie za pomocą słów kluczowych „NOT IN” odrzucane są wszystkie rekordy, które posiada ten użytkownik.

Właśnie za pomocą takiego zabiegu udało mi się uzyskać oczekiwany efekt.

```
CREATE FUNCTION func_get_all_Zasady_Niepasujące
(
    @id INT
)
RETURNS TABLE
AS
    RETURN
        SELECT z.[Nr],
               z.[Nazwa],
               z.[Opis]
        FROM   view_select_all_Zasady AS z,
               view_select_all_Użytkownicy AS us
        WHERE  us.[Nr] = @id
              AND
               z.[Nr] NOT IN (SELECT [Nr zasady] FROM
view_select_all_Uprawnienia WHERE [Nr użytkownika] = @id)
```

func_get_Użytkownicy_Nazwa

Funkcja, która zwraca nazwę użytkownika, a jako parametr przyjmuje jego numer identyfikacyjny.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-16
-- Description:  Returns user name
-- =====
CREATE FUNCTION func_get_Użytkownicy_Nazwa
(
    @id VARCHAR(30)
)
RETURNS TABLE
AS
    RETURN SELECT [Nazwa użytkownika] FROM [dbo].[view_select_all_Użytkownicy] WHERE
[Nr] = @id;
```


func_get_Użytkownicy_Nr

Funkcja, która zwraca numer identyfikacyjny użytkownika, a jako parametr przyjmuje jego nazwę.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Marek Kurańda
-- Create date: 2021-01-16
-- Description:      Returns user id
-- =====
CREATE FUNCTION func_get_Użytkownicy_Nr
(
    @name VARCHAR(30)
)
RETURNS TABLE
AS
    RETURN SELECT [Nr] FROM [dbo].[view_select_all_Użytkownicy] WHERE [Nazwa
użytkownika] = @name;
```

Funkcje skalarne

func_check_rule

Zwraca bit (1 -prawda, 0 - fałsz) dla zdania logicznego, czy istnieje określona zasada.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Marek Kurańda
-- Create date: 2021-01-20
-- Description:      Returns bit if a rule exists
-- =====
CREATE FUNCTION func_check_rule
(
    @nr          INT = -1,
    @nazwa VARCHAR(30) = NULL
)
RETURNS VARCHAR(30)
AS
BEGIN
    DECLARE @r BIT;
    SET @r = 0;

    -- Jeśli ustawiono numer
    IF (@nr != -1 AND @nr != 0)
        SET @r = (SELECT COUNT(*) FROM [Zasady] WHERE [Nr] = @nr);
    -- Jeśli ustawiono nazwę
    ELSE
        BEGIN
            IF (LEN(@nazwa) > 4)
                SET @r = (SELECT COUNT(*) FROM [Zasady] WHERE [Nazwa] =
@nazwa);
            END;

    -- Zwróć rezultat
    RETURN @r;
END
```

func_check_user_rule

Zwraca bit (1 -prawda, 0 - fałsz) dla zdania logicznego, czy określony użytkownik posiada pewne uprawnienia.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-20
-- Description: Returns bit if a user has a rule
-- =====
CREATE FUNCTION func_check_user_rule
(
    @nr_user      INT,
    @nr_rule      INT
)
RETURNS BIT
AS
BEGIN
    DECLARE @r BIT;
    SET @r = (SELECT COUNT(*) FROM [Uprawnienia] WHERE [Nr zasady] = @nr_rule AND
[Nr użytkownika] = @nr_user);

    -- Zwróć rezultat
    RETURN @r;
END
```

func_get_osoby_count

Przyjmuje jeden parametr – ilość rekordów na stronę. Wylicza zatem, ile stron zawiera tabela „Osoby”.

```
CREATE FUNCTION func_get_osoby_count (@count INT)
RETURNS INT
BEGIN
    DECLARE @c AS INT;
    SET @c = (SELECT COUNT(*) FROM [Osoby]);
    RETURN (@c / @count) + 1;
END
```

func_get_table_count

Oblicza na ile stron można podzielić tabelkę.

```
USE [Domowe_Wydatki]
GO
/***** Object: UserDefinedFunction [dbo].[func_get_table_count]    Script Date:
2021-01-28 18:14:28 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[func_get_table_count] (@table VARCHAR(50), @count INT)
RETURNS INT
BEGIN
    DECLARE @c AS INT;
    DECLARE @q NVARCHAR(MAX);
    SET @q = 'SELECT COUNT(*) FROM [' + @table + ']';
    exec sp_executeSQL @q, @c;
    RETURN (@c / @count) + 1;
END
```

func_get_Zasady_Nr

Zwraca numer identyfikacyjny zasady

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Marek Kurańda
-- Create date: 2021-01-20
-- Description:      Returns id of rule
-- =====
CREATE FUNCTION func_get_Zasady_Nr
(
    @nazwa VARCHAR(30)
)
RETURNS INT
AS
BEGIN
    DECLARE @r INT;
    SET @r = (SELECT [Nr] FROM [Zasady] WHERE [Nazwa] = @nazwa);

    -- Zwróć rezultat
    RETURN @r;
END
```

Triggery

trig_delete_rule

Blokuje usuwanie rekordów dla tabeli „Zasady”.

```
CREATE TRIGGER trig_delete_rule
    ON [dbo].[Zasady]
    FOR DELETE
AS
    PRINT 'Nie możesz usuwać danych z tej tabeli!';
ROLLBACK;
```

trig_insert_rule

Blokuje wstawianie nowych zasad.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-29
-- Description:  Forbids to insert records in "Zasady"
-- =====
CREATE TRIGGER [dbo].[trig_insert_rule]
    ON [dbo].[Zasady]
    AFTER INSERT
AS
    PRINT 'Nie możesz dodawać nowych zasad!';
GO
```


trig_update_rule

Blokuje modyfikację zasad w tabeli „Zasady”.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Marek Kurańda
-- Create date: 2021-01-29
-- Description:  Forbids to change record in "Zasady"
-- =====
CREATE TRIGGER [dbo].[trig_update_rule]
    ON [dbo].[Zasady]
    AFTER DELETE
AS
BEGIN
    PRINT 'Nie możesz zmieniać danych w tej tabelce!';
    ROLLBACK;
END
GO
```

WSTĘP	2
PARĘ SŁÓW.....	3
SPOSÓB DZIAŁANIA	4
Podstrona: /settings	5
Podstrona: /main.....	6
Logowanie	9
OPIS OD STRONY PROGRAMISTY	16
Struktura katalogów	17
Budowa pliku index.php:	19
Plik manager.php.....	21
Budowa pliku header.php	22
Plik footer.php	25
Plik error404.php	26
Plik form.php	27
Plik report.php.....	29
Plik insert_new_record.php.....	34
Plik update_record.php	36
Plik delete_record.php	38
Plik manager.js	39
Plik login.js.....	43
Plik form.js.....	45
Plik update.js	50
Plik settings.js	54
FOLDER DATA	59
actions.js.....	60
colors.json	62
fields.json	64

form.json	66
settings.json.....	68
PLIKI COOKIE.....	69
SCHEMAT BAZY DANYCH	70
TABELKI	71
Kategorie	71
Logi	72
Osoby.....	73
Powiązania.....	74
Sklepy	75
Uprawnienia	76
Użytkownicy	77
Wydatki	78
Zasady.....	79
WIDOKI.....	80
view_select_all_Kategorie	80
view_select_all_Logi	81
view_select_all_Nousers	82
view_select_all_Osoby	83
view_select_all_Sklepy	84
view_select_all_Uprawnienia	85
view_select_all_Użytkownicy	86
view_select_all_Wydatki	87
view_select_all_Zasady	88
PROCEDURY.....	89
proc_add_to_Kategorie	89
proc_add_to_Logi	90

proc_add_to_Osoby	91
proc_add_to_Sklepy	92
proc_add_to_Uprawnienia	93
proc_add_to_Uzytkowicy	94
proc_add_to_Wydatki	95
proc_delete_row	96
proc_get_table_count	97
proc_paging	98
proc_select_datetime	99
proc_select_record	100
proc_select_time	101
proc_update_to_Kategorie.....	102
proc_update_to_Osoby.....	103
proc_update_to_Sklepy.....	104
 FUNKCJE TABELARYCZNE	 105
func_get_all_Kategorie.....	105
func_get_all_Nousers	106
func_get_all_Osoby	107
func_get_all_Sklepy.....	108
func_get_all_Uzytkownicy.....	109
func_get_all_Zasady	110
func_get_all_Zasady_Niepasujace.....	111
func_get_Uzytkownicy_Nazwa.....	112
func_get_Uzytkownicy_Nr.....	113
 FUNKCJE SKALARNE.....	 114
func_check_rule.....	114
func_check_user_rule.....	115
func_get_osoby_count	116

func_get_table_count.....	117
func_get_Zasady_Nr.....	118
TRIGGER119	
trig_delete_rule.....	119
trig_insert_rule.....	120
trig_update_rule.....	121