Nazwa kwalifikacji: **Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych**

Oznaczenie kwalifikacji: **EE.09**

Numer zadania: **05**

Kod arkusza: **EE.09-05-22.01-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 obejmuje cały obszar ekranu, z widocznym paskiem zadań, a zapytanie ma charakter uniwersalny dla każdego zestawu danych. Nie należy oceniać wykadrowanych zrzutów ekranu
R.1.1	Wykonano import tabel do bazy *portal* , czynność udokumentowano plikiem o nazwie *egzamin* w formacie JPEG
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 wybierające jedynie pole haslo z tabeli uzytkownicy dla użytkownika o loginie Justyna. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT haslo FROM uzytkownicy WHERE login = 'Justyna';` oraz wynik działania zapytania udokumentowano zrzutem, na którym widoczne jest jedynie jedno hasło wynikające z zapytania
R.1.4	Utworzono zapytanie 2 liczące liczbę wierszy w tabeli dane. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT COUNT(*) FROM dane;` możliwa dowolna nazwa pola zamiast * oraz wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest efekt poprawnie wykonanego zapytania - wartość 11
R.1.5	Utworzono zapytanie 3 wybierające jedynie pole login z tabeli uzytkownicy i odpowiadające mu pola rok_urodz, przyjaciol, hobby, zdjecie z tabeli dane dla użytkownika o loginie Justyna. Tabele połączone są relacją 1..1, za pomocą kluczy głównych. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT login, rok_urodz, przyjaciol, hobby, zdjecie FROM dane JOIN uzytkownicy ON dane.id = uzytkownicy.id WHERE login = 'Justyna';` oraz wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest rekord: Justyna, 1995, 10, moda, o6.jpg
R.1.6	Utworzono zapytanie 4 dodające do tabeli dane kolumnę stanowisko dowolnego typu napisowego. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `ALTER TABLE dane ADD stanowisko TEXT;` możliwe ADD COLUMN oraz dowolny typ napisowy oraz wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest efekt poprawnie wykonanego zapytania
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	Wysokość obrazu *o1.jpg* wynosi dokładnie 200 px, a szerokość 400 px
R.2.2	Witrynę internetową zapisano w pliku o nazwie *uzytkownicy.php*
R.2.3	Zastosowano właściwy standard kodowania polskich znaków dla witryny
R.2.4	Nadano tytuł strony: "Portal społecznościowy”
R.2.5	Wygląd strony zdefiniowano dzieląc ją na bloki: dwa bloki banera, blok lewy i prawy, stopka. Zastosowano znaczniki sekcji
R.2.6	W lewym bloku banera zapisano nagłówek h2: "Nasze osiedle", w bloku lewym h3 "Logowanie", w bloku prawym h3 "Wizytówka"
R.2.7	W bloku lewym umieszczono formularz z dwoma polami edycyjnymi i przyciskiem
R.2.8	W bloku prawym umieszczono dwa obrazy. Pierwszy ma tekst alternatywny o treści „osoba”, drugi wyświetla obraz serca (za pomocą skryptu lub w HTML)

R.2.9	W stopce umieszczono napis „Stronę wykonał: ”, dalej wstawiony numer PESEL zdającego lub w przypadku jego braku inny numer, którym został podpisany arkusz.
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	**Układ bloków** po uruchomieniu strony w przeglądarce jest zgodny z Obrazem 2 w arkuszu egzaminacyjnym (prawidłowo zastosowano właściwości CSS układające bloki na stronie, np. float i clear albo display flex albo grid)
R.3.2	Na stronie umieszczono odnośnik zawierający przycisk o treści „Więcej informacji”, którego kliknięcie powoduje przejście do strony *dane.html*
R.3.3	Formularz wysyła dane do tej samej strony metodą post (poprawne również bez atrybutu action)
R.3.4	Pole edycyjne do wpisywania hasła jest typu password oraz przycisk wysyła dane z formularza (input z typem submit lub button)
R.3.5	Strona zawiera działające połączenie z zewnętrznym arkuszem stylów o nazwie *styl5.css* , formatowanie pochodzi jedynie z tego arkusza Uwaga: kryterium to nie jest spełnione również, gdy źle przypisano 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
R.4.1	Ustawiono domyślny krój czcionki dla całej strony na Tahoma (dla selektora * lub body lub html lub kontenera całej strony)
R.4.2	Ustawiono kolor tła #003d33 dla obu bloków banera i stopki, #26a69a dla bloku lewego i prawego
R.4.3	Ustawiono biały kolor czcionki dla obu bloków banera i stopki
R.4.4	Ustawiono wysokość bloków: 50px dla obu bloków banera, 550px dla bloku lewego i prawego
R.4.5	Ustawiono szerokość banera lewego i bloku prawego 70% oraz banera prawego i bloku lewego 30%
R.4.6	Ustawiono wyrównanie tekstu do środka dla obu bloków banera i bloku wizytówki
R.4.7	Ustawiono marginesy zewnętrzne (margin) dla bloku wizytówki 30px
R.4.8	Ustawiono pozostałe cechy bloku wizytówki: szerokość 400px, obramowanie linią ciągłą 1 px koloru #003d33
R.4.9	Ustawiono styl dla przycisku w bloku wizytówki (dowolną metodą, tak aby tylko ten przycisk był formatowany stylem) kolor tła #003d33, biały kolor czcionki, szerokość 100%, wysokość 30 px, bez obramowania
R.4.10	Gdy kursor myszy znajdzie się na przycisku w bloku wizytówki jego tło zamienia się na #004c40, np. za pomocą pseudoklasy hover
R.5	**Rezultat 5: Skrypt**
	Uwaga: Rezultat sprawdzić w wersji XAMPP 8.0.7 jeśli witryna nie uruchamia się z powodu błędów PHP - sprawdzić 5.1 ÷ 5.3 i 5.8, 5.9 wg kryteriów w kodzie źródłowym strony. Kryteria uznajemy za spełnione jeżeli zastosowano biblioteki MySQLi lub PDO
R.5.1	Skrypt realizuje połączenie z serwerem bazy danych i wybór bazy *portal*
R.5.2	Ostatnią operacją na bazie jest jej zamknięcie (zgodne ze składnią)
R.5.3	Skrypt 1 wysyła do bazy zapytanie 2 **lub** skrypt 2 wysyła zapytanie 1 i 3 zmodyfikowane tak, że w klauzuli warunku sprawdzony jest login przesłany z formularza
R.5.4	Skrypt 1 wyświetla liczbę użytkowników znajdujących się w bazie (rozwiązanie ma charakter uniwersalny) według wzoru: „Liczba użytkowników portalu: <zwrocona_wartosc>”
R.5.5	Skrypt 2 wyświetla komunikat "login nie istnieje", **jeżeli** podany formularzem login nie istnieje w bazie
R.5.6	Skrypt 2 wyświetla komunikat "hasło nieprawidłowe", **jeżeli** dla loginu nie zgadza się hasło podane formularzem z hasłem w bazie - zastosowano funkcję sha1 do porównania szyfrowanego hasła
R.5.7	Skrypt 2, **tylko jeżeli** login i hasło się zgadzają wyświetla dane odpowiadające loginowi w bloku wizytówka: obraz, login, hobby, liczba przyjaciół
R.5.8	Skrypt 2 oblicza wiek na podstawie roku urodzenia z bazy i aktualnej daty. Np. `$wiek = date('Y') - $wiersz[1];` (dopuszcza się jawne użycie roku 2022)

R.5.9	W skrypcie 1 komunikat jest zapisany w nagłówku h5 oraz w skrypcie 2 login zapisany jest w nagłówku h4, hobby w paragrafie, liczba przyjaciół w nagłówku h1