

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

PHP i MySQL. 8 komponentów dla kreatywnych webmasterów

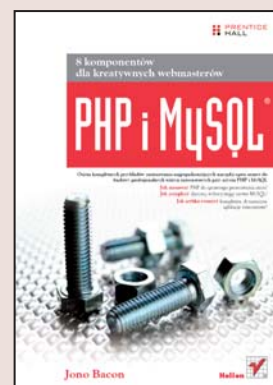
Autor: Jono Bacon

Tłumaczenie: Piotr Pilch

ISBN: 978-83-246-0957-4

Tytuł oryginału: [Practical PHP and MySQL\(R\):
Building Eight Dynamic Web Applications](#)

Format: B5, stron: 528



Osiem kompletnych przykładów zastosowania najpopularniejszych narzędzi open Source do budowy profesjonalnych witryn internetowych przy użyciu PHP i MySQL

- Jak stosować PHP do sprawnego generowania stron?
- Jak zarządzać danymi, wykorzystując serwer MySQL?
- Jak szybko tworzyć kompletne, dynamiczne aplikacje internetowe?

Czas stron statycznych się skończył! Blogi, fora dyskusyjne, sklepy internetowe, serwisy aukcyjne i informacyjne – dziś o popularności danej strony decydują przede wszystkim aplikacje dynamiczne pozwalające użytkownikom na komunikację z witryną. Większość stron bazuje na dającej ogromne możliwości platformie obejmującej język PHP i bazy danych MySQL. Bezpłatny dostęp do tej technologii, jej potencjał i wydajność przyczyniły się do niesamowitej popularności wśród webmasterów.

„PHP i MySQL. 8 komponentów dla kreatywnych webmasterów” to wprowadzenie w świat popularnej platformy do tworzenia aplikacji sieciowych od strony praktycznej. Krótka część teoretyczna, dzięki której dowiesz się, jak zainstalować PHP i MySQL, oraz poznasz podstawy korzystania z tych narzędzi, to jedynie wstęp do książki. W części głównej przedstawiono techniki stosowania PHP i MySQL na przykładzie ośmiu kompletnych, funkcjonalnych aplikacji, które decydują o atrakcyjności witryny. Pracując na tych przykładach nauczysz się sprawnie zarządzać bazami danych, dynamicznie aktualizować informacje na stronach, obsługiwać rejestrację i logowanie, stosować pakiety z repozytorium PEAR oraz wykonywać wiele innych operacji przydatnych na profesjonalnych stronach internetowych.

- Wprowadzenie do języka PHP i serwera MySQL
- Tworzenie dynamicznych aplikacji sieciowych
- Projektowanie interfejsu użytkownika
- Zarządzanie bazami danych
- Obsługa rejestracji i uwierzytelniania
- Administrowanie witrynami
- Stosowanie technologii Ajax
- Zarządzanie treścią witryn
- Używanie pakietów z repozytorium PEAR

Bądź kreatywnym webmasterem i buduj dynamiczne strony internetowe!



Spis treści

O autorze	9
Przedmowa	11
Wprowadzenie	13
Rozdział 1. Nowa metodyka	19
Technologia	20
Działanie dynamicznej aplikacji internetowej	23
Podsumowanie	27
Rozdział 2. Język PHP i serwer MySQL — wprowadzenie	29
Konfigurowanie interpretera języka PHP i serwera MySQL	30
Rozpoczęcie korzystania z języka PHP	32
Instalowanie serwera MySQL	50
Łączenie się z serwerem MySQL za pomocą skryptu PHP	59
Podsumowanie	65
Rozdział 3. Uruchamianie projektów	67
Zastosowanie środowiska XAMPP	67
Podsumowanie	70
Rozdział 4. Dziennik internetowy	71
Przegląd projektu — przypadek użycia dziennika Blogtastic	72
Definiowanie bazy danych	73
Rozpoczęcie pisania kodu	76

Przeglądanie wybranych wpisów	90
Tworzenie przeglądarki kategorii	100
Decydowanie, kto może się zalogować	103
Definiowanie własnych kategorii	109
Tworzenie nowych wpisów blogu	111
Aktualizowanie wpisu blogu	114
Podsumowanie	119
Rozdział 5. Fora dyskusyjne	121
Co jest pod maską?	121
Tworzenie własnego forum	124
Zanim rozpocznie się realizację projektu	124
Tworzenie bazy danych	125
Projektowanie wyglądu witryny	131
Wyświetlanie forów	133
Zarządzanie logowaniem użytkowników	143
Wiadomości i odpowiadanie na nie	161
Tworzenie stron administracyjnych	169
Podsumowanie	178
Rozdział 6. Koszyk zakupów	181
Przegląd projektu	181
Tworzenie bazy danych	182
Rozpoczęcie pisania kodu	187
Zarządzanie logowaniem użytkowników	192
Wyświetlanie i wybieranie produktów	196
Płacenie za produkty	208
Strony administracyjne	219
Podsumowanie	227
Rozdział 7. Witryna aukcji internetowych	229
Przegląd projektu	230
Tworzenie bazy danych	230
Rozpoczęcie pisania kodu	233
Wyświetlanie pozycji aukcji	236
Obsługa użytkowników	252
Dodawanie pozycji	255
Obsługa licytacji	271
Planowanie wczytywania strony	275
Podsumowanie	276
Rozdział 8. Kalendarz internetowy	277
Przegląd projektu	277
Tworzenie bazy danych	278
Rozpoczęcie pisania kodu	280
Przeglądanie wydarzeń	288
Podsumowanie	314

Rozdział 9. System zarządzający treścią pytań i odpowiedzi	315
Przegląd projektu	316
Projektowanie bazy danych	317
Rozpoczęcie tworzenia kodu	322
Wyświetlanie pytań	333
Obsługa logowania	343
Dodawanie i ocenianie pytań	354
Zarządzanie tematami	369
Zarządzanie wątkami	374
Prawo własności do tematu	378
Podsumowanie	391
Rozdział 10. Projekt wielokrotnego użycia	393
Przegląd projektu	394
Projektowanie bazy danych	396
Inteligentne używanie katalogów	399
Rozpoczęcie pisania kodu — tworzenie szkieletu	400
Pobieranie dystrybucji	408
Przeglądanie zrzutów ekranu	410
Przeglądarka dostępnych projektów	411
Zarządzanie projektami	413
Modyfikowanie ogólnych ustawień	420
Zarządzanie plikami do pobrania	422
Usuwanie dystrybucji	428
Zarządzanie zrzutami ekranu	430
Usuwanie obrazów	434
Dodawanie nowego projektu	435
Wdrażanie projektu w obrębie aplikacji	436
Podsumowanie	438
Rozdział 11. Witryna informacyjna	441
Przegląd projektu	442
Instalowanie modułów PEAR	442
Projektowanie bazy danych	445
Rozpoczęcie tworzenia kodu	448
Obsługa logowania użytkowników	456
Przeglądanie i ocenianie informacji	462
Zarządzanie informacjami	468
Zarządzanie kategoriami	475
Tworzenie mechanizmu wyszukiwania	479
Podsumowanie	485
Dodatek A Projektowanie witryny WWW	487
Skorowidz	517

1

Nowa metodyka

Opanowanie umiejętności programowania zawsze było trudną sztuką. Pomimo zadziwiająco dużej dostępności dokumentacji, podręczników, stron internetowych, materiałów wideo, książek i innych zasobów w gruncie rzeczy nauka programowania w dalszym ciągu nie jest prosta, zwłaszcza gdy nie jest się studentem wydziału informatycznego.

Jedną z głównych przyczyn jest to, że zwykle analizowanie kodu źródłowego przebiega w nienaturalny sposób. Większość książek i podręczników wydaje się podążać ścieżką dokładnego objaśniania mniej istotnych szczegółów języka, a następnie przedstawiania na ich podstawie bardziej złożonych zagadnień. Taka metoda przypomina przygotowywanie się do egzaminów. Ciężko zapamiętać w odpowiedniej kolejności poszczególne kawałki informacji i wiedzieć, jak są ze sobą powiązane.

Ta książka jest inna. Choć większość książek korzysta z wyżej omówionej metodyki, ta pokazuje diametralne zmiany w sposobie przekazywania wiedzy. Zamiast zawierać 300 stron przytłaczającej teorii, książka przede wszystkim koncentruje się na kilku praktycznych projektach, które Czytelnik zrealizuje we własnym zakresie. Projekty te dotyczą aplikacji internetowych. Ponadto, zdobywając wiedzę o tworzeniu różnych aplikacji, Czytelnik nie tylko lepiej zrozumie język PHP i serwer MySQL, lecz również w większym stopniu nabędzie umiejętność projektowania *rzeczywistych* aplikacji.

Projekty, które będą realizowane w książce, dotyczą następujących zastosowań internetowych:

- zwykła witryna internetowa,
- dziennik internetowy,
- witryna aukcyjna,
- koszyk zakupów,
- fora dyskusyjne,
- witryna zarządzająca listą często zadawanych pytań FAQ (*Frequently Asked Questions*),
- witryna informacyjna,
- niezależna aplikacja PHP,
- prosty kalendarz w technologii AJAX.

Każdy projekt daje możliwość opanowania nowych umiejętności i koncentruje się na konkretnych wyzwaniach.

Technologia

Nie powinno być zaskoczeniem, że technologia wykorzystana w książce do tworzenia aplikacji internetowych to język PHP i serwer MySQL. Jeśli Czytelnik znalazł książkę w dziale księgarni dotyczącym technologii ASP (*Active Server Pages*), obawiam się, że ktoś sobie w niemylły sposób zażartował.

Gdy język PHP i serwer MySQL połączy się ze sobą, uzyska się solidny szkielet pozwalający projektować elastyczne aplikacje internetowe o dużych możliwościach. Powodem dobrej współpracy obu technologii jest to, że każda zapewnia rozbudowany zestaw internetowych narzędzi programistycznych. W przypadku projektowania dowolnej aplikacji internetowej przede wszystkim trzeba mieć odpowiednie środowisko umożliwiające tworzenie dynamicznych stron oraz funkcji obsługujących i przetwarzających dane, łączących się z zasobami, zarządzających użytkownikami i wykonujących inne zadania. Język PHP spełnia te wymagania. Jest to wyjątkowo elastyczny język oferujący pokaźny wachlarz funkcji radzących sobie z typowymi wyzwaniami towarzyszącymi projektowaniu aplikacji internetowych, z których wiele przedstawiono w ramach projektów zamieszczonych w książce. Następnie trzeba mieć rozwiązanie pozwalające magazynować spore ilości danych, które będą wyświetlane, aktualizowane, usuwane, modyfikowane i w inny sposób przetwarzane. W tym celu należy skorzystać z bazy danych. MySQL jest niezawodnym, elastycznym i łatwym w obsłudze serwerem bazodanowym.

Zanim przyjrzymy się modelowi działania aplikacji internetowych, w niniejszym rozdziale bardziej szczegółowo zostaną przedstawione niezbędne narzędzia pracy.

Język PHP

PHP jest popularnym językiem skryptowym wysokiego poziomu wykorzystywanym przez wiele organizacji i projektantów. Pierwotnie język PHP, stworzony w ramach niewielkiego projektu opartego na języku Perl pod koniec 1995 r. przez Rasmusa Lerdorfa, miał ułatwić swojemu twórcy zaprojektowanie własnej strony internetowej. W związku z tym Lerdorf nadał językowi nazwę PHP (ang. *Personal Home Page* — osobista strona domowa).

Gdy Lerdorf zaczął pracować dla uczelni University of Toronto i projektować system obsługujący połączenia telefoniczne dające studentom dostęp do internetu, nie dysponował żadnymi środkami pozwalającymi łączyć strony internetowe z bazami danych. Dlatego Lerdorf zastąpił kod języka Perl interfejsem napisanym w języku C, który oferował możliwość nawiązania połączenia między stronami internetowymi a serwerem bazodanowym MySQL. Gdy początkowo niewielki projekt Lerdorfa zaczął się rozrastać, autor udostępnił go w internecie w wersji *open source*, licząc na udoskonalenia dokonywane przez innych programistów zainteresowanych językiem PHP. W późniejszym czasie skrót PHP przyjął aktualne rekurencyjne rozwinięcie, czyli *PHP Hypertext Preprocessor* (ukute przez Zeeva Suraskiego i Andiego Gutmansa, którzy w 1997 r. ponownie napisali analizator składni). Oprogramowanie cały czas było rozwijane i obecnie ma postać rozbudowanej platformy.

PHP jest solidnym i dobrze zdefiniowanym językiem uwzględniającym między innymi obsługę programowania zorientowanego obiektowo, warunków, plików i operacji arytmetycznych. Pod względem semantycznym język PHP przypomina język skryptowy powłoki rozbudowany o prostsze elementy języka C.

Język PHP wpisuje się w filozofię języków programowania, które oferują rozbudowaną i uniwersalną standardową bibliotekę. Ponadto język ten w dużym zakresie obsługuje mnóstwo rozwiązań, takich jak pliki *cookies*, formularze, sesje, pliki nagłówkowe, gniazda sieciowe, poczta elektroniczna oraz protokoły LDAP, IRC i inne. Obsługa baz danych uwzględnia nie tylko serwer MySQL, ale również wiele innych, między innymi PostgreSQL, Oracle, MS SQL, dBase, Sybase i DB2. Tak rozbudowana obsługa baz danych okaże się przydatna, gdy kiedykolwiek trzeba będzie przystosować aplikację do współpracy z inną bazą.

PHP można zastosować nie tylko do tworzenia aplikacji internetowych, ale także w roli języka skryptowego powłoki. Oznacza to, że za pomocą jednego języka można projektować aplikacje internetowe i pisać skrypty powłoki zarządzające pracą komputerów. Możliwe jest nawet użycie języka PHP do tworzenia aplikacji przeznaczonych dla stacji roboczych. Choć to zastosowanie zwykle jest związane z bardziej ekscentryczną częścią entuzjastów języka PHP, coraz więcej programistów się do niego przekonuje.

Język PHP nie ogranicza się do własnych funkcji i obsługuje zewnętrzne funkcje za pośrednictwem biblioteki PEAR (*PHP Extension and Application Repository*). Biblioteka ta działa podobnie do modułów CPAN języka Perl i zapewnia dodatkową funkcjonalność, która jest łatwo dostępna za pomocą kilku wbudowanych niezależnych modułów rozwiązujących określone problemy. Te specjalne moduły można dodać do własnej aplikacji i uzyskać dostęp do oferowanych przez

nie możliwości. Jeśli na przykład za pomocą aplikacji internetowej trzeba wysłać wiadomość pocztową, można użyć specjalnej funkcji pocztowej biblioteki PEAR, która rozbudowuje obsługę poczty elektronicznej przez język PHP. Dzięki temu język PHP lepiej obsługuje zewnętrzne rozszerzenia. W efekcie pojawiła się ogromna liczba darmowych modułów PEAR.

Serwer MySQL

MySQL jest rozbudowanym relacyjnym serwerem bazodanowym o dużych możliwościach, w pierwotnej wersji stworzonym przez Davida Axmarka, Allana Larssona i Michaela „Monty” Wideniusa. Założona przez nich komercyjna organizacja MySQL AB zajmuje się rozwijaniem i sprzedażą serwera MySQL, a także towarzyszących mu produktów. Choć początkowo projekt serwera MySQL był realizowany na zasadach *open source*, jego twórcy byli przeświadczeni, że mogą rozpocząć działalność biznesową opartą na tym produkcie. Dzięki firmie projektanci mogą w pełnym zakresie zajmować się oprogramowaniem, co przynosi korzyści zarówno społeczności *open source*, jak i komercyjnym użytkownikom serwera MySQL. Pod względem funkcjonalnym darmowe i komercyjne wersje serwera MySQL są jednakowe. Jedyna różnica tkwi w sposobie licencjonowania (firmy muszą nabyć licencję, jeśli zamierzają zastosować serwer MySQL w celach komercyjnych). Serwer MySQL zyskał ogromną popularność. Korzystają z niego takie firmy, jak Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics i Texas Instruments.

MySQL jest w pełni funkcjonalnym serwerem bazodanowym wykorzystującym otwarte standardy, takie jak ANSI SQL 99 (umożliwia komunikację z bazami danych za pomocą języka SQL — *Structured Query Language*). Standard ten zapewnia środki pozwalające wstawiać, aktualizować i pobierać informacje z bazy danych przy użyciu języka będącego standardem przemysłowym. Język SQL jest stosowany zarówno w przypadku produktów bazodanowych, jak i innych. Serwer MySQL obsługuje kilka dodatkowych instrukcji języka SQL. Serwer MySQL nie tylko ma status standardu, jest też wieloplatformowy. Oznacza to, że można go uruchomić nie tylko na Linuksie, ale także na innych systemach operacyjnych, takich jak Windows, Mac OS X, BSD lub inne warianty Uniksa.

Sam serwer bazodanowy uwzględnia interaktywnego klienta trybu wiersza poleceń, który umożliwia komunikowanie się z nim. Choć klient jest przydatny, może wywołać obawy u początkujących użytkowników, niezaznajomionych z wierszem poleceń. Na szczęście istnieje kilka graficznych klientów bazujących na przeglądarce internetowej, dzięki którym można uniknąć kontaktu z wierszem poleceń (w książce intensywnie wykorzystuje się jednego z takich klientów o nazwie `phpMyAdmin`). Serwer MySQL obsługuje też kilka języków programowania pozwalających uzyskać dostęp do bazy danych i pobrać z niej informacje. Spośród tych języków wymienimy: PHP, Pythona, Perla, C, C++ i Javę. Choć na początku w celu wysyłania zapytań bazodanowych można korzystać wyłącznie z języka PHP, obsługa wielu języków będzie przydatna, gdy konieczne okaże się utworzenie modułów i aplikacji za pomocą innych języków.

Działanie dynamicznej aplikacji internetowej

Na najbardziej podstawowym poziomie język PHP i serwer MySQL zapewniają środki umożliwiające dystrybucję dynamicznej treści do urządzenia sieciowego. Można tu przytoczyć przykłady komputera wujka Janka łączącego się ze stroną internetową, usługi dostarczania łączącej się bezprzewodowo ze swoją siecią śledzącą lub użytkownika uzyskującego dostęp do skrzynki pocztowej za pośrednictwem witryny WWW. Każde z tych różnych zastosowań w zasadzie korzysta z takich samych składników programowych, które komunikują się ze sobą ponad podziałami sprzętowymi. *Projektowanie klient-serwer* jest technicznym określeniem typu programowania, który stosuje się w przypadku języka PHP i serwera MySQL. Aby Czytelnik w pełni zrozumiał to zagadnienie, w rozdziale przedstawiono kilka przykładów i objaśniono, jak informacja jest przekazywana między składnikami internetu.

Można sobie wyobrazić dowolną stronę internetową. Niech Czytelnik trochę pofantazjuje i pomyśli, że jest tajnym szpiegiem i tak się składa, że dysponuje witryną WWW z ulubionymi „zabawkami” szpiegowskimi. Załóżmy, że witryna wyświetla podstawowy kod HTML zrozumiały dla przeglądarki internetowej i nic ponadto. Strona internetowa zawiera wyłącznie listę uwzględniającą szpiegowski materiał i nie oferuje interaktywności. Po prostu należy nawiązać połączenie z witryną WWW o adresie *www.adres.com*. W efekcie zostanie wyświetlona jej zawartość. Na rysunku 1.1 pokazano typ interakcji występujący w przypadku tego przykładu.

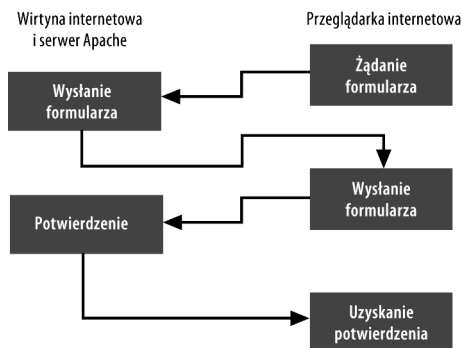


Rysunek 1.1. Metoda łączenia się przeglądarki internetowej z witryną WWW

W przykładzie klient łączy się z serwerem WWW (w tym przypadku z serwerem Apache) i żąda strony HTML. Po nawiązaniu połączenia serwer Apache zwraca klientowi żadaną stronę. Serwer Apache pełni rolę pośrednika odbierającego żądania i wysyłającego odpowiedzi klientowi.

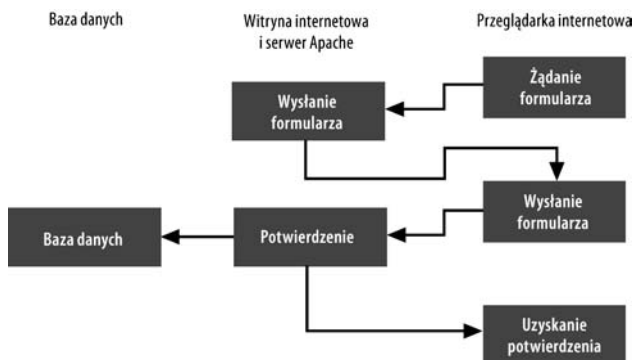
Rysunek 1.2 przedstawia trochę bardziej złożony przykład tego, jak funkcjonuje formularz HTML pobierający dane.

Jeśli analizę diagramu rozpocząć od jego prawego górnego narożnika, można zauważyć, że w pierwszej kolejności przeglądarka internetowa żąda formularza HTML. Serwer Apache odpowiada i odsyła jej formularz. Użytkownik następnie wypełnia formularz i przekazuje go z powrotem serwerowi Apache, który wysyła klientowi stronę potwierdzającą. Kierując się wskazówkami, można zorientować się, jak przebiega proces przekazywania informacji między klientem i serwerem.



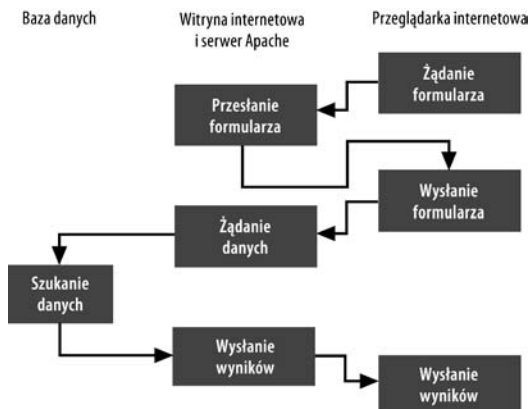
Rysunek 1.2. Obsługa formularza HTML przez klienta i serwer

Następny przykład, przedstawiony na rysunku 1.3, demonstruje, co się stanie, gdy poprzedni przykład zostanie rozbudowany o serwer bazodanowy. W tym przypadku nie tylko użytkownik wypełni formularz, ale również jego zawartość zostanie zapisana w bazie danych. Aby tak było, po pobraniu przez witrynę WWW zawartości formularza trzeba będzie uwzględnić dodatkowy krok. Witryna musi wysłać stronę potwierdzającą i umieścić dane w bazie. Ponieważ druga z wymienionych operacji jest jednokierunkowa, na rysunku nie ma widocznej strzałki z dwoma grotami.



Rysunek 1.3. Po dodaniu bazy danych przepływ informacji jest trochę inny

Model widoczny na rysunku 1.3 jest niemal taki sam jak pokazany na rysunku 1.2. Różni się tym, że dodano dodatkowy etap. Poza umieszczeniem informacji w bazie danych trzeba też mieć możliwość ich pobrania w celu wyświetlenia wyników. Na rysunku 1.4 widać klienta żądającego formularza (na przykład wyszukującego), a następnie wysyłającego go z wprowadzonymi szczegółami. Wyniki zwrócone w odpowiedzi na zapytanie skierowane do bazy danych są przekazywane klientowi.

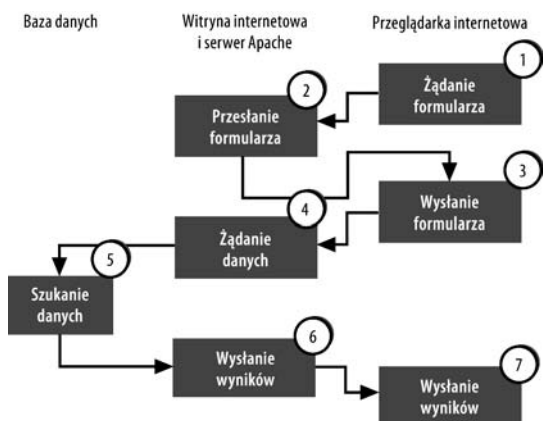


Rysunek 1.4. Typowy przypadek: użytkownik wypełnia formularz i uzyskuje dane z bazy

W tym przykładzie najpierw od serwera żądany jest formularz. Serwer przekazuje formularz przeglądarce, a następnie użytkownik umieszcza w nim wymagane informacje i wysyła go. W dalszej kolejności serwer odbiera szukany ciąg i żąda od bazy danych pobrania pozycji, które zawierają szukane dane. Serwer bazodanowy wyszukuje odpowiednie dane i przekazuje je serwerowi Apache, który przesyła je klientowi.

Przedstawione przykłady modelu klient-serwer korzystają z dość luźnych definicji różnych składników (baza danych, serwer WWW i przeglądarka) wchodzących w ich skład. Chodzi o to, aby można było skupić się na tym, w jaki sposób dane są przekazywane między różnymi podstawowymi elementami architektury klient-serwer.

Ostatni przykład (rysunek 1.5) szczegółowo wyjaśnia, jak każdy krok przebiega w każdym z wyżej wymienionych składników.



Rysunek 1.5. Każdy numer odnosi się do niżej zamieszczonego punktu

1. Użytkownik wprowadza adres w oknie przeglądarki internetowej (klient), aby uzyskać dostęp do witryny WWW. W ramach nawiązywanego połączenia jest też żądana strona HTML zawierająca formularz.
2. Przeglądarka łączy się z serwerem Apache, który przechowuje pliki HTML i PHP tworzące witrynę WWW. Apache obsługuje żądanie (zgodnie z regułami zdefiniowanymi w jego pliku konfiguracyjnym, które umożliwiają zlokalizowanie odpowiedniej strony i jej wysłanie) i odsyła klientowi stronę internetową uwzględniającą formularz HTML.
3. Użytkownik wypełnia formularz wyświetlony w oknie przeglądarki, a następnie przesyła go serwerowi.
4. Serwer Apache odbiera wypełniony formularz i otwiera odpowiedni plik w celu dostarczenia formularza. Plik zawiera kod języka PHP realizujący połączenie z bazą danych. Kod ten jest przekazywany przez serwer Apache interpreterowi języka PHP, który go wykonuje. Kod łączy się z serwerem MySQL (może znajdować się na tym samym lub innym komputerze; nie jest to istotne). Po połączeniu się za pomocą języka SQL (służy do komunikowania się z bazami danych) jest tworzone żądanie pobrania informacji.
5. Serwer bazodanowy MySQL otrzymuje żądanie SQL i szuka danych. Po ich zlokalizowaniu wynik jest przekazywany skryptowi PHP, który wygenerował żądanie.
6. Skrypt PHP otrzymuje od serwera MySQL wynik i zanim wyśle go przeglądarce internetowej (klient), tworzy stronę HTML zawierającą rezultat.
7. Przeglądarka odbiera wynik zapytania w postaci strony HTML, którą wyświetla użytkownikowi.

W związku z tym procesem należy zapamiętać kilka istotnych rzeczy. Warto zapoznać się z poniższymi kluczowymi uwagami, a następnie ponownie przeanalizować proces w celu wyeliminowania wszelkich niejasności.

Przede wszystkim należy podkreślić, że dla przeglądarki internetowej zrozumiałe jest język HTML, lecz nie PHP (oczywiście przyjęto, że przykład uwzględnia wszystkie przeglądarki, choć niektóre z nich mogą obsługiwać języki skryptowe, takie jak JavaScript; w przykładzie nie wzięto tego pod uwagę). Komunikacja realizowana przez przeglądarkę internetową bazuje na języku HTML (stąd w kroku 6. wyniki zapytania wysłanego do serwera MySQL zostały zamienione na hipertekst). Następną kwestią jest ściśle powiązanie języka PHP z serwerem Apache. Wszystkie połączenia i zapytania bazodanowe są umieszczane w skrypcie PHP. Tak silna integracja ma na celu uzyskanie wysokiej wydajności. Język PHP można potraktować jak pośrednika przy uzyskiwaniu dostępu do baz danych, plików, danych XML itp. Pośrednik ten wysyła wszystko z komputera, korzystając z serwera Apache. Serwer MySQL powinien być uważany wyłącznie za urządzenie magazynujące dane, które będzie przydatne tylko wtedy, gdy zostanie nawiązane z nim połączenie w celu dodania, usunięcia, aktualizacji lub pobrania informacji.

Serwer MySQL może być odpowiednikiem dysku twardego komputera, który okaże się przydatny, tylko jeśli oprogramowanie uzyska do niego dostęp i zapisze dane. Język PHP można przyrównać do takiego oprogramowania.

Podsumowanie

Podobnie jak w przypadku dowolnego materiału, również nauka projektowania przy wykorzystaniu języka PHP i serwera MySQL wiąże się z koniecznością zrozumienia kilku zasadniczych zagadnień, zanim przejdzie się do konkretów. W celu zapewnienia solidnych podstaw w niniejszym rozdziale przedstawiono główne zagadnienia.

W następnym rozdziale zostaną zaprezentowane podstawy języka PHP i serwera MySQL. Czytelnik zapozna się z fundamentalnymi składnikami języka, a następnie od razu zacznie tworzyć aplikację. Każda aplikacja pozwoli zdobyć nowe umiejętności, pogłębić wiedzę i poznać nowe pojęcia.