

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

Tworzenie stron WWW z wykorzystaniem Ajaksa. Projekty

Autor: Larry Ullman

Tłumaczenie: Radosław Meryk

ISBN: 978-83-246-1553-7

Tytuł oryginału: [Building a Web Site
with Ajax: Visual QuickProject Guide](#)

Format: 170x230, stron: 168



Stwórz interaktywny serwis WWW, korzystający z technologii AJAX!

- Jakich narzędzi potrzebujesz, by stworzyć stronę WWW?
- Jak wzbogacić witrynę dzięki technologii AJAX?
- Jak obsługiwać dane w formacie XML?

AJAX jest skrótem od Asynchronous JavaScript and XML. Technologia ta stała się trzonem rewolucji Web 2.0. Dziś trudno sobie wyobrazić nowoczesną aplikację WWW, która nie korzystałaby z dobrodziejstw tego rozwiązania. Nie bez powodu! Intuicyjny interfejs użytkownika, wysoki poziom interaktywności, wygoda oraz dynamika serwisów korzystających z tego rozwiązania gwarantują, że jeszcze długo będzie ono na topie technik tworzenia stron WWW.

Dzięki książce „Tworzenie stron WWW z wykorzystaniem AJAKSA. Projekty” poznasz proces tworzenia strony WWW, korzystającej z tej technologii. Autor krok po kroku wyjaśnia zasadę działania oraz sposób tworzenia serwisu i każdego z elementów projektowanych dla niego aplikacji. Pierwsze zadanie, które stawia on przed Tobą, to zbudowanie bazy danych. Później nauczysz się implementować funkcjonalności takie, jak przeglądanie danych, dodawanie nowych wpisów czy też wyszukiwanie informacji. Po przeczytaniu niniejszej książki nie będzie dla Ciebie problemem połączenie na przykład technologii HTML, JavaScript, CSS, XML i PHP w celu uzyskania dynamicznego, interaktywnego oraz efektownego serwisu WWW. Zobaczysz także, jak poszczególne funkcjonalności aplikacji WWW działają bez wykorzystania technologii AJAX, a następnie jak zyskują na atrakcyjności po dodaniu tej technologii. Teraz wszystko zależy tylko od Twojej wyobraźni!

- Zasada działania AJAX
- Narzędzia konieczne w procesie projektowania, wytwarzania i testowania
- Przygotowanie bazy danych
- Obsługa żądań z wykorzystaniem PHP
- Przygotowanie stron HTML
- Tworzenie i obsługa formularzy
- Dynamiczna weryfikacja danych wprowadzanych przez użytkownika
- Testowanie statycznych rozwiązań
- Testowanie dynamicznych rozwiązań, korzystających z AJAX
- Wykorzystanie języka JavaScript
- Przeglądanie informacji
- Wyszukiwanie informacji
- Dodawanie nowych wpisów do bazy
- Zastosowanie formatu XML
- Wykorzystanie formatu JSON

Stwórz swoją własną, interaktywną stronę WWW!



spis treści

wprowadzenie 9

jak działa ajax?	10	witryna internetowa	
czego można nauczyć się z tej książki?	11	towarzysząca książce	17
jak zorganizowana jest ta książka?	12	następny krok	18
potrzebne narzędzia	14		

1. tworzenie bazy danych 19

dostęp do bazy danych MySQL	20	wypełnienie tabel danymi	25
utworzenie bazy danych	21	informacje dodatkowe	27
utworzenie tabel	23		

2. przeglądanie pracowników 29

co będziemy robić?	30	nawiązanie połączenia z bazą danych	36
utworzenie strony HTML	31	utworzenie arkusza stylów	37
utworzenie formularza HTML	32	wykorzystanie arkusza stylów	38
utworzenie skryptu PHP	33	testowanie stron	39
wyświetlenie listy pracowników	34	informacje dodatkowe	40
zgłaszanie błędów	35		

3. przeglądanie z wykorzystaniem technologii ajax 45

co będziemy robić?	46	wyświetlenie listy pracowników	53
utworzenie funkcji	47	obsługa odpowiedzi	55
sprawdzenie obsługi technologii Ajax	48	wyświetlenie wyników	56
wywołanie funkcji	49	modyfikacja kodu HTML	57
konfiguracja Ajaksa	50	testowanie warstwy Ajaksa	58
rozpoczęcie strony PHP	52	informacje dodatkowe	59

spis treści

4. dodawanie rekordów

63

co będziemy robić?	64	walidacja danych formularza	70
utworzenie strony HTML	65	aktualizacja bazy danych	73
utworzenie formularza	66	wyświetlenie komunikatów o błędach	74
utworzenie elementów formularza	67	testowanie wersji nieajaksowej	75
rozpoczęcie strony PHP	69	informacje dodatkowe	77

5. dodawanie rekordów z wykorzystaniem technologii ajax

79

co będziemy robić?	80	aktualizacja bazy danych	92
dodanie elementów technologii Ajax	82	uzupełnienie danych w formacie XML	93
wykorzystanie warstwy Ajaksa	83	obsługa odpowiedzi	95
konfiguracja Ajaksa	84	przygotowanie strony	96
przygotowanie danych formularza	85	obsługa danych w formacie XML	97
dokończenie funkcji	86	wyświetlenie wyników	99
przygotowanie do przesyłania wyników w formacie XML	87	testowanie warstwy Ajaksa	100
walidacja danych formularza	88	informacje dodatkowe	102

6. wyszukiwanie

107

co będziemy robić?	108	wyświetlenie wyników	114
utworzenie strony HTML	109	wyświetlenie komunikatów o błędach	115
utworzenie formularza	111	testowanie wersji nieajaksowej	116
rozpoczęcie strony PHP	112	informacje dodatkowe	117
zapytanie do bazy danych	113		

7. wyszukiwanie z wykorzystaniem ajaksa

119

co będziemy robić?	120	przygotowanie strony	132
dodanie elementów technologii Ajax	122	obsługa danych w formacie XML	133
wykorzystanie warstwy Ajaksa	123	wyświetlenie wyników	134
konfiguracja Ajaksa	125	obsługa nazwisk	136
dokończenie funkcji	126	obsługa wydziałów	137
przygotowanie do przesyłania		obsługa adresów e-mail	139
wyników w formacie XML	127	wyświetlenie komunikatu	
zapytanie do bazy danych	128	o braku wyników	141
pobranie wyników	129	dokończenie funkcji	142
dokończenie skryptu PHP	130	testowanie warstwy Ajaksa	143
obsługa odpowiedzi	131	informacje dodatkowe	144

dodatek A następne kroki

149

problemy związane z Ajaxem	150	zasoby w internecie	159
alternatywy dla Ajaksa	151	zestawy Framework	
wprowadzenie do JSON	152	dla języka JavaScript	160
wysyłanie danych w formacie JSON	154	biblioteki PHP-Ajax	161
dostęp do danych w formacie JSON	156	debugowanie kodu JavaScript	162
korzystanie z danych w formacie JSON	157	informacje dodatkowe	166

skorowidz

169



dodawanie rekordów

W pierwszym rozdziale utworzyliśmy strukturę bazy danych dla witryny WWW wyświetlającej listę pracowników. W poprzednich dwóch rozdziałach opracowaliśmy ajaksową i nieajaksową wersję aplikacji wyświetlającej listę pracowników wybranego wydziału. W rozdziale niniejszym opracujemy funkcję dodawania pracowników do bazy danych. Oczywiście rozpoczniemy od wersji nieajaksowej gwarantującej dostęp dla wszystkich użytkowników, a następnie — w kolejnym rozdziale — opracujemy wersję bazującą na technologii Ajax.

Rozpoczniemy od utworzenia **formularza HTML**, który pobiera wszystkie niezbędne dane. Następnie napiszemy skrypt PHP, który obsługuje przesyłanie formularza bez wykorzystania technologii Ajax. Będzie to bardzo prosty skrypt PHP obsługujący formularz HTML.

co będziemy robić?

1 Najpierw, na stronach 65 – 68, utworzymy **stronę HTML z formularzem**.

W formularzu są elementy reprezentujące wszystkie dane dotyczące pracowników zapisane w bazie danych.

Gdy w przeglądarkach bez obsługi Ajaxa użytkownik kliknie *Dodaj*, formularz zostanie przesłany do skryptu PHP.

2 Na stronach 69 – 74 napiszemy **skrypt PHP**, który obsługuje formularz HTML. **Skrypt dokonuje walidacji danych formularza**, a następnie zwraca wyniki.

W przypadku nieprawidłowego wypełnienia dowolnego pola skrypt PHP wyświetla komunikat o błędzie.

3 Na koniec, tak jak wcześniej, **przetestujemy** napisane elementy aplikacji, aby uzyskać pewność, że wszystko działa poprawnie.

Ten formularz służy do wprowadzania danych dotyczących pracownika (wszystkie pola są obowiązkowe):

IMIE

NAZWISKO

ADRES E-MAIL

WYDZIAŁ

NUMER WEWNĘTRZNY

Nowy pracownik - Mozilla Firefox

plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Nowy pracownik

Wystąpiły następujące błędy :

- Należy podać prawidłową wartość w polu: imię.
- Należy podać prawidłową wartość w polu: nazwisko.
- Należy podać prawidłową wartość w polu: adres email.
- Należy podać prawidłową wartość w polu: numer wewnętrzny.

Na końcu tego rozdziału, w punkcie „Informacje dodatkowe”, umieścimy dodatkowe dane, wskazówki i zalecenia związane z wykonanymi czynnościami.

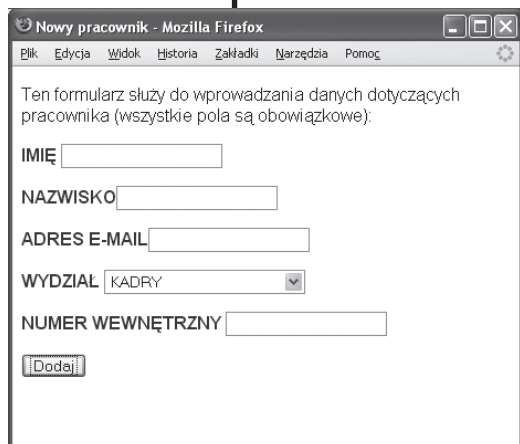
W następnym rozdziale do utworzonych stron dodamy elementy technologii Ajax.

utworzenie strony html

Rozpocniemy od utworzenia **nowej strony HTML** w dowolnym edytorze tekstu.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <meta http-equiv="content-type" content="text/html;
↳ charset=windows-1250" />
  <title>Nowy pracownik</title>
  <style type="text/css" media="all">@import "style.css";</style>
</head>
<body>
</body>
</html>
```

Element `<title>` opisuje przeznaczenie strony.



The screenshot shows a browser window with the title 'Nowy pracownik - Mozilla Firefox'. The address bar is empty. The menu bar includes 'Plik', 'Edycja', 'Widok', 'Historia', 'Zakładki', 'Narzędzia', and 'Pomoc'. The main content area displays a form with the following fields and controls:

- Text: 'Ten formularz służy do wprowadzania danych dotyczących pracownika (wszystkie pola są obowiązkowe):'
- Text input: 'IMIĘ'
- Text input: 'NAZWISKO'
- Text input: 'ADRES E-MAIL'
- Dropdown menu: 'WYDZIAŁ' with 'KADRY' selected
- Text input: 'NUMER WEWNĘTRZNY'
- Submit button: 'Dodaj'

Arkusze stylów nadaje stronie taki sam wygląd jak pozostałym stronom w serwisie.

utworzenie formularza

1 Zdefiniuj komunikat informujący o tym, do czego służy formularz i jak należy z niego korzystać.

```
...  
<body>  
<p>Ten formularz służy do wprowadzania danych dotyczących pracownika  
> (wszystkie pola są obowiązkowe):</p>  
<form action="dodaj_pracownika.php" method="post" id="prac_form">  
</form>  
</body>  
...
```

2 W przypadku przeglądarek nieobsługujących Ajaksa formularz będzie przesłany do skryptu *dodaj_pracownika.php*, który napiszemy w dalszej części rozdziału.

3 Formularz wykorzystuje metodę POST, a nie GET (zobacz „Informacje dodatkowe” na stronie 77).

4 Dla warstwy ajaksowej istotne znaczenie ma **wartość atrybutu id formularza (prac_form)**.

utworzenie elementów formularza

1 Do wprowadzania imienia, nazwiska i adresu e-mail pracownika należy utworzyć pola tekstowe.

```
...  
<form action="dodaj_pracownika.php" method="post" id="prac_form">  
<p><label class="tytul" id="imie_etykieta">Imię<input type="text"  
↳ id="imie" name="imie"/></label> </p>  
<p><label class="tytul" id="nazwisko_etykieta">Nazwisko<input  
↳ type="text" id="nazwisko" name="nazwisko" /></label> </p>  
<p><label class="tytul" id="email_etykieta">Adres e-mail<input  
↳ type="text" id="email" name="email" /></label>  
</p>
```

utworzenie elementów formularza cd.

2 Wydział pracownika można wybrać za pomocą **rozwijanego menu**.

```
<p><label class="tytul" id="wydzial_id_etykieta">Wydział<select  
↳ id="wydzial_id" name="wydzial_id">  
  <option value="1">Kadry</option>  
  <option value="2">Księgowość</option>  
  <option value="3">Marketing</option>  
  <option value="4">Wydział pomocniczy</option>  
</select></label> </p>
```

3 Kolejne pole tekstowe służy do wprowadzania numeru telefonu wewnętrznego pracownika.

```
<p><label class="tytul" id="telefon_wewn_etykieta">Numer  
↳ wewnętrzny<input type="text" id="telefon_wewn" name="telefon_  
↳ wewn" /></label> </p>  
<p><input name="dodaj" type="submit" value="Dodaj" /></p>  
</form>  
...
```

4 Etykiety informują o przeznaczeniu poszczególnych elementów (zobacz „Informacje dodatkowe” na stronie 77).

5 Przycisk przesyłania formularza ma wartość *Dodaj*.

6 Skrypt należy zapisać w pliku *dodaj_pracownika.html* i umieścić w tym samym katalogu co pozostałe strony aplikacji.

Nowy pracownik - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Ten formularz służy do wprowadzania danych dotyczących pracownika (wszystkie pola są obowiązkowe):

IMIĘ

NAZWISKO

ADRES E-MAIL

WYDZIAŁ

NUMER W

Dodaj

rozpoczęcie strony php

Pierwszy skrypt PHP obsługuje operację przesłania formularza bez wykorzystania technologii Ajax. Powinien to być jednocześnie prawidłowy dokument HTML, dlatego należy go rozpocząć od poniższych znaczników:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<head>
  <meta http-equiv="content-type" content="text/html;
↳ charset=windows-1250" />
  <title>Nowy pracownik</title>
  <style type="text/css" media="all">@import "style.css";</style>
</head>
<body>
<h1>Nowy pracownik</h1>
<?php # dodaj_pracownika.php
require_once('mysql.inc.php');
mysql_close($dbc);
?>
</body>
</html>
```

- Wszystkie własności strony zostaną zdefiniowane pomiędzy znacznikami PHP.

Strona musi nawiązać połączenie z serwerem MySQL, dlatego należy załadować skrypt *mysql.inc.php*.

Przed zakończeniem działania skryptu połączenie MySQL musi być zamknięte.

Skrypt należy zapisać w pliku *dodaj_pracownika.php* i umieścić w tym samym katalogu co plik *dodaj_pracownika.html*.

walidacja danych formularza

Przed wykorzystaniem w zapytaniu danych wprowadzonych w formularzu należy przeprowadzić ich walidację. W skrypcie PHP należy wprowadzić poniższy kod:

```
...
require_once('mysql.inc.php');
$bledy = array();
if (!empty($_POST['imie'])) {
    $fn = mysql_real_escape_string($_POST['imie'], $dbc);
} else {
    $bledy[] = 'imię';
}
```

1 W tablicy **\$bledy** będą zapisane wszystkie błędy, jakie wystąpiły podczas walidacji danych w formularzu.

2 Walidacja pól tekstowych polega na **sprawdzeniu, czy nie są one puste** (zobacz „Informacje dodatkowe” na stronie 77).

```
if (!empty($_POST['nazwisko'])) {
    $ln = mysql_real_escape_string($_POST['nazwisko'], $dbc);
} else {
    $bledy[] = 'nazwisko';
}
```

```
if (!empty($_POST['email'])) {
    $e = mysql_real_escape_string($_POST['email'],$dbc);
} else {
    $bledy[] = 'adres email';
}
```

3 W celu uzyskania pewności, że dane można bezpiecznie wykorzystać w zapytaniu, są one przetwarzane w funkcji `mysql_real_escape_string()`.

walidacja danych formularza cd.

```
if (isset($_POST['wydzial_id']) && is_numeric($_POST['wydzial_id']) &&
↳ ($_POST['wydzial_id']>0)) {
    $idw = (int) $_POST['wydzial_id'];
} else {
    $bledy[] = 'wydział';
}
```

4 **Wartości liczbowe**, takie jak identyfikator wydziału i numer telefonu, **muszą być liczbami dodatnimi**.

```
if (isset($_POST['telefon_wewn']) && is_numeric($_POST['telefon_wewn'])
↳ && ($_POST['telefon_wewn'] > 0)) {
    $wewn = (int) $_POST['telefon_wewn'];
} else {
    $bledy[] = 'numer wewnętrzny';
}
mysql_close($dbc);
...
```

5 Aby można było je bezpiecznie wykorzystywać w zapytaniu, **dla wartości numerycznych wykonuje się rzutowanie typów** do danych typu integer.

6 Każda nieudana operacja walidacji powoduje dodanie wpisu do tablicy \$bledy.

aktualizacja bazy danych

Zakładając, że dane wprowadzone w formularzu pomyślnie przeszły przez procedury walidacji, należy uruchomić zapytanie INSERT.

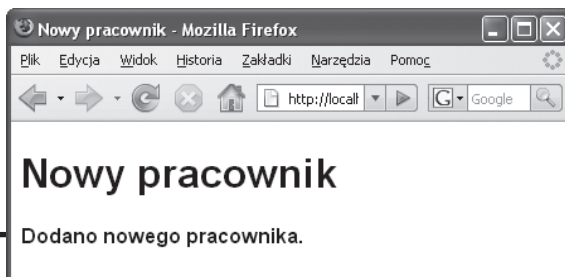
- 1 Jeśli nie było błędów, to ten warunek będzie prawdziwy (ponieważ zmienna \$bledy będzie pusta).

```
...
    $bledy[] = 'numer wewnętrzny';
}
if (!$bledy) {
    $q = "INSERT INTO pracownicy VALUES (NULL, $idw, '$fn', '$ln',
    'e', $wewn)";
    $r = mysql_query($q, $dbc);
    if (mysql_affected_rows($dbc) == 1) {
        echo '<p><strong>Dodano nowego pracownika.</strong></p>';
    }
    mysql_close($dbc);
    ...
}
```

- 2 Instrukcja INSERT dodaje nowego pracownika do bazy danych, wykorzystując dane przetworzone za pomocą procedur walidacyjnych (zobacz „Informacje dodatkowe” na stronie 78).

- 3 Funkcja `mysql_affected_rows()` zwraca liczbę rekordów, których dotyczyło zapytanie. W przypadku zapytania wykorzystanego w tym skrypcie liczba ta będzie wynosiła 1, bowiem dodano jeden nowy rekord.

- 4 Wyniki działania skryptu są wyświetlane na ekranie.



wyświetlenie komunikatów o błędach

Na koniec, jeśli jest taka potrzeba, należy wyświetlić użytkownikom informacje o błędach, które wystąpiły.

1 Pierwsza klauzula `else` dotyczy sytuacji, kiedy zapytanie nie dotyczyło jednego wiersza. Zazwyczaj jest to sygnał wystąpienia błędu składniowego (zobacz „Informacje dodatkowe” na stronie 78).

```
...
    echo '<p><strong>Dodano nowego pracownika.</strong></p>';
  } else { // Wykonanie zapytania nie powiodło się.
    echo '<p class="blad">Nie można dodać pracownika z powodu błędu
↳ systemowego.</p>';
  }
} else { // Błędy!
  echo '<p>Wystąpiły następujące błędy :</p><ul class="blad">';
  foreach ($bledy as $e) {
    echo "<li>Należy podać prawidłową wartość w polu:$e.</li>\n";
  }
  echo '</ul>';
}
mysql_close($dbc);
...
```

2 Druga klauzula `else` uruchamia się, jeśli dane nie przejdą pomyślnie przez wszystkie testy walidacji.

3 Ponieważ błędy są zapisane w tablicy, najłatwiejszym sposobem uzyskania dostępu do wszystkich błędów jest przetwarzanie ich w pętli.

Nowy pracownik

Wystąpiły następujące błędy :

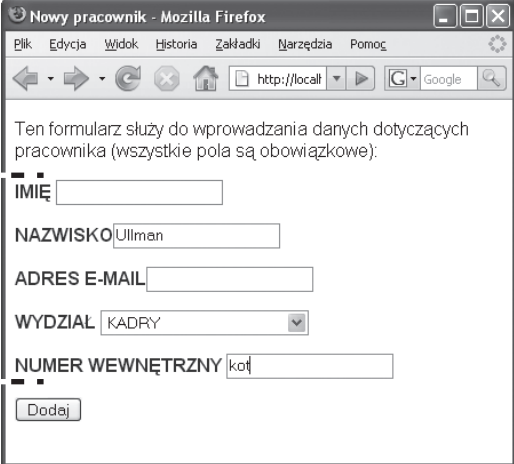
- Należy podać prawidłową wartość w polu:imię.
- Należy podać prawidłową wartość w polu:nazwisko.
- Należy podać prawidłową wartość w polu:adres email.
- Należy podać prawidłową wartość w polu:numer telefonu.

testowanie wersji nieajksowej

1 Załaduj stronę HTML w przeglądarce WWW, aby przetestować aplikację w dotychczasowym kształcie.

Aby strona zadziałała, adres **musi zaczynać się od http://**.

2 Wypełnij **częściowo** formularz i kliknij *Dodaj*.



Nowy pracownik - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

http://local

Ten formularz służy do wprowadzania danych dotyczących pracownika (wszystkie pola są obowiązkowe):

IMIĘ

NAZWISKO

ADRES E-MAIL

WYDZIAŁ

NUMER WEWNĘTRZNY



Nowy pracownik - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

http://localhost/ajax1/dodaj_pracownika.php

Nowy pracownik

Wystąpiły następujące błędy :

- Należy podać prawidłową wartość w polu:imię.
- Należy podać prawidłową wartość w polu:adres email.
- Należy podać prawidłową wartość w polu:numer wewnętrzny.

3 Skrypt PHP powinien wyświetlić wyniki.

testowanie wersji nieajaksowej cd.

4 Powróć do formularza, wypełnij go całkowicie i kliknij *Dodaj*.

Nowy pracownik - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

http://localhost/ Google

Ten formularz służy do wprowadzania danych dotyczących pracownika (wszystkie pola są obowiązkowe):

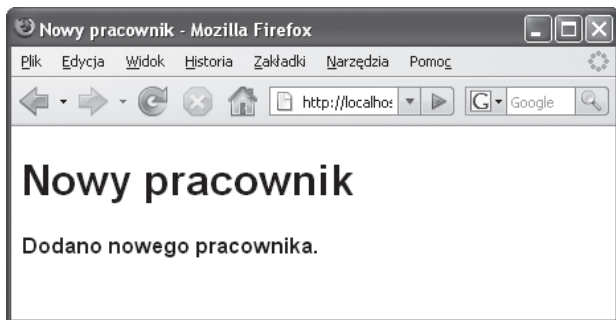
IMIĘ

NAZWISKO

ADRES E-MAIL

WYDZIAŁ

NUMER WEWNĘTRZNY



5 Tak jak wcześniej, skrypt PHP wyświetla wyniki.

utworzenie formularza str. 66

- Metody POST, ogólnie rzecz biorąc, powinno się używać w przypadku, gdy operacja przesyłania formularza oddziałuje na witrynę. Na przykład użycie formularza *Nowy pracownik* dodaje nowy rekord do bazy danych.
- Metodę GET zazwyczaj wykorzystuje się w celu zażądania informacji. Na przykład powinno się ją zastosować przy żądaniu informacji o pracownikach w określonym wydziale.

utworzenie elementów formularza str. 67

- Etykiety formularza spełniają dwa cele. Po pierwsze, informują użytkowników o przeznaczeniu określonych elementów. Po drugie, będą modyfikowane z wykorzystaniem JavaScript w celu wskazania błędów w ajaksowej wersji formularza.

walidacja danych formularza str. 70

- Sposób walidacji danych formularza zależy od typu danych (liczby, ciągi znaków itp.) oraz spodziewanych wartości (liczba dodatnia, adres e-mail itp.). Dla potrzeb walidacji nazwiska osoby często wystarczy się upewnić, czy wprowadzono jakąkolwiek wartość.
- Dokładniejsza walidacja adresu e-mail obejmuje potwierdzenie, że spełnia on **wzorzec wyrażenia regularnego**. Tego rodzaju walidację należy przeprowadzać znacznie częściej dla adresów e-mail niż dla nazwisk, ponieważ adresy e-mail muszą spełniać ściśle reguły.
- Aby poprawić bezpieczeństwo tego systemu, można by zastosować funkcję `strip_tags()` dla operacji wprowadzania danych tekstowych. Wykorzystanie tej funkcji pozwala zapobiegać atakom za pomocą skryptów krzyżowych (ang. *cross-site scripting attacks* — XSS).
- Funkcja `mysql_real_escape_string()` zapewnia specyficzną dla języka ochronę dotyczącą wartości tekstowych wykorzystywanych w zapytaniach.

aktualizacja bazy danych str. 73

- Istnieje dokładniejszy sposób zapisania zapytania INSERT. W tej składni należy określić kolumny, których dotyczy operacja. W przypadku zapytania używanego w tym rozdziale byłaby to instrukcja **INSERT INTO pracownicy (wydzial_id, imie, nazwisko, email, telefon_wewn) VALUES (\$idw, '\$fn', '\$ln', '\$e', \$wewn).**

wyświetlenie komunikatów o błędach str. 74

- Aby debugować problemy dotyczące PHP i MySQL, należy **wyświetlić zapytanie** w celu przeanalizowania jego składni. Należy również wywołać funkcję `mysql_error()`, aby przekonać się, jakie problemy zgłasza baza danych.
- Witryny produkcyjne nigdy nie powinny ujawniać szczegółowych komunikatów o błędach, na przykład uruchamianych zapytań lub błędów bazy MySQL. Informacje te można wykorzystywać dla potrzeb debugowania, ale potem należy je ukryć w wersji produkcyjnej (która powinna być pozbawiona błędów).