

Bazy danych 2022 - Laboratoria: Aplikacja

Bartosz Brzostowski

20 kwietnia 2022

1 Aplikacja w PHP

Do realizacji aplikacji w PHP (co jest zalecaną formą) polecam pakiet XAMPP <https://www.apachefriends.org/pl/index.html>. Jest to darmowa, wieloplatformowa (Windows, Linux i OS X) dystrybucja serwera Apache'a, bazy danych MariaDB i interpreterów dla skryptów napisanych w PHP i Perlu.

MariaDB to system zarządzania relacyjną bazą danych w dużej mierze zgodny z MySQL. System ten stworzony został przez współtwórców MySQL i autorzy dbają o utrzymanie kompatybilności z wcześniejszymi wersjami MySQL.

Oczywiście istnieje wiele innych pakietów zapewniających podobne możliwości. Przykładowo dla platformy Windows jest to EasyPHP <https://www.easyphp.org/>. Inna wieloplatformowa paczka to AMPPS <http://www.ampps.com/>. Proszę nie tracić czasu na wyszukiwanie kolejnych alternatywnych dystrybucji. Szczególnie, że dostępnych jest sporo mało aktualnych (kilkuletnie wersje stabilne) wersji oprogramowania.

Użytkowników systemów Linux-owych może kusić instalacja z pakietów poszczególnych potrzebnych składników. To rozwiązanie jest jednak dla trochę bardziej zaawansowanych, gdyż konfiguracja może przysporzyć nieoczekiwanych problemów.

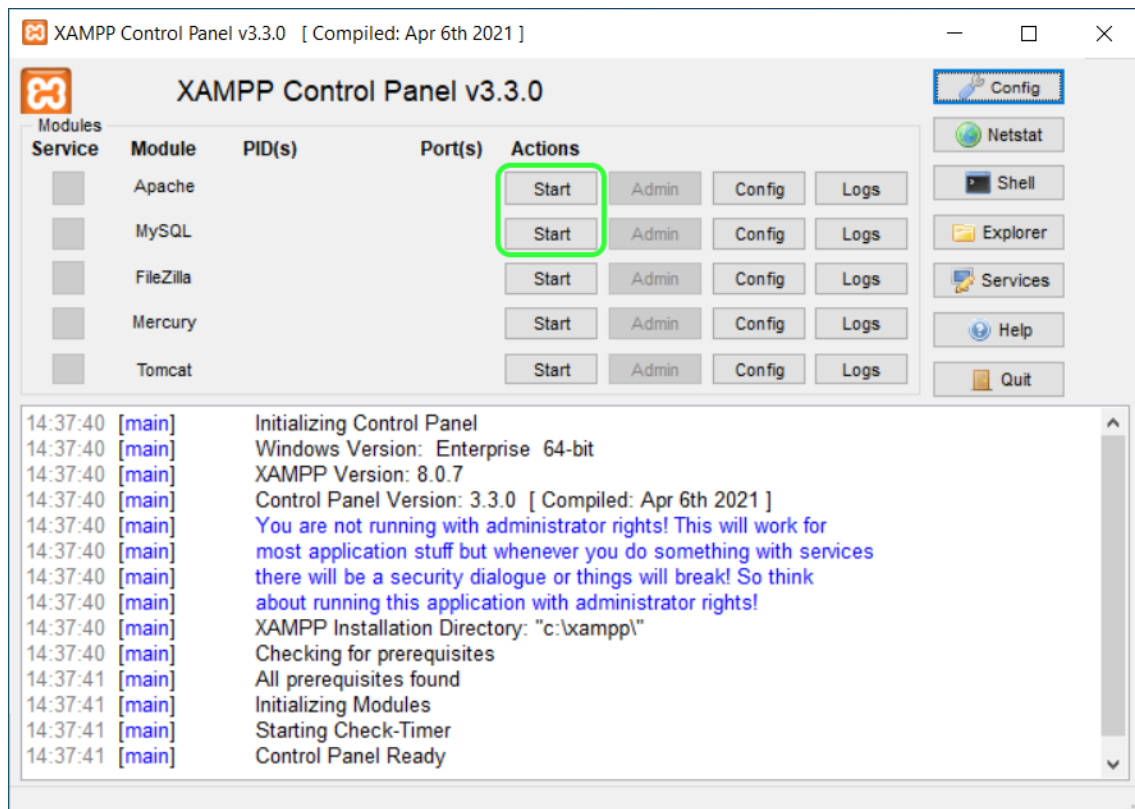
Dla osób zdecydowanych na oddzielną instalację komponentów – MySQL można pobrać ze strony <https://www.mysql.com/>.

W razie wątpliwości proszę kontaktować się z prowadzącym zajęcia.

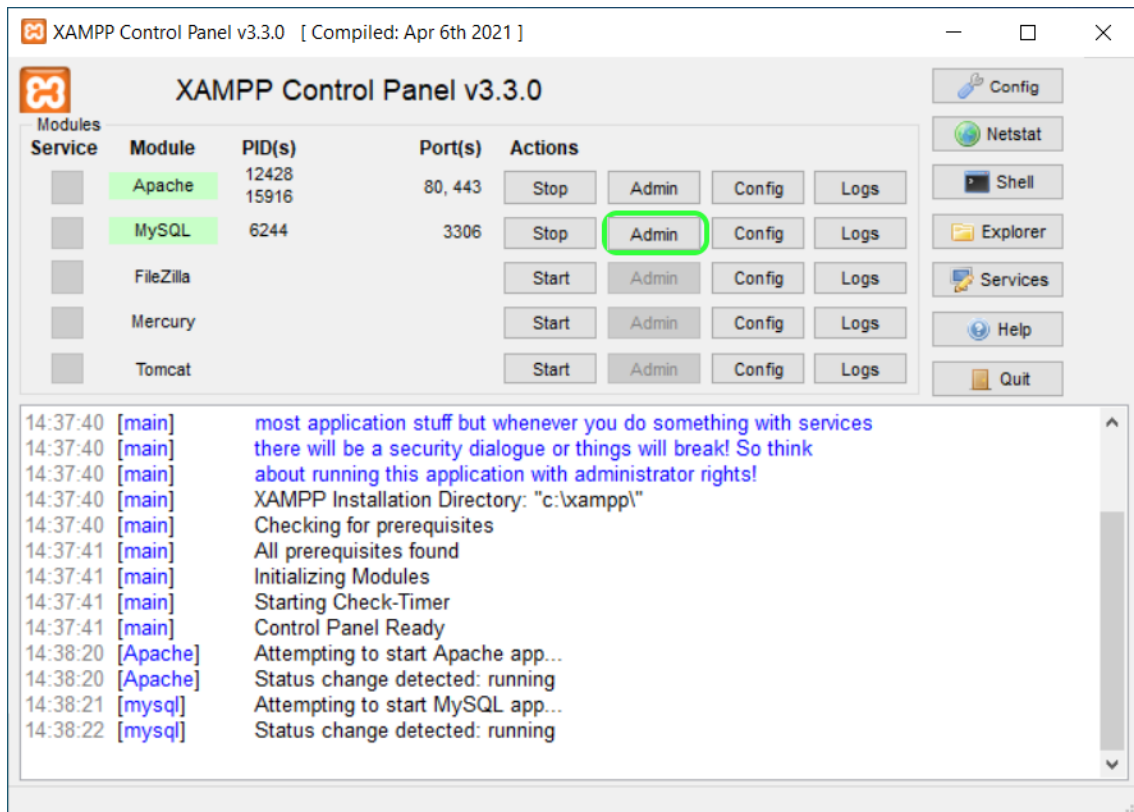
2 Korzystanie z XAMPP

W przypadku korzystania z PHPMyAdmin z pakietu XAMPP obsługa jest dość intuicyjna. Moja sugestia aby używać tego pakietu wynikała z tego, że jest dostępny na wiele platform.

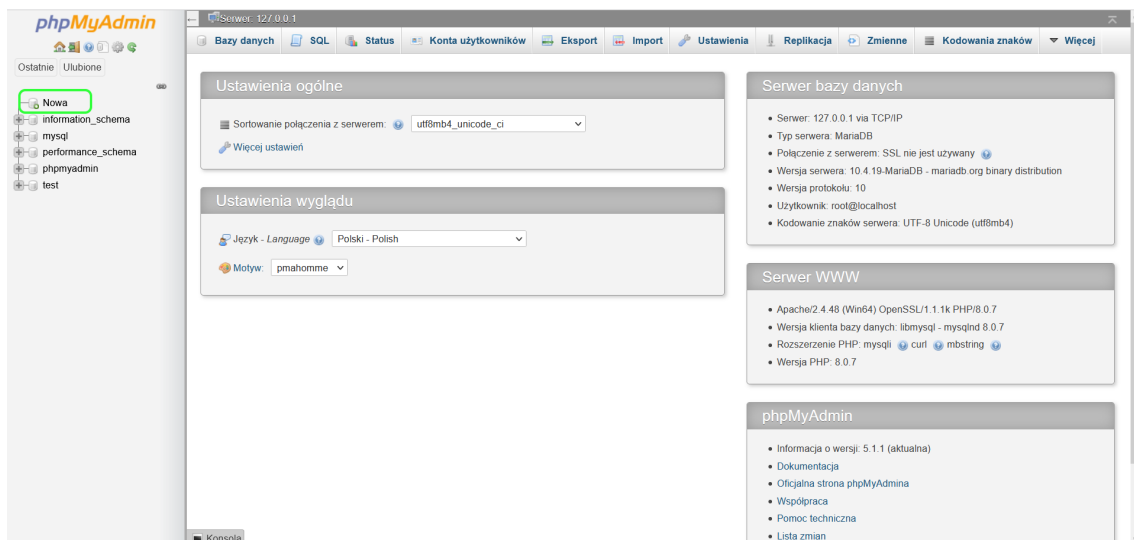
To co potrzebne jest do pracy nad projektem to serwery Apache i MySQL. Uruchamiamy je w panelu kontrolnym przyciskami START.



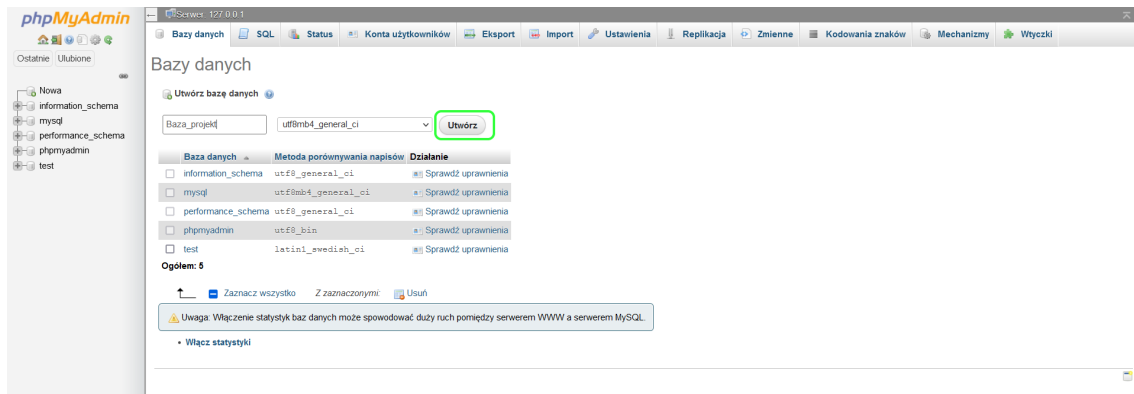
I uruchamiamy PHPMyAdmin wybierając przycisk ADMIN dla serwera MySQL



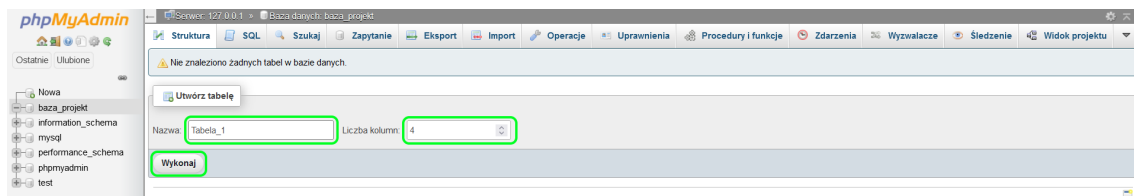
W przeglądarce internetowej otwiera się PHPMyAdmin i możemy przejść do zakładania nowej bazy danych wybierając przycisk NOWA.



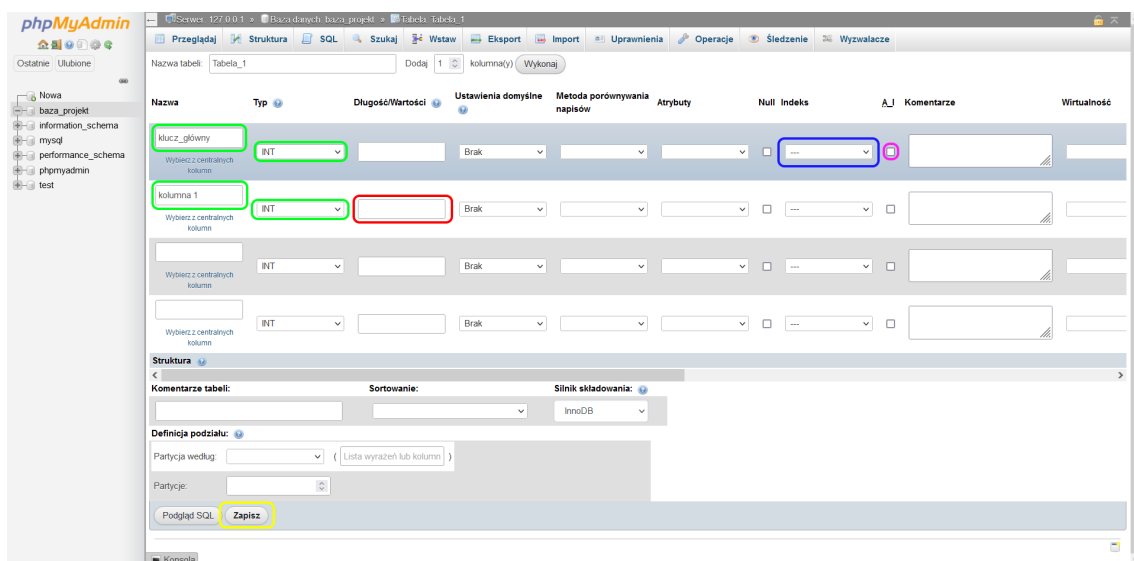
Wpisujemy nazwę nowej bazy i wybieramy przycisk Utwórz, aby założyć bazę.



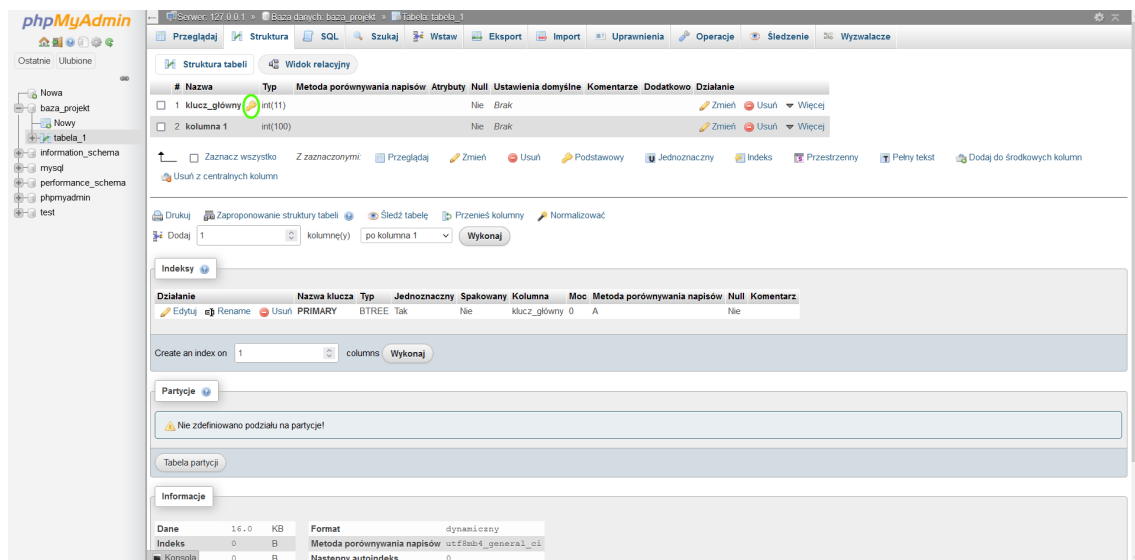
Po utworzeniu bazy możemy dodawać do niej tabele. Podajemy nazwę tabli oraz liczbę kolumn i wybieramy przycisk utwórz.



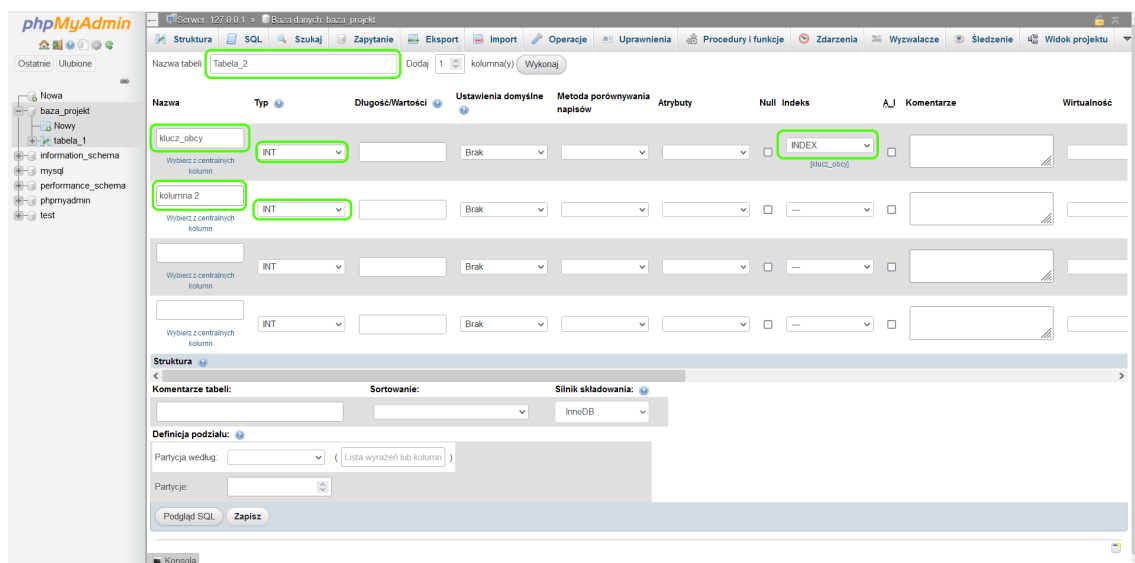
W tabeli dodajemy kolumny. Podajemy nazwy oraz typ . Dla typu INT można nie podać długości będzie ustawiona na domyślną wartość. W przypadku VARCHAR obowiązkowo podajemy długość. Dla klucza głównego w kolumnie indeks z rozwijanego menu wybieramy PRIMARY. Po wybraniu PRIMARY pojawi się okno dialogowe, które zatwierdzamy przyciskiem WYKONAJ. Po czym zaznaczamy autoinkrementację (A.I) ☐. Jak uzupełnimy wszystkie kolumny, które planujemy mieć w tabeli wybieramy przycisk ZAPISZ .



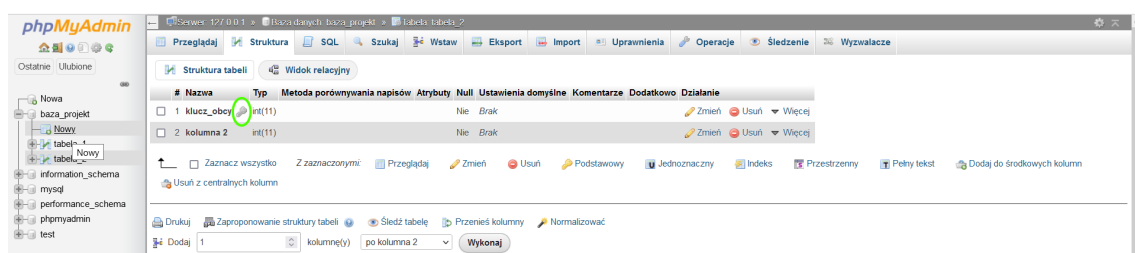
Po zapisaniu mamy widok struktury tabeli z zaznaczonym symbolem klucza głównego. W tym widoku można poprawiać/uzupełniać strukturę tabeli.







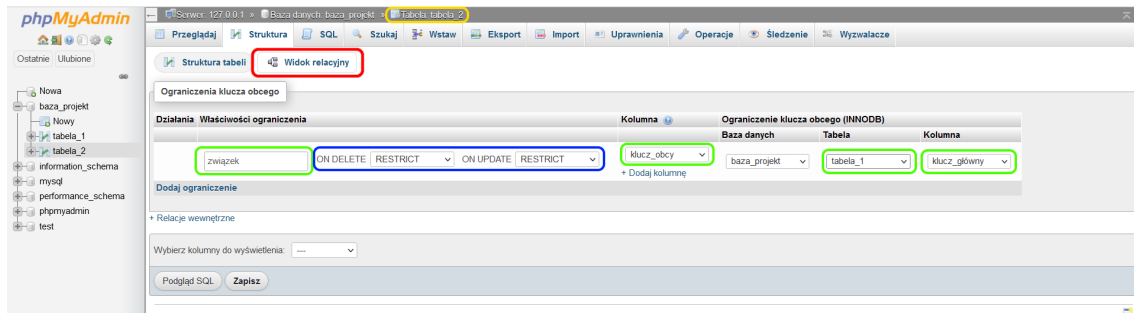
Analogicznie tworzymy kolejną tabelę podając nazwę tabeli, nazwy kolumn i typy danych. Dla klucza obcego w kolumnie indeks z rozwijanego menu wybieramy INDEX.



Po zapisaniu zobaczymy widok struktury drugiej tabeli z zaznaczonym symbolem klucza obcego.



W widoku struktury pojawia się przycisk WIDOK RELACYJNY , gdy którakolwiek z tabel ma zdefiniowany klucz. Gdy żadna z tabel nie ma przypisanego klucza to tego przycisku nie ma! (Na poprzednich rysunkach widać go, gdyż pierwsza tabela miała zdefiniowany klucz główny). Po wybraniu tego przycisku można przystąpić do definiowania relacji (w przykładzie przycisk był wybrany z widoku struktury drugiej tabeli ). Podajemy nazwę związku oraz wybieramy z którą tabelą i którą kolumną tabeli powiązana jest kolumna z tabeli_2 . Należy też odpowiednio ustawić opcje na wypadek usuwania lub modyfikacji krotek .



Po zapisaniu więzów pojawi się odpowiedni komunikat, że zostało to zrobione poprawnie bądź komunika o błędach.

Przedstawiłem Państwu powyżej najbardziej podstawowe informacje o tworzeniu bazy w PHPMyAdmin. Poruszanie się po zakładkach pomoże Państwu lepiej zorientować się w dostępnych funkcjonalnościach.

Jeśli chodzi o samą aplikację to w katalogu XAMPP-a znajduje się podkatalog *htdocs*. Przykładowo w systemie Windows może to być *C:\xampp\htdocs*. W katalogu *htdocs* można założyć katalog na Państwa projekt np. katalog *ProjektBD* (czyli pełna ścieżka to *C:\xampp\htdocs\ProjektBD*). W każdym systemie operacyjnym należy zapewnić prawa dostępu do katalogu *htdocs* dlatego przykładowo w systemie Windows nie należy instalować do katalogu *Program Files*, a systemie Linux zapewnić odpowiednie prawa dostępu (*chmod*).

W katalogu *ProjektBD* można umieścić plik(i) PHP Państwa aplikacji. Działanie sprawdzamy w przeglądarce wpisując adres *https://localhost/ProjektDB/start.php* bądź też *https://127.0.0.1/ProjektDB/start.php*. Przy czym plik *start.php* jest tu plikiem podstawowym aplikacji (oczywiście można użyć nazwy *index.php*). Aby całość działała muszą być uruchomione serwery Apache i MySQL. A w samym pliku PHP połączenie z bazą realizujemy jako root bez hasła, o ile użyją Państwo domyślnych ustawień. Czyli:

```
mysqli("localhost", "root", , "nazwa_bazy");
```

Dla osób, które nie chcą/nie mogą nic instalować na swoim komputerze istnieje możliwość pracy zdalnej na serwerze 'panoramx'. Logując się można utworzyć całą bazę i aplikację na serwerze. Samą bazę tworzymy logując się do MySQL. Natomiast aplikację w PHP należy umieścić w katalogu *public_html* znajdującym się w katalogu domowym użytkownika (*/home/nazwa_użytkownika/public_html*). Katalog *public_html* i pliki w nim powinny mieć odpowiednio ustawione prawa dostępu (pliki 644, a katalog 611).

Samo łącznie z bazą wymaga podania hasła i nazwy użytkownika. Proponuję rozwiązać to w następujący sposób - w głównym pliku PHP załączać plik zawierający funkcje obsługujące połączenie. W przypadku bazy używanej na zajęciach wyglądało by to tak:

```
<?php
include 'open_db.php';
$conn = OpenCon();
echo "Connected Successfully";
echo "<pre>";
$result = $conn->query("SELECT * FROM klienci");
while ($row = $result->fetch_row())
print_r($row);
$result->close();
echo "</pre>";
CloseCon($conn);
?>
```

natomiast plik "open_db.php" z funkcjami obsługującymi łącznie z bazą:

```
<?php
function OpenCon()
{
$dbhost = "{}localhost";
$dbuser = "{}nazwa_użytkownika";
$dbpass = "{}hasło_do_MySQL";
$db = "{}nazwa_bazy";
$conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or
die("Connect failed: %s\n". $conn -> error);
return $conn;
}
function CloseCon($conn)
{
$conn -> close();
}
?>
```

W powyższym przykładzie "nazwa_użytkownika" to Państwa login, hasło do MySQL przechowywane jest w pliku *.mysql* w katalogu domowym użytkownika. Zgodnie z regułami nazwa bazy rozpoczyna się od loginu. Ten przykład wypisuje oczywiście zawartość tabeli 'klienci'.

Alternatywnie można wykorzystać oddzielny plik (przykładowo o nazwie "polacz.php") do przechowania danych dostępowych:

```
<?php
$dbhost = "{}localhost";
$dbuser = "{}nazwa_użytkownika";
$dbpass = "{}hasło_do_MySQL";
$db = "{}nazwa_bazy";
?>
```

I połączenie realizować w głównym pliku PHP:

```
<?php
require_once "polacz.php";

$conn = new mysqli($dbhost, $dbuser, $dbpass,$db);
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();}

$que = "SELECT * FROM produkty1";
$result = mysqli_query($conn, $que) or die($conn->error);

echo "<pre>";
while ($row = $result->fetch_row())
    print_r($row);
$result->close();
echo "</pre>";
...
$conn->close();
?>
```