

Obraz 1. Tabela zadania

idZadania	MI: INTEGER
dataZadania	DATE
wpis	TEXT
miesiac	TEXT
rok	INTEGER

Obraz 1. Tabela zadania

Za pomocą narzędzia phpMyAdmin wykonaj podane operacje na bazie danych:

- Utwórz bazę danych o nazwie *kalendarz*
- Do bazy *kalendarz* zaimportuj tabelę z pliku *baza5.sql* z rozpakowanego archiwum *import*. Nie kadruj zrzutów.
- Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w formacie PNG pod nazwą *zrzut_import_tabel*. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel
- Wykonaj zapytania SQL działające na bazie *kalendarz*. Zapytania zapisz w formacie JPEG i nadaj im nazwy *kw1*, *kw2*, *kw3*, *kw4*. Zrzuty powinny obejmować cały ekran monitora, z widocznym paskiem zadań
- Zapytanie 1: wybierające jedynie pola *dataZadania* i *wpis* dla zadań, z miesiąca sierpnia
- Zapytanie 2: wybierające jedynie pola *miesiac* i *rok* dla zadania z dnia 7 sierpnia 2020 r.
- Zapytanie 3: wybierające jedynie nazwę *miesiac* oraz licząc dla każdego miesiąca ile jest wpisów rozpoczynających się od słowa „Weterynarz” w kolumnie *wpis*
- Zapytanie 4: aktualizujące pole *wpis* dla zadania z datą 9 sierpnia 2020, nowy wpis to „Wycieczka: Zalew Mietkowski”

Grafika:

- Plik *logo2.png*, wypakowany z archiwum, 120 px

Cechy witryny:

- Składa się ze stron
- Zapisana w języku PHP
- Jawnie zastosowane
- Zadeklarowane
- Tytuł strony w
- Arkusz stylów
- Podział strony
- skryptem b
- aby po un
- Zawartość
- Zawartość
- zawier
- P
-
- Za
- Z
- 7

Organizer: SIERPIEŃ

Zadaj wydarzenie:

2020-08-01	2020-08-02	2020-08-03	2020-08-04	2020-08-05	2020-08-06
2020-08-07	2020-08-08	2020-08-09	2020-08-10	2020-08-11	2020-08-12
2020-08-13	2020-08-14	2020-08-15	2020-08-16	2020-08-17	2020-08-18
2020-08-19	2020-08-20	2020-08-21	2020-08-22	2020-08-23	2020-08-24
2020-08-25	2020-08-26	2020-08-27	2020-08-28	2020-08-29	2020-08-30
2020-08-31					

Stronę wykonał: 0000000000

Obraz 2. Witryna internetowa, wysłano formularz z wartością „zakupy”, kursor znajduje się na zmodyfikowanym bloku

Grafika:

- Plik *logo2.png*, wypakowany z archiwum, należy przeskalować dokładnie do wymiarów 120 px na 120 px

Cechy witryny:

- Składa się ze strony o nazwie *organizer.php*
- Zapisana w języku HTML 5
- Jawnie zastosowany właściwy standard kodowania polskich znaków
- Zadeklarowany język witryny: polski
- Tytuł strony widoczny na karcie przeglądarki: „Sierpniowy kalendarz”
- Arkusz stylów w pliku o nazwie *styl5.css* prawidłowo połączony z kodem strony
- Podział strony na bloki: trzy bloki banera, poniżej blok główny, w którym znajdują się generowane skryptem bloki kalendarza, na dole blok stopki. Podział zrealizowany za pomocą znaczników sekcji tak, aby po uruchomieniu w przeglądarce wygląd układu bloków był zgodny z Obrazem 2
- Zawartość pierwszego bloku banera: nagłówek pierwszego stopnia o treści „Organizer: SIERPIEŃ” zawierający:
 - Pole edycyjne podpisane: „Zapisz wydarzenie: ”
 - Przycisk o treści „OK” wysyłający zawartość formularza
- Zawartość trzeciego bloku banera: *logo2.png* z tekstem alternatywnym o treści: „sierpień”
- Zawartość bloku głównego: efekt działania skryptu 1
- Zawartość stopki: Paragraf (akapit) o treści: „Stronę wykonał: ”, dalej wstawiony numer zdającego

Styl CSS witryny internetowej

- Styl CSS zdefiniowany w całości w zewnętrznym pliku o nazwie `style5.css`. Arkusz CSS zawiera formatowanie:
- Ustawione domyślne wartości dla wszystkich selektorów stylu CSS: krój czcionki `Tahoma`
 - Dla wszystkich bloków banera: kolor tła `#7DA453`, biały kolor czcionki, wysokość 120 px
 - Dla wszystkich pierwszych i drugiego bloku banera: szerokość 200 px, wysokość 100 px, marginesy zewnętrzne 7 px, obramowanie o szerokości 1 px, linią ciągłą o kolorze `#7DA453`
 - Dla bloku kalendarza: kolor tła `#AED581`, szerokość o kolorze `#7DA453`, czcionki na biały
 - Dla kursora znajduje się na bloku kalendarza, kolor tła bloku zmienia się na `#7DA453`, czcionki na biały
 - Dla stopki: kolor tła `#7DA453`, biały kolor czcionki, marginesy wewnętrzne 5 px
 - Dla selektora nagłówka piątego stopnia: wyrównanie tekstu do środka
 - Dla selektora formularza: margines wewnętrzny górny: 50 px

Uwaga: style CSS dla nagłówka piątego stopnia i formularza należy zdefiniować wyłącznie przy pomocy selektora dla znaczników nagłówka piątego stopnia i formularza. Jest to uwarunkowane projektem późniejszej rozbudowy witryny.

Skrypt połączenia z bazą

W tabeli 1 podano wybór funkcji PHP do obsługi bazy danych. Wymagania dotyczące skryptu:

- Napisany w języku PHP, w pliku `organizer.php`
- Skrypt łączy się z serwerem bazodanowym na `localhost`, użytkownik `root` bez hasła, baza danych o nazwie `kalendarz`
- Działanie skryptu obsługującego formularz:
 - Skrypt pobiera dane z pola edycyjnego
 - Następnie za pomocą zmodyfikowanego zapytania 4 uaktualnia dane w bazie danych tak, że dane w polu wpis zostały pobrane z formularza. Aby uprościć zadanie skrypt wprowadza wpis zawsze do rekordu o dacie 2020-08-09.
- Działanie skryptu 1:
 - Wysyła do bazy danych zapytanie 1
 - Za pomocą znacznika sekcji definiuje blok, a w nim wyświetla dane przesłane skryptem dotyczące jednego wiersza. Blok jest zgodny z Obrazem 3 oraz jest w nim wyświetlone:
 - Data zadania w nagłówku piątego stopnia
 - Pole wpis w paragrafie (akapicie)
 - Liczba wygenerowanych bloków odpowiada liczbie wierszy zwróconych zapytaniem
- Na końcu działania skrypt zamyka połączenie z serwerem.

2020-08-11

Weterynarz - Dika

Tabela 1. Wybór funkcji języka PHP do obsługi bazy MySQL i MariaDB

Funkcje biblioteki MySQLi	Zwracana wartość
<code>mysqli_connect(serwer, użytkownik, hasło, nazwa_bazy)</code>	id połączenia lub FALSE, gdy niepowodzenie
<code>mysqli_select_db(id_połączenia, nazwa_bazy)</code>	TRUE/FALSE w zależności od stanu operacji
<code>mysqli_error(id_połączenia)</code>	Tekst komunikatu błędu
<code>mysqli_close(id_połączenia)</code>	TRUE/FALSE w zależności od stanu operacji
<code>mysqli_query(id_połączenia, zapytanie)</code>	Wynik zapytania
<code>mysqli_fetch_row(wynik_zapytania)</code>	Tablica numeryczna odpowiadająca wierszowi zapytania
<code>mysqli_fetch_array(wynik_zapytania)</code>	Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania
<code>mysqli_num_rows(wynik_zapytania)</code>	Liczba wierszy w podanym zapytaniu
<code>mysqli_num_fields(wynik_zapytania)</code>	Liczba kolumn w podanym zapytaniu