Logowanie w PHP - część druga (wykorzystanie bazy danych)

W tej części postaram się wyjaśnić, jak w prosty sposób stworzyć logowanie na stronie z wykorzystaniem bazy danych MySQL.

Aby tego dokonać będziemy potrzebowali czterech rzeczy - serwera z obsługą php i bazą danych MySQL, oraz trzech plików: db.php, który stworzy nam odpowiednie tabele w bazie danych, login.php w którym umieścimy formularz logowania, oraz secret.php, który będzie odpowiedzialny za sprawdzenie, czy użyszkodnik użytkownik ma dostęp do tajnych zasobów.

Zaczynamy! Tworzymy plik login.php z formularzem logowania o treści identycznej jak w poprzednim przykładzie:

http://ideone.com/4Wmqm

Do połączenia i wykonywania wszystkich operacji na bazie danych wykorzystamy bibliotekę PDO (która powinna być uprzednio zainstalowana na serwerze). Następnie musimy utworzyć plik db.php, w którym umieścimy kod łączący się z bazą danych oraz tworzący w niej odpowiednią tabelę:

http://ideone.com/52z94

Wyjaśnienie: na początku otwieramy sekcję try..catch łapiącą wszelkie wyjątki, które może wyrzucić klasa PDO, wewnątrz try{} kolejno:

- otwieramy połączenie z bazą danych, gdzie: host_bazy to domena/adres IP serwera bazy danych (jeśli to jest ta sama maszyna, można wpisać localhost); nazwa_bazy - jest to nazwa bazy danych z której będziemy korzystać; user i pass to odpowiednio nazwa użytkownika i jego hasło do bazy danych
- tworzymy tabelę users zawierającą dane o użytkownikach
- dodajemy użytkownika admin z takim samym jak poprzednio haszem hasła - czyli utworzonym z 'adminpasswd'
- Wyświetlamy komunikat

Natomiast w bloku catch łapiemy ewentualne wyjątki oraz wyświetlamy je jeśli wystąpią. W tym wypadku hasło tak samo jak w wypadku logowania z użyciem plików tekstowych jest haszowane. Do wygenerowania haszu z hasła można skorzystać z dostępnych w Internecie narzędzi, np. http://www.insidepro.com/hashes.php. Po utworzeniu odpowiednich tabel można (a nawet dobrze by było) usunąć ten plik, gdyż nie jest on nam do niczego potrzebny. Jeśli to już zostało zrobione, pozostał już tylko 1 plik - secret.php zawierający zasoby, do których dostęp chcemy ograniczyć. Oto, co powinien zawierać:

- Sprawdzenie, czy użytkownik istnieje w bazie danych i czy podał prawidłowe dane
- Zapisanie w sesji informacji o tym, że użytkownik ma prawo do dostępu (aby nie trzeba było się logować po każdym odświeżeniu stron

- Wyświetlenie naszej tajnej treści

Zaczynamy - tworzymy plik secret.php o następującej treści:
http://ideone.com/xDC0d

Teraz wyjaśnienie co się dzieje linijka po linijce: standardowo otwieramy sekcję kodu PHP, w następnej linii rozpoczynamy sesję, po czym sprawdzamy, czy: a) użytkownik wysłał uprzednio jakieś dane przez formularz, b) jest zalogowany. Jeśli żadna z tych sytuacji nie zaistniała - kończymy działanie skryptu (dyrektywa die()). Gdy już wiemy, że użytkownik jest zalogowany/przesłał dane przez formularz otwieramy połączenie z bazą danych - host i nazwa bazy danych, jak i hasło muszą być atkie same jak podczas tworzenia tabeli uzytkowników. Następne 3 linie to tworzenie zapytania wyszukującego użytkownika w bazie danych. :user oraz :passwd oznaczają odpowiednio użytkownika i hasło. Następnie pod wspomniane :user i :passwd podpisujemy odpoiednio nazwę użytkownika, jak i hasz hasła podanego przez użytkownika (podanie danych w ten sposób zamiast bezpośrednio w zapytaniu zabezpiecza nas przed błędami typu SQL Injection; patrz: mój artykuł o bezpieczeństwie danych osobowych w serwisach internetowych, a w szczególności sekcje dot. odzyskiwania haseł i błędów SQL Injection).

Warunek if sprawdza, ile rekordów zwróciło zapytanie; jeśli 1, oznacza to, że użyszkodnik użytkownik z takim hasłem i nazwą został odnaleziony w bazie danych i możemy ustawić wartość $_SESSION['logged_in'] (informującej o zalogowaniu) na true. Ostatnia linia w sekcji try to zamknięcie połączenia pobranych wyników. W sekcji catch, podobnie jak w pliku db.php łapiemy ewentualny wyjątek i wyświetlamy go. Sekcji wewnątrz ciała dokumentu nie będę opisywał, gdyż było to zrobione w poprzedniej części.

Jeszcze obiecane porównanie:

Logowanie z wykorzystaniem plików tekstowych:
Zalety:
- Szybkość
- Niewielkie wymagania wobec serwera

Wady:
- Skomplikowane zarządzanie użytkownikami (edycja/usuwanie)

Logowanie z wykorzystaniem bazy danych:

Zalety:
- Łatwość zarządzania użytkownikami

Wady:
- Mniejsza szybkość
- Większe wymagania o serwera

Jak widać, logowanie z użyciem bazy danych ma mniej zalet, lecz ta jedna zwykle przeważa - utrata wydajności nie jest tak duża, a zarządzanie jest wiele prostsze i wydajniejsze.

//edit: paczka z kodem do pobrania