Nazwa kwalifikacji:	**Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych**
Oznaczenie kwalifikacji:	**EE.09**
Numer zadania:	**04**
Kod arkusza:	**EE.09-04-SG_zo**

Lp.	Elementy podlegające ocenie/kryteria oceny
R.1	**Rezultat 1: Operacje na bazie danych**
	Uwaga: W przypadku oceny zrzutów należy uznać za prawidłowe jeżeli widoczny jest cały obszar ekranu, a zapytanie ma charakter uniwersalny dla każdego zestawu danych. Nie należy oceniać wykadrowanych zrzutów ekranu
R.1.1	Wykonano import tabel do bazy danych *egzamin* , czynność udokumentowano plikiem o nazwie *import* w formacie PNG
R.1.2	Zapisano plik tekstowy o nazwie *kwerendy* zawierający co najmniej jedno zapytanie SQL, wynikające z treści zadania
R.1.3	Utworzono zapytanie 1 wstawiające do tabeli *wynik* rekord o danych: klucz główny nadawany automatycznie, bmi_id = 1, data_pomiaru = 2020-05-20, wynik = 15. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `INSERT INTO wynik(bmi_id, data_pomiaru, wynik) VALUES (1, "2020-05-20", 15);` (możliwe bez pierwszego nawiasu lub przez listę przypisań) **oraz** wynik działania zapytania udokumentowano zrzutem na którym widoczne jest poprawne zapytanie lub dodany rekord w tabeli
R.1.4	Utworzono zapytanie 2 wybierające jedynie pola id, informacja i wart_min z tabeli *bmi* . W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT id, informacja, wart_min FROM bmi;` **oraz** wynik działania zapytania udokumentowano zrzutem, na którym są widoczne dokładnie cztery rekordy o id= 1, 2, 3, 4, jedynie pola id, informacja, wart_min
R.1.5	Utworzono zapytanie 3 wybierające jedynie pola wynik z tabeli *wynik* oraz informacja z tabeli *bmi* . Wykorzystano relację. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT wynik, bmi.informacja FROM wynik JOIN bmi ON wynik.bmi_id = bmi.id;` możliwe INNER JOIN lub bez JOIN (porównanie kluczy w warunku) **oraz** wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest co najmniej jeden wiersz jedynie z danymi: 20, waga prawidlowa. większa liczba wierszy wynika z wykonania kwerendy 1 lub wstawienia rekordów przez stronę ale **nie** z powodu nieprawidłowego przypisania kluczy w zapytaniu
R.1.6	Utworzono zapytanie 4 wybierające jedynie pola wart_min i wart_max z tabeli *bmi* dla niedowagi. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT wart_min, wart_max FROM bmi WHERE informacja="niedowaga";` **oraz** wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest dokładnie jeden wiersz jedynie z danymi 0, 18
R.2	**Rezultat 2: Zawartość witryny internetowej**
	Uwaga: W napisach widocznych na stronie dopuszcza się drobne błędy literowe (nie zmieniające sensu tekstu), błędy wielkości liter i znaków diakrytycznych, tekst może być pisany w cudzysłowach lub bez *Jeśli witryna nie uruchamia się z powodu błędów PHP – sprawdzić wg kryteriów w kodzie źródłowym strony*
R.2.1	Plik *rys1.png* przeskalowano z zachowaniem proporcji do wysokości 300 px, odpowiadająca temu szerokość wynosi 364 px (± 5 px); przezroczystość została zachowana
R.2.2	Witrynę internetową zapisano w pliku PHP o nazwie *waga* oraz zastosowano właściwy standard kodowania polskich znaków
R.2.3	Nadano tytuł strony: "Twój wskaźnik BMI"
R.2.4	Wygląd strony zdefiniowano dzieląc ją na bloki: baner, logo, blok lewy i prawy, główny oraz stopka. Zastosowano znaczniki sekcji
R.2.5	W bloku banera zapisano nagłówek h2: "Oblicz wskaźnik BMI", w bloku prawym nagłówek h1: "Podaj dane"

Więcej materiałów znajdziesz na zawodowe.edu.pl

R.2.6	W bloku logo umieszczono obraz *wzor.png* oraz w bloku lewym umieszczono obraz *rys1.png.* Przynajmniej jeden z obrazów ma tekst alternatywny: "liczymy BMI" (dla *wzor.png*) lub "zrzuć kalorie!" (dla *rys1.png*), gdy obraz nie istnieje powinien wyświetlić się tekst alternatywny
R.2.7	W bloku prawym umieszczono formularz, zastosowano znacznik <form> oraz jego zamknięcie
R.2.8	W formularzu umieszczono dwa pola edycyjne oraz przycisk "Licz BMI i zapisz wynik"
R.2.9	W bloku głównym umieszczono tabelę o trzech kolumnach i przynajmniej jednym wierszu, użyto znaczniki <table>, <tr>, <td> i ich zamknięcia, generowane skryptem lub HTML
R.2.10	Pierwszy wiersz tabeli jest wierszem nagłówkowym, zastosowano znacznik <th> (opcjonalnie <thead> i <td>) oraz jego zamknięcie
R.3	**Rezultat 3: Działanie witryny internetowej**
	Uwaga: Jeśli witryna nie uruchamia się z powodu błędów PHP – uruchomić plik lokalnie z dysku lub sprawdzić wg kryteriów w kodzie źródłowym strony
R.3.1	Wygląd **układu bloków** po uruchomieniu strony w przeglądarce jest w pełni zgodny z obrazem 2 w arkuszu egzaminacyjnym
R.3.2	Przycisk wysyła dane z formularza oraz przynajmniej jedno pole do wprowadzania danych jest typu liczbowego
R.3.3	Formularz wysyła dane metodą POST do tego samego pliku (poprawne również bez atrybutu action)
R.3.4	Kliknięcie odsyłacza otwiera dowolny plik ze zrzutem ekranu
R.3.5	Strona zawiera działające połączenie z zewnętrznym arkuszem stylów o nazwie *styl4.css* , formatowanie pochodzi jedynie z tego arkusza Uwaga: kryterium to nie jest spełnione również, gdy zdający źle przypisał co najmniej jeden selektor do bloku (np. kropka zamiast #)
R.4	**Rezultat 4: Styl CSS witryny internetowej**
	Uwaga: W przypadku, gdy nie jest spełnione kryterium 3.5, kryteria w rezultacie R.4 należy ocenić w kodzie CSS (w pliku CSS lub znaczniku <style>). Składnia musi być zgodna ze specyfikacją CSS *styl znacznika tabeli odpowiada selektorowi table lub selektorowi identyfikatora przypisanego do znacznika table lub selektorowi klasy jeżeli jest przypisana tylko do table; podobnie dla znacznika form*
R.4.1	Ustawiono krój czcionki Helvetica dla znacznika body lub kontenera całej strony
R.4.2	Ustawiono kolor tła #FFE4B5 dla znacznika body lub kontenera całej strony, oraz kolor tła #4B0082 dla bloków banera, głównego i stopki
R.4.3	Ustawiono biały kolor czcionki dla banera, stopki i znacznika tabeli
R.4.4	Ustawiono wysokość bloku logo i banera na 100px, bloku lewego i prawego: 300px
R.4.5	Ustawiono szerokość bloku logo na 25%, banera na 75%, bloku lewego na 45%, prawego na 55%, znacznika tabeli 90%
R.4.6	Ustawiono wyrównanie tekstu do środka dla banera, bloku prawego i znacznika tabeli
R.4.7	Ustawiono marginesy wewnętrzne (padding) 60px dla bloku głównego oraz 40px dla stopki oraz marginesy zewnętrzne (margin) 40px dla znacznika form
R.4.8	Ustawiono obramowanie o szerokości 1px linią kreskowaną, żółtą (dowolny odcień) dla znacznika tabeli
R.4.9	W momencie najechania kursorem na pojedynczy wiersz tabeli kolor tła wiersza zmienia się na #9370DB a czcionki na żółty (dowolny odcień) (np. za pomocą pseudoklasy hover)
R.4.10	Ustawiono rozmiar czcionki 150% dla banera
R.5	**Rezultat 5: Skrypt połączenia z bazą**
	Uwaga: jeśli witryna nie uruchamia się z powodu błędów PHP - sprawdzić 5.1 - 5.6, 5.8 wg kryteriów w kodzie źródłowym strony. Kryteria uznajemy za spełnione jeżeli zastosowano biblioteki MySQLi lub PDO
R.5.1	Skrypt napisano w języku PHP i zawiera on co najmniej jedną instrukcję poprawną składniowo

Pobierz więcej arkuszy z zawodowe.edu.pl

R.5.2	Skrypt realizuje połączenie z serwerem bazy danych i wybór bazy *egzamin*
R.5.3	Po wykonaniu operacji na bazie zastosowano funkcję zamknięcia bazy
R.5.4	W skrypcie działającym na danych przesłanych przez formularz zastosowano tablicę $_POST, zapis musi być poprawny składniowo i logicznie
R.5.5	Skrypt 1 wysyła do bazy danych zapytanie 1 zmodyfikowane tak, że dane wstawiane do tabeli są wyliczone z danych formularza lub zwrócone funkcją PHP np. date('Y-m-d') lub Skrypt 2 wysyła do bazy zapytanie 2
R.5.6	Skrypt 1 liczy wskaźnik BMI na podstawie danych z formularza, wzór np. `$waga / ($wzrost * $wzrost) * 10000;`
R.5.7	Skrypt 1 wyświetla dane i obliczenia według wzoru: „Twoja waga: <podana_waga>; Twój wzrost: <podany_wzrost> BMI wynosi: <obliczona_wartość>”
R.5.8	Skrypt 1 wyznacza przedział BMI, zgodnie z tabelą *bmi* w bazie danych i przypisuje wartość 1 dla niedowagi, 2 dla wagi prawidłowej, 3 dla nadwagi i 4 dla otyłości
R.5.9	Skrypt 2 wyświetla otrzymane dane w komórkach tabeli zgodnie z obrazem 2 arkusza egzaminacyjnego