Nazwa kwalifikacji:	**Tworzenie i administrowanie stronami i aplikacjami internetowymi oraz bazami danych**
Oznaczenie kwalifikacji:	**INF.03**
Numer zadania:	**07**
Kod arkusza:	**INF.03-07-25.06-SG**
Wersja arkusza:	**SG**

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 z widocznym paskiem zadań, a zapytanie ma charakter uniwersalny dla każdego zestawu danych. Nie należy oceniać wykadrowanych zrzutów ekranu. Jeżeli część rekordów nie jest widoczna, sprawdzić w phpMyAdmin
R.1.1	Wykonany import tabel do bazy danych *wyprawy*, czynność udokumentowana w pliku *import.png*
R.1.2	Zapisany plik *kwerendy.txt* zawierający co najmniej jedno zapytanie SQL, wynikające z treści zadania
R.1.3	Wykonane zapytanie 1 wybierające jedynie nazwy wszystkich miejscowości posortowane rosnąco. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT nazwa FROM miejsca ORDER BY nazwa;` oraz wynik działania zapytania udokumentowany zrzutem, na którym widocznych jest dokładnie 21 rekordów: pierwszy Brno, ostatni Wiedeń
R.1.4	Wykonane zapytanie 2 wybierające jedynie cenę wycieczki do Malborka. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT cena FROM miejsca WHERE nazwa = "Malbork";` oraz wynik działania zapytania udokumentowany zrzutem, na którym widoczna jest wartość 2000.00
R.1.5	Wykonane zapytanie 3 wybierające jedynie nazwę, cenę i link obrazu z tabeli *miejsca* dla wierszy, w których link obrazu rozpoczyna się od 0 (zero). W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT nazwa, cena, link_obraz FROM miejsca WHERE link_obraz LIKE "0%";` oraz wynik działania zapytania udokumentowany zrzutem, na którym widocznych jest dokładnie 9 rekordów, pierwszy Malbork 2000, *01.jpg*; ostatni Olsztyn 660 09*.jpg*
R.1.6	Wykonane zapytanie 4 wybierające jedynie nazwy wszystkich miejsc i odpowiadające im obliczone średnie liczby osób dorosłych, zgrupowane według nazw miejsc. Średnia liczba osób jest zaokrąglona do liczby całkowitej. W pliku z kwerendami lub na zrzucie istnieje zapis np.: `SELECT nazwa, ROUND(AVG(liczba_doroslych)) FROM miejsca JOIN wycieczki USING(id_miejsca) GROUP BY nazwa;` *(lub porównanie kluczy po ON lub WHERE, dowolna metoda zaokrąglenia do liczby całkowitej)* oraz wynik działania zapytania udokumentowany zrzutem, na którym widocznych jest dokładnie 16 rekordów dla Brna, Gdańska, ..., Warszawy, Wenecji

R.1.7	Wykonane zapytanie 5 tworzące tabelę *klienci* z kolumnami: Klucz podstawowy: id_klienta jako liczba całkowita dodatnia, autoinkrementowany, nie może przyjmować wartości NULL, imie typu tekstowego o długości 20 znaków, nazwisko typu tekstowego o długości 40 znaków, data_ur typu daty. W pliku z kwerendami lub na zrzucie istnieje zapis np.:
	`CREATE TABLE klienci (`
	`id_klienta INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,`
	`imie VARCHAR(20), nazwisko VARCHAR(40), data_ur DATE );`
	oraz wynik działania zapytania udokumentowany zrzutem, na którym jest komunikat o poprawnym wykonaniu zapytania lub struktura tabeli *klienci*

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	Grafika *02.jpg* przeskalowana z zachowaniem proporcji do szerokości 200 px, odpowiadająca temu wysokość wynosi 134 px (± 5 px)
R.2.2	W pliku index.php zapisana deklaracja HTML5 <!DOCTYPE HTML>, deklaracja języka zawartości: polski np. <html lang="pl">, w sekcji head znajduje się jawnie podany standard kodowania polskich znaków
R.2.3	Nadany stronie tytuł: "Biuro turystyczne"
R.2.4	W sekcji body zdefiniowany jest podział na bloki za pomocą semantycznych znaczników HTML 5: nav, main->aside+section, footer
R.2.5	Umieszczony w bloku bocznym <h3>: „Twój cel wyprawy” i <h4>: „Koszt wycieczki", w sekcji <h3>: „Wycieczki” oraz paragraf z numerem zdającego w stopce
R.2.6	W bloku nawigacyjnym umieszczona jedna lista punktowana z trzema elementami, zastosowane znaczniki ,
R.2.7	W bloku bocznym umieszczony formularz z listą rozwijaną, 3 polami edycyjnymi i przyciskiem „Symulacja ceny”
R.2.8	W sekcji umieszczony przynajmniej jeden blok z przypisaną klasą *wycieczka (w skrypcie lub HTML)*
R.2.9	W sekcji umieszczona przynajmniej jedna grafika z tekstem alternatywnym "zdjęcie z wycieczki" *(w skrypcie lub HTML)*
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 ilustracją 3 w arkuszu egzaminacyjnym *(prawidłowo zastosowane właściwości CSS decydujące o układzie bloków na stronie, np. float i clear albo display flex albo grid albo position)*
R.3.2	Strona zawiera działające połączenie z zewnętrznym arkuszem stylów o nazwie *styl.css* formatowanie CSS pochodzi jedynie z tego arkusza
R.3.3	Utworzone 3 odnośniki : Kliknięcie odnośnika „Wczasy” otwiera podstronę *wczasy.html,* kliknięcie odnośnika „Wycieczki” otwiera podstronę *wycieczki.html,* kliknięcie odnośnika „All inclusive” otwiera podstronę *allinclusive.html*
R.3.4	Formularz wysyła dane metodą POST do tego samego pliku *(poprawne również bez atrybutu action)*
R.3.5	Przycisk realizuje funkcję wysłania danych z formularza *(input lub button z typem submit lub button bez typu)*
R.3.6	Przynajmniej jedna etykieta jest powiązana z kontrolką, dwa pola edycyjne są typu number, jedno typu date
R.4	**Rezultat 4: Styl CSS witryny internetowej**
	Uwaga: Kryteria muszą działać na stronie (sprawdzić w inspektorze). Dopuszcza się sprawdzenie w kodzie CSS tylko w przypadku, gdy formatowany element nie istnieje w HTML lub plik CSS nie został powiązany z witryną, a kod CSS jest zgodny ze składnią (np. w HTML nie zdefiniowano atrybutu klasy lub id lub brakuje elementu formatowanego selektorem typu)
R.4.1	Ustawione domyślnie dla wszystkich selektorów (*): krój czcionki "Comic Sans MS", sans-serif *(tylko taka kolejność*) oraz wyrównanie tekstu do środka
R.4.2	Ustawiony kolor tła dla bloku nawigacyjnego i stopki #00838F, dla bloku bocznego #B2DFDB
R.4.3	Ustawiony kolor czcionki dla bloku nawigacyjnego, stopki i selektora a #B2DFDB
R.4.4	Ustawione dla bloku bocznego i sekcji wysokość 600px oraz szerokość 50% *(width lub grid-template-columns lub flex)*
R.4.5	Ustawiona dla bloku bocznego i sekcji właściwość `overflow: auto;`

R.4.6	Ustawione marginesy wewnętrzne *(padding)* dla bloku nawigacyjnego i stopki 10px oraz marginesy zewnętrzne *(margin):* 7px dla selektora img i jedynie lewy 50px dla odnośnika
R.4.7	Ustawione dla obrazu opływanie po jego lewej stronie *(float: right;)* oraz zaokrąglenie rogów 30px
R.4.8	Ustawione dla selektora li: `display: inline-block;` punktory nie są widoczne oraz dla odnośnika styl bez podkreślenia `text-decoration: none;`
R.4.9	Ustawiony styl klasy dla wycieczki: kolor tła #00838F z przezroczystością 0.7 lub RGBA(0,131,143,0.7) lub #00838fb3, wysokość 150px, marginesy zewnętrzne: górny i dolny 10px, boczne 100px, zaokrąglenie rogów 30px
R.4.10	Gdy kursor znajdzie się na odnośniku, jego kolor czcionki zmienia się na biały i dodawane jest podkreślenie **lub** gdy kursor znajdzie się na bloku ostylowanym klasą wycieczka, przezroczystość wynosi 1
R.5	**Rezultat 5: Skrypt połączenia z bazą**
	Uwaga: rezultat R.5 sprawdzić w wersji XAMPP 8.2.12. Jeśli witryna nie uruchamia się z powodu błędów PHP - sprawdzić kryteria 5.1, 5.2, 5.4, 5.5, 5.8, 5.10 w kodzie źródłowym strony. Instrukcje muszą być zgodne ze składnią oraz zawierać dane wynikające z zadania. Kryteria 5.1 ÷ 5.4, 5.6 ÷ 5.9 są sprawdzane tylko w przypadku użycia bibliotek MySQLi lub PDO
R.5.1	Skrypt realizuje połączenie z serwerem bazy danych i wybór bazy *wyprawy* oraz ostatnią operacją na bazie jest jej zamknięcie
R.5.2	Skrypt 1 zawiera instrukcję wysyłającą do bazy danych zapytanie 1 lub Skrypt 3 zawiera instrukcję wysyłającą do bazy danych zapytanie 3
R.5.3	Skrypt 1 wyświetla kolejne miejsca wycieczki jako elementy listy rozwijanej
R.5.4	Skrypt 2 zawiera instrukcję wysyłającą do bazy danych zmodyfikowane zapytanie 2, w ten sposób, że sprawdzane jest miejsce wycieczki wysłane z formularza
R.5.5	Skrypt 2 wylicza wartość wycieczki na podstawie wysłanej z formularza liczby dorosłych i dzieci oraz ceny zwróconej zapytaniem, przy czym dzieci płacą połowę ceny wycieczki
R.5.6	Skrypt 2 wyświetla tekst „W dniu: <termin>”, gdzie pole <> zostało wysłane z formularza
R.5.7	Skrypt 2 wyświetla tekst „<wartość> złotych”, gdzie pole <> jest wyliczoną wartością kosztu wycieczki
R.5.8	Skrypt 3 dla każdego zwróconego zapytaniem wiersza tworzy blok. W bloku wyświetlane są miejsce i cena wycieczki
R.5.9	Skrypt 3 wyświetla obrazy których nazwy zostały zwrócone zapytaniem
R.5.10	W skryptach zastosowana przynajmniej jedna zmienna oraz nazwy wszystkich zmiennych są w języku polskim lub angielskim i odzwierciedlają cel zastosowania