

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.03-01-23.05-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 nauczycielowi przez podniesienie ręki.
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

Uwaga. Katalog z rezultatami pracy oddać zgodnie z poleceniem prowadzącego próbny egzamin.

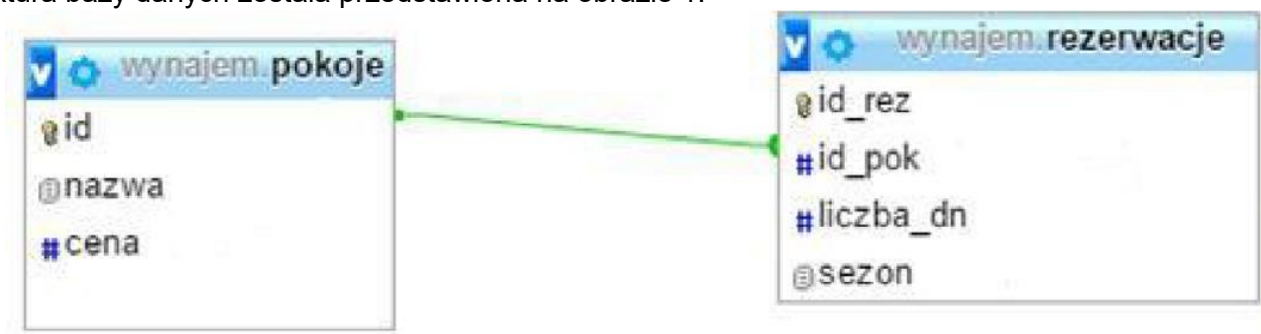
Wykonaj aplikację internetową pensjonatu, wykorzystując pakiet XAMPP oraz edytor zaznaczający składnię. Aby wykonać zadanie, należy zalogować się na konto **Egzamin** bez hasła. Na pulpicie znajduje się archiwum ZIP o nazwie *zad2.zip* zabezpieczone hasłem: **MN52dc%%**

Archiwum należy rozpakować.

Na pulpicie konta **Egzamin** należy utworzyć folder. Jako nazwy folderu użyj swojego nazwiska, imienia oraz nazwy klasy. Rozpakowane pliki należy umieścić w tym folderze. Po skończonej pracy wszystkie wyniki należy zapisać w tym folderze.

Operacje na bazie danych

Struktura bazy danych została przedstawiona na obrazie 1.



Obraz 1. Tabele bazy danych.

Za pomocą narzędzia phpMyAdmin wykonaj operacje na bazie danych:

- utwórz bazę danych o nazwie *wynajem*;
- do bazy *wynajem* zaimportuj tabele z pliku *wynajem.sql* z rozpakowanego archiwum;
- wykonaj zrzut ekranu po imporcie. Zrzut zapisz w formacie PNG i nazwij *import*. Nie kadruj zrzutu. Powinien on obejmować cały ekran monitora, z widocznym paskiem zadań. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel;
- wykonaj zapytania SQL działające na bazie *wynajem*. Zapytania zapisz w pliku *kwerendy.txt*. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie JPEG i nadaj im nazwy *kw1*, *kw2*, *kw3*, *kw4*, *kw5*. Zrzuty powinny obejmować cały ekran monitora z widocznym paskiem zadań.
 - Zapytanie 1: wybierające wszystkie wiersze z tabeli *pokoje*.
 - Zapytanie 2: wybierające z tabeli *rezerwacje* jedynie id pokoju oraz sezon dla rezerwacji o liczbie dni przekraczającej tydzień.
 - Zapytanie 3: liczące sumę dni rezerwacji zimą.
 - Zapytanie 4: aktualizujące liczbę dni rezerwacji. Minimalna liczba dni rezerwacji to 4, dla rekordów, w których liczba dni jest mniejsza od 4 liczba ta jest zmieniana na 4.
 - Zapytanie 5: tworzące użytkownika **najemca** z hasłem **n@JEM1@!**

Grafika

Obrazy *1.jpg*, *2.jpg*, *3.jpg* należy przeskalować bez zachowania proporcji do wymiarów 300 px na 300 px, następnie zapisać w odcieniach szarości jako *1szary.jpg*, *2szary.jpg*, *3szary.jpg*, odpowiednio.

Witryna internetowa



Obraz 2. Witryna internetowa, strona *index.html*, kursor znalazł się na odnośniku **CENNIK**.

Cechy witryny:

- składa się ze stron *index.html*, *cennik.php*, *kalkulator.html*,
- na stronie *kalkulator.html* należy jedynie zapisać „strona w trakcie budowy”,
- strony *index.html* i *cennik.php* różnią się jedynie blokiem środkowym.

Cechy wspólne dla stron *index.html* i *cennik.php*:

- zapisane w języku HTML5,
- jawnie zastosowany właściwy standard kodowania polskich znaków,
- zadeklarowany język zawartości witryny: polski,
- tytuł strony widoczny na karcie przeglądarki: „Wynajem pokoi”,
- arkusz stylów w pliku o nazwie *styl2.css* prawidłowo połączony z kodem strony,
- podział strony na bloki: baner, poniżej obok siebie blok lewy, środkowy i prawy, na dole stopka. Podział zrealizowany za pomocą znaczników sekcji tak, aby po uruchomieniu w przeglądarce wygląd układu bloków był zgodny z obrazem 2,
- zawartość banera: nagłówek pierwszego stopnia o treści „Pensjonat pod dobrym humorem”,
- zawartość bloku lewego:
 - odnośnik „GŁÓWNA” do pliku *index.html*,
 - obraz *1szary.jpg* z tekstem alternatywnym „pokoje”;
- zawartość bloku prawego:
 - odnośnik „KALKULATOR” do pliku *kalkulator.html*,
 - obraz *3szary.jpg* z tekstem alternatywnym „pokoje”;
- zawartość stopki: napis „Stronę opracował(a): <informacje>”, gdzie <informacje> to pole zawierające Twoje imię, nazwisko i nazwę klasy.

Zawartość bloku środkowego dla strony *index.html*:

- odnośnik „CENNIK” do pliku *cennik.php*,
- obraz *2szary.jpg* z tekstem alternatywnym „pokoje”.

Zawartość bloku środkowego dla strony *cennik.php*:

- odnośnik „CENNIK” do pliku *cennik.php*
- tabela o trzech kolumnach wypełniana skryptem

<u>CENNIK</u>		
1	jednoosobowy	100
2	dwuosobowy	170
3	trzyosobowy	210

Obraz 3. Blok środkowy strony *cennik.php*.

Styl CSS witryny internetowej

Styl CSS zdefiniowany w całości w zewnętrznym pliku o nazwie *styl2.css*. Arkusz CSS zawiera formatowanie:

- ustawione domyślne wartości dla wszystkich selektorów stylu CSS: krój czcionki *Garamond*, wyrównanie tekstu do środka,
- dla selektora ciała strony: kolor tła *Lavender*,
- wspólne dla banera i stopki: kolor tła #AF8EB5, biały kolor czcionki, marginesy wewnętrzne 10 px,
- wspólne dla bloków lewego, środkowego i prawego: kolor tła #E1BEE7, szerokość 30%, wysokość 500 px, margines wewnętrzny górny 40 px, marginesy zewnętrzne 15 px, cień bloku o przesunięciu 10 px w obu osiach, rozmyciu 10 px i kolorze *DimGray*; informacje o właściwości CSS dotyczące cienia umieszczono w tabeli *Tabela 3*,
- dla selektora odnośnika: kolor czcionki *Indigo*, rozmiar czcionki 150%, marginesy wewnętrzne 10 px, czcionka pogrubiona,
- gdy kursor znajdzie się na odnośniku, jego kolor tła zmienia się na *Indigo*, a kolor czcionki na biały,
- wspólne dla selektora obrazu i tabeli: obramowanie linią ciągłą o szerokości 1 px, szerokość 80%, marginesy zewnętrzne górny i dolny 40 px, lewy i prawy automatycznie wyliczany przez przeglądarkę.

Uwaga. Style CSS dla odnośnika, obrazu i tabeli należy zdefiniować wyłącznie przy pomocy selektora dla znaczników odnośnika, obrazu i tabeli. Jest to uwarunkowane projektem późniejszej rozbudowy witryny.

Skrypt po stronie serwera

W tabeli *Tabela 1* podano wybór funkcji PHP do obsługi bazy danych.

Wymagania dotyczące skryptu:

- napisany w języku PHP,
- łączy się z serwerem bazy danych na *localhost*, użytkownik *root* bez hasła, baza danych o nazwie *wynajem*,
- po wykonaniu operacji na bazie danych skrypt zamyka połączenie z serwerem,
- skrypt wysyła do bazy danych zapytanie 1,
- dane z każdego zwróconego zapytaniem wiersza wypisywane są w kolejnym wierszu tabeli.

Skrypt po stronie klienta

W tabeli *Tabela 2* podano wybrane pola i metody modelu DOM języka JavaScript. Wymagania dotyczące skryptu:

- obsługuje akcje wywoływane kliknięciem na zdjęciach za pomocą języka JavaScript,
- po kliknięciu na **dowolnym zdjęciu szarym** jest ono podmieniane na **odpowiednie** zdjęcie kolorowe. Po ponownym kliknięciu – powraca do zdjęcia szarego i tak na zmianę – realizowane za pomocą funkcji. Np. po kliknięciu na zdjęcie *1szare.jpg* jest ono podmieniane na *1.jpg*, a po kliknięciu na zdjęcie *1.jpg* jest ono podmieniane na *1szare.jpg*.

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. 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>document.replaceChild(<i>element</i>)</code> <code>document.write(<i>text</i>)</code>		background-color color font-size font-style = "normal italic oblique initial inherit" font-weight = "normal lighter bold bolder value initial inherit" list-style-type = "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, endIndex</i>) replace(<i>textToReplace, newText</i>) toUpperCase() toLowerCase()

Tabela 3. Składnia właściwości CSS box-shadow.

Składnia właściwości CSS box-shadow
box-shadow: none <i>h-offset v-offset blur spread color</i> inset initial inherit;

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

W folderze powinny znajdować się pliki: *1.jpg, 2.jpg, 3.jpg, 1szary.jpg, 2szary.jpg, 3szary.jpg, cennik.php, index.html, kalkulator.html, kw1.jpg, kw2.jpg, kw3.jpg, kw4.jpg, kw5.jpg, kwerendy.txt, przeglądarka.txt, styl2.css, import.png*, ewentualnie inne przygotowane pliki.

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,
- skrypt po stronie serwera,
- skrypt po stronie klienta.