

Zadanie egzaminacyjne

Wykonaj aplikację internetową Atlasu Zwierząt.

Wykorzystaj pakiet XAMPP jako środowisko bazowo-aplikacyjne, edytor zaznaczający składnię.

Operacje na bazie danych

Baza danych jest zgodna ze strukturą przedstawioną na rysunku 1. Tabela Zwierzeta ma dwa klucze obce, łączące je z tabelami Odzywianie i Gromady. Pole czy_zagrozony tabeli Zwierzeta przybiera wartość 0, gdy gatunek nie jest zagrożony oraz 1, gdy gatunek jest zagrożony

Uruchom usługi MySQL i Apache za pomocą XAMPP Control Panel. Za pomocą narzędzia phpMyAdmin wykonaj podane operacje na bazie danych:

Utwórz nową bazę danych o nazwie „baza”.

Do bazy „baza” zaimportuj tabele z pliku baza2.sql z rozpakowanego archiwum.

Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w folderze z numerem PESEL, w formacie PNG i nazwij import. Nie kadruj zrzutu. Powinien on obejmować cały ekran, z widocznym paskiem zadań. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel.

Zapisz i wykonaj zapytania SQL działające na bazie „baza”. Zapytania zapisz w pliku kwerendy.txt w folderze z numerem PESEL. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie JPG i nadaj im nazwy kw1, kw2, kw3, kw4, kw5, kw6. Zrzuty powinny obejmować cały ekran monitora z widocznym paskiem zadań .

Zapytanie 1: wybierające id i nazwę z tabeli Gromady.

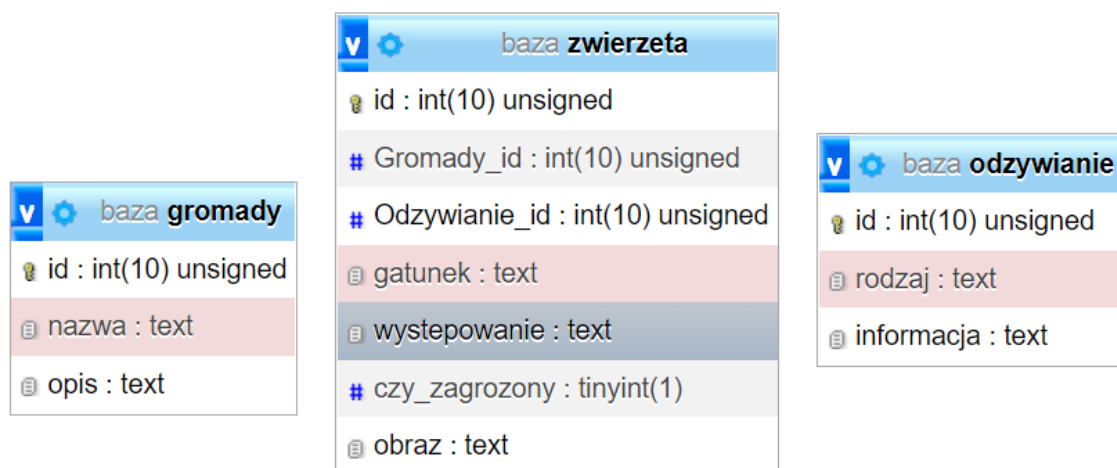
Zapytanie 2: dodające rekord do tabeli Odzywianie

Zapytanie 3: wybierające jedynie pola: gatunek i wystepowanie z tabeli Zwierzeta jedynie dla ryb.

Zapytanie 4: wybierające jedynie pola id oraz gatunek z tabeli Zwierzeta oraz odpowiadające im pole nazwa z tabeli Gromady. Należy postużyć się relacją.

Zapytanie 5: zliczające liczbę rekordów w tabeli Zwierzeta jedynie dla ssaków.

Zapytanie 6: Tworzące użytkownika KatarzynaNowak o hasło K4tNow



Rysunek 1 Baza danych

Witryna internetowa

Cechy witryny:

Składa się ze stron: index.php oraz dodaj.php. Plik dodaj.php nie ma formatowania i służy tylko do obsługi drugiego formularza.

Poniższe informacje dotyczą tylko pliku index.php

Zastosowany właściwy standard kodowania polskich znaków.

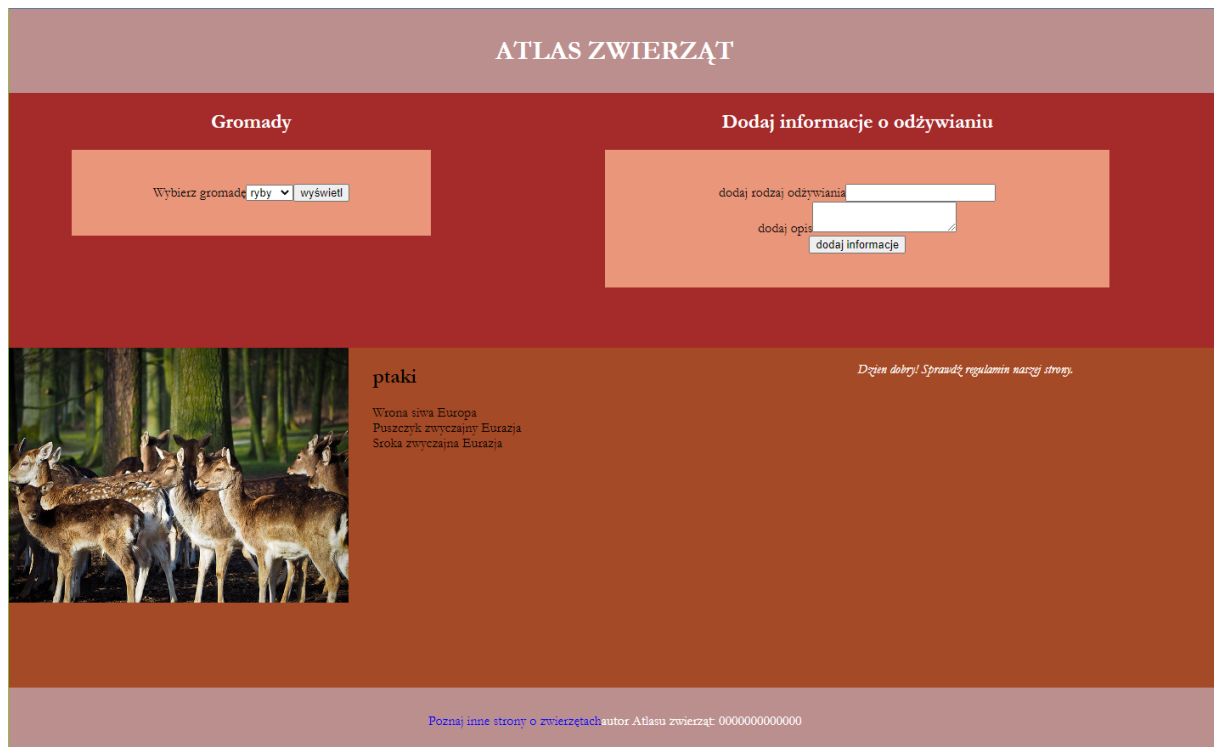
Tytuł strony widoczny na karcie przeglądarki: "Dane o zwierzętach".

Arkusze stylów w pliku o nazwie styl3.css prawidłowo połączone z kodem strony

Podział strony na bloki: na górze baner, poniżej dwa bloki z formularzami: lewy i prawy (to znaczy każdy blok zawiera jeden formularz i jeden nagłówek)

poniżej trzy bloki główne: lewy, środkowy i prawy, na dole blok stopki.

Podział zrealizowany za pomocą znaczników sekcji tak, aby po uruchomieniu w przeglądarce wygląd układu bloków był zgodny z rysunkiem 2



Rysunek 2 Widok strony wraz z działaniem mechanizmu ciasteczek

Zawartość banera: nagłówek pierwszego stopnia "ATLAS ZWIERZĄT".

Zawartość lewego bloku z formularzem:

- Nagłówek drugiego stopnia o treści "Gromady" .
- Formularz wysyłający dane metodą POST z elementami:
 - Tekst "Wybierz gromadę" - lista rozwijana, której elementy są wynikiem działania skryptu nr 1
 - Przycisk wysyłający dane o nazwie "Wyświetl".

Zawartość prawego bloku z formularzem:

- Nagłówek drugiego stopnia o treści "Dodaj informacje o odżywianiu" .
- formularz wysyłający dane metodą „ POST” z elementami
 - Tekst „dodaj rodzaj odżywiania”
 - Pole typu tekstowego
 - Poniżej tekst „dodaj opis”
 - Obszar tekstowy
 - Poniżej przycisk „dodaj informacje”
 - Formularz wysyła dane do pliku „dodaj.php”. (realizacja skryptu nr 2)

Zawartość bloku głównego

- lewego: obraz zwierzeta.jpg z wypakowanego archiwum z tekstem alternatywnym "dzikie zwierzęta".

Zawartość bloku głównego - środkowego: -

Efekt wykonania skryptu nr 3

Zawartość bloku głównego - prawego

- Efekt wykonania skryptu nr 4(mechanizm ciasteczek)

Zawartość stopki:

- Odnośnik o treści "Poznaj inne strony o zwierzętach", prowadzący do strony atlas-zwierzat.pl, odnośnik otwiera stronę w osobnym oknie
- Napis o treści: "autor Atlasu zwierząt: ", dalej wstawiony numer PESEL zdającego

Styl css witryny internetowej

Cechy formatowania działające na stronie:

Domyślne dla całej strony: krój czcionki Garamond

Dla bloku banera: kolor tła RosyBrown, biały kolor czcionki, wyrównanie tekstu do środka, marginesy wewnętrzne 10px.

Dla lewego bloku z formularzem: kolor tła brown, szerokość:40% wysokość 300px;

Dla prawego bloku z formularzem: kolor tła brown, szerokość:60% , wysokość 300px;

Formatowanie dla znacznika formularza: kolor tła: darksalmon, szerokość 60%, formularz wyśrodkowany w poziomie, wyrównanie tekstu do srodka, margines wewnętrzny:40px;

Wspólnie dla bloku głównego lewego i prawego: : kolor tła: RGB 165,74,38;, wysokość 400px, szerokość 30%.

Dla bloku głównego - środkowego: kolor tła: RGB 165,74,38; , wysokość 400px, szerokość 40%.

Dla stopki: kolor tła RosyBrown, biały kolor czcionki, wyrównanie tekstu do środka, marginesy wewnętrzne 30px

Odnośnik w stopce-brak podkreślenia

Skrypt połączenia z bazą

Wymagania dotyczące skryptów:

Napisany w języku PHP

Skrypt łączy się z serwerem bazodanowym na localhost, użytkownik root bez hasła, baza danych o nazwie „baza”.

Na końcu skrypt zamyka połączenie z serwerem bazodanowym.

Działanie skryptu nr 1: - Skrypt wysyła do bazy zapytanie numer 1 - W kolejnych elementach listy rozwijanej znajdują się nazwy gromady; (zwracaną wartością może być id lub nazwa gromady).

Działanie skryptu nr 2: - Skrypt wysyła do bazy zapytanie numer 2. Dane są pobrane z formularza i przesłane do pliku „dodaj.php”. Jeżeli co najmniej jedno z pól nie jest wypełnione, w pliku „dodaj.php” wypisuje się informacja „musisz wypełnić wszystkie pola formularza”. Jeżeli oba pola są wypełnione i dodano rekord do bazy, wyświetla się komunikat „dodano dane do bazy”.

Działanie skryptu nr 3: - Skrypt pobiera dane z listy rozwijanej (formularz pierwszy) - W zależności od wartości pola wypisuje w nagłówku drugiego stopnia nazwę gromady, a poniżej wyniki zmodyfikowanego zapytania nr3 (należy posłużyć się relacją), w ten sposób, że w sekcji warunku sprawdza czy gromada jest zgodna z wartością pobraną z pola select. Wartości zwrócone zapytaniem są wypisywane w osobnych wierszach, w formacie: "<gatunek>, <występowanie>". Jeżeli formularz nie został wysłany -wyświetla się komunikat: „nie wybrano informacji z bazy”.

Działanie skryptu nr 4:

W oparciu o mechanizm ciasteczek, skrypt wypisuje na stronie komunikaty:

-gdy odwiedzający wejdzie na stronę pierwszy raz zakładane jest ciasteczko z czasem trwania 1 godzina od utworzenia oraz wyświetlony jest w paragrafie czcionką pochyloną komunikat: „Dzien dobry! Sprawdź regulamin naszej strony”.

-jeżeli odwiedzający w ciągu godziny od poprzedniego wejścia wejdzie ponownie na stronę – wyświetlany jest w paragrafie, czcionką pogrubioną, komunikat: „Miło nam, że nas znowu odwiedziłeś.”

Ocenie będzie podlegać sześć rezultatów:

Operacje na bazie danych

Zawartość witryny internetowej

Działanie witryny internetowej

Styl CSS witryny internetowej

Skrypty połączenia z bazą

Mechanizm ciasteczek

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


```
bool setcookie ( string $name , string $value = "" , int $expire = 0 , string $path = "" , string $domain = "" , bool $secure = false , bool $httponly = false)
```

setcookie() defines a cookie to be sent along with the rest of the HTTP headers. Like other headers, cookies must be sent *before* any output from your script (this is a protocol restriction). This requires that you place calls to this function prior to any output, including `<html>` and `<head>` tags as well as any whitespace.

Once the cookies have been set, they can be accessed on the next page load with the [\\$ COOKIE](#) array. Cookie values may also exist in [\\$ REQUEST](#).

name

The name of the cookie.

value

The value of the cookie. This value is stored on the clients computer; do not store sensitive information. Assuming the `name` is `'cookieName'`, this value is retrieved through [\\$ COOKIE\['cookieName'\]](#).

expire

The time the cookie expires. This is a Unix timestamp so is in number of seconds since the epoch. In other words, you'll most likely set this with the [time\(\)](#) function plus the number of seconds before you want it to expire. Or you might use [mktime\(\)](#). `time()+60*60*24*30` will set the cookie to expire in 30 days. If set to 0, or omitted, the cookie will expire at the end of the session (when the browser closes).

path

The path on the server in which the cookie will be available on. If set to `/`, the cookie will be available within the entire `domain`. If set to `'/foo/'`, the cookie will only be available within the `/foo/` directory and all sub-directories such as `/foo/bar/` of `domain`. The default value is the current directory that the cookie is being set in.

domain

The (sub)domain that the cookie is available to. Setting this to a subdomain (such as `'www.example.com'`) will make the cookie available to that subdomain and all other sub-domains of it (i.e. `w2.www.example.com`). To make the cookie available to the whole domain (including all subdomains of it), simply set the value to the domain name (`'example.com'`, in this case).

secure

Indicates that the cookie should only be transmitted over a secure HTTPS connection from the client. When set to `TRUE`, the cookie will only be set if a secure connection exists. On the server-side, it's on the programmer to send this kind of cookie only on secure connection (e.g. with respect to [\\$ SERVER\["HTTPS"\]](#)).

httponly

When `TRUE` the cookie will be made accessible only through the HTTP protocol. This means that the cookie won't be accessible by scripting languages, such as JavaScript. It has been suggested that this setting can effectively help to reduce identity theft through XSS attacks (although it is not supported by all browsers), but that claim is often disputed. Added in PHP 5.2.0. **TRUE** or **FALSE**

Source: PHP Manual