

Zadanie egzaminacyjne

Wykonaj aplikację internetową dla salonu pielęgnacji psów i kotów. Wykorzystaj pakiet XAMPP jako środowisko bazodanowo – aplikacyjne. Stronę internetową zbuduj przy wykorzystaniu edytora zaznaczającego składnię.

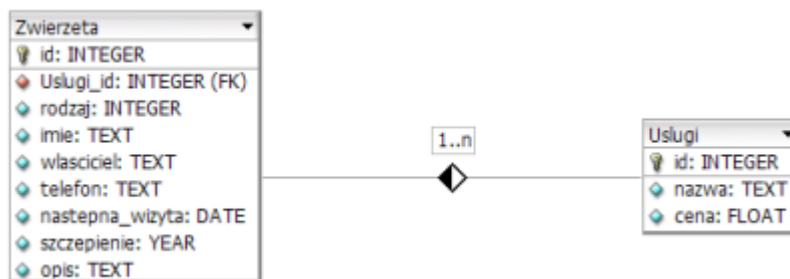
Wykorzystaj archiwum ZIP o nazwie *dane2.zip*, zabezpieczone hasłem: **Zwi3rz3ta**

Plik należy rozpakować.

Wyniki swojej pracy zapisz w folderze. Jako nazwy folderu użyj swojego imienia i nazwiska. Rozpakowane pliki umieść w tym folderze.

Operacje na bazie danych

Na obrazie 1 przedstawiono fragment bazy Salon. Pole rodzaj z tabeli Zwierzeta określa rodzaj zwierzęcia i przyjmuje wartość 1 dla psa lub wartość 2 dla kota. Pole Usługi_id tabeli Zwierzeta jest kluczem obcym wiążącym do klucza głównego tabeli Usługi



Obraz 1. Fragment bazy Salon

Uruchom usługi MySQL i Apache z XAMPP Control Panel i przejdź do narzędzia phpMyAdmin. Następnie wykonaj operacje na bazie danych:

- Utwórz nową bazę danych o nazwie Salon
- Do bazy Salon zaimportuj tabele z pliku *zwierzeta.sql* z rozpakowanego archiwum
- Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w folderze w formacie PNG i nazwij *import-salon.png*. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import bazy. Nie skaluj, ani nie przycinaj obrazu
- Utwórz następujące zapytania SQL do bazy Salon i sprawdź poprawność ich działania:
 - Zapytanie 1: wybierające jedynie pola nazwa i cena z tabeli Usługi
 - Zapytanie 2: wybierające jedynie pola imie, rodzaj, następna_wizyta, telefon z tabeli Zwierzeta dla tych rekordów, dla których następna_wizyta jest różna od 0
 - Zapytanie 3: korzystające z relacji i wybierające jedynie pola rodzaj z tabeli Zwierzeta oraz odpowiadające im pola nazwa z tabeli Usługi
 - Zapytanie 4: zwracające minimalną cenę spośród wszystkich usług zapisanych w tabeli Usługi
- Utworzone zapytania zapisz w folderze, w pliku *kwerendy.txt*. Zapytania ponumeruj stosując format zapisu: „zapytanie 1: ... treść zapytania ...” Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie PNG i nadaj im nazwy *kw1*, *kw2*, *kw3*, *kw4*. Zrzuty powinny być czytelne i wykonane bez skalowania i kadrowania, a także z widocznym paskiem zadań i godziną ich wykonania.

Witryna internetowa



Obraz 2. Witryna internetowa

Przygotowanie grafiki:

- Wykonaj obraz i jego miniaturę na podstawie grafiki rys.png wypakowanej z archiwum



Plik rys.png

- Obraz przytnij / kadruj do ramki widocznej na rys.png. W skadrowanym obrazie ramka ta nie powinna występować (powinna być odcięta)
- Skaluj obraz z zachowaniem proporcji do szerokości 400 px i zapisz tak przygotowany obraz jako pies w formacie JPEG
- Skaluj obraz z zachowaniem proporcji do szerokości 120 px i zapisz tak przygotowany obraz jako pies-mini w formacie JPEG

Cechy witryny:

- Nazwa strony: *salon.php*
- Zastosowany właściwy standard kodowania polskich znaków
- Tytuł strony, widoczny na karcie przeglądarki: „Salon pielęgnacji”
- Arkusz stylów w pliku o nazwie *salon.css*, prawidłowo połączony z kodem strony
- Podział strony na bloki: baner, obok siebie panele lewy, środkowy, prawy zrealizowany za pomocą znaczników sekcji
- Zawartość banera: nagłówek pierwszego stopnia o treści: „SALON PIELEGNACJI PSÓW I KOTÓW”
- Zawartość panelu lewego:
 - Nagłówek trzeciego stopnia o treści: „SALON ZAPRASZA W DNIACH”,
 - Lista punktowana z dwoma punktami: „Poniedziałek, 12:00 - 18:00” i „Wtorek, 12:00 - 18:00”
 - Obraz *pies-mini*, który jest jednocześnie odnośnikiem prowadzącym do grafiki *pies*, obraz powinien opływać tekst następnego akapitu z lewej strony (nie należy wykorzystywać tabeli, opływanie należy zdefiniować w kodzie CSS)
 - Tekst akapitu (paragrafu) o treści „Umów się telefonicznie na wizytę lub po prostu przyjdź!”
- Zawartość panelu środkowego
 - Nagłówek trzeciego stopnia o treści: „PRZYPOMNIENIE O NASTĘPNEJ WIZYCIE”
 - Wyniki działania skryptu nr 1
- Zawartość panelu prawego:
 - Nagłówek trzeciego stopnia o treści: „USŁUGI”
 - Wyniki działania skryptu nr 2

Styl CSS witryny internetowej

Plik *salon.css* zawiera formatowanie dla:

- Banera: kolor RGB tła (30, 137, 157), biały kolor czcionki, rozmiar czcionki 120%, wyrównanie tekstu do prawej strony, wysokość 100 px
- Panelu lewego i prawego: kolor RGB tła (58, 159, 178), szerokość 20%, wysokość 550 px
- Panelu środkowego: kolor RGB tła (143, 212, 225), szerokość 60%, wysokość 550 px
- Nagłówka trzeciego stopnia: biały kolor czcionki
- Obrazu: zdefiniowane opływanie do lewej strony

Niewymienione właściwości obiektów przybierają wartości domyślne.

Skrypt połączenia z bazą

W tabeli 1 podano wybór funkcji PHP do obsługi bazy danych. Wymagania dotyczące skryptu:

- Napisany w języku PHP
- Nie jest wymagane sprawdzenie czy operacja na bazie powiodła się
- Skrypt łączy się z serwerem bazodanowym na *localhost*, użytkownik **root** bez hasła, baza danych o nazwie *salon*

- Na końcu działania skryptu powinno zostać obsłużone zamknięcie połączenia z serwerem
- Działanie skryptu nr 1:
 - Skrypt wysyła do bazy zapytanie 2 (z pliku *kwerendy.txt*)
 - Z każdego zwróconego rekordu wyświetlane są informacje według wzoru (tekst napisany w nawiasach < > oznacza zwrócone zapytaniem pole):
 - Jeżeli pole rodzaj ma wartość 1 w pierwszej linii wypisane jest: „Pies: <imie>”
 - Jeżeli pole rodzaj ma wartość 2 w pierwszej linii wypisane jest: „Kot: <imie>”
 - Druga linia: „Data następnej wizyty: <nastepna_wizyta>, telefon właściciela: <telefon>”
- Działanie skryptu nr 2:
 - Skrypt wysyła do bazy zapytanie 1 (z pliku *kwerendy.txt*)
 - Każdy zwrócony rekord jest wyświetlany w osobnej linii

Tabela 1. Wybór funkcji języka PHP do obsługi bazy MySQL i MariaDB

Funkcje biblioteki mysql	Funkcje biblioteki mysqli	Zwracana wartość
mysql_connect(<i>serwer, użytkownik, hasło</i>)	mysqli_connect (<i>serwer, użytkownik, hasło, nazwa_bazy</i>)	id połączenia lub FALSE, gdy niepowodzenie
mysql_select_db (' <i>nazwa_bazy</i> ' [, <i>id_polaczenia</i>])	mysqli_select_db (<i>id_polaczenia, nazwa_bazy</i>)	TRUE/FALSE w zależności od stanu operacji
mysql_error([<i>id_polaczenia</i>])	mysqli_error (<i>id_polaczenia</i>)	Tekst komunikatu błędu
mysql_close([<i>id_polaczenia</i>])	mysqli_close (<i>id_polaczenia</i>)	TRUE/FALSE w zależności od stanu operacji
mysql_query(<i>zapytanie</i> [, <i>id_polaczenia</i>])	mysqli_query (<i>id_polaczenia, zapytanie</i>)	Wynik zapytania
mysql_fetch_row(<i>wynik_zapytania</i>)	mysqli_fetch_row (<i>wynik_zapytania</i>)	Tablica numeryczna odpowiadająca wierszowi zapytania
mysql_fetch_array(<i>wynik_zapytania</i>)	mysqli_fetch_array (<i>wynik_zapytania</i>)	Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania
mysql_num_rows(<i>wynik_zapytania</i>)	mysqli_num_rows (<i>wynik_zapytania</i>)	Liczba wierszy w podanym zapytaniu
mysql_num_fields (<i>wynik_zapytania</i>)	mysqli_num_fields (<i>wynik_zapytania</i>)	Liczba kolumn w podanym zapytaniu