

Nazwa kwalifikacji:	Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych
Oznaczenie kwalifikacji:	EE.09

Numer zadania: **05**

Kod arkusza: **EE.09-05-21.01-SG**

Lp.	Elementy podlegające ocenie/kryteria oceny
R.1	Rezultat 1: Operacje na bazie danych
	<i>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</i>
R.1.1	Wykonano import tabel do bazy danych <i>egzamin</i> , czynność udokumentowano plikiem o nazwie <i>import</i> w formacie PNG
R.1.2	Zapisano plik tekstowy o nazwie <i>kwerendy</i> z zapytaniami SQL zawierający co najmniej jedno zapytanie SQL, wynikające z treści zadania
R.1.3	Utworzono zapytanie 1 wybierające jedynie pola czas, kierunek, nr_rejsu i status_lotu z tabeli <i>przyloty</i> posortowane rosnąco według czasu. W pliku z kwerendami lub na zrzucie istnieje zapis np.: SELECT czas, kierunek, nr_rejsu, status_lotu FROM przyloty ORDER BY czas; (też gdy zapisano ASC) <u>oraz</u> wynik działania zapytania udokumentowano zrzutem, na którym jest widocznych dokładnie 8 rekordów, jedynie pola czas, kierunek, nr_rejsu, status_lotu. Rekordy są posortowane rosnąco wg. czasu
R.1.4	Utworzono zapytanie 2 zliczające przyloty, które odbyły się w godzinach 10:00 – 10:59. W pliku z kwerendami lub na zrzucie istnieje zapis np.: SELECT COUNT(*) FROM przyloty WHERE czas LIKE "10%"; możliwe inne warianty, np. z BETWEEN <u>oraz</u> wynik działania zapytania udokumentowano zrzutem, na którym jest widoczna wartość 7, wynikająca z zadanego zapytania
R.1.5	Utworzono zapytanie 3 wybierające jedynie pola nr_rejsu z tabeli <i>przyloty</i> oraz typ z tabeli <i>samoloty</i> , dla lotów których kierunkiem jest Warszawa. Wykorzystano relację. W pliku z kwerendami lub na zrzucie istnieje zapis np.: SELECT nr_rejsu, samoloty.typ FROM przyloty JOIN samoloty ON przyloty.samoloty_id = samoloty.id WHERE kierunek = "Warszawa"; możliwe INNER JOIN lub bez JOIN (porównanie kluczy w warunku) <u>oraz</u> wynik działania zapytania udokumentowano zrzutem, na którym widoczny jest dokładnie jeden rekord jedynie z danymi: W63454, Boening 737
R.1.6	Utworzono zapytanie 4 aktualizujące w tabeli <i>odloty</i> pole status_lotu na: „opoznienie 20 min” dla lotu o numerze rejsu LX5673. W pliku z kwerendami lub na zrzucie istnieje zapis np.: UPDATE odloty SET status_lotu="opoznienie 20 min" WHERE nr_rejsu="LX5673"; <u>oraz</u> wynik działania zapytania udokumentowano zrzutem na którym widoczne jest poprawne zapytanie lub widoczne są rekordy ze zmienionymi danymi
R.2	Rezultat 2: Zawartość witryny internetowej
	<i>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</i> <i>Jeśli witryna nie uruchamia się z powodu błędów PHP – sprawdzić wg kryteriów w kodzie źródłowym strony</i>

R.2.1	Plik <i>zad5.png</i> przeskalowano z zachowaniem proporcji do wysokości 200 px, odpowiadająca temu szerokość wynosi 305 px (± 5 px); przezroczystość została zachowana
R.2.2	Witrynę internetową zapisano w pliku PHP o nazwie <i>lotnisko</i> oraz zastosowano właściwy standard kodowania polskich znaków
R.2.3	Nadano tytuł strony: "Port Lotniczy"
R.2.4	Wygląd strony zdefiniowano dzieląc ją na bloki: trzy bloki banera, blok główny, dwa bloki stopki. Zastosowano znaczniki sekcji
R.2.5	W drugim bloku banera zapisano nagłówek h1: "Przyloty", w trzecim bloku banera nagłówek h3: "przydatne linki" oraz w pierwszym bloku stopki zapisano paragraf (w kodzie HTML lub skrypcie)
R.2.6	W pierwszym bloku banera umieszczono obraz <i>zad5.png</i> z tekstem alternatywnym: "logo lotnisko", gdy obraz nie istnieje powinien wyświetlać się tekst alternatywny
R.2.7	W bloku głównym umieszczono tabelę o czterech kolumnach, zastosowano znaczniki <code><table></code> , <code><tr></code> , <code><td></code> i ich zamknięcia, tabela generowana skryptem lub HTML
R.2.8	Umieszczono w tabeli wiersz nagłówkowy, zastosowano cztery znaczniki <code><th></code> (opcjonalnie <code><thead></code> i <code><td></code>) oraz ich zamknięcia wewnątrz pierwszego wiersza tabeli
R.2.9	W pierwszym bloku stopki gdy wyświetlany jest tekst "Dzień dobry!..." czcionka jest pogrubiona <u>lub</u> gdy wyświetlony jest tekst "Witaj ponownie..." czcionka jest pochylona (dowolną metodą)
R.3	Rezultat 3: Działanie witryny internetowej
	<i>Uwaga: Jeśli witryna nie uruchamia się z powodu błędów PHP – uruchomić plik lokalnie z dysku lub sprawdzić wg kryteriów w kodzie źródłowym strony</i>
R.3.1	Wygląd <u>układu bloków</u> po uruchomieniu strony w przeglądarce jest w pełni zgodny z obrazem 2 w arkuszu egzaminacyjnym
R.3.2	Na stronie umieszczono odnośnik, którego wybranie prowadzi do pliku tekstowego zawierającego kwerendy
R.3.3	Odnośnik otwiera się w nowym oknie lub karcie
R.3.4	Strona zawiera działające połączenie z zewnętrznym arkuszem stylów o nazwie <i>styl5.css</i> formatowanie pochodzi jedynie z tego arkusza (poza pochyleniem i pogrubieniem) 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
	<i>Uwaga: W przypadku, gdy nie jest spełnione kryterium 3.4, kryteria w rezultacie R.4 należy ocenić w kodzie CSS (w pliku CSS lub znaczniku <code><style></code>). Składnia musi być zgodna ze specyfikacją CSS $rgb(30, 144, 255) = \#1E90FF$; $rgb(0, 0, 128) = \#000080$; $rgb(192, 192, 192) = \#C0C0C0$</i>
R.4.1	Ustawiono krój czcionki Garamond dla selektora body lub kontenera całej strony
R.4.2	Ustawiono kolor tła $rgb(30, 144, 255)$ dla selektora body lub kontenera całej strony, oraz kolor tła $rgb(0, 0, 128)$ dla bloku głównego
R.4.3	Ustawiono biały kolor czcionki dla selektora body lub kontenera całej strony
R.4.4	Ustawiono wysokość wszystkich bloków banera na 200px
R.4.5	Ustawiono szerokość pierwszego i trzeciego bloku banera na 25%, drugiego bloku banera oraz obu bloków stopki na 50% oraz szerokość tabeli na 100%

R.4.6	Ustawiono wyrównanie tekstu do środka dla drugiego bloku banera, pierwszego bloku stopki oraz wyrównanie do prawej strony dla drugiego bloku stopki
R.4.7	Ustawiono marginesy wewnętrzne (padding) na: 60px dla bloku głównego, 20px dla selektora paragrafu, 10px dla selektora td lub th oraz jedynie margines wewnętrzny górny (padding-top) 100px dla drugiego bloku stopki
R.4.8	Ustawiono obramowanie selektora p oraz td lub th: 1px linią ciągłą rgb(192, 192, 192) oraz połączone obramowanie dla selektora tabeli (border-collapse: collapse;)
R.4.9	W momencie najechania kursorem na paragraf, jego kolor tła zmienia się na rgb(192, 192, 192) (np. hover)
R.4.10	Ustawiono rozmiar czcionki 250% dla drugiego bloku banera, 200% dla selektora p oraz tekst wielkimi literami (text-transform: uppercase;) dla bloku głównego
R.5	Rezultat 5: Skrypt połączenia z bazą
	<i>Uwaga: jeśli witryna nie uruchamia się z powodu błędów PHP - sprawdzić 5.1 - 5.4, 5.6, 5.7 wg kryteriów w kodzie źródłowym strony. Kryteria uznajemy za spełnione jeżeli zastosowano biblioteki MySQLi lub PDO</i>
R.5.1	Skrypt napisano w języku PHP i zawiera on co najmniej jedną instrukcję poprawną składniowo
R.5.2	Skrypt 1 realizuje połączenie z serwerem bazy danych i wybór bazy <i>egzamin</i>
R.5.3	Po wykonaniu operacji na bazie zastosowano funkcję zamknięcia bazy
R.5.4	Skrypt 1 wysyła do bazy zapytanie nr 1
R.5.5	Dane otrzymane z bazy są wyświetlane w kolejnych wierszach tabeli, w odpowiednich komórkach, tak jak na Obrazie 2
R.5.6	Skrypt 2 wykorzystuje mechanizm ciasteczek, zastosowano funkcje setcookie, isset oraz tablicę \$_COOKIE, zapis musi być poprawny składniowo i logicznie
R.5.7	Gdy odwiedzający jest na stronie pierwszy raz zakładane jest ciasteczko z czasem trwania 2 godziny od utworzenia, np. time() + 7200
R.5.8	Gdy odwiedzający jest na stronie pierwszy raz wyświetlany jest komunikat "Dzień dobry! Strona lotniska używa ciasteczek" w przeciwnym wypadku komunikat „Witaj ponownie na stronie lotniska"