

- *** **5** Utwórz nową bazę danych o nazwie *gatunki_filmowe*. Zaimportuj do niej dane otrzymane od nauczyciela (np. w plikach *filmy_5.txt*, *gatunki.txt*). Utwórz związek między tabelami tak, żeby atrybut *ID_gatunku* w tabeli *filmy* był kluczem obcym i wskazywał na atrybut *IDG* z tabeli *gatunki*. Napisz zapytanie, które wyświetli zestawienie wszystkich komedii.
- *** **6** Utwórz nową bazę danych o nazwie *seanse_kinowe*. Zaimportuj do niej dane otrzymane od nauczyciela (np. w plikach *filmy_6.txt*, *kina.txt*, *seanse.txt*). Utwórz związek między tabelami tak, żeby atrybuty *ID_kina* i *ID_filmu* w tabeli *seanse* wskazywały odpowiednio na: *IDK* z tabeli *kina* oraz *IDF* z tabeli *filmy*. Napisz zapytanie wyświetlające zestawienie tytułów filmów, które będą grane w kinie Luna. Dane posortuj alfabetycznie według tytułów filmów.
- *** **7** Utwórz nową bazę danych o nazwie *matura*. Ze strony Głównego Urzędu Statystycznego (stat.gov.pl) pobierz plik CSV zawierający informacje o średnich wynikach egzaminu maturalnego. Zaimportuj plik do bazy danych. Napisz zapytanie wyświetlające te egzaminy maturalne, których średni wynik zawierał się między 70% a 100%.

18. Tworzenie aplikacji korzystającej z sieciowej bazy danych

Jeśli znasz języki SQL oraz HTML, już tylko krok dzieli cię od stworzenia własnej aplikacji internetowej. Tym krokiem jest poznanie podstaw języka PHP, który połączy to, co widać w przeglądarce, z tym, co jest niewidoczne, czyli bazą danych. W tym temacie nauczysz się tworzyć aplikację, która pozwoli na gromadzenie informacji i zarządzanie nimi.

Cele lekcji

- Poznasz etapy projektowania aplikacji internetowej.
- Zaprojektujesz wygląd witryny, wykorzystując znaczniki języka HTML.
- Połączysz interfejs użytkownika z bazą danych dzięki instrukcjom języka PHP.

W praktyce zawodowej duże projekty informatyczne są przedsięwzięciem, które wymaga zaangażowania zespołu wykwalifikowanych fachowców. Wiele czasu poświęca się na dokładne przeanalizowanie potrzeb klienta, aby produkt jak najlepiej spełniał swoje funkcje. Ważnym etapem jest również testowanie i udoskonalanie stworzonej aplikacji. My jednak proces tworzenia aplikacji internetowej sprowadzimy do możliwie prostego i użytecznego przykładu, który pokaże, jak powstają tego typu programy.

18.1. Projektowanie aplikacji internetowej

Aplikacja działająca w przeglądarce internetowej jest zbudowana z wielu elementów. Dlatego jej przygotowanie wymaga ułożenia i wdrożenia planu działania. Wyróżnimy następujące etapy pracy nad aplikacją:

1. ustalenie sposobu działania aplikacji,
2. wybór technologii, konfiguracja oprogramowania, przygotowanie bazy danych,
3. wykonanie interfejsu użytkownika i stworzenie oprogramowania,
4. zapewnienie komunikacji serwisu z bazą danych.

Realizację pierwszego punktu warto rozpocząć od ustalenia szczegółów pomysłu. My wykorzystamy bazę danych z poprzednich dwóch tematów. Zawiera ona listy gatunków drzew i ulic miasta oraz informacje o tym, ile jakich drzew rośnie przy poszczególnych ulicach. Stworzymy aplikację dla pracowników zarządu zieleni miejskiej. Główną funkcją tej aplikacji będzie możliwość dodawania drzew i ulic do bazy danych oraz wyświetlania aktualnych list ulic i drzew.

Dobra rada

Kiedy projektujesz aplikację, postaraj się jak najdokładniej określić założenia projektu. Opisz możliwe funkcjonalności. Możesz nawet na początku naszkicować projekt na kartce. Dzięki temu będziesz wiedzieć, co dokładnie trzeba zrobić w kolejnych etapach pracy.

Back-end i front-end

Aplikacje internetowe składają się z dwóch głównych części: back-end i front-end. Od ich prawidłowej współpracy zależy sprawne, bezpieczne i poprawne funkcjonowanie wielu serwisów, z których korzystamy na co dzień.

Back-end

Technologie, których nie widzi użytkownik. Do back-endu zalicza się system zarządzania bazą danych zainstalowany na serwerze oraz skrypty wymieniające informacje między bazą danych a użytkownikiem.

Front-end

Technologie, które widzi i z których aktywnie korzysta użytkownik. Front-end odpowiada za wygląd strony internetowej, na której pracuje aplikacja bazodanowa, oraz za funkcjonalności serwisu, z którymi bezpośrednio styka się użytkownik.



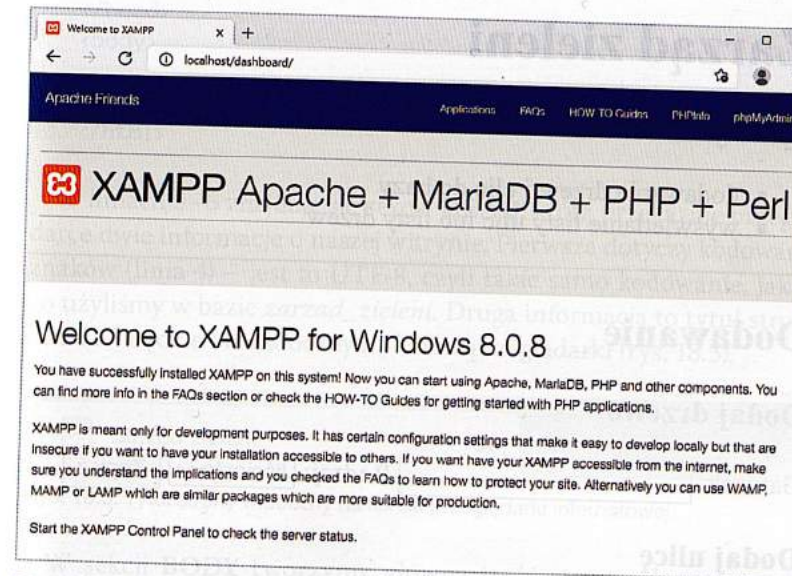
Wybór technologii i konfiguracja oprogramowania

W budowie aplikacji internetowych można wyróżnić dwie części. Pierwsza z nich to **back-end**, czyli elementy niewidoczne dla użytkownika aplikacji. Drugą częścią jest **front-end**, który obejmuje to, z czego użytkownik aktywnie korzysta.

W naszym przypadku back-end to baza danych oraz skrypty pośredniczące w komunikacji między bazą a interfejsem użytkownika. Za front-end natomiast uznamy interfejs aplikacji i oprogramowanie odpowiedzialne za interakcję z nią.

Baza danych na potrzeby naszego projektu jest już gotowa. Do jej utworzenia wykorzystaliśmy **język SQL** oraz narzędzia dostępne w pakiecie XAMPP. Zanim przejdziemy do kolejnych etapów, otworzymy panel kontrolny tego pakietu (XAMPP Control Panel) i włączymy serwer stron internetowych (serwer webowy) Apache oraz serwer baz danych MySQL. Sprawdzimy, czy serwer internetowy jest uruchomiony, wpisując w przeglądarce internetowej adres *localhost* (rys. 18.1).

Język SQL,
s. 289



Rys. 18.1. Aktywny serwer internetowy Apache

Oba te serwery są niezbędne do działania aplikacji. Serwer Apache będzie odpowiadał za wyświetlanie jej interfejsu, a serwer MySQL pozwoli na obsługę **zapytań** do bazy danych wysyłanych przez serwer webowy.

Potrzebujemy jeszcze folderu, w którym umieścimy pliki aplikacji. Nazwiemy go *zarzad_zieleni* i zapiszemy w katalogu *htdocs*, który znajduje się w miejscu zainstalowania pakietu XAMPP. Domyślnie jest to folder *xampp* na głównym dysku twardym naszego komputera. Pełna ścieżka utworzonego folderu wygląda więc następująco: *C:\xampp\htdocs\zarzad_zieleni*.

Zapytanie do bazy danych,
s. 290

Warto wiedzieć

Jeśli w folderze z plikami aplikacji znajduje się też plik *index.php*, to ma on wyższy priorytet niż *index.html* i jest znajdowany jako pierwszy po wpisaniu adresu w przeglądarce internetowej.

Język HTML, podręcznik *Informatyka na czasie 1. Zakres rozszerzony*, s. 230–244 [↗](#)
Język CSS, podręcznik *Informatyka na czasie 1. Zakres rozszerzony*, s. 245–248 [↗](#)

Jeśli teraz w pasku adresu przeglądarki internetowej wpisujemy adres *localhost/zarząd_zieleni*, przeglądarka poszuka w tym folderze pliku *index.html* i jeśli go znajdzie, wyświetli jego zawartość jako stronę główną serwisu. Dlatego w folderze *zarząd_zieleni* utworzymy również plik *index.html*.

18.2. Realizacja oprogramowania po stronie front-end

Widoczne w aplikacji teksty i obiekty wstawimy za pomocą znaczników **języka HTML**. Nie będziemy się tu skupiać na wyglądzie strony WWW, ponieważ najważniejszym celem jest zaprogramowanie elementów współpracujących z sieciową bazą danych. Można jednak dowolnie zmodyfikować interfejs, wykorzystując możliwości **języka CSS**, aby uatrakcyjnić aplikację.

Wygląd naszej aplikacji, czyli jej interfejs, przedstawia rysunek 18.2.

Zarząd zieleni

Aplikacja umożliwia:

- dodawanie drzew i ulic do bazy
- wyświetlanie listy ulic lub listy drzew

Dodawanie

Dodaj drzewo

Gatunek: Rodzaj: liściaste ▾ Zapisz

Dodaj ulicę

Ulica: Zapisz

Wyświetlanie

Wyświetl listę drzew lub ulic

Wybierz, co chcesz wyświetlić: drzewa ▾ Wyświetl

Rys. 18.2. Interfejs aplikacji do zarządzania informacjami w bazie danych zarządu zieleni

Interfejs aplikacji

Do utworzenia pliku HTML można użyć dowolnego edytora tekstu, np. systemowego Notatnika. Najlepiej jednak wykorzystać program, który będzie kolorował znaczniki języka HTML, taki jak Notepad++. Po jego uruchomieniu zapisujemy w utworzonym wcześniej folderze plik o nazwie *index.html*. Plik na razie jest pusty, jednak dzięki dodaniu rozszerzenia *html* program automatycznie włączy kolorowanie składni. Jeśli tak się nie stanie, możemy ręcznie włączyć kolorowanie, wybierając język HTML z menu **Składnia**.

Zacniemy od utworzenia podstawowej struktury dokumentu HTML.

```

1. <!DOCTYPE html>
2. <html lang="pl">
3. <head>
4.     <meta charset="UTF-8" />
5.     <title>Zarząd zieleni</title>
6. </head>
7. <body>
8.
9. </body>
10. </html>

```

W liniach 3–6 znajduje się sekcja **HEAD**, która przekazuje przeglądarce dwie informacje o naszej witrynie. Pierwsza dotyczy kodowania znaków (linia 4) – jest to UTF-8, czyli takie samo kodowanie, jakiego użyliśmy w bazie *zarząd_zieleni*. Druga informacja to tytuł strony (linia 5). Będzie on widoczny na karcie przeglądarki (rys. 18.3).



Rys. 18.3. Tytuł strony widoczny na karcie przeglądarki internetowej

W sekcji **BODY** tworzymy główną treść strony. Zacniemy od wyświetlenia nazwy i opisu aplikacji.

```

7. <body>
8. <h1>Zarząd zieleni</h1>
9.     <p>Aplikacja umożliwia:</p>
10.     <ul>
11.         <li>dodawanie drzew i ulic do bazy</li>
12.         <li>wyświetlanie listy ulic lub listy drzew</li>
13.     </ul>
14.     <hr />
15. </body>

```

Dobra rada

W ustawieniach programu Notepad++ przejdź do kategorii **Autouzupełnianie** i zaznacz autowstawianie i zaznaczenie tagów zamykających języka HTML. Usprawni ci to pracę – kiedy wpiszesz jakiś znacznik otwierający, program automatycznie wstawi jego zamykający odpowiednik.

W linii 8 tworzymy nagłówek pierwszego poziomu, w którym wpisujemy nazwę aplikacji. W liniach 9–13 dodajemy akapit (znaczniki `<p>` i `</p>`) i listę wypunktowaną (`` i ``), aby opisać działanie aplikacji. Ten fragment kodu kończymy w linii 14 znacznikiem `<hr />`, który zamyka tę część strony poziomą linią (rys. 18.4).

Zarząd zieleni

Aplikacja umożliwia:

- dodawanie drzew i ulic do bazy
- wyświetlanie listy ulic lub listy drzew

Rys. 18.4. Pierwsza część strony z nazwą aplikacji i opisem jej działania

Formularze

Teraz dodamy pierwszy formularz, który pozwoli użytkownikowi na dodanie drzewa do bazy. Poniższy kod umieszczamy po znaczniku `<hr>`.

```
15. <h2>Dodawanie</h2>
16. <h3>Dodaj drzewo</h3>
17. <form action="dodaj_drzewo.php" method="POST">
18.   Gatunek:
19.   <input type="text" name="gatunek" maxlength="50" />
20.   Rodzaj:
21.   <select name="rodzaj">
22.     <option>liściaste</option>
23.     <option>iglaste</option>
24.   </select>
25.   <input type="submit" value="Zapisz" />
26. </form>
```

Warto wiedzieć

Oprócz pól tekstowych, list rozwijanych i przycisków formularz może zawierać inne elementy, np. pola wyboru czy przyciski opcji. Formularz może też służyć do przesyłania plików na serwer.

Przed pierwszym formularzem dodajemy dwa nagłówki: drugiego stopnia (linia 15) i trzeciego stopnia (linia 16). Oznaczamy w ten sposób, do czego służy ta część aplikacji. W linii 17 od znacznika `<form>` zaczynamy formularz.

Pierwszym elementem formularza jest pole tekstowe – za jego wstawienie odpowiada znacznik `<input>` w linii 19. Drugim elementem jest lista rozwijana (linie 21–24). Trzecim jest przycisk typu `submit`, którego naciśnięcie wysyła dane z formularza do pliku wskazanego za pomocą atrybutu `action` w znaczniku `<form>` (linia 17). Znacznik `<form>` ma jeszcze atrybut `method`, który wskazuje sposób przesyłania danych do pliku. Jego wartość ustawiona na `POST` oznacza, że dane nie zostaną dopisane do adresu URL i ich przekazanie nie będzie widoczne dla użytkownika.

Drugim sposobem przesyłania danych z formularza jest metoda `GET`, która dane te dodaje na końcu adresu podanego w atrybucie `action`. Tę metodę wykorzystuje się np. w wyszukiwarkach internetowych.

Elementy formularza mają swoje unikalne nazwy, nadane za pomocą atrybutu `name`. Zostaną one wykorzystane przez plik PHP do jednoznacznego określenia, jaka informacja została przesłana.

Stworzymy teraz kolejny formularz, którego zadaniem będzie dodawanie nowych ulic do bazy danych.

```
27. <h3>Dodaj ulicę</h3>
28. <form action="dodaj_ulice.php" method="POST">
29.   Ulica:
30.   <input type="text" name="ulica" maxlength="50" />
31.   <input type="submit" value="Zapisz" />
32. </form>
33. <hr />
```

Formularz ten jest bardzo podobny do poprzedniego. Składa się z pola tekstowego i przycisku `Zapisz`. Po jego naciśnięciu dane są przesyłane do pliku `dodaj_ulice.php` (linia 28). Na końcu formularza wstawiamy poziomą linię za pomocą znacznika `<hr />` w linii 33 (rys. 18.5).

Dodawanie

Dodaj drzewo

Gatunek: Rodzaj:

Dodaj ulicę

Ulica:

Rys. 18.5. Formularze pozwalające na dodawanie drzew i ulic do bazy danych

W ostatniej części dokumentu `index.html` utworzymy formularz pozwalający określić, jakie dane chcemy wyświetlić. Z listy rozwijanej będziemy mogli wybrać, czy mają to być drzewa czy ulice.

Rys. 18.6. Lista rozwijana umożliwiająca wyświetlenie drzew lub ulic

Następnie za pomocą przycisku `Wyświetl` wyślemy nasz wybór do pliku `wyswietl.php`.

Warto wiedzieć

Pole tekstowe formularza pozwala przesłać praktycznie każdy tekst, w tym zapytanie SQL. Wprawny użytkownik może więc przesłać zapytanie usuwające rekordy, tabele lub nawet bazę danych (atak `SQL injection`). Aby się przed tym uchronić, warto zablokować niektóre uprawnienia do bazy danych i zastosować funkcję zabezpieczającą języka PHP, np. `addslashes`.

```

34. <h2>Wyświetlanie</h2>
35. <h3>Wyświetl listę drzew lub ulic</h3>
36. <form action="wyswietl.php" method="POST">
37.     Wybierz, co chcesz wyświetlić:
38.     <select name="lista">
39.         <option>drzewa</option>
40.         <option>ulice</option>
41.     </select>
42.     <input type="submit" value="Wyświetl" />
43. </form>

```

W linii 38 definiujemy listę rozwijaną, w liniach 39 i 40 – elementy tej listy, a w linii 42 – przycisk typu `submit`, wysyłający dane (rys. 18.7).

Wyświetlanie

Wyświetl listę drzew lub ulic

Wybierz, co chcesz wyświetlić:

Rys. 18.7. Formularz odpowiedzialny za wybranie, co ma zostać wyświetlone

Po zapisaniu zmian w pliku otwieramy przeglądarkę internetową i wpisujemy w niej adres `localhost/zarząd_zieleni`. Zobaczymy utworzoną właśnie stronę internetową z formularzami. Jeśli strona się nie wyświetliła, sprawdźmy, czy plik `index.html` znajduje się w folderze `zarząd_zieleni` oraz czy uruchomiony został serwer Apache.

Ćwiczenie 1

Otwórz panel kontrolny XAMPP i uruchom serwery Apache oraz MySQL. Utwórz folder `zarząd_zieleni` na dysku głównym swojego komputera w folderze `C:\xampp\htdocs`. W dowolnym edytorze tekstu za pomocą znaczników HTML utwórz interfejs aplikacji taki jak na rysunku 18.2. Plik zapisz pod nazwą `index.html` w folderze `zarząd_zieleni`.

Zapamiętaj

Współczesne strony internetowe to najczęściej aplikacje działające w przeglądarce i składające się z dwóch głównych części: technologii odpowiedzialnych za interfejs użytkownika (front-end) i technologii ukrytych przed wzrokiem użytkownika (back-end). Wygląd aplikacji projektuje się za pomocą znaczników HTML oraz CSS.

Dobra rada

Jeśli zechcesz utworzyć kolejną aplikację, po prostu załóż dla niej nowy folder w katalogu `C:\xampp\htdocs`. Pamiętaj, aby w tym folderze umieścić plik strony głównej `index.html` lub `index.php`.

18.3. Realizacja oprogramowania po stronie back-end

Po utworzeniu interfejsu aplikacji czas zająć się jej silnikiem, czyli elementami, które odpowiadają za prawidłowe działanie programu oraz komunikację z bazą danych.

W pliku HTML utworzyliśmy trzy formularze i określiliśmy w nich pliki, do których mają być przekazywane dane. Są to kolejno:

- ▶ `dodaj_drzewo.php` – po wprowadzeniu do formularza gatunku i rodzaju nowego drzewa skrypt z tego pliku doda je jako nowy rekord w tabeli drzewa,
- ▶ `dodaj_ulice.php` – po wprowadzeniu do formularza nazwy ulicy skrypt z tego pliku doda ją jako nowy rekord w tabeli ulice,
- ▶ `wyswietl.php` – po wybraniu z listy rozwijanej słowa `drzewa` lub `ulice` skrypt z tego pliku wyświetli rekordy z odpowiedniej tabeli.

Powyższe trzy pliki utworzymy, wykorzystując możliwości języka PHP. Jest to skryptowy język programowania, czyli taki, którego kod źródłowy zawiera skrypty wykonywane po stronie serwera. W chwili wywołania skryptu z pliku PHP interpreter zainstalowany na serwerze internetowym na bieżąco wykonuje kod i zwraca użytkownikowi wyniki jego działania. W naszym przypadku skrypty w plikach PHP będą przyjmowały dane z formularzy umieszczonych w pliku `index.html`. Dzięki zapytaniom SQL zawartym w skryptach PHP przyjęte dane będą zapisywane w bazie lub zostaną wyświetlone rekordy z wybranej tabeli.

Podobnie jak w przypadku pliku HTML, trzy pliki PHP utworzymy w folderze `zarząd_zieleni`. Zaczniemy od pliku `dodaj_drzewo.php`. Wykorzystamy w nim również znaczniki HTML. Plik będzie zawierał podstawową strukturę dokumentu HTML, a w niej – w sekcji **BODY** – znacznik otwierający `<?php` i zamykający `?>`. Wszystko, co znajdzie się między tymi znacznikami, serwer internetowy przekaże do interpretera PHP, aby ten wykonał kod i zwrócił wynik.

Przyjmowanie danych z formularza

Powiedzieliśmy wcześniej, że formularz przyjmuje dane od użytkownika, a po naciśnięciu przycisku typu `submit` przekazuje je do pliku docelowego PHP. W przypadku formularza służącego do dodawania drzewa były to dwie informacje: gatunek i rodzaj. Tak też nazwalimy pola formularza przyjmującego te dane – odpowiada za to ich atrybut `name`.

W pliku PHP utworzymy dwie zmienne, które posłużą do przechowywania wartości przyjętych z pól formularza:

1. `$gatunek = $_POST["gatunek"];`
2. `$rodzaj = $_POST["rodzaj"];`

Warto wiedzieć

Silnik to potoczna nazwa niewidocznych elementów aplikacji. Można je porównać do silnika samochodu, który również jest ukryty, ale stanowi najważniejszy element odpowiedzialny za napęd pojazdu.

Język PHP

Dobra rada

W plikach PHP możesz używać dowolnych znaczników HTML, a także tworzyć linki do plików CSS. W ten sposób możesz zmieniać wygląd strony zawierającej skrypty.

Nazwy zmiennych w języku PHP poprzedzamy znakiem dolara. Następnie po znaku równości przypisujemy tym zmiennym wartości. W naszym przypadku wskazujemy, że pochodzą one z metody POST zastosowanej w formularzu HTML. W nawiasach kwadratowych i cudzysłowach wpisujemy nazwy pól formularza, z których tworzone zmienne mają pobrać wpisane wartości. Każdą instrukcję języka PHP kończymy średnikiem.

Połączenie z bazą danych

Kiedy wartości z formularza zostaną przypisane do zmiennych, należy wskazać, co interpreter ma z nimi zrobić. W naszym przypadku wartości te zostaną przekazane do bazy danych. Jednak zanim to nastąpi, skrypt musi nawiązać połączenie z bazą danych. W tym celu stworzymy kolejną zmienną, która będzie nowym **obiekt** należącym do **klasy** `mysqli`. Dzięki niej będziemy mogli korzystać z języka SQL w skrypcie PHP.

Obiekt,
s. 226

Klasa `mysqli`

```
3. $polaczenie = new mysqli("localhost", "admin",
4. "admin1Str@t0r", "zarzad_zieleni");
```

Atrybut,
s. 226

W nawiasach wpisujemy **atrybuty** nowego obiektu. Są to kolejno:

- ▶ nazwa i adres serwera – w naszym przypadku jest to nasz komputer, dlatego wpisujemy `localhost`,
- ▶ użytkownik – wykorzystamy utworzone w temacie 16 konto użytkownika `admin`,
- ▶ hasło – wpisujemy hasło przypisane użytkownikowi `admin`,
- ▶ nazwa bazy danych – w naszym przypadku jest to `zarzad_zieleni`.

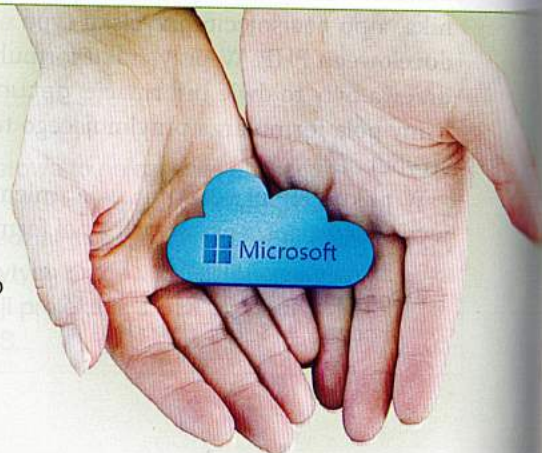
Dodawanie rekordu do bazy danych

Po przypisaniu do zmiennych informacji pochodzących z formularza i połączeniu skryptu z bazą danych pozostaje jeszcze wpisanie tych danych jako nowego rekordu w bazie.

A to ciekawe

System operacyjny w chmurze

Coraz więcej programów komputerowych jest przenoszonych do chmury. W 2021 r. firma Microsoft zaprezentowała system Windows działający w chmurze i obsługiwany przez przeglądarkę internetową z dowolnego miejsca na świecie. Każdy użytkownik takiego systemu może go skonfigurować według własnych preferencji i przechowywać w nim swoje zasoby.



```
5. $polaczenie->query
6. ("INSERT INTO
7. drzewa(Gatunek, Rodzaj)
8. VALUES
9. ('$gatunek', '$rodzaj')");
```

W linii 5 wywołujemy **metodę** `query` obiektu `$polaczenie`. Jej zadaniem jest wykonanie **zapytania do bazy danych**. Zapytanie umieszczamy jako parametr metody `query` w nawiasach okrągłych i w cudzysłowie. Nasze zapytanie wykorzystuje **klauzulę** `INSERT INTO` (linia 6), aby atrybutom `Gatunek` i `Rodzaj` w tabeli `drzewa` przypisać wartości (linia 8) znajdujące się w zmiennych `$gatunek` i `$rodzaj` (linia 9).

Metoda `query`

Zapytanie do bazy danych,
s. 290

Klauzula,
s. 291

Zamykanie połączenia i wyświetlenie komunikatu

Po wykonaniu zapytania należy zamknąć aktywne połączenie z bazą danych. W tym celu w 10 linii pliku wywołujemy **metodę** `close` obiektu `$polaczenie`.

Metoda `close`

```
10. $polaczenie->close();
```

Metoda `close` nie przyjmuje żadnych parametrów.

Skrypt PHP powinien już działać prawidłowo, czyli dodawać nowy rekord do bazy danych. Jednak aby aplikacja była kompletna, dopiszemy jeszcze komunikat dla użytkownika i wstawimy hiperłącze umożliwiające powrót do strony głównej.

```
11. echo "Dodano drzewo $rodzaj $gatunek<br />";
12. echo "<a href='index.html'>Wróć</a>";
```

Do wyświetlania napisów w przeglądarce używamy w języku PHP **słowa kluczowego** `echo`. Po nim w cudzysłowie wpisujemy frazę, która ma zostać wyświetlona. Zauważ, że możemy również wyświetlać aktualne wartości zmiennych – w tym celu po prostu podajemy ich nazwy (linia 11) – oraz wykorzystywać znaczniki HTML, np. tworzące hiperłącza (linia 12).

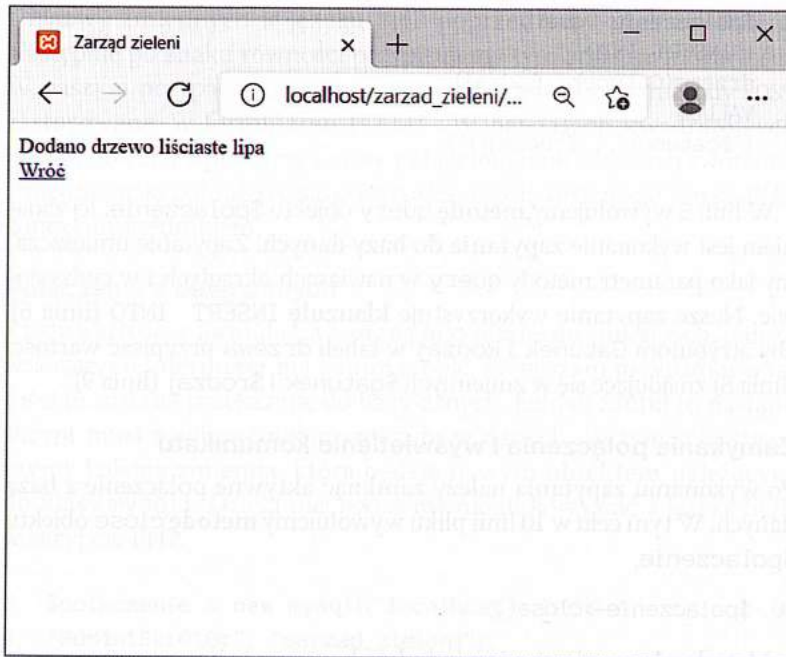
Słowo kluczowe `echo`

Testowanie aplikacji

Po zapisaniu zmian w pliku `dodaj_drzewo.php` możemy już przetestować działanie gotowej części aplikacji. W przeglądarce internetowej wpisujemy adres `localhost/zarzad_zieleni`, aby uruchomić aplikację. W pierwszym formularzu widocznym na stronie wprowadzamy nazwę nowego drzewa i wybieramy, jakiego jest rodzaju. Zatwierdzamy przyciskiem **Zapisz** i zostajemy przeniesieni do pliku `dodaj_drzewo.php`. Skrypt zawarty w tym pliku doda nowe drzewo do tabeli bazy danych i wyświetli odpowiedni komunikat (rys. 18.8, s. 332).

Dobra rada

Zanim przejdziesz do kolejnego etapu tworzenia aplikacji, możesz sprawdzić, czy skrypt na pewno dodał drzewo do bazy. W tym celu wykorzystaj zapytanie do bazy danych – napisz je w `phpMyAdmin` lub użyj Wiersza poleceń.



Rys. 18.8. Okno przeglądarki z komunikatem potwierdzającym dodanie nowego drzewa

Warto wiedzieć

Programiści umieszczają w pliku PHP również kod, który ma zadziałać w razie błędu (np. braku połączenia z bazą danych). Najczęściej taki kod przenosi na inną stronę w serwisie.

Ćwiczenie 2

W dowolnym edytorze tekstu, korzystając z instrukcji języka PHP, utwórz skrypt:

- dodający drzewo do bazy danych (plik zapisz pod nazwą *dodaj_drzewo.php* w folderze *zarząd_zieleni*),
- dodający ulicę do bazy danych (plik zapisz pod nazwą *dodaj_ulice.php* w folderze *zarząd_zieleni*).

Zapamiętaj

Za pomocą instrukcji języka PHP można tworzyć skrypty przekazujące informacje do bazy danych. Skrypty są wykonywane po stronie serwera.

Wyświetlanie danych

Skrypt dodający nowy rekord do tabeli ulice jest bardzo podobny do skryptu dodającego rekord do tabeli drzewa, dlatego nie będziemy go omawiać. Przejdziemy do ostatniego pliku PHP, czyli *wyswietl.php*.

Plik przyjmuje z formularza HTML jedną wartość, zapisaną pod nazwą *lista*. W znany już sposób przypiszemy tę wartość do zmiennej i utworzymy połączenie z bazą danych.

- `$lista = $_POST["lista"];`
- `$polaczenie = new mysqli("localhost",`
- `"admin", "@dm1n1Str@t0r", "zarząd_zieleni");`

Zmienna `$lista` może przyjąć jedną z dwóch wartości: drzewa lub ulice. Dzieje się tak, ponieważ w formularzu źródłowym użyliśmy listy rozwijanej zawierającej tylko te dwie opcje. W zależności od tego, jaka to będzie wartość, powinny zostać wyświetlone inne dane. Do wyboru ścieżki działania wykorzystamy **instrukcję warunkową if**.

• Instrukcja warunkowa if

- `if($lista == "drzewa"){`
- `$wyswietl = $polaczenie->query("SELECT`
- `Gatunek, Rodzaj FROM drzewa");`
- `while($wiersz = $wyswietl->fetch_array()){`
- `echo "Gatunek: ".$wiersz["Gatunek"]." Rodzaj: "`
- `$wiersz["Rodzaj"]."
";`
- `}`
- `}`

W nawiasie po instrukcji **if** sprawdzamy, czy zmienna `$lista` ma wartość drzewa. Jeśli warunek jest spełniony, wykonywane są instrukcje w nawiasach klamrowych.

W linii 5 dla zmiennej (obiektu) `$polaczenie` wywoływana jest metoda `query`, wysyłająca zapytanie do bazy danych. Zapytanie ma zwrócić wartości atrybutów `Gatunek` i `Rodzaj` z tabeli drzewa, a wynik ma zostać przypisany do zmiennej tablicowej `$wyswietl`.

W linii 7 wykorzystujemy **pętlę while**, aby wyświetlić kolejne rekordy z tabeli drzewa. W warunku działania pętli widzimy przypisanie do zmiennej `$wiersz` wyniku działania **metody** `fetch_array`, wywołanej dla zmiennej tablicowej `$wyswietl`. Metoda `fetch_array` pobiera kolejne wartości ze zmiennej `$wyswietl` i dzięki instrukcji zawartej w pętli wyświetla wartości indeksów `Gatunek` i `Rodzaj`. Instrukcja w pętli będzie powtarzana, dopóki istnieją jakiegokolwiek nieodczytane wartości.

Warto wiedzieć

Język PHP jest bardzo podobny do języka C++. Korzysta z takich samych operatorów i obowiązują w nim podobne reguły, np. średnik na końcu każdej instrukcji.

• Pętla while

• Metoda `fetch_array`

Zmienna `$wiersz` jest interpretowana jako zmienna tablicowa o dwóch polach, ponieważ może przyjmować dwie wartości. Indeksy zmiennej tablicowej mają nazwy odpowiadające nazwom atrybutów tabeli (`Gatunek`, `Rodzaj`).

Prześledźmy jeszcze działanie instrukcji w pętli. Słowo kluczowe **echo** spowoduje wyświetlenie kolejnych wartości, które są połączone ze sobą za pomocą **operatora konkatencji**. W języku PHP operatorem tym jest **Operator konkatencji** kropka. Pierwsza pojawi się wartość pola z indeksem `Gatunek` zmiennej tablicowej `$wiersz`. Potem zostanie wyświetlona spacja, a następnie wartość pola z indeksem `Rodzaj` zmiennej tablicowej `$wiersz`. Na końcu zostanie wstawiony znacznik końca wiersza `
`.

• Operator konkatencji

Podobną instrukcję warunkową konstruujemy, aby wyświetlić nazwy ulic. Na końcu pliku należy zamknąć połączenie z bazą danych i wyświetlić hiperłącze pozwalające na powrót do strony głównej. Efekt działania formularza po wybraniu wartości „drzewa” jest widoczny na rysunku 18.9.



Rys. 18.9. Wynik działania skryptu z pliku `wyswietl.php`

Ćwiczenie 3

W dowolnym edytorze tekstu, korzystając z instrukcji języka PHP, utwórz skrypt pozwalający na wyświetlenie listy drzew lub listy ulic z bazy danych `zarząd_zieleni`. Plik zapisz pod nazwą `wyswietl.php` w folderze `zarząd_zieleni`.

Zapamiętaj

Język PHP jest podobny do innych interpretowanych języków programowania. Można w nim wykorzystywać instrukcje warunkowe oraz pętle. Pozwala on również na przyjmowanie danych z formularzy HTML oraz przekazywanie ich do bazy danych dzięki zapytaniom języka SQL.

Podsumowanie

- Przed wykonaniem aplikacji sieciowej należy wybrać środowisko do jej uruchomienia (serwery WWW i bazodanowe) oraz technologie do tworzenia strony WWW i skryptów (HTML, CSS, PHP).
- Za wygląd strony i oprogramowanie zadań wykonywanych przez przeglądarkę odpowiedzialna jest część aplikacji po stronie front-end.
- Za zadania polegające na komunikacji z bazą danych odpowiedzialna jest część aplikacji po stronie back-end.
- Dane do aplikacji internetowej są przesyłane za pośrednictwem formularzy metodą **POST** lub metodą **GET**.
- Skryptowy język programowania PHP wykorzystuje instrukcje znane z wielu innych języków programowania, m.in. instrukcje warunkowe i pętle.
- Jednym z kluczowych zadań w procesie tworzenia aplikacji jest jej testowanie oraz wprowadzanie poprawek i ulepszeń.

Zadania

- * **1** Wykorzystaj znaczniki HTML oraz CSS, aby uatrakcyjnić wygląd utworzonej w temacie aplikacji. Dodaj kolory, zmień krój czcionki, wyrównaj elementy na stronie.
- ** **2** Do aplikacji utworzonej w temacie dodaj skrypty umożliwiające przypisanie gatunków drzew do ulic oraz określenie liczby drzew przy każdej ulicy.
- ** **3** Do aplikacji utworzonej w temacie dodaj skrypty pozwalające na wyświetlanie informacji o tym, jakie gatunki drzew i w jakiej liczbie rosną przy wybranej ulicy.
- ** **4** Zmodyfikuj aplikację utworzoną w temacie, rozdzielając jej poszczególne funkcjonalności na oddzielne podstrony. Na stronie głównej umieść nazwę aplikacji i jej opis oraz menu, które pozwoli na poruszanie się między podstronami.
- *** **5** Utwórz aplikację, która pozwoli ci na zapisywanie swoich ocen z poszczególnych przedmiotów. Przygotuj bazę danych złożoną z przynajmniej dwóch tabel, zaprojektuj interfejs aplikacji oraz napisz skrypty umożliwiające komunikację z bazą danych.
- *** **6** Do serwera SQL zaimportuj bazę danych przekazaną przez nauczyciela (np. w pliku `artykuly_autorzy.accdb`). Baza zawiera informacje o artykułach na blog i ich autorach. Utwórz aplikację pozwalającą na dodawanie informacji do bazy danych oraz wyświetlanie listy artykułów i ich autorów.

Wskazówka: Data w programie Access domyślnie zapisywana jest w formacie 01.09.2021. Z kolei język SQL wymaga daty w formacie 2021-09-01. Aby po wyeksportowaniu danych daty w pliku tekstowym były w odpowiednim formacie, w kreatorze eksportu kliknij przycisk **Zaawansowane** i w otwartym oknie zmień **Kolejność dat** na **RMD** oraz **Ogranicznik daty** na **-**. Więcej informacji o funkcjach daty i czasu w języku SQL znajdziesz w dodatku 11 na s. 435.