

KUMASI TECHNICAL UNIVERSITY

COMPUTER SCIENCE DEPARTMENT

BACK – END WEB DEVELOPMENT TECHNOLOGY

COURSE CODE: BCT 205

MERCY VICENTIA ADU GYAMFI (MRS)

COURSE OUTLINE

- Course Assessment
- Course Objectives
- Recommended Books
- Motivation
- Course Work
 1. PHP programming environment
 2. MySQL environment

COURSE ASSESSMENT

COURSE ACTIVITY	ALLOCATED MARKS
Attendance	5
Class Test 1- Arranged	5
Class Test 2- Pop Quiz	5
Take-home Assignment	5
Mid-Semester/ Practical	20
Examinations	60

RECOMMENDED BOOKS

- Jon Duckett. (2022). PHP & MySQL: Server-side Web Development, 1st Edition. Wiley
- Robin Nixon. (2021). Learning PHP, MySQL & JavaScript. A Step-by-Step Guide to Creating Dynamic Websites, 6th Edition. O'Reilly Media, Inc.
- Kenneth E. Marks, Oscar Merida, Kara Ferguson. (2021). PHP Web Development with MySQL: A Hands On Approach to Application Programming, php[architect].
- Patrick M. Carey. (2017). New Perspectives HTML5 and CSS3: Comprehensive, 7th Edition. Cengage Learning.

MOTIVATION

- “Learning to code is useful no matter what your career ambitions are.”
- “You will definitely enjoy what you’ve worked hard for — you’ll be happy; and things will go well for you.”
- Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program. (Linus Torvalds)

OBJECTIVES

Students will learn how

- To make pages dynamic based upon user interaction
- Interacting with HTML forms, store and retrieve information from local data sources which include a database
- Learn how PHP can be combined with MySQL to create a very powerful online database engine

RESOURCES NEEDED

Students will should have the following resources :

- Xampp : <https://www.apachefriends.org/>
- Git :<https://git-scm.com/>
- GitHub : <https://github.com/>

BACK-END DEVELOPMENT

- Means working on server-side software, which focuses on everything you can't see on a website.
- Back-end developers ensure the website performs correctly, focusing on databases, back-end logic, application programming interface (APIs), architecture, and servers.
- Full-stack developers are comfortable working with both the front and back ends.

BACK-END DEVELOPMENT TOOLs

- Programming languages:

- Python
- PHP
- JavaScript
- Ruby
- Java
- C#

- Frameworks:

- Laravel
- Django
- Spring
- Ruby on Rails
- Meteor
- Node.js

BACK-END DEVELOPMENT TOOLs

- Databases:

- MongoDB
- MySQL
- Oracle

- Servers:

- Apache
- NGINX
- Lighttpd
- Microsoft IIS

Static Vs. Dynamic Websites

- HTML alone only allows you to create static web pages.
- This approach is fine if you just want to display images of your picture.
- Dynamic website is a web application with the ability to interact with a database and for your web pages to update based on content that can change.
- PHP comes in:
 - The PHP interpreter runs in the webserver.
 - PHP will deliver HTML to the webserver,.
 - PHP is also a programming language, it allows you to build the HTML dynamically.

PHP INTRODUCTION

PHP started out as a small **open source** project that evolved as more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- **Other Names : Personal Home Page, Professional Home Page**
- PHP is a ***server side scripting language*** that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like and C++.

Common uses of PHP

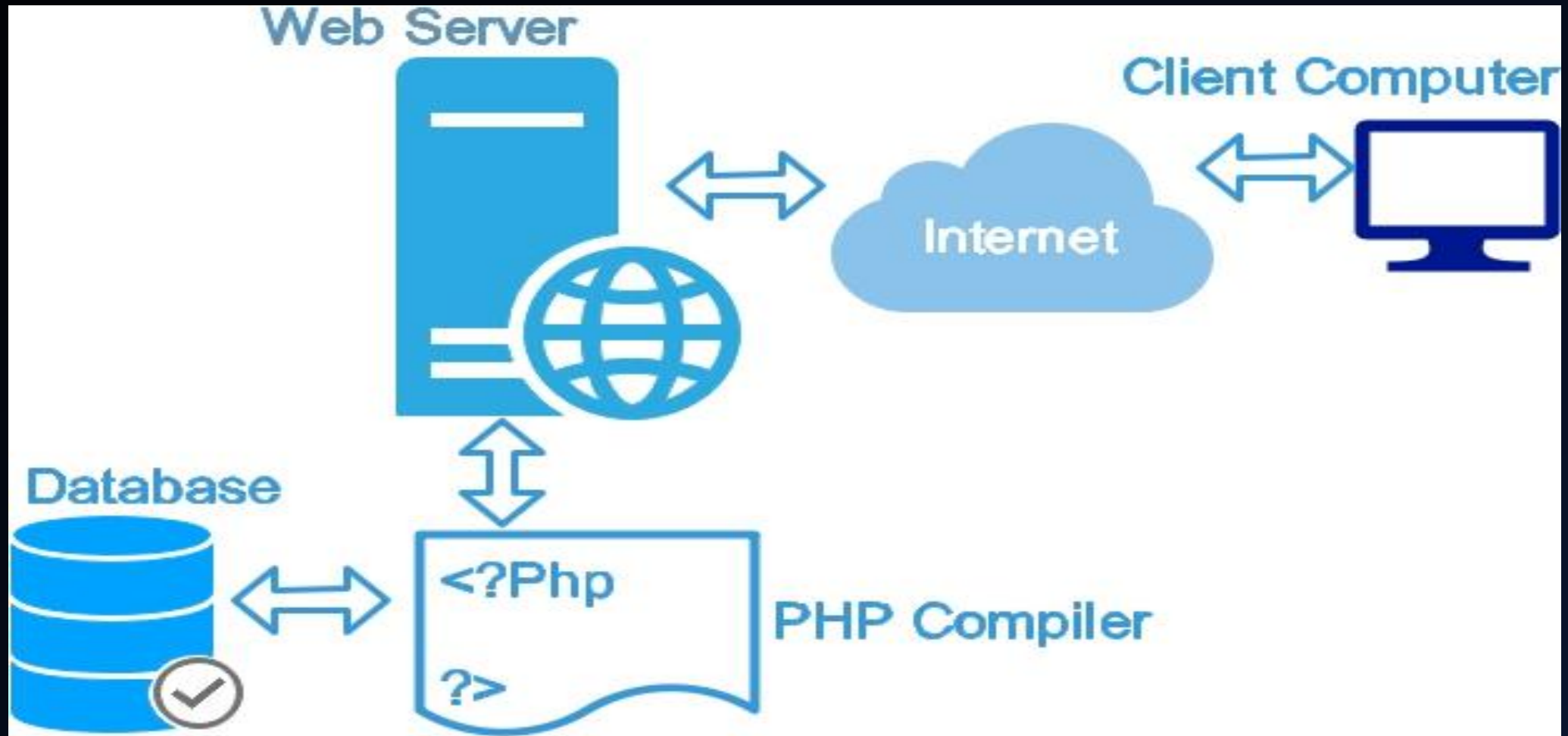
- PHP performs system functions, *i.e.* from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, *i.e.* gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

WEB SERVER



Setting Up a Development Server

- WAMP, MAMP, XAMPP and LAMP are abbreviations for “Windows, Apache, MySQL, and PHP,” “Mac, Apache, MySQL, and PHP,” and “Linux, Apache, MySQL, and PHP.”
- These abbreviations describe a fully functioning setup used for developing dynamic Internet web pages.
- XAMPP is the title used for a compilation of free software. The name is an acronym, with each letter representing one of the five key components. The software packet contains the webserver Apache, the relational database management system MySQL (or MariaDB), and the scripting languages Perl and PHP.
- The initial X stands for the cross operating systems that it works with Linux, Windows, and Mac OS X.

Setting Up a Development Server

- **Apache:** The open source web server Apache is the most widely used server worldwide for delivery of web content. The server application is made available as a free software by the Apache Software Foundation.
- **MySQL/MariaDB:** in MySQL, XAMPP contains one of the most popular relational database management systems in the world In combination with the web server Apache and the scripting language PHP, MySQL offers data storage for web services Current XAMPP versions have replaced MySQL with MariaDB (a community developed fork of the MySQL project, made by the original developers)
- **PHP:** the server side programming language PHP enables users to create dynamic websites or applications. PHP can be installed on all platforms and supports a number of diverse database systems.

Setting Up a Development Server

- **Perl:** the scripting language Perl is used in system administration, web development, and network programming. Like PHP, Perl also enables users to program dynamic web applications.
- Alongside these core components, this free to use Apache distribution contains some other useful tools, which vary depending on your operating system. These tools include the
 - mail server **Mercury** ,
 - the database administration tool **phpMyAdmin** ,
 - the web analytics software solutions **Webalizer** **OpenSSL** , and Apache
 - **Tomcat**
 - the **FTP** servers **FileZilla** or **ProFTPd**

Setting Up a Development Server

- An XAMPP server can be installed and used with a single executable file quickly and easily, functioning as a local test system for Linux, Windows, and Mac OS X
- The software packet contains the same components that are found on common web servers Developers have the chance to test out their projects locally and to transfer them easily to productive systems

Setting Up a Development Server Installing

- **Download:** XAMPP is a release made available by the non profit project Apache Friends and are available for download at:
<https://www.apachefriends.org/download.html>
- **Run .exe file:** Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending .exe
- **Deactivate any antivirus software:** Since an active antivirus program can negatively affect the installation process, it's recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed

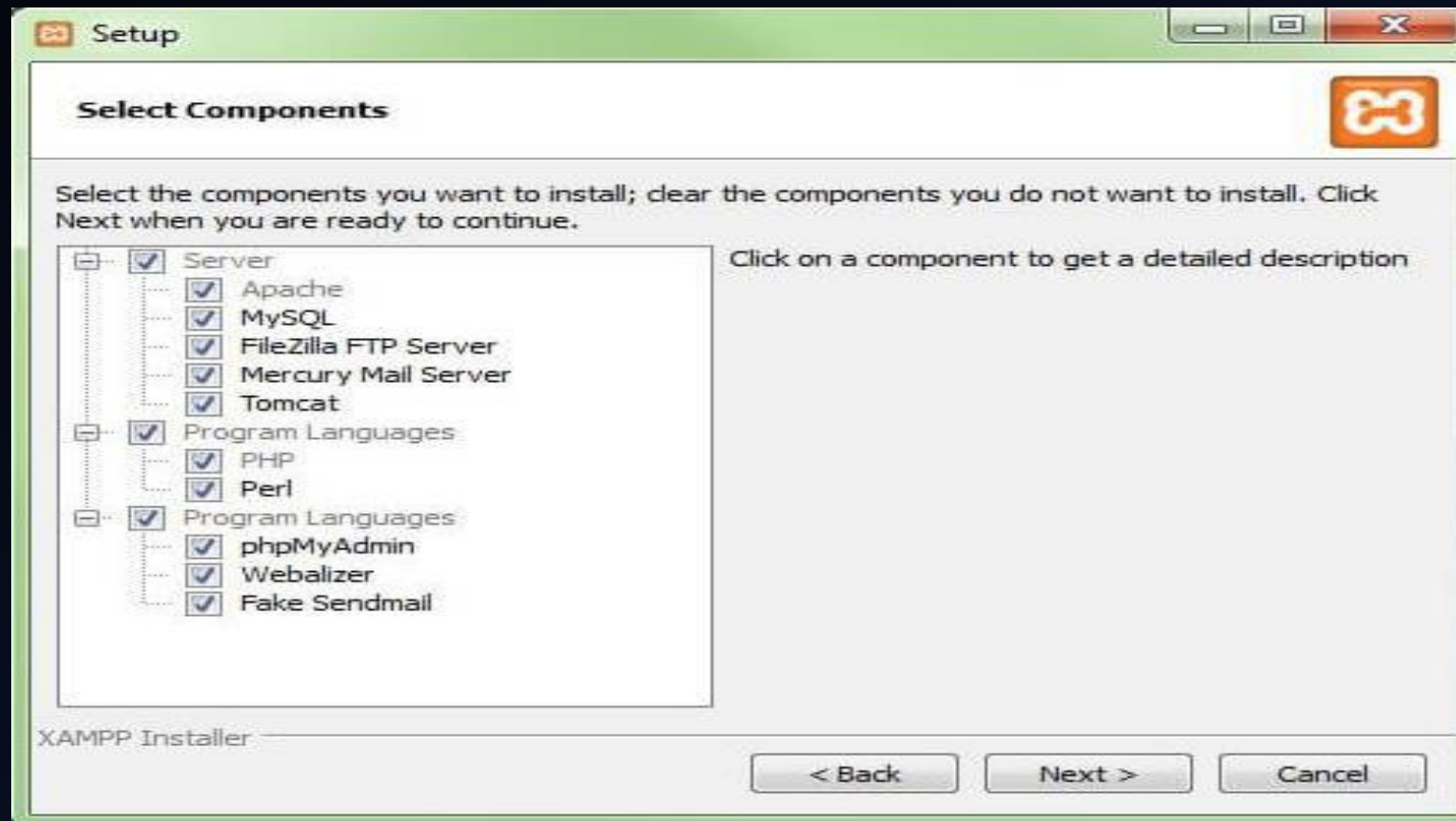
Setting Up a Development Server

- **Start the setup wizard:** After you've opened the .exe file (after deactivating your antivirus program(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on 'Next' to configure the installation settings.



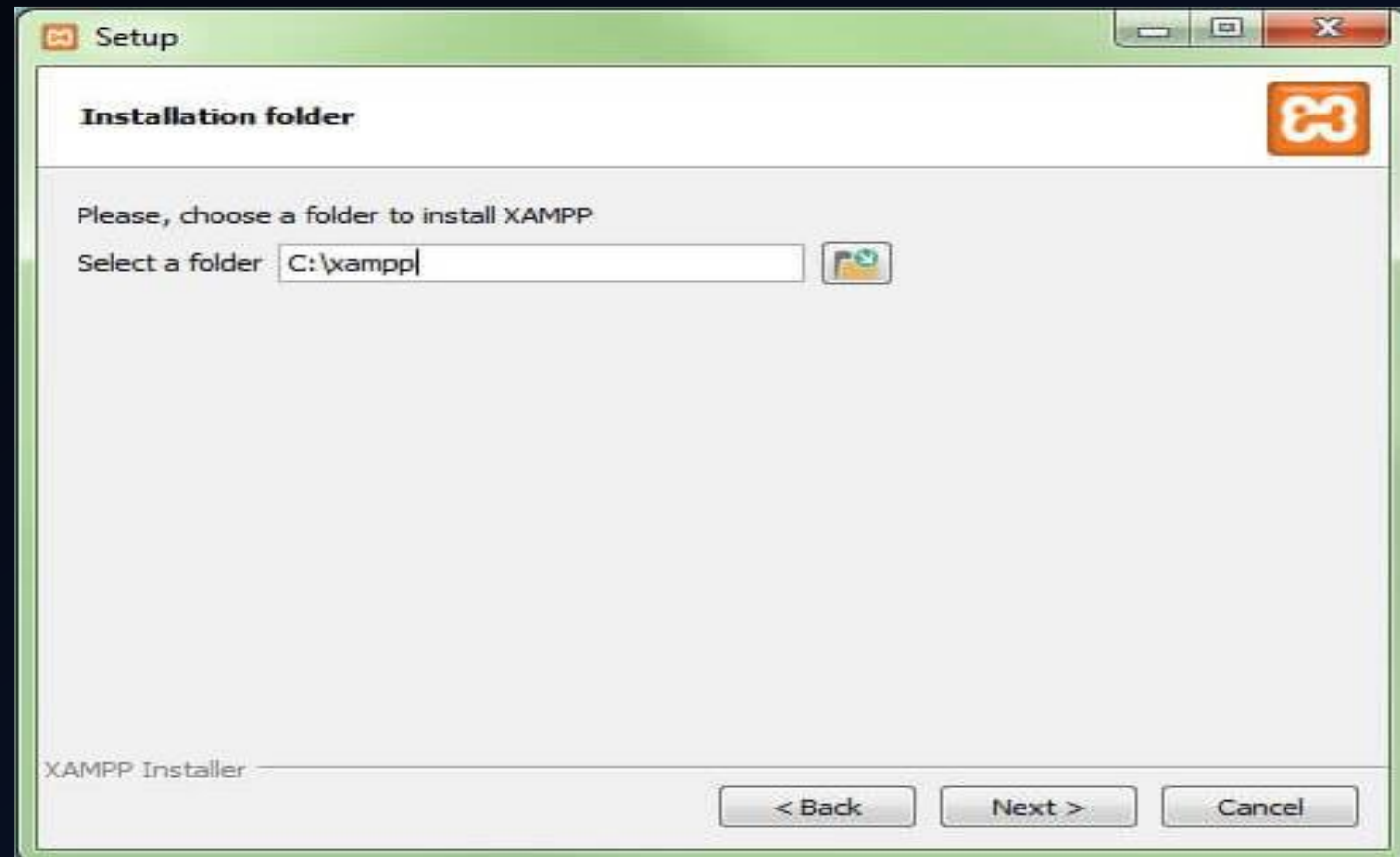
Setting Up a Development Server

- **Choose software components:** Under 'Select Components', you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click 'Next'.



Setting Up a Development Server

- **Choose the installation directory:** In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C for you. After you've chosen a location, click 'Next'.



Setting Up a Development Server

- **Start the installation process:** Once all the aforementioned preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. Choose the installation directory In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed If you opt for the standard setup, then a folder with the name XAMPP will be created under C for you After you've chosen a location, click



Setting Up a Development Server

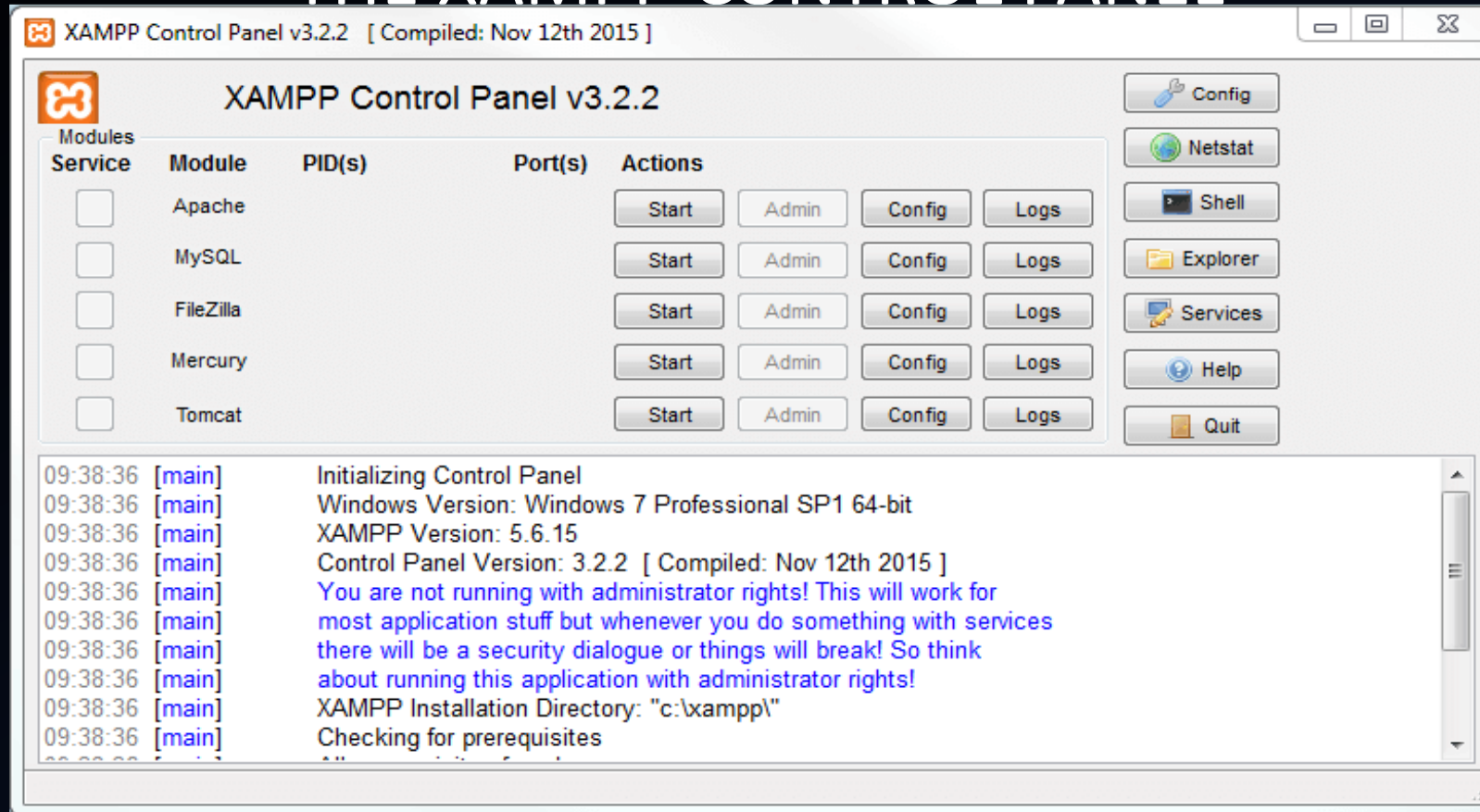
- **Complete installation:** Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.



Setting Up a Development Server

- Controls for the individual components of your test server can be reached through the XAMPP Control Panel The clear user interface logs all actions and allows you to start or stop individual modules with a single The XAMPP Control Panel also offers you various other buttons, including:
 - **Config:** allows you to configure the XAMPP as well as the individual components
 - **Netstat :** shows all running processes on the local computer
 - **Shell:** opens a UNIX shell
 - **Explorer:** opens the XAMPP folder in Windows Explorer
 - **Services:** shows all services currently running in the background
 - **Help:** offers links to user forums
 - **Quit:** closes the XAMPP Control Panel

THE XAMPP CONTROL PANEL



Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under ' You can see which modules have been started because their names are highlighted green under the ' title

Setting Up a Development Server

- SETTING UP XAMPP
- A common source of error connected with Apache is blocked ports . If you're using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it's likely that the Tomcat port is being blocked, meaning the web server can't be started. There are three ways to solve this issue:
- **Change the conflicting port:** Let's assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the program and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.

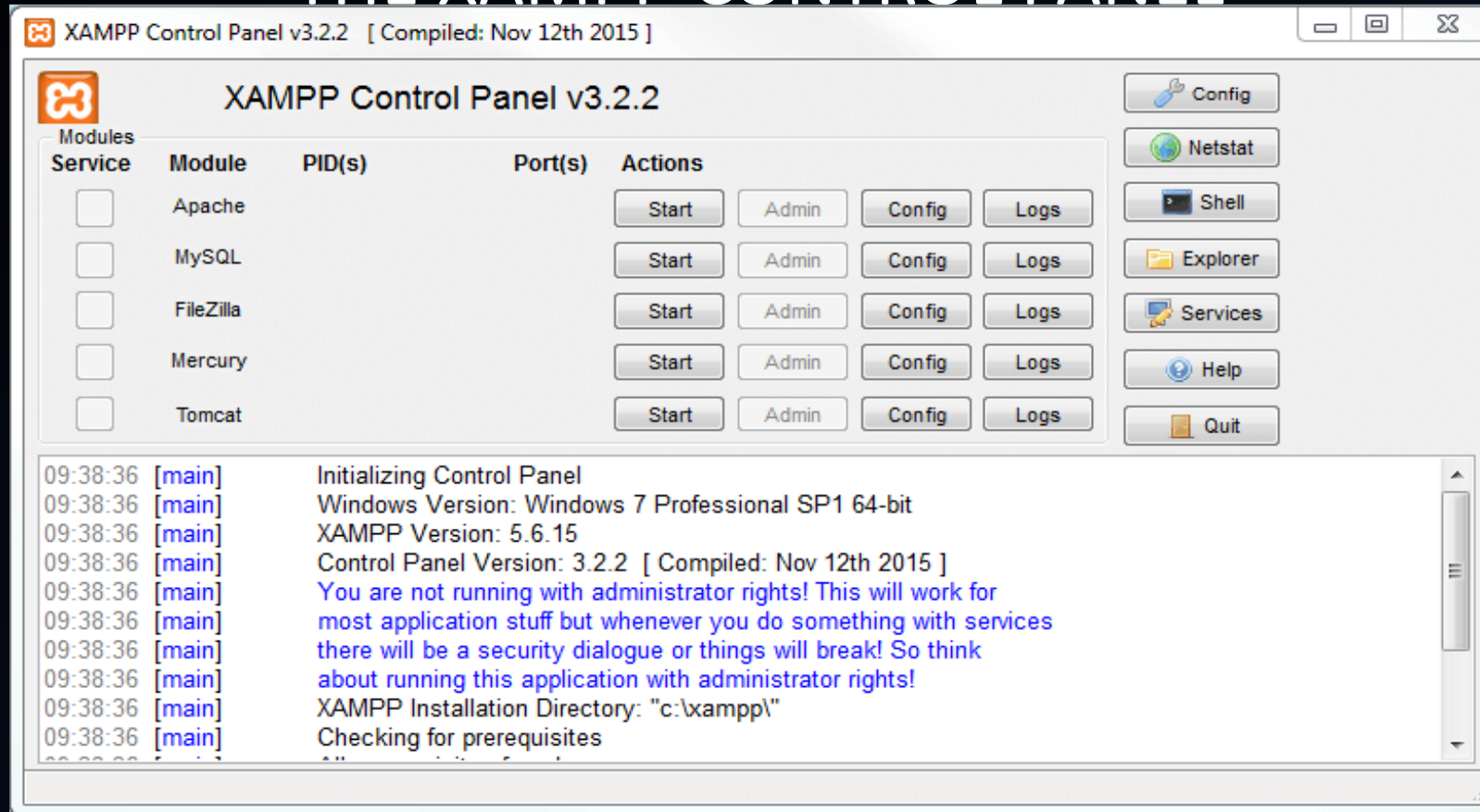
Setting Up a Development Server

- SETTING UP XAMPP
- **Change the conflicting port:** Let's assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the program and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.
- **End the conflicting program:** The simplest way to avoid port conflicts in the short term is to end the conflicting program (Skype in this case) If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved

Setting Up a Development Server

- SETTING UP XAMPP
- A common source of error connected with Apache is blocked ports . If you're using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it's likely that the Tomcat port is being blocked, meaning the web server can't be started. There are three ways to solve this issue:
- **Change the conflicting port:** Let's assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the program and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.

THE XAMPP CONTROL PANEL



- Modules that can't be started will be shown in red The user will also receive an error report to help solve the problem
- **Testing the web server** Open your Web browser, Type `http ://` (or `http 127 0 0 1` in the Address box, click Enter



DEVELOPMENT SERVER

PHP

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP Code</h1>

<?php

echo "Hello World!";

define("LEARNING", "Welcome to!");

function myFungsi() {
    echo LEARNING;
}

myFungsi();

$waktu = date("H");

if ($t < "25") {
    echo "Prpare for paydat!";
} else {
    echo "Have a good night!";
}

?>

</body>
</html>
```


FIRST PHP PAGE

Creating BASIC PHP Scripts

- The Web page generated from the PHP code, and HTML or XHTML elements found within the PHP file, is returned to the client.
- A PHP file that does not contain any PHP code should have an .html extension.
- .php is the default extension that most Web servers use to process PHP scripts

FIRST PHP PAGE

1. How to escape from HTML and enter PHP mode

PHP parses a file by looking for one of the special tags that tells it to start interpreting the text as PHP code. The parser then executes all of the code it finds until it runs into a PHP closing tag.

FIRST PHP PAGE

2. Simple HTML Page with PHP

The following is a basic example to output text using PHP.

```
<html><head>  
<title>My First PHP Page</title>  
</  
<? php  
echo "Hello World!";  
>  
</body></html>
```

Copy the code onto your web server and save it as “test.php”.

You should see “Hello World!” displayed.

Notice that the semicolon is used at the end of each line of PHP code to signify a line break. Like HTML, PHP ignores whitespace between lines of code. (An HTML equivalent is
)

FIRST PHP PAGE

- The PHP code is usually in files with extension ".php"

<?php denotes start of PHP code

?> denotes end of PHP code

- The PHP code can be mixed with HTML code

```
<html>
<head><title>Hello world page</title>
<body>
<?php echo "Hello PHP!"; ?>
</body>
</html>
```

PHP statements must be ended with a semicolon.

FIRST PHP PAGE

The diagram illustrates the execution of a PHP page in a web browser. It consists of three main components:

- Web Browser:** The top-left window shows the browser's address bar with `http://localhost/hello.php`. The 'View' menu is open, showing options like 'Toolbars', 'Quick tabs', 'Explorer bars', 'Go to', 'Stop', 'Refresh', 'Zoom (150%)', 'Text size', 'Encoding', 'Style', 'Caret browsing', 'Source', 'Security report', 'International website address', 'Webpage privacy policy...', and 'Full screen'. A blue arrow points from the 'Source' option to the 'Original Source' window.
- Original Source:** The top-right window, titled 'http://localhost/hello.php - Original Source', displays the raw HTML and PHP code:

```
1 <html>
2 <head><title>Hello world page</title></head>
3 <body>
4 Hello HTML!<br>
5 Hello PHP!</body>
6 </html>
7
```
- Rendered Output:** The bottom-left window shows the rendered HTML output:

```
Hello HTML!
Hello PHP!
```

A blue box with white text explains the process: **PHP code is never sent to a client's Web browser; only the HTML output of the processing is sent to the browser.**

Below the rendered output, the final HTML code is shown, highlighting the PHP code that was processed on the server:

```
<html>
<head><title>Hello world
page</title></head>
<body>
Hello HTML!<br>
<?php echo "Hello PHP!"; ?>
</body>
</html>
```

FIRST PHP PAGE

- HTML code is processed by browsers as web pages are loading. (client-side)
- PHP code is preprocessed by PHP Web servers that parse requested web pages as they are being passed out to the browser. (Server-side)
- You can embed sections of PHP inside HTML:

```
<BODY>
<p>
<?php $test = "Hello World!";
echo $test; ?>
</p>
</BODY>
```

- Or, you can call HTML from PHP :

```
<?php
echo "<html><head><title>Hello</title>
...
?>
```

FIRST PHP PAGE

General PHP Format

There are a few general rules required by PHP:

- PHP code is enclosed by the `<?php` and `?>` tags
`<?php`
...
`?>`
- Every PHP statement must end in a semicolon ;
`echo 'Thanks for submitting the form.';`
- PHP variables must begin with a dollar sign \$
`$email = $_POST['email'];`
 - A variable name must be at least one character in length.
 - The first character after the dollar sign \$ can be a letter or an underscore _, and characters after that can be a letter, an underscore, or a number.
 - Spaces and special characters other than \$ at the beginning and _ are not allowed in any part of a variable name.

General PHP Format

Comments

- PHP allows you to create single and multi-line comments,
- When we comment our code, our comments should reflect why our code exists instead of how our code works. The how is most often explained by the actual code.

```
<?php
```

```
//Single line comment
```

```
# Also a single line comment
```

```
//comment after code
```

```
/* Multi line block, you can add as many lines as you want, but don't write a book */
```

Bad Commenting Example

```
/* We are formatting phone numbers by stripping all characters then putting parens around the first 3 numbers adding a space then grouping the next 3 numbers adding a dash then the last 4 numbers */
```

```
<?php
```

```
/* The business has a requirement that all phone numbers be formatted with the (123) 123-1234 format */
```

```
function formatPhoneNumber($phone_string)
```

```
{ if (preg_match('/^\+\d(\d{3})(\d{3})(\d{4})$/ ', $data, $matches))
```

```
{ $result = "($matches[1]) $matches[2]-$matches[3]"; return $result; }
```


THANK YOU !

NO MATTER WHICH FIELD OF WORK YOU WANT TO GO
IN, IT IS OF GREAT IMPORTANCE TO LEARN AT LEAST
ONE PROGRAMMING LANGUAGE.

