

SCR - LAMP & PHP

Maksymilian Kadukowski ind. 248974

Łukasz Chwistek ind. 243662

10.05.20

1 Wstęp

1.1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie z podstawowymi zasadami instalacji i konfiguracją serwera WWW, bazy danych MySQL oraz parsera PHP. W celu wykonania ćwiczenia wykorzystano pakiet LAMP na systemie o jądrze linuxowym.

1.2 Język php

PHP jest skryptowym językiem programowania wykorzystywanym najczęściej do tworzenia stron internetowych. Po napotkaniu znacznika PHP, następuje analiza składniowa danych wejściowych z założeniem pewnej struktury tych danych i wyodrębnienie ich do postaci w której dane da się wykorzystać w oprogramowaniu. Skrypt wykonywany jest po stronie serwera, co oznacza, że jest niewidoczny dla użytkowników. Pozwala na m.in. dynamiczne generowanie strony, tworzenie i modyfikowanie plików na serwerze, ograniczanie dostępu do danych podstron na twojej stronie, szyfrowanie danych czy stosowanie ciasteczek. Kod PHP może być przeplatany z kodem HTML, jednak nie jest to dobra praktyka.

Najprostszym przykładem jego wykorzystania jest obsługa formularzy na stronach internetowych. PHP pozwala także na obsługę baz danych przy znajomości SQL.

W PHP najczęściej tworzone są systemy zarządzania treścią, systemy forów dyskusyjnych, aplikacje pocztowe oraz klienty baz danych.

1.3 MySQL

Bazy danych MySQL stanowią ważny element wielu witryn www, przechowując niezbędne do jej działania dane (np. konta użytkowników, w tym loginy i hasła).

MySQL to tzw. system zarządzania bazą danych DBMS (ang. Database Management System). Natomiast SQL (ang. Structured Query Language) to język, który służy do pisania zapytań do tej bazy (zapytania najczęściej umieszczamy wewnątrz skryptów PHP).

1.4 Apache

Apache HTTP serwer jest programem który pracuje w tle. Jego głównym zadaniem jest komunikacja poprzez sieć używając protokołu TCP/IP.

Przy uruchomieniu jego zachowanie jest ustawiane poprzez pliki konfiguracyjne. W stanie bezczynności słucha on adresów IP które ma podane w pliku httpd.conf. W wypadku gdy na którychś z nich przyjdzie zapytanie, analizuje znajdujące się w nim nagłówki i podąża za wytycznymi odpowiadającymi tym nagłówkom znajdującym się w pliku konfiguracyjnym.

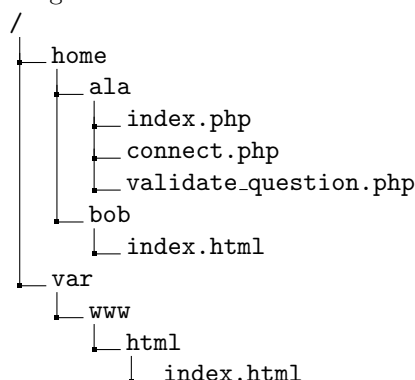
Dla zapytań statycznych (w formie URL) "tłumaczy" on to zapytanie na odpowiedni plik znajdujący się na jego dysku lokalnym i wysyła go przez internet klientowi użytkownika (zazwyczaj przeglądarce internetowej).

W przypadku zapytania dynamicznego uruchamia odpowiedni program z przesłanymi informacjami i po jego wykonaniu wysyła output tego programu do jednostki która to zapytanie wysłała. Jeśli w jakimś przypadku serwer nie będzie w stanie przerobić zapytania użytkownika to wyśle mu odpowiedni komunikat błędu.

2 Ćwiczenie

2.1 Hierarchia plików

Podstrony Ali i Boba znajdują się w ich folderach domowych, tak jak zostało to przedstawione na diagramie.



Dla poprawnego działania strony konieczne było utworzenia linków symbolicznych do folderów `/home/ala` oraz `/home/bob` w folderze `/var/www/html`.

Dodatkowo, pliki te mają wyłączone permisje czytania przez użytkowników zaliczanych jako **others**. Aby umożliwić dostęp do tych plików serwerowi Apache2 należało stworzyć specjalną grupę (np. `apache-server-group`) do której zostaną dodani użytkownicy `ala`, `bob` oraz specjalne konto `www-data`. W przypadku mojego systemu (Linux Mint 19.3) serwer Apache2 działa jako proces tego konta. Następnie użytkownicy `ala` i `bob` muszą zmienić grupę swoich plików `index.php/html` na wcześniej stworzoną.

Uporządkowanie permisji w ten sposób pozwala ukryć źródła stron przed innymi użytkownikami systemu, jednocześnie pozwalając serwerowi Apache2 na dostęp i udostępnienie już po uruchomieniu skryptów PHP.

2.2 MySQL

W bazie danych MySQL projektu znajduje się tylko jedna tablica zawierająca pytania do quizu. Oprócz obowiązkowej kolumny **id** będącego tzw. wartością kluczową znajdują się w niej 2 kolumny typu VARCHAR trzymające samo pytanie oraz odpowiedź.

2.3 Język php

2.3.1 Główna logika

Dla większej czytelności główna logika programu została rozbita na następujące funkcje znajdujące się w pliku `connect.php`:

- Funkcja `sum_array`

```
1 function sum_array($array)
2 {
3     $result = 0;
4
5     foreach($array as $val)
6     {
7         if ($val)
8             $result += 1;
9     }
10
11     return $result;
12 }
```

Zwraca ilość zmiennych o wartości `true` z tablicy `$array`.

- Funkcja `getRandomQuestionsIdx`

```
1 function getRandomQuestionsIdx($count, $no)
2 {
3     $array = array();
4
5     for ($i = 0; $i < $count; $i++)
6         array_push($array, false);
7
8     while (sum_array($array) != $no)
9         $array[rand(0, $count - 1)] = true;
10
11     $result = array();
12     for ($i = 0; $i < $count; $i++)
13         if ($array[$i])
14             array_push($result, $i);
15
16     shuffle($result);
17
18     return $result;
19 }
```

Zwraca tablicę o wielkości `$no` z losowymi i unikalnymi numerami całkowitymi z przedziału `[0, $count)`.

- Funkcja `getRandomQuiz`

```
1 function getRandomQuiz($noOfQuestions)
2 {
3     global $host;
4     global $db_user;
5     global $db_pass;
6     global $db_name;
7
8     $db_connection = @new mysqli($host, $db_user, $db_pass, $db_name);
9     $result = false;
10
11     if ($db_connection->connect_errno != 0)
12     {
13         echo "Error: " . $db_connection->connect_errno;
14     }
15     else
16     {
17         $sql_result = $db_connection->query("SELECT * FROM questions");
```

```

18         if (!$sql_result)
19         {
20             echo "Error: query not able to execute";
21         }
22         else if ($sql_result->num_rows == 0)
23         {
24             echo "Error: questions not found";
25         }
26         else
27         {
28             $questions = $sql_result->fetch_all(MYSQLI_BOTH);
29             $result = array();
30             $randIdx = getRandomQuestionsIdx(count($questions),
31 $noOfQuestions);
32             foreach ($randIdx as $idx)
33                 array_push($result, $questions[$idx]);
34             $sql_result->free();
35         }
36         $db_connection->close();
37
38         return $result;
39     }

```

Zwraca listę pytań w postaci tablic asocjacyjnych.

2.3.2 Pozostałe pliki

Reszta plików występująca w programie:

- */var/www/html/index.html*

W pliku znajduje się strona główna zawierająca łącza do stron domowych Ali oraz Boba. W pliku nie znajduje się żaden kod języka PHP.

- */home/ala/index.php*

Plik wyświetla dla użytkownika Quiz z losowo wybranymi pytaniami z bazy danych. Poszczególne pytania i pola na odpowiedź użytkownika wyświetlane są w odpowiednio wydzielonych polach.

W pliku w pierwszej kolejności uruchamiana jest sesja i definiowane są zmienne sesyjne w tablicy `$_SESSION`, gdzie określa się ilość wyświetlanych pytań oraz losuje pytania z przygotowanej bazy.

```

1 <?php
2 session_start();
3 require_once("connect.php");
4
5 $noOfQuestions = 5;
6 $_SESSION['q_array'] = getRandomQuiz($noOfQuestions);
7
8 function dumpQuestion($question, $idx)
9 {
10     echo '<tr><td>Pytanie:</td><td>' . $question['question'] . '</td></tr>';
11     echo '<tr><td>Odpowiedz:</td><td> <input type="text" name="answer' . $idx
12         . '" style="width: 100%;"></td></tr>';
13 }
14 ?>

```

Udzielone przez użytkownika odpowiedzi są przekazywane metodą POST do pliku `/var/www/html/validate_question.php`, który sprawdza ich poprawność, a następnie wywoływana jest strona podsumowująca jego wynik.

```
1 <form action="validate_question.php" method="POST">
2   <table border=0 cellpadding=10 >
3     <?php
4       for ($i = 0; $i < count($_SESSION['q_array']); $i++)
5       {
6         dumpQuestion($_SESSION['q_array'][$i], $i);
7         echo '<tr><td colspan=2><hr style="width: 100%;"></td></tr>';
8       }
9     ?>
10    <tr><td colspan=2><input type="submit" style="width: 100%;" value="
    Odpowiedz!"><td></tr>
11  </table>
12 </form>
```

- `/home/bob/index.html`

Plik zawiera stronę domową boba. W pliku nie znajduje się żaden kod języka PHP.

- `/var/www/html/connect.php`

Oprócz wcześniej wspomnianych funkcji w pliku zdefiniowane są również zmienne trzymające adres, nazwę, oraz dane logowania do bazy danych MySQL.

```
1 $host = "localhost";
2 $db_user = "root";
3 $db_pass = "password";
4 $db_name = "scrphp";
```

- `/var/www/html/validate_question.php`

Plik zajmuje się sprawdzeniem poprawności odpowiedzi udzielonych w quizie oraz wyświetlaniem wyniku. W pliku znajduje się, jeszcze przed jakimkolwiek elementem języka HTML, sprawdzenie poprawności sesji oraz metody POST.

```
1 <?php
2   function goToIndex()
3   {
4       header("Location: index.php");
5       exit();
6   }
7
8   session_start();
9
10  if (!isset($_SESSION['q_array']))
11      goToIndex();
12
13  $questionsSize = count($_SESSION['q_array']);
14  $nOfGoodAnswers = 0;
15
16  for ($i = 0; $i < $questionsSize; $i++)
17  {
18      if (!isset($_POST['answer' . $i]))
19          goToIndex();
20  }
```

```

21         if ($_POST['answer' . $i] === $_SESSION['q_array'][$i]['answer'])
22             $nOfGoodAnswers++;
23     }
24     ?>

```

Plik uruchamia sesję aby mieć dostęp do aktualnego zestawu pytań zdefiniowanego w poprzednim pliku `/home/ala/index.php`. Sprawdza, czy rzeczywiście dane pole w tablicy `$_SESSION` zostało zdefiniowane, i jeśli tak, to iteruje się po każdym pytaniu, jednocześnie sprawdzając czy w tablicy `$_POST` zostało zdefiniowane odpowiednie pole odpowiedzi.

Reszta pliku to banalne wyświetlenie wyniku w podstawowym szablonie HTML.

```

1  <!DOCTYPE HTML>
2  <html lang="pl">
3      <head>
4          <meta charset="UTF-8">
5          <title>scr-php</title>
6      </head>
7      <body>
8          Liczba dobrych odpowiedzi: <?php echo $nOfGoodAnswers; ?> <br>
9          Liczba złych odpowiedzi: <?php echo $questionsSize - $nOfGoodAnswers;
10         ?> <br>
11         Procent: <?php echo ($nOfGoodAnswers / $questionsSize) * 100 . "%";
12         ?> <br>
13         <br><br>
14         <a href="index.php">Powrot</a>
15     </body>
16 </html>

```

2.3.3 Połączenie z bazą danych MySQL

Projekt dynamicznie łączy się z bazą danych MySQL. Przez połączenie jest w stanie wyświetlić użytkownikowi losowe pytania z wcześniej wspomnianej bazy danych.

Podstawowe połączenie zastosowano w funkcji `getRandomQuiz()`. Tak jak zostało to przedstawione we wcześniejszej funkcji, podstawowe połączenie i zapytanie do bazy danych z języka PHP jest uzyskiwane przez użycie rozszerzenia `mysqli`.

Opis połączenia wykonywanego w programie:

1. Stworzenie nowego obiektu `mysqli` zajmującego się połączeniem z bazą danych. Parametry konstruktora to, jak widać, nazwa hosta, nazwa użytkownika oraz hasło i nazwa bazy danych.

Warto wspomnieć, że wcześniejsze użycia dyrektywy `global` pozwoliło na 'import' zmiennych z globalnej przestrzeni nazw do funkcji, co umożliwia jednolite i deskryptywne podanie argumentów.

```

1 $db_connection = @new mysqli($host, $db_user, $db_pass, $db_name);

```

2. Wypisanie numeru błędu w przypadku błędnego połączenia.

```

1 if ($db_connection->connect_errno != 0)
2 {
3     echo "Error: " . $db_connection->connect_errno;
4 }

```

3. Uruchomienie zapytania języka SQL.

```
1 $sql_result = $db_connection->query("SELECT * FROM questions");
```

4. Sprawdzenie poprawności wyniku zapytania. Pierwszy if informuje o poprawności samego zapytania i wywołą się np. dla niepoprawnej składni SQLa czy przez problem z połączeniem. Drugi if natomiast sprawdza, czy zapytanie w ogóle dało jakiekolwiek wyniki.

```
1 if (!$sql_result)
2 {
3     echo "Error: query not able to execute";
4 }
5 else if ($sql_result->num_rows == 0)
6 {
7     echo "Error: questions not found";
8 }
```

5. Konwersja wyniku zapytania do postaci listy tablic asocjacyjnych oraz wyciągnięcie losowych pytań do zmiennej \$result.

```
1 else
2 {
3     $questions = $sql_result->fetch_all(MYSQLI_BOTH);
4     $result = array();
5     $randIdx = getRandomQuestionsIdx(count($questions), $noOfQuestions);
6     foreach ($randIdx as $idx)
7         array_push($result, $questions[$idx]);
8 }
```

6. Uwolnienie pamięci oraz zwrócenie listy losowych pytań.

```
1     $sql_result->free();
2 }
3 $db_connection->close();
4
5 return $result;
6 }
7
```

3 Podsumowanie i wnioski

Ćwiczenie umożliwiło zapoznanie się z językiem PHP, bazą danych MySQL oraz serwer HTTP Apache. Ta wiedza wraz z poprzednimi ćwiczeniami (HTML, Javascript) umożliwia stworzenie w pełni działającego serwisu internetowego, od przysłowiowego back-endu do front-endu.