

# ARKUSZ EE.09 : 2020 - CZERWIEC - ZAD. 01 (PHP)

📄 OZNACZENIE: EE.09-01-20.06-SG ⌚ LIMIT CZASU: 150 MINUT

## Zadanie egaminacyjne

Wykonaj aplikację internetową organizera, wykorzystując pakiet XAMPP oraz edytor zaznaczający składnię.

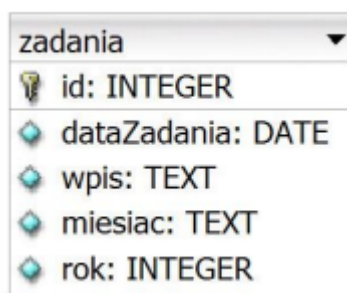
Aby wykonać zadanie zaloguj się na konto Egzamin bez hasła. Na pulpicie znajdziesz archiwum ZIP o nazwie egzamin5.zip zabezpieczone hasłem: kaL(end)arz%

Archiwum należy rozpakować.

Na pulpicie konta Egzamin utwórz folder. Jako nazwy folderu użyj swojego numeru PESEL. Umieść w tym folderze rozpakowane pliki.

Po skończonej pracy wyniki zapisz również w tym folderze.

### Obraz 1. Tabela zadania



zadania	
💡	id: INTEGER
💎	dataZadania: DATE
💎	wpis: TEXT
💎	miesiac: TEXT
💎	rok: INTEGER

## Operacje na bazie danych

Do wykonania zadania należy użyć tabeli zadania przedstawionej na Obrazie 1.

Uruchom usługi MySQL i Apache za pomocą XAMPP Control Panel. Za pomocą narzędzia phpMyAdmin wykonaj podane operacje na bazie danych:



- Utwórz bazę danych o nazwie `egzamin6`
- Do bazy `egzamin6` zaimportuj tabele z pliku `baza5.sql` z rozpakowanego archiwum
- Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w formacie PNG pod nazwą `import` w folderze z numerem PESEL. Nie kadruj zrzutu. Zrzut powinien obejmować cały ekran monitora, z widocznym paskiem zadań. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel
- Zapisz i wykonaj zapytania SQL działające na bazie `egzamin6`. Zapytania zapisz w pliku `kwerendy.txt`, w folderze z numerem PESEL. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie JPG i nadaj im nazwy `kw1`, `kw2`, `kw3`, `kw4`. Zrzuty powinny obejmować cały ekran monitora, z widocznym paskiem zadań
  - + Zapytanie 1: wybierające jedynie pola `dataZadania`, `miesiac` i `wpis` z tabeli `zadania` dla zadań, których miesiąc to sierpień
  - + Zapytanie 2: wybierające jedynie pola `miesiac` i `rok` z tabeli `zadania` dla zadania z datą `2020-08-01`
  - + Zapytanie 3: wybierające jedynie pole `wpis` i wyświetlające je bez powtórzeń, z tabeli `zadania` dla wpisów kończących się literą „o”
  - + Zapytanie 4: zmieniające pole `wpis` w tabeli `zadania` dla zadania z datą `2020-08-27`, nowy wpis to „Wycieczka: Karkonosze”


## Witryna internetowa

### Obraz 2. Witryna internetowa



MÓJ ORGANIZER

Wpis wydarzenia: 



2020-08-01, sierpień Remont	2020-08-02, sierpień Remont	2020-08-03, sierpień Remont	2020-08-04, sierpień Remont	2020-08-05, sierpień	2020-08-06, sierpień
2020-08-07, sierpień Rower	2020-08-08, sierpień Rower	2020-08-09, sierpień	2020-08-10, sierpień	2020-08-11, sierpień Weterynarz - Dika	2020-08-12, sierpień
2020-08-13, sierpień Sopot	2020-08-14, sierpień Sopot	2020-08-15, sierpień	2020-08-16, sierpień	2020-08-17, sierpień Wyjazd na wakacje!	2020-08-18, sierpień Tatry
2020-08-19, sierpień Tatry	2020-08-20, sierpień Tatry	2020-08-21, sierpień Tatry	2020-08-22, sierpień Tatry	2020-08-23, sierpień Tatry	2020-08-24, sierpień Tatry
2020-08-25, sierpień	2020-08-26, sierpień	2020-08-27, sierpień	2020-08-28, sierpień	2020-08-29, sierpień Weterynarz - Dika	2020-08-30, sierpień
2020-08-31, sierpień					

miesiąc: sierpień, rok: 2020  
Stronę wykonał: 0000000000

Przygotowanie grafiki:

- o Plik logo2.png , wypakowany z archiwum, należy przeskalować dokładnie do wymiarów 100 px na 100 px

Cechy witryny:

- o Składa się ze strony o nazwie organizer.php
- o Zastosowany właściwy standard kodowania polskich znaków
- o Tytuł strony widoczny na karcie przeglądarki: „Organizer”
- o Arkusz stylów w pliku o nazwie styl6.css prawidłowo połączony z kodem strony
- o Podział strony na bloki: trzy bloki banera , poniżej blok główny , w którym znajdują się bloki wyświetlające dane kolejnych dni, na dole blok stopki . Podział zrealizowany za pomocą znaczników sekcji tak, aby po uruchomieniu w przeglądarce wygląd układu bloków był zgodny z obrazem 2
- o Zawartość pierwszego bloku banera: nagłówek drugiego stopnia o treści „MÓJ ORGANIZER”
- o Zawartość drugiego bloku banera: formularz wysyłający dane metodą POST do tej samej strony, zawierający:
  - + Pole edycyjne podpisane: „Wpis wydarzenia: ”
  - + Przycisk wysyłający zawartość formularza z etykietą „ZAPISZ”



- Zawartość trzeciego bloku banera: logo2.png z tekstem alternatywnym o treści: „Mój organizer”
- Zawartość bloku głównego: efekt działania skryptu 1
- Zawartość stopki:
  - Efekt działania skryptu 2
  - Paragraf (akapit) o treści: „Stronę wykonał: ”, dalej wstawiony numer PESEL zdającego

## Styl CSS witryny internetowej

Cechy formatowania CSS:

- Domyślne dla całej strony: krój czcionki Georgia
- Dla pierwszego i drugiego bloku banera: kolor tła #3CB371, biały kolor czcionki, wysokość 100 px, szerokość 40%
- Dla trzeciego bloku banera: kolor tła #3CB371, wysokość 100 px, szerokość 20%
- Dla bloku wyświetlającego dane jednego dnia: kolor tła #FFDAB9, szerokość 170 px, wysokość 90 px, marginesy zewnętrzne 5 px, obramowanie o szerokości 1 px, linią ciągłą o kolorze #3CB371
- Dla stopki: kolor tła #3CB371, biały kolor czcionki
- Dla znacznika nagłówka szóstego stopnia: wyrównanie tekstu do prawej strony
- Dla znacznika formularza: marginesy wewnętrzne 30 px

## Skrypt połączenia z bazą

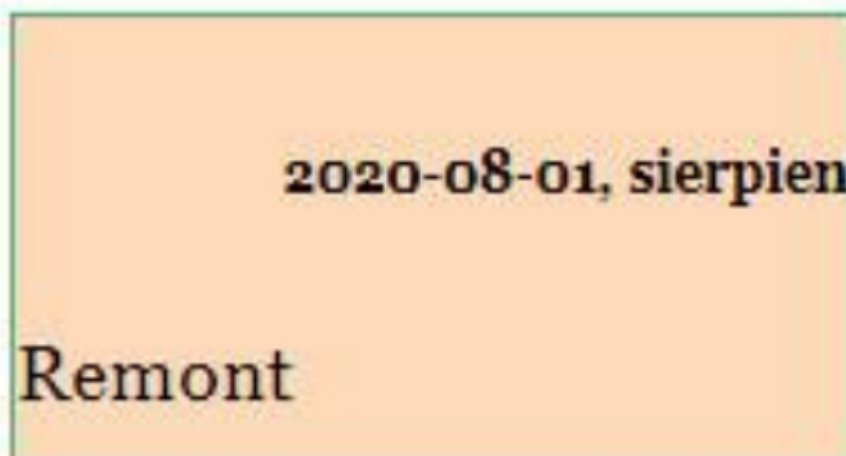
W Tabeli 1 podano wybór funkcji PHP do obsługi bazy danych.

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 egzamin6
- Działanie skryptu, który odbiera dane przesłane z formularza:
  - + Skrypt pobiera dane z pole edycyjnego
  - + Następnie uaktualnia dane w bazie danych. Aby uprościć zadanie skrypt wprowadza wpis zawsze dla rekordu o dacie 2020-08-27. Do uaktualnienia danych wykorzystaj zapytanie 4, modyfikując w nim wartość pola wpis



### Obraz 3. Pierwszy blok



- Działanie skryptu 1 :
  - + Wysła do bazy danych zapytanie 1
  - + Definiuje blok (za pomocą znaczników sekcji), a w nim wyświetla dane przesłane skryptem dotyczące jednego wiersza. Blok jest zgodny z obrazem 3 oraz jest w nim wyświetlone:
    - + W nagłówku szóstego stopnia data zadania i miesiąc oddzielone przecinkiem
    - + Pole wpis w paragrafie (akapicie)
  - + Liczba wygenerowanych bloków powinna odpowiadać liczbie wierszy zwróconych zapytaniem
- Działanie skryptu 2 :
  - + Wysła do bazy danych zapytanie 2
  - + Wyświetla w nagłówku pierwszego stopnia tekst: „miesiąc: <nazwa\_miesiąca>, rok: <rok>”, gdzie pola w nawiasach <...> oznaczają dane pobrane skryptem
  - + Na końcu działania skrypt zamyka połączenie z serwerem

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

Funkcje biblioteki mysql	Funkcje biblioteki mysqli	Zwracana wartość
mysql_connect(serwer, użytkownik, hasło)	mysqli_connect(serwer, użytkownik, hasło, nazwa_bazy)	id połączenia lub FALSE, gdy niepowodzenie

<b>Funkcje biblioteki mysql</b>	<b>Funkcje biblioteki mysqli</b>	<b>Zwracana wartość</b>
mysql_select_db ( <i>'nazwa_bazy'</i> [, <i>id_polaczenia</i> ])	mysqli_select_db( <i>id_polaczenia</i> , <i>nazwa_bazy</i> )	TRUE / FALSE, w zależności od stanu operacji
mysql_error([ <i>id_polaczenia</i> ])	mysqli_error( <i>id_polaczenia</i> )	Tekst komunikatu błędu
mysql_close([ <i>id_polaczenia</i> ])	mysqli_close( <i>id_polaczenia</i> )	TRUE / FALSE, w zależności od stanu operacji
mysql_query( <i>zapytanie</i> [, <i>id_polaczenia</i> ])	mysqli_query( <i>id_polaczenia</i> , <i>zapytanie</i> )	Wynik zapytania
mysql_fetch_row ( <i>wynik_zapytania</i> )	mysqli_fetch_row( <i>wynik_zapytania</i> )	Tablica numeryczna odpowiadająca wierszowi zapytania
mysql_fetch_array ( <i>wynik_zapytania</i> )	mysqli_fetch_array( <i>wynik_zapytania</i> )	Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania
mysql_num_rows ( <i>wynik_zapytania</i> )	mysqli_num_rows( <i>wynik_zapytania</i> )	Liczba wierszy w podanym zapytaniu
mysql_num_fields ( <i>wynik_zapytania</i> )	mysqli_num_fields ( <i>wynik_zapytania</i> )	Liczba kolumn w podanym zapytaniu ^