



DO NOWEJ  
PODSTAWY PROGRAMOWEJ

## Część 4

Tworzenie aplikacji internetowych

### Kwalifikacja EE.09

Programowanie, tworzenie  
i administrowanie stronami  
internetowymi i bazami danych



**Podręcznik do nauki zawodu  
technik informatyk**

Jolanta Pokorska

**Helion**

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba

Projekt okładki: Jan Paluch

Fotografia na okładce została wykorzystana za zgodą Shutterstock.

Helion SA

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

[http://helion.pl/user/opinie?e094ti\\_ebook](http://helion.pl/user/opinie?e094ti_ebook)

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-7052-4

Copyright © Helion 2020

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• Poleć książkę na Facebook.com</li> <li>• Kup w wersji papierowej</li> <li>• Ocen książkę</li> </ul> | <ul style="list-style-type: none"> <li>• Księgarnia internetowa</li> <li>• Lubię to! » Nasza społeczność</li> </ul> |
|--|---|

# Spis treści

<b>Wstęp</b>	5
<b>Rozdział 1. Aplikacje internetowe</b>	7
1.1. Wprowadzenie	7
1.2. Wzorce projektowe	8
1.3. Serwery aplikacji internetowych	10
1.4. Pakiet XAMPP	10
1.5. Pytania i zadania	15
<b>Rozdział 2. Język JavaScript</b>	17
2.1. Wprowadzenie	17
2.2. Struktura języka JavaScript	17
2.3. Składnia języka JavaScript	20
2.4. Instrukcje sterujące	31
2.5. Funkcje	41
2.6. Obiekty	49
2.7. Obiekty wbudowane języka JavaScript	56
2.8. Obiekty DOM	65
2.9. Obsługa zdarzeń	80
2.10. Wykorzystanie skryptów na stronie internetowej	85
2.11. Walidacja formularzy	91
2.12. Pytania i zadania	102
<b>Rozdział 3. Biblioteka jQuery</b>	105
3.1. Opis biblioteki	105
3.2. Zdarzenia biblioteki jQuery	115
3.3. Zastosowanie biblioteki jQuery na stronie internetowej	118
3.4. Walidacja formularzy	134
3.5. Pytania i zadania	137
<b>Rozdział 4. Język PHP</b>	139
4.1. Wprowadzenie	139
4.2. Struktura języka PHP	141
4.3. Składnia języka PHP	143



4.4. Instrukcje sterujące .....	160
4.5. Funkcje .....	173
4.6. Funkcje wbudowane .....	181
4.7. Funkcje obsługi plików .....	195
4.8. Obsługa formularzy .....	208
4.9. Pliki cookies i sesje .....	219
4.10. Bazy danych w PHP .....	235
4.11. Biblioteka PDO .....	258
4.12. Pytania i zadania .....	261
<b>Rozdział 5. Walidacja kodu aplikacji .....</b>	<b>263</b>
5.1. Wprowadzenie .....	263
5.2. Testy aplikacji .....	264
5.3. Debugowanie aplikacji .....	266
5.4. Pytania i zadania .....	270
<b>Rozdział 6. Dokumentowanie aplikacji .....</b>	<b>271</b>
6.1. Komentarze .....	271
6.2. Tworzenie dokumentacji programu .....	274
6.3. Automatyczne generowanie dokumentacji użytkownika .....	277
6.4. Pytania i zadania .....	279
<b>Bibliografia .....</b>	<b>280</b>
<b>Skorowidz .....</b>	<b>281</b>

# Wstęp

*Kwalifikacja EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Część 4. Tworzenie aplikacji internetowych. Podręcznik do nauki zawodu technik informatyk jest czwartym z grupy podręczników przygotowanych dla kwalifikacji EE.09. Programowanie, tworzenie i administrowanie stronami internetowymi i bazami danych. Kwalifikacja ta wraz z kwalifikacją EE.08. Montaż i eksploatacja systemów komputerowych, urządzeń peryferyjnych i sieci obejmuje efekty kształcenia, których osiągnięcie jest niezbędne do uzyskania dyplomu potwierdzającego kwalifikacje zawodowe w zawodzie technik informatyk.*

Treści zawarte w czwartej części podręcznika zostały oparte na podstawie programowej kształcenia w zawodach wprowadzonej rozporządzeniem Ministra Edukacji Narodowej z dnia 31 marca 2017 r.

Obejmują one zagadnienia teoretyczne prowadzące do uzyskania wymienionych w podstawie programowej efektów kształcenia, projekty różnych zadań oraz ich realizację praktyczną. Tak skonstruowany podręcznik pomaga uczniowi w zdobywaniu wymaganej wiedzy oraz umożliwia mu samodzielne poszerzanie umiejętności.

Podręcznik składa się z sześciu rozdziałów. Ich budowa pozwala na realizację treści programowych w sposób wybrany przez nauczyciela oraz umożliwia samodzielne rozwijanie umiejętności przez uczniów.

Rozdział 1., „Aplikacje internetowe”, zawiera omówienie podstawowych zagadnień z zakresu użycia aplikacji internetowych. Dotyczy efektów związanych z tworzeniem oraz wykorzystywaniem aplikacji webowych. Efektami tymi są: rozpoznawanie technologii działających po stronie klienta i serwera, identyfikowanie wzorców projektowych, korzystanie z frameworków, instalowanie serwerów WWW.

Rozdział 2., „Język JavaScript”, zawiera omówienie zagadnień z zakresu stosowania skryptów wykonywanych po stronie klienta przy tworzeniu aplikacji internetowych. Dotyczy efektów związanych ze stosowaniem funkcji, obiektów i metod w języku JavaScript, z tworzeniem własnych funkcji, obiektów i metod oraz testowaniem powstającej aplikacji i modyfikowaniem jej kodu źródłowego. Efektami tymi są: identyfikowanie operatorów arytmetycznych, bitowych i logicznych, stosowanie obsługi zdarzeń myszy i klawiatury, definiowanie skryptów obsługujących formularze i kontrolki HTML, wykorzystywanie mechanizmów walidacji formularzy HTML, korzystanie z funkcji modelu DOM, analizowanie poprawności tworzonych obiektów i metod.

Rozdział 3., „Biblioteka jQuery”, zawiera omówienie zagadnień z zakresu wykorzystania biblioteki jQuery do tworzenia własnych aplikacji. Dotyczy efektów związanych ze stosowaniem funkcji, metod i obiektów biblioteki jQuery, tworzeniem własnych funkcji, obiektów i metod z użyciem biblioteki jQuery oraz wykorzystywaniem ich

na stronie internetowej. Efektami tymi są: identyfikowanie i stosowanie instrukcji, metod, obiektów i zdarzeń biblioteki jQuery, używanie odpowiednich instrukcji, procedur i metod biblioteki jQuery do tworzenia efektów na stronie internetowej oraz do walidacji formularzy.

Rozdział 4., „Język PHP”, zawiera omówienie podstawowych zagadnień z zakresu stosowania skryptów wykonywanych po stronie serwera przy tworzeniu aplikacji internetowych. Dotyczy efektów związanych z wykorzystaniem wbudowanych typów danych, stosowaniem funkcji i tworzeniem własnych funkcji w języku PHP oraz testowaniem powstającej aplikacji i modyfikowaniem jej kodu źródłowego. Efektami tymi są: identyfikowanie operatorów arytmetycznych, bitowych i logicznych, tworzenie własnych typów danych, stosowanie instrukcji i funkcji języka PHP do określonych zadań, wykorzystywanie operacji na plikach i katalogach, tworzenie i walidacja formularzy, wysyłanie danych z formularza, używanie plików cookie i sesji, stosowanie bibliotek do obsługi bazy danych, analizowanie poprawności tworzonych aplikacji.

Rozdział 5., „Walidacja kodu aplikacji”, zawiera omówienie zagadnień z zakresu testowania i debugowania kodu aplikacji internetowej. Dotyczy efektów związanych z analizą testów tworzonej aplikacji oraz sposobami debugowania aplikacji. Efektami tymi są: analiza błędów w kodzie źródłowym programu, poprawianie błędów, stosowanie debugera w przeglądarce internetowej, rozpoznawanie rodzajów testów.

Rozdział 6., „Dokumentowanie aplikacji”, zawiera omówienie zagadnień z zakresu dokumentowania tworzonej aplikacji. Dotyczy efektów związanych z opracowywaniem dokumentacji programu. Efektami tymi są: identyfikowanie sposobów dokumentowania powstającej aplikacji, tworzenie dokumentacji programu, przygotowywanie instrukcji użytkownika, stosowanie komentarzy w celu automatycznego generowania dokumentacji użytkownika.



# Aplikacje internetowe



## 1.1. Wprowadzenie

**Aplikacja internetowa** to program, który pracuje na serwerze i komunikuje się z użytkownikiem poprzez sieć komputerową z wykorzystaniem przeglądarki internetowej. Przeglądarka internetowa w takim przypadku pełni funkcję interfejsu użytkownika. W aplikacji internetowej zakłada się interakcję z użytkownikiem, korzystanie z baz danych i innych usług, często rozproszonych, umieszczonych na wielu różnych serwerach.

Istotnymi cechami aplikacji internetowej są łatwość i szybkość dotarcia do informacji (dowolny komputer podłączony do internetu lub sieci LAN i przeglądarka stron internetowych) oraz bezpieczeństwo danych (login i hasło, szyfrowanie połączenia oraz system uprawnień).

Zalety aplikacji internetowych to:

- dostępność dla wszystkich zainteresowanych bez ograniczeń związanych z czasem i miejscem,
- praktycznie brak konieczności instalowania dodatkowego oprogramowania; aby korzystać z aplikacji internetowej, wystarczy przeglądarka internetowa,
- brak konieczności zakupu dodatkowego sprzętu komputerowego; aplikacja internetowa może zostać zainstalowana na serwerze dostępnym w internecie,
- łatwość utrzymania i modernizacji, ponieważ zmiany i aktualizacje w aplikacji są wykonywane na serwerze bez udziału użytkowników aplikacji,
- łatwość integracji aplikacji internetowej z innymi usługami,
- niższe koszty wytworzenia, uruchomienia i utrzymania aplikacji internetowej w porównaniu z innymi rozwiązaniami; większość technologii stosowanych do tworzenia aplikacji internetowych jest bezpłatna.



## 1.2. Wzorce projektowe

### 1.2.1. Wprowadzenie

**Wzorce projektowe** to sprawdzone w praktyce uniwersalne rozwiązania problemów często pojawiających się podczas projektowania aplikacji. Wzorce projektowe tworzone są najczęściej na bazie programowania obiektowego. Pokazują powiązania i zależności pomiędzy klasami oraz obiektami i ułatwiają tworzenie oraz modyfikację kodu źródłowego. Wprowadzają standaryzację kodu oraz zwiększają jego wydajność i niezawodność. Wzorzec projektowy powinien zawierać oprócz rozwiązania problemu także dokumentację, która wyjaśni cel, sposób działania i zalety danego rozwiązania.

Dokumentacja wzorca projektowego powinna obejmować:

- unikatową nazwę, która pozwoli odwoływać się do wzorca,
- opis celu, czyli czym należy się kierować podczas wyboru wzorca,
- opis sytuacji, w których wzorzec można stosować,
- graficzną reprezentację wzorca w postaci diagramu klas lub interakcji,
- listę klas i obiektów stosowanych we wzorcu,
- opis wzajemnej interakcji klas i obiektów wykorzystywanych we wzorcu,
- wykaz wyników i efektów ubocznych, jakie występują podczas używania wzorca,
- wskazówki dotyczące zastosowania wzorca,
- przykładowy kod w jednym z języków programowania pokazujący zastosowanie wzorca.

Wzorzec projektowy nie jest gotową implementacją rozwiązania. Przypomina szablon, który może być zastosowany w wielu różnych sytuacjach. Wzorce oparte są na praktycznych rozwiązaniach, które zostały zaimplementowane w wybranym języku obiektowym. Mogą przyspieszyć proces projektowania aplikacji przez wykorzystanie wypróbowanych rozwiązań dla sformułowanego problemu.

### 1.2.2. Wzorzec projektowy MVC

**MVC** (ang. *Model-View-Controller*) jest jednym z najczęściej stosowanych wzorców projektowych w aplikacjach internetowych. Głównym założeniem tego wzorca jest podzielenie kodu aplikacji na trzy moduły:

- **Model** — reprezentuje dane (na przykład pobierane z bazy danych).
- **Widok** — reprezentuje interfejs użytkownika.
- **Kontroler** — reprezentuje logikę sterującą aplikacją.

### Model

**Model** zapewnia zestandaryzowany sposób dostępu do danych. Najczęściej stosowany jest do pobierania i modyfikowania rekordów z bazy danych. Pozostała część aplikacji staje się niezależna od tego, skąd i w jaki sposób dane są pobierane. Dane mogą być pobierane z różnych źródeł (na przykład z bazy lub pliku) i nie jest to istotne pod kątem logiki sterującej aplikacją. Tworząc tę logikę, wywołujemy tylko odpowiednie funkcje modelu i w wyniku dostajemy dane do przetworzenia. Model jest elementem opcjonalnym, gdyż nie zawsze korzystamy z danych pobieranych z bazy czy plików.

### Widok

**Widok** reprezentuje to, co będzie widoczne dla użytkownika. Widok oddzielony od logiki aplikacji pozwala na bezproblemową zmianę oprawy graficznej w dowolnym momencie. Zadaniem kontrolera jest przekazanie danych do widoku. Programista tworzący widok nie musi nic wiedzieć na temat logiki programu. Otrzymuje on tylko dane, które należy sformatować. Widok można zmienić, nie ingerując w kod programu.

### Kontroler

**Kontroler** zawiera najważniejszą część kodu, która steruje aplikacją. Odpowiada między innymi za przetwarzanie danych pobranych za pomocą modelu i przekazanie ich poprzez widok użytkownikowi, a także za wykonywanie odpowiednich akcji w zależności od działań użytkownika.

Zaletą korzystania ze wzorca MVC jest uzyskanie gotowej struktury aplikacji oraz uporządkowanie struktury kodu szczególnie przy tworzeniu dużych aplikacji. Wzorzec ten pozwala też na uniknięcie problemów ze znalezieniem określonej funkcjonalności. Umożliwia stosowanie jasnego podziału prac nad projektem. Zmiana bazy danych czy wyglądu strony przestaje być problemem.

Wzorzec MVC często jest traktowany jako wzorzec złożony, który może być rozbijany na prostsze wzorce. Na bazie wzorca MVC powstało również wiele wzorców pochodnych. Obecnie częściej stosowane są wzorce pochodne niż oryginalny wzorzec.

Jedną z pierwszych adaptacji MVC na potrzeby środowiska WWW była platforma Struts przeznaczona dla technologii Java Server Pages.

Firma Microsoft wykorzystuje adaptację MVC zbudowaną na bazie technologii ASP.NET w postaci platformy ASP.NET MVC, która umożliwia tworzenie aplikacji z uwzględnieniem wyraźnie odseparowanych modeli, kontrolerów oraz widoków. Platforma ta udostępnia dużą liczbę specyficznych rozszerzeń oraz konstrukcji programistycznych.

Istnieje wiele frameworków wspomagających programowanie i udostępniających gotową architekturę opartą na MVC. Należą do nich: Zend Framework, Code Igniter, CakePHP, Symfony.

## 1.3. Serwery aplikacji internetowych

**Serwery WWW** (ang. *web server*) to programy, których zadaniem jest obsługa żądań protokołu komunikacyjnego HTTP. Przeważnie serwery WWW instalowane są na przeznaczonych do tego komputerach odgrywających rolę serwerów internetowych. Można również instalować serwer WWW na stacji roboczej (lokalnie) w celu na przykład testowania tworzonych aplikacji. Aby pobrać określoną stronę internetową, klient łączy się z serwerem stron WWW za pośrednictwem przeglądarki internetowej. Do udostępniania dynamicznie utworzonych stron internetowych oraz danych przechowywanych w bazach danych serwer WWW może korzystać z dodatkowego oprogramowania, na przykład z PHP (i za jego pomocą z MySQL).

Podstawowe zadania serwera WWW to:

- przyjęcie połączenia TCP,
- dekodowanie żądań protokołu HTML,
- realizacja żądania (przesłanie zasobu, wykonanie programu),
- obudowanie dokumentu nagłówkami HTTP,
- wysłanie danych do klienta.

### 1.3.1. Serwer Apache

Apache jest najpopularniejszym i bezpłatnym serwerem WWW dostępnym dla wielu systemów operacyjnych (Linux, Windows, macOS). Razem z interpreterem języka skryptowego PHP i bazą danych MySQL stanowi jedno z najpopularniejszych środowisk — tak zwaną platformę AMP.

Najważniejsze cechy serwera Apache to:

- wielowątkowość,
- skalowalność,
- bezpieczeństwo,
- kontrola dostępu (uwierzytelnianie),
- uruchamianie witryn internetowych generowanych dynamicznie,
- niezależność od oprogramowania i sprzętu,
- dostępność jako open source.

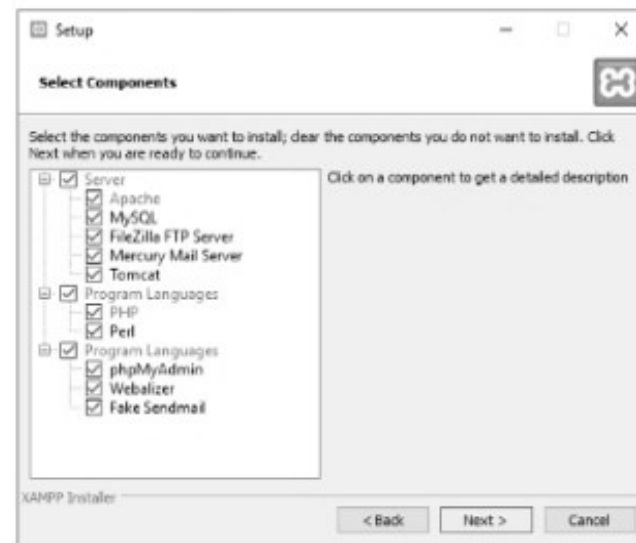
Serwer Apache jest podzielony na wiele modułów. Podczas instalowania serwera można wybrać tylko potrzebne moduły lub zainstalować wszystkie.

## 1.4. Pakiet XAMPP

Serwer stron internetowych Apache z serwerem baz danych MySQL lub MariaDB oraz językiem PHP tworzy wydajną i niezależną od systemu platformę do budowania dynamicznych stron WWW. Platforma ta określana jest jako platforma AMP. W systemach

Linux znana jest pod nazwą **LAMP**, a w systemie Windows pod nazwą **WAMP**. Istnieje również wieloplatformowa wersja, nazywana **XAMPP**. Pakiet ten może być wykorzystywany do testowania działania aplikacji na lokalnym komputerze.

Poza serwerem aplikacji internetowych Apache, serwerem baz danych MySQL i językiem skryptowym PHP w pakiecie XAMPP jest dostępny język Perl oraz zestaw narzędzi, między innymi phpMyAdmin (rysunek 1.1).



Rysunek 1.1. Komponenty dostępne w pakiecie XAMPP

Plik instalacyjny w wersji dla systemów Linux oraz Windows można pobrać ze strony <http://www.apachefriends.org/en/xampp.html>.

### 1.4.1. Instalacja XAMPP w systemie Linux

Przed zainstalowaniem pakietu XAMPP należy sprawdzić, czy dla systemu Linux wcześniej nie zostały zainstalowane serwery Apache i MySQL oraz język PHP. Po pobraniu pliku instalacyjnego (na przykład *xampp-linux-x64-7.3.9-0-installer.run*) należy go uruchomić. W tym celu trzeba wpisać polecenia:

```
chmod +x xampp-linux-x64-7.3.9-0-installer.run
./xampp-linux-x64-7.3.9-0-installer.run
```

W wyniku wykonania podanych poleceń nastąpi uruchomienie pliku instalacyjnego XAMPP. W pierwszym oknie instalatora trzeba wybrać komponenty do zainstalowania. W kolejnych należy klikać przycisk *Next*, aż pakiet zostanie zainstalowany. Po tych czynnościach pojawi się okno z panelem sterującym (rysunek 1.2). Należy wybrać zakładkę *Manage Servers*, aby sprawdzić, czy serwer Apache został automatycznie uruchomiony.





Rysunek 1.2. Panel kontrolny pakietu XAMPP

By uruchomić bazę MySQL, trzeba wybrać opcję *MySQL Database* i kliknąć przycisk *Start*. Po tych czynnościach pakiet jest przygotowany do pracy.

Poprawność instalacji pakietu i uruchomienia serwerów można sprawdzić, wpisując w dowolną przeglądarkę internetową adres <http://localhost/dashboard/>. Nastąpi przekierowanie na serwer i pojawi się widok podobny do pokazanego na rysunku 1.3, co oznacza, że serwer WWW działa prawidłowo.



Rysunek 1.3. Test działania pakietu XAMPP

W pakiecie XAMPP głównym katalogiem serwera WWW jest */opt/lampp/htdocs/*. W tym katalogu powinny być umieszczane tworzone skrypty oraz pliki HTML.

Ponieważ domyślnie zasoby pakietu XAMPP są dostępne dla każdego, należy zabezpieczyć je, tworząc konta użytkowników i definiując hasła dostępu. W tym celu z konsoli trzeba wpisać polecenie:

```
sudo /opt/lampp/lampp security
```

W ramach konfigurowania zabezpieczeń należy zdefiniować oddzielnie hasła dostępu do:

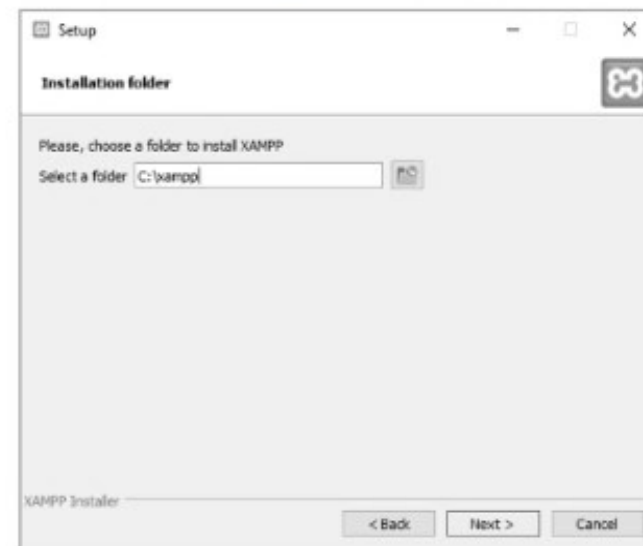
- panelu konfiguracyjnego pakietu XAMPP (dostępny pod adresem <http://opt/lampp/>, konto użytkownika to *lampp*),
- oprogramowania phpMyAdmin (zarządzanie bazą danych przez serwer WWW),
- konta administratora bazy danych MySQL,
- serwera FTP.

Po tych działaniach elementy składowe XAMPP będą w pełni zabezpieczone.

### 1.4.2. Instalacja XAMPP w systemie Windows

Przed zainstalowaniem pakietu XAMPP należy wyłączyć program antywirusowy i sprawdzić ustawienia firewalla. Po pobraniu pliku instalacyjnego (na przykład *xampp-windows-x64-7.3.9-0VC15-installer.exe*) należy go uruchomić.

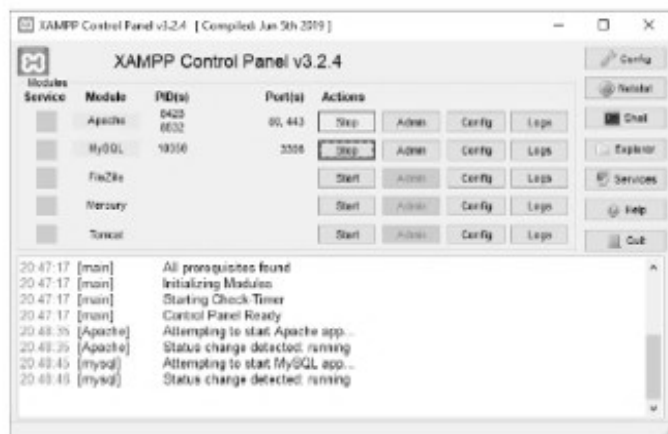
Po uruchomieniu pliku instalacyjnego można wybrać serwery, języki i narzędzia, które będą instalowane. Następnie należy określić folder, w którym pakiet zostanie zainstalowany. Domyślnie jest to folder *xampp* na dysku *C:* (rysunek 1.4).



Rysunek 1.4. Folder instalacyjny pakietu XAMPP

Po wskazaniu folderu docelowego nastąpi wypakowanie plików instalacyjnych i zainstalowanie pakietu. Po zakończeniu instalowania pojawi się pytanie, czy chcemy uruchomić panel kontrolny XAMPP Control Panel, którego zadaniem jest skonfigurowanie zainstalowanego oprogramowania. W panelu tym dostępne są (rysunek 1.5):

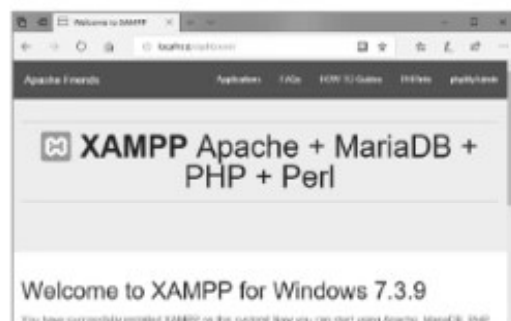
- serwer WWW Apache z interpreterem skryptów w PHP i Perl,
- serwer bazy danych MySQL,



Rysunek 1.5. Panel kontrolny XAMPP Control Panel

- serwer FTP FileZilla,
- serwer poczty elektronicznej Mercury,
- środowisko serwera dla Javy (Java Server Pages) — Java Tomcat.

W celu przetestowania poprawności działania serwera należy w dowolnej przeglądarce wpisać adres <http://localhost/>. Jeżeli pojawi się widok podobny do pokazanego na rysunku 1.6, oznacza to, że serwer WWW działa prawidłowo.



Rysunek 1.6. Test działania pakietu XAMPP

W celu przetestowania działania skryptów na serwerze trzeba odszukać folder *htdocs*. Znajduje się on w folderze instalacyjnym pakietu XAMPP. Standardowo jest to folder *C:\xampp\htdocs*. To katalog główny serwera WWW. W tym folderze będą umieszczane tworzone skrypty oraz pliki HTML. Jeżeli XAMPP został zainstalowany w innej lokalizacji, trzeba znaleźć odpowiednią ścieżkę do katalogu *htdocs*.

## 1.5. Pytania i zadania

### 1.5.1. Pytania

1. Wymień podstawowe zalety wykorzystywania aplikacji internetowych.
2. Omów narzędzia stosowane podczas pracy z aplikacją internetową po stronie klienta.
3. Omów narzędzia stosowane podczas pracy z aplikacją internetową po stronie serwera.
4. Wymień programowe warstwy funkcjonalne, które wchodzi w skład rozszerzonej architektury aplikacji internetowej.
5. Jakie zadania spełnia logika prezentacji w architekturze 4-warstwowej?
6. Z jakich modułów składa się wzorzec projektowy MVC?
7. Wymień podstawowe funkcje realizowane przez serwer WWW.
8. Wymień popularne serwery WWW.
9. Omów podstawowe cechy serwera Apache.
10. Omów podstawowe cechy serwera IIS.