

Nazwa kwalifikacji: **Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych**

Oznaczenie kwalifikacji: **INF.03**

Numer zadania: **01**

Wersja arkusza: **SG**

Czas trwania egzaminu: **150 minut**

TE-WRF-INF.03-01-23.12-SG

PRÓBNY EGZAMIN ZAWODOWY

Rok 2023

CZĘŚĆ PRAKTYCZNA

**PODSTAWA PROGRAMOWA
2019**

Instrukcja dla zdającego

1. Sprawdź, czy plik PDF z arkuszem egzaminacyjnym zawiera 6 stron i nie zawiera błędów. Ewentualny brak stron lub inne usterki zgłoś przez podniesienie ręki nauczycielowi.
2. Zapoznaj się z treścią zadania oraz stanowiskiem egzaminacyjnym. Masz na to 10 minut. Czas ten nie jest wliczany do czasu trwania egzaminu.
3. Czas rozpoczęcia i zakończenia pracy nauczyciel zapisze w widocznym miejscu.
4. Wykonaj samodzielnie zadanie egzaminacyjne. **Nie możesz korzystać z żadnych zasobów niedostarczonych wraz z tym arkuszem, w szczególności z zasobów Internetu.** Przestrzegaj zasad bezpieczeństwa i organizacji pracy.
5. Po zakończeniu wykonania zadania postępuj zgodnie z informacjami od nauczyciela dotyczącymi przesyłania pracy do oceny.
6. Po uzyskaniu zgody nauczyciela możesz opuścić salę/miejsce przeprowadzania egzaminu.

Powodzenia!

Zadanie egzaminacyjne

Wykonaj aplikację internetową weryfikującą poprawność wpisywanych do bazy danych pracowników firmy. Wykorzystaj do tego celu edytor zaznaczający składnię oraz program XAMPP. Na pulpicie znajdziesz archiwum o nazwie materiały.zip zabezpieczone hasłem: EgZ@Min

Na pulpicie utwórz folder. Jako nazwy folderu użyj swojego imienia i nazwiska oraz klasy. Umieść w tym folderze rozpakowane pliki. Po skończonej pracy wyniki zapisz również w tym folderze.

Operacje na bazie danych

Do wykonania operacji na bazie należy wykorzystać przedstawioną na Obrazie 1 tabelę.



firma	
id	int(11)
nazwisko	varchar(20)
imie	varchar(20)
PESEL	varchar(11)

Obraz 1 Tabela bazy danych firma

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

- utwórz bazę danych o nazwie *firma*
- do bazy zaimportuj tabele z pliku *pracownicy.sql*. Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w formacie JPEG o nazwie *import* w folderze. 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 *firma*. Zapytania zapisz w pliku *zapytania.txt*, w folderze. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie JPG i nadaj im nazwy *zap1*, *zap2*, *zap3*, *zap4*. Zrzuty powinny obejmować cały ekran monitora, z widocznym paskiem zadań.
- Zapytanie 1: wybierające wszystkie dane pracowników z tabeli *pracownicy*.
- Zapytanie 2: dodające nowego użytkownika lokalnego o nazwie *boss* i hasło *acdc*.
- Zapytanie 3: zliczające i podające w kolumnie o nazwie *ilość pracowników* ilość rekordów z danymi znajdującymi się w tabeli *pracownicy*.
- Zapytanie 4: wstawiające do tabeli *pracownicy* rekord z danymi pracownika: *Rak, Monika, 00101843690*, gdzie odpowiednio danymi są nazwisko, imię, PESEL.

Przygotowanie witryny internetowej

Projekt witryny powinien składać się ze strony głównej *index.php* oraz *formularz.php*.



Obraz 2 Strona główna aplikacji – plik *index.php*

Cechy wspólne dla obu plików strony:

- zapiane w języku HTML5
- jawnie zastosowany właściwy standard kodowania polskich znaków
- tytuł strony *Dane osobowe* widoczny w karcie przeglądarki
- podłączony plik z arkuszem stylów *style.css*
- podział stron na bloki: baner, poniżej blok lewy i prawy, na dole stopka. Podział dokonywany jest za pomocą znaczników sekcji. Rezultat powinien być podobny do strony z Obrazu 2.
- baner z nagłówkiem pierwszego stopnia o treści „*Dane osobowe pracowników*”
- w bloku lewym lista punktowana z odnośnikami:
 - *Wyświetl dane* powiązany z plikiem *index.php*
 - *Wpisz dane* powiązany z plikiem *formularz.php*
- stopka składa się z nagłówka piątego stopnia o treści: „*AUTOR strony:* ” i imienia i nazwiska zdającego

Cechy indywidualne dla pliku *index.php*:

- w bloku prawym:
 - nagłówek drugiego stopnia o treści „*Dane osobowe*”
 - tabela generowana skryptem 1

Obraz 3 Strona główna aplikacji – plik formularz.php

Cechy indywidualne dla pliku *formularz.php* (jak na obrazie 3):

- w bloku prawym:
 - nagłówek drugiego stopnia o treści „*Podaj dane*”
 - formularz składający się z pól edycyjnych wraz z etykietami *Nazwisko*, *Imię* i *PESEL* oraz przycisku *Sprawdź i zapisz* wysyłającymi wpisane dane do skryptu 2
 - panel z wynikami błędów lub informacją o zapisie do bazy danych generowany skryptem 2

Styl CSS projektowanej witryny

Plik *style.css* zawiera formatowanie dla:

- całej strony: krój czcionki *Garamond*, wyśrodkowanie tekstu
- bloku banera i stopki: kolor tła #0451b0, kolor czcionki biały, wysokość 70 px, rozmiar czcionki 150%
- bloku lewego: kolor tła #5ca4fd, wysokość 600 px, szerokość 35%
- bloku prawego: kolor tła #8ec0fd, wysokość 600 px, szerokość 65%
- znacznika tabeli: margines zewnętrzny auto, obramowanie w kolorze czarnym, ciągłe o grubości 1 px
- znacznika komórki tabeli: obramowanie w kolorze czarnym, ciągłe o grubości 1 px
- znacznika komórki nagłówkowej tabeli: obramowanie w kolorze czarnym, ciągłe o grubości 1 px
- listy punktowanej: brak symboli punktatorów, szerokość 180 px
- odnośników w liście wypunktowanej: kolor czcionki czarny, margines wewnętrzny górny i dolny 8 px, margines wewnętrzny prawy i lewy 16 px, margines zewnętrzny 10 px, brak podkreślenia odnośnika, obramowanie w kolorze czarnym, ciągłe o grubości 1 px, zaokrąglenie rogów obramowania 10 px
- po najechaniu myszą na dowolny odnośnik jego tło zmienia się na #0451b0
- panelu z wynikami błędów lub informacją o zapisie do bazy danych: margines zewnętrzny górny 100 px, obramowanie w kolorze czarnym, ciągłe o grubości 1 px

Wymagania dotyczące skryptu 1:

- wykonywany po stronie serwera
- łączy się z serwerem bazodanowym na *localhost*, użytkownik *root* bez hasła, baza danych o nazwie *firma*
- wysyła do bazy zapytanie 1 i pobiera z bazy danych dane pracowników (patrz „Operacje na bazie danych”).
- generuje tabelę z wierszem nagłówkowym z komórkami *Id*, *Nazwisko*, *Imię*, *PESEL* oraz wypełnia tabelę danymi pobranymi z bazy danych (jak na obrazie 2)

Wymagania dotyczące skryptu 2:

- wykonywany po stronie serwera
- łączy się z serwerem bazodanowym na *localhost*, użytkownik *root* bez hasła, baza danych o nazwie *firma*
- w przypadku niewpisania w formularzu nazwiska wyświetla napis o treści *brak nazwiska*
- w przypadku niewpisania w formularzu imienia wyświetla napis o treści *brak imienia*
- w przypadku niewpisania w formularzu numeru PESEL wyświetla napis o treści *brak numeru PESEL*
- weryfikuje poprawność wpisanego numeru PESEL wyliczając sumę kontrolną - 11 cyfrę PESEL. (Algorytm wyliczania sumy kontrolnej znajduje się poniżej)
- po poprawnej weryfikacji wyświetla napis „Zapisuję do bazy <nazwisko> <imię> <PESEL>” (gdzie <nazwisko> <imię> <PESEL> są danymi pobranymi z formularza) i wysyła odpowiednio zmodyfikowane (uzupełnione danymi pobranymi z formularza) zapytanie 4 (patrz „Operacje na bazie danych”).

Algorytm wyliczania sumy kontrolnej numeru PESEL

PESEL składa się z 11 cyfr. Ostatnia 11 cyfra jest sumą kontrolną obliczaną według algorytmu.

Krok 1

Policz sumę iloczynów poszczególnych cyfr i odpowiadających im wag wg wzoru:

$$S = 1 \cdot \text{cyfra1} + 3 \cdot \text{cyfra2} + 7 \cdot \text{cyfra3} + 9 \cdot \text{cyfra4} + 1 \cdot \text{cyfra5} + 3 \cdot \text{cyfra6} + 7 \cdot \text{cyfra7} + 9 \cdot \text{cyfra8} + 1 \cdot \text{cyfra9} + 3 \cdot \text{cyfra10}$$

Krok 2

Oblicz resztę z dzielenia przez 10 z otrzymanej sumy S (operacja modulo) $M = S \% 10$

Krok 3

Przypadek 1

Jeśli liczba M równa jest 0 to suma kontrolna (11 cyfra PESEL) równa jest 0

Przypadek 2

Jeśli liczba M jest różna od 0 to suma kontrolna jest wynikiem odejmowania $10 - M$

Przykład 1 dla PESEL 00101843690

$$S = 1 \cdot 0 + 3 \cdot 0 + 7 \cdot 1 + 9 \cdot 0 + 1 \cdot 1 + 3 \cdot 8 + 7 \cdot 4 + 9 \cdot 3 + 1 \cdot 6 + 3 \cdot 9 = 120$$

$$M = 120 \% 10 = 0 \text{ (M równa 0 więc przypadek 1)}$$

11 cyfra PESEL = 0 Pesel jest prawidłowy

Przykład 2 dla PESEL 04061656044

$$S = 1 \cdot 0 + 3 \cdot 4 + 7 \cdot 0 + 9 \cdot 6 + 1 \cdot 1 + 3 \cdot 6 + 7 \cdot 5 + 9 \cdot 6 + 1 \cdot 0 + 3 \cdot 4 = 186$$

$$M = 186 \% 10 = 6 \text{ (M różne od 0 więc przypadek 2)}$$

11 cyfra PESEL = $10 - M = 10 - 6 =$ 4 Pesel jest prawidłowy

Skrypt wykonywany po stronie klienta

Wymagania dotyczące skryptu 3:

- wykonywany po stronie klienta
- po kliknięciu na baner znajdujący się na stronie *index.php* kolor czcionki zmienia się na #EEEEEE
- po ponownym kliknięciu na baner kolor czcionki zmienia się ponownie na kolor biały
- strona wczytywana jest zawsze z białym kolorem czcionki

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

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

Tabela 2. Wybór funkcji języka PHP do obsługi łańcuchów znakowych.

Funkcja	Opis
<code>strlen(\$a)</code>	Zwraca długość łańcucha
<code>\$a == \$b</code>	Porównuje zawartość dwóch łańcuchów
<code>\$a[i]</code>	Zwraca i-ty (indeksując od zera) znak łańcucha

Tabela 3. Wybrane metody modelu DOM języka JavaScript.

Wyszukiwanie elementów	Zmiana elementów
<code>document.getElementById(<i>id</i>)</code> <code>document.getElementsByTagName(<i>TagName</i>)</code> <code>document.getElementsByClassName(<i>ClassName</i>)</code> <code>document.getElementsByName(<i>ElementName</i>)</code>	<code>element.innerHTML = "nowa zawartość"</code> <code>element.attribute_name = "nowa zawartość"</code> <code>element.setAttribute(<i>atrybut, wartość</i>)</code> <code>element.style.property_name = "nowa wartość"</code>
Operacje na elementach dokumentu	Wybrane właściwości obiektu style
<code>document.createElement(<i>element</i>)</code> <code>document.removeChild(<i>element</i>)</code> <code>document.appendChild(<i>element</i>)</code>	<code>backgroundColor</code> <code>color</code> <code>fontSize</code>

document.replaceChild(<i>element</i>) document.write(<i>text</i>)		fontStyle = "normal italic oblique initial inherit" fontWeight = "normal lighter bold bolder value initial inherit" listStyleType = "circle decimal disc none square initial..."
Wybrane zdarzenia HTML		
Zdarzenia myszy	Zdarzenia klawiatury	Zdarzenia obiektów
onclick ondblclick onmouseover onmouseout	onkeydown onkeypress onkeyup	onload onresize onfocusin onfocusout / onblur
Elementy formularzy		Metody i pola obiektu string (JS)
Ważniejsze typy pola input: button, checkbox, numer, password, radio, text Inne elementy: select, textarea		length indexOf(<i>text</i>) search(<i>text</i>) substr(<i>startIndex</i> , <i>endIndex</i>) replace(<i>textToReplace</i> , <i>newText</i>) toUpperCase() toLowerCase()

Uwaga. Po zakończeniu pracy utwórz w stworzonym przez siebie folderze plik tekstowy o nazwie przegladarka.txt. Zapisz w nim nazwę przeglądarki internetowej, w której weryfikowałeś poprawność działania witryny.

Rezultaty swojej pracy wyślij do oceny zgodnie z instrukcją od nauczyciela. Twoje rozwiązanie powinno zawierać:

- pliki graficzne: *import.jpg*, *zap1.jpg*, *zap2.jpg*, *zap3.jpg*, *zap4.jpg*
- plik tekstowy z zapytaniami SQL: *zapytania.txt*
- plik z arkuszem stylów: *styl.css*
- pliki z kodem witryny: *index.php* i *formularz.php*

Czas przeznaczony na wykonanie zadania wynosi 150 minut.

Ocenie będzie podlegać 6 rezultatów:

- operacje na bazie danych,
- zawartość witryny internetowej,
- działanie witryny internetowej,
- styl CSS witryny internetowej,
- skrypty działające po stronie serwera,
- skrypt działający po stronie klienta.