የሥራና ክህሎት ሚኒስቴር
Ministry of Labor and Skills

Web Development and Database Administration Level-IV

Based on November 2023, Curriculum Version II

Module Title: Producing server-side script for dynamic web pages

Module code: EIS WDDBA4 M04 0322

Nominal duration: 100 Hours

Prepared by: Ministry of Labor and Skill

November, 2023

Addis Ababa, Ethiopia

Acknowledgment

Ministry of Labor and Skills wish to extend thanks and appreciation to the many representatives of TVET instructors and respective industry experts who donated their time and expertise to the development of this Teaching, Training and Learning Materials (TTLM).

Table of Contents

Acronym

CSS ---------- Cascading Style Sheets

DTD ---------- Document Type Definition

HTML ---------- Hypertext Markup Language

JS ---------- JavaScript

MySQL ---------- My Structured Query Language

PHP ---------- Hypertext Preprocessor

WAF ---------- Web Application Firewall

XHTML ---------- EXtensible HyperText Markup Language

Introduction to the Module

Server-side scripting for dynamic web pages involves the use of programming languages and frameworks to generate content on the server that is then delivered to users' browsers. Unlike client-side scripting, which runs in the user's browser, server-side scripting allows for the execution of code on the server, enabling dynamic and personalized content generation.

Module covers the units:

- Web Document Requiring server-side dynamic interaction
- Server-side script
- Produce Web Documents
- Test scripts and debug
- Set up Security

Learning Objective of the Module

- Understand the concept of server-side dynamic interaction
- Identify the different types of scrip languages and their application
- Design Server-Side Scripts
- Test Scripts and Debug Effectively
- Implement Security Measures

Module Instruction

For effective use these modules trainees are expected to follow the following module instruction:

1. Read the information written in each unit
2. Accomplish the Self-checks at the end of each unit
3. Perform Operation Sheets which were provided at the end of units
4. Do the “LAP test” giver at the end of each unit and
5. Read the identified reference book for Examples and exercise

Unit One: Web Document Requiring server-side dynamic interaction

This unit is developed to provide you the necessary information regarding the following content coverage and topics:

- Dynamic Functionality of a Website
- Web Scripting Language
- Web Document Requirements

This unit will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- Determine Dynamic Functionality
- Explore and Evaluate Web Scripting Languages
- Understand Web Document Requirements

1.1. Dynamic Functionality of a website

The dynamic functionality of a website falls into specific categories. Website managers should define which one theirs fits and manage it accordingly. Too often, websites attempt to serve multiple categories, which confuse users. By defining a website within one of these categories, website managers will help search engines respond with better results while offering opportunities for allied websites to back link to their website because they know exactly what to expect.

Big corporations are as guilty as smaller companies in failing to serve specific and dynamic functionality to users of their websites, often presenting a mishmash of hard-to-find portals, outdated news feeds, antiquated customer support sections and pedantic marketing information.

To define the dynamic functionality of a website, a website manager should ask the simple question whether it changes or not. **Does the website change or customize itself frequently and automatically?** We're not talking about rotating banners here, mind you! We're talking about real functionality. For instance, a dynamic website could automatically and continually display user posts, relevant news, and product updates. But other dynamic functionality is innovative and unique so that the service is the purpose of the website rather than a benefit of a larger agenda they're presenting.

Categories for Dynamic Functionality Websites

- **Activity** – A place for users to find others with common interests who have planned activities or are interested in creating new events to serve those interests.
- **Affiliate** – A third-party website whose purpose is to sell or present someone's product or service, usually for the purpose of generating advertising revenue.
- **Archive** – Searchable data, usually pictures or documents, presenting information that is legally required to be stored or is otherwise valuable to the users.
- **Blog** – A website journal used to post online diaries from individuals.
- **Comedy** – Where users can trade or find jokes on specific or a variety of topics.
- **Community** – This website allows like-minded individuals to participate in newsgroups or research news and user posts for the common purpose of building interest among its visitors.
- **Company** – Providing information about a corporation or business that assists customers and investors in their research.
- **Dating** – A place for users to find relationships.

- **Gambling** – A place to speculate on games or make spreads on arbitrary activities where odds influence the outcome of agreements made between two parties.
- **Investing** – Website where research is presented with the opportunity to invest in speculative commodities or instruments that will increase or decrease in value over time, depending on established markets and their influences.
- **News** – Exactly what it implies.
- **Personal** – Often a family or individual website where blogs, pictures or other posts appear for friends and family.
- **Political** – Propaganda site produced to influence perspectives on an individual or cause.
- **Products** – Website for selling items, typically in shopping carts.
- **Religious** – A place where users can find spiritual and practical inspiration, support and assistance regarding life's larger issues, often leading to community involvement.
- **Search** – A research destination where information is aggregated and presented based on user queries.
- **Social** – Networking websites used by individuals to communicate with others who designate shared interests

If the website is not dynamic stop what you're doing and add dynamic functionality immediately! Whether it's daily news feeds presenting unique content, quizzes, ongoing surveys and pole results, or even a web camera displaying video of the warehouse where employees are shipping products to customers, give users a reason to come back for more of something that's interesting. And remember that it does not have to be useful to be dynamic and interesting.

While the dynamic functionality of a website may cross into many different categories, most website managers will find by narrowing their mission to one, then he or she can research others who fit into that same category, borrowing dynamic functionality and attracting users similarly.

1.2. Web Scripting Language

When building an online application or simply adding some additional dynamics to a website, there is a special resource you need: scripting languages. Learning about them and knowing which ones to use is essential to enjoy all that the modern web has to offer.

Scripting languages are types of programming languages where the instructions are written for a run-time environment, to bring new functions to applications, and integrate or communicate complex systems and other programming languages. You have probably already heard of PHP, Python, JavaScript, and jQuery.

These are just a few examples of scripting languages that power the web and plenty of applications you and millions of other Internet users execute every day.

While all scripting languages are programming languages, not all programming languages are scripting languages. This means that certain capabilities are exclusive to either of them, which requires the right amount of knowledge to use the appropriate ones.

A scripting language is a programming language that executes tasks within a special run-time environment by an interpreter instead of a compiler. They are usually short, fast, and interpreted from source code or bytecode. Such environments include software applications, web pages, and even embedded systems in operating system shells and games.

Most modern structures support these languages, which gives them wide support while also being developed through an open-source process.

Using a scripting language is preferable depending on your goals and environment. As they are a series of commands executed with no need for a compiler, they are cross-platform and do not require special software to be installed in order to run except for a web browser, of course.

1.2.1. Types of Scripting Languages

There are two main types of scripting languages: server-side and client-side. They differ on where the code is run from, which affects not only the actual languages chosen but also the performance and their capabilities.

I. Server-side scripting language

The term server-side scripting language refers to those that run off a web server. Since it performs from the back-end side, the script is not visible to the visitor. Because of that, it is a more secure approach.

They are often used to create dynamic websites and platforms, handle user queries, and generate and provide data and others. A famous example of server-side scripting is the use of PHP in WordPress.

Examples: PHP, Python, Node.js, Perl, and Ruby.

II. Client-side scripting language

Unlike the above, client-side scripting languages run off the user's browser. It is usually performed at the front end, which makes it visible to visitors and makes it less vulnerable to exploits and leaks. As such, it is often used to build user interfaces and lighter functionality such as that.

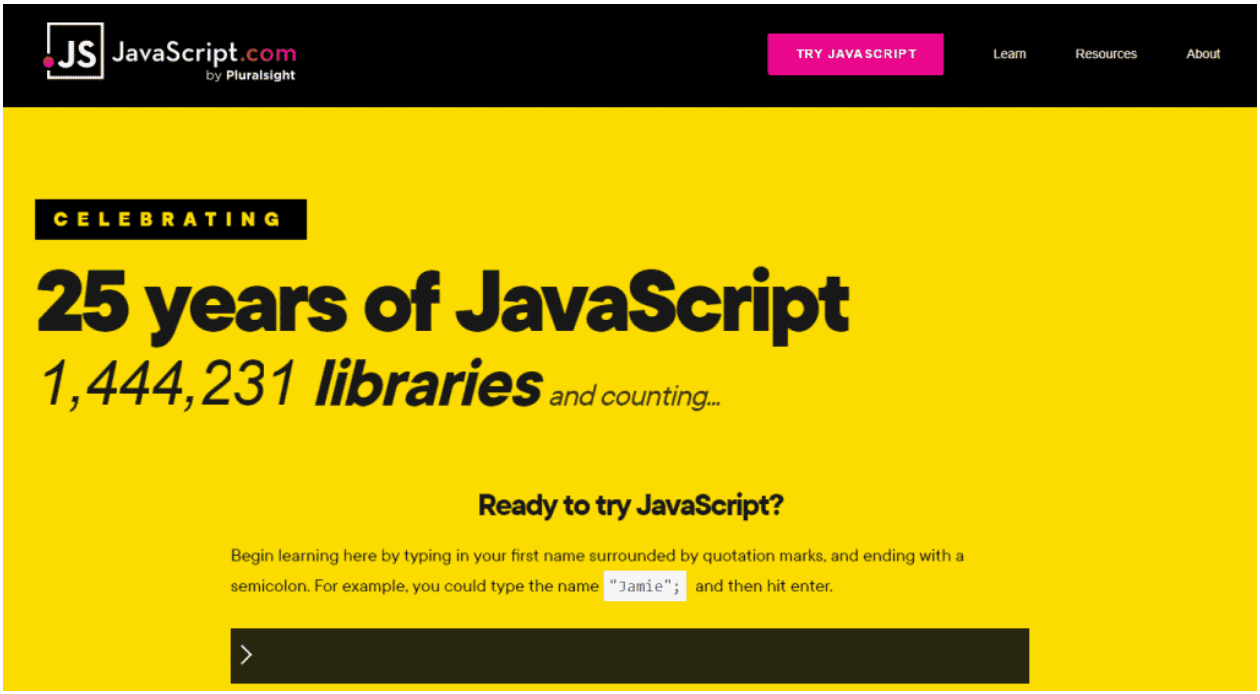
Since it runs locally, they usually provide better performance and, therefore, do not strain your server.

Examples: HTML, CSS, jQuery, and JavaScript.

1.2.2. Scripting Languages Examples

Now that you know the types, let's take a closer look at some of the different scripting languages out there to choose from. Each has its own unique advantages and potential uses.

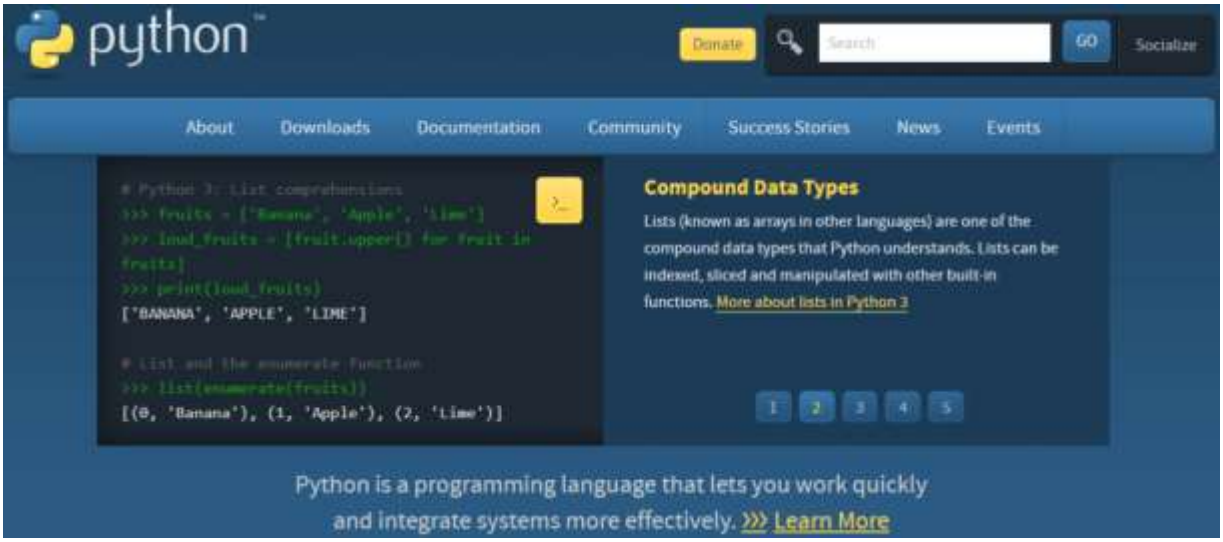
I. JavaScript



Also sometimes abbreviated as simply "JS", JavaScript is probably the best-known scripting language, as it's a pillar of the web as we know it (right along with HTML and CSS). In fact, about 98 percent of websites currently on the web use JavaScript.

JavaScript is considered a subdialect of the ECMAScript language, so it utilizes the same syntax. It also supports key features like first-class functions, dynamic typing, prototype-based object orientation, and more.

II. Python



After JavaScript, Python is easily the most popular, best-known scripting language in use today. Programmers love it for its sheer ease of use and concise syntax systems, as they can create code significantly more quickly and with less actual typing involved.

Python is also free and open-source, making it a highly accessible scripting language. Features supported by Python include functional programming paradigms, object-oriented programming, and more.

III. PHP

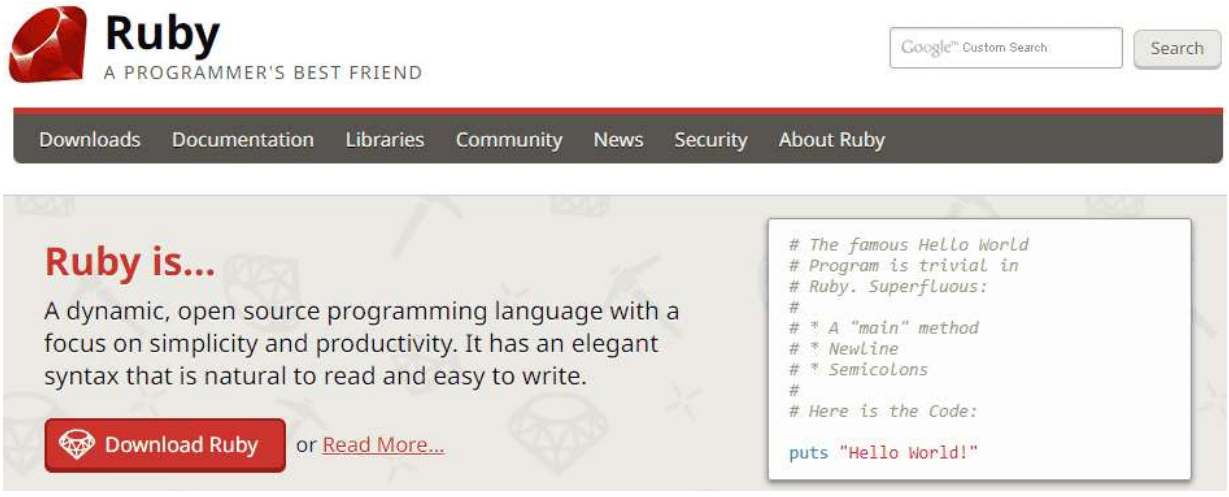
PHP is an open-source scripting language commonly used by backend web developers.

The name began as an acronym for "personal home page" a callback to PHP's origins as a way to make static HTML pages more functional and dynamic.

However, modern PHP is very much its own standalone scripting language.

PHP features object-oriented programming options and can be easily embedded into HTML documents of all types. Input is also fairly loose and easy, meaning programmers don't need to declare variable data types.

IV. Ruby

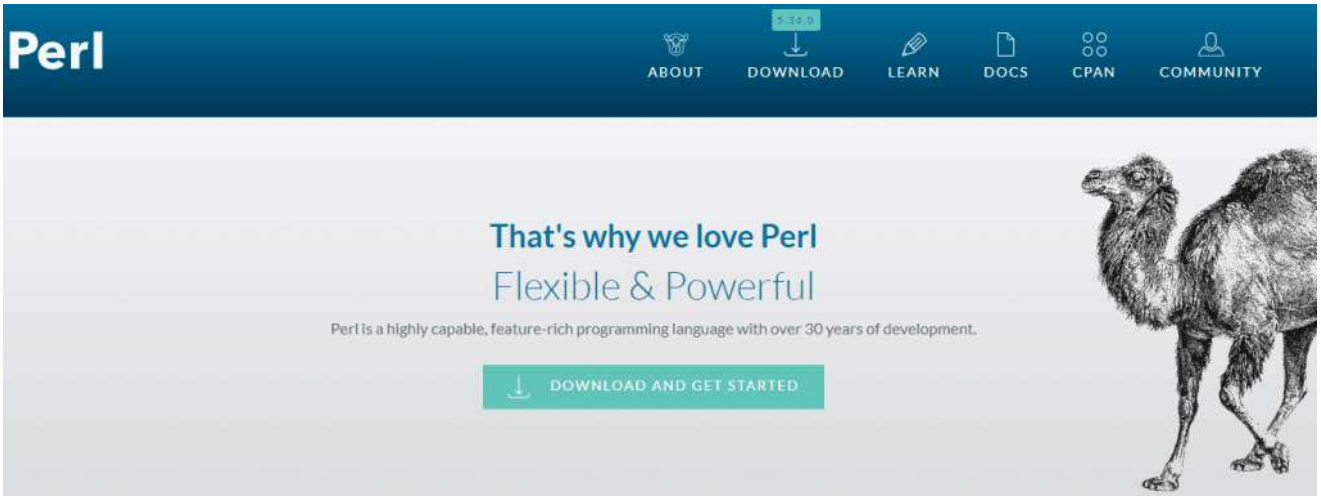


Ruby's claim to fame is its flexibility, making it a favorite among web developers of all types.

Among other things, it takes so much of the guesswork out of creating truly innovative software. It's also incredibly easy to learn, thanks to its clean syntax, making Ruby an especially popular pick for beginning coders.

Ruby is a strictly object-oriented scripting language, so everything becomes an object when working with it. This is even the case for factors like integers or Booleans which are usually more primitive.

V. Perl



Perl is a general scripting language that's been around a very long time since December of 1987, to be exact. It started out as a UNIX language primarily used to process reports. (Its name even originates from the phrase "practical extraction and reporting language".)

Perl really began to gain traction throughout the 1990s when coders used it heavily for CGI (common gateway interface), a specification most often seen today on legacy websites.

However, it remains fairly popular because of its innovation and suitability for text manipulation tasks.

1.2.3. Advantages of Using Scripting Languages

Scripting languages can be highly beneficial to your projects, as they add several new features to applications and web pages.

Learning about how to use them might give you an edge on your capabilities without the heavy requirements of time and resources of traditional programming languages.

Here are the most important benefits of using scripting languages in your processes.

- **Easy to learn and use**

They are often pointed out as great starting points for those interested in learning programming. That is because they are considerably easier to learn and use than traditional languages. This means that you can quickly implement the scripts you need without them requiring a lot of time and resources to be invested in them.

- **Open-source and free**

Anyone can use scripting languages without any restrictions. All they have to do is learn them and implement their capabilities into their structure. They are all open-source, which means anybody can contribute to their development on a global scale. This also contributes to the security of the systems that use them.

- **Portable and cross-platform**

Since scripting languages run off a remote server or from the visitor's web browser, they have another highly valuable benefit: they are portable and cross-platform. This means no additional software needs to be installed to run them and any browser can execute their functions under any operating system and platform such as WordPress.

- **Lighter memory requirement**

Unlike what happens with traditional programming, scripting languages do not require compilers to store an executable file to be run.

Instead, they use interpreters, which contributes to a much smaller memory requirement on the system running them either the server or the user's local machine.

1.2.4. Disadvantages of Using Scripting Languages

Despite being the optimal choice for certain cases, scripting languages are not the right fit for others. As with any tool, you should know its disadvantages to make a proper decision when planning your project. Below are the most notable disadvantages of using scripting languages in your business.

- **Lack of optimizations**

Unlike what happens to most programming languages, scripting languages have each statement line individually analyzed by the compiler without any optimizations. This might cause an impact on performance.

- **Slower execution**

Additionally, most scripting language interpreters are slower than compilers. That is because traditional programming languages are usually converted to machine code before being compiled, thus offering faster execution for the users.

- **Organization requirements**

When taking advantage of the benefits of these languages, you might reach a point where your application or website is filled with them. This requires extra attention to organize everything and keep track of which scripts you are using. Otherwise, it will be impractical to maintain them.

- **Time commitment on updates**

In order for your scripts to continue to function properly, they need to be updated regularly. This is necessary whenever browsers receive significant new versions or when your own website changes in some radical way.

1.3. Web Document Requirement

A website requirements document is an essential pioneer to initiating any website design project for business owners. It serves as a comprehensive outline detailing the precise needs of the new website. Providing a clear roadmap ensures that the website designer or developer constructs a site that aligns with the owner's objectives.

Having worked with several clients over the years, we believe the website requirements document holds paramount significance in the web design process. It facilitates effective communication of the owner's vision to the designer, establishing mutual understanding regarding the website's overall objectives and essential features.

To ensure a comprehensive website requirement document, you must ensure you have these five points checked off:

I. Outline a clear purpose and goals for the website.

II. Define your target audience.
III. Determine technical requirements and specifications.
IV. Define content requirements.
V. Include wireframes and site maps.

I. Outline a clear purpose and goals for the website

For a business, your website goals can be any of the following:

- **Educate potential customers about your products and services**: You want visitors to understand what you offer and how to meet their needs.
- **Encourage contact from leads**: The site should make it easy for interested visitors to contact you by phone, email, or online contact form.
- **Build your brand and credibility:** The site's overall look, feel, and messaging should convey your brand personality and inspire trust in your company.
- **Generate more sales and revenue:** Ultimately, you want the site to drive more people to purchase from you, either directly on the site or by contacting you.

With these goals defined, your website designer can determine how best to achieve them when building the site.

II. Define your target audience

As the website owner, you need to know and understand your target audience.

Who are the people that will be visiting and using your website?

Knowing your target audience is one of the crucial requirements for website design because it'll determine many of the design decisions for your website.

III. Determine technical requirements and specifications.

When determining the technical website development requirements, several vital things must be considered. As the client, you know your business and audience best, so providing as much detail as possible about what you envision for your site will help ensure the result meets your needs. Here are six technical requirements and specs to consider:

- Functionality
- Site Accessibility

- Integrations
- Site search
- Analytics
- Site styles

IV. Define content requirements.

When defining the content requirements for website design, several vital things must be considered.

Here's a 4-point checklist outlining what to look into:

I. Determine what pages are needed to achieve your business goals.
II. Decide what written content is needed for each page.
III. Determine what visual content will be required.
IV. Define interactive elements.

Self-Check 1

Part-I: Choose the correct answer

1. One of the following is not the types of Server-side scripting language.
    A. PHP
    B. JavaScript
    C. Ruby
    D. Python
2. A document that is essential to initiating any web design project for business owners.
    A. Website Design
    B. Website Technical documentation
    C. Website Requirement
    D. Website Script
3. Which one of the following is not the advantages of using scripting languages?
    A. Slower execution
    B. Open-source
    C. Portable
    D. Easy to use

Part-II: Answer the following questions accordingly

1. List some of the Dynamic website's functionalities.
2. Explain the purpose of scripting language.
3. Write down the difference between Server side and client-side scripting.

Unit Two: Server-Side Scripts

This unit to provide you the necessary information regarding the following content coverage and topics:

- Development Environment
- Basic syntax of Server-side scripts
- Forms and user input
- Working with Database
- Security Features in web development

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Set up a development environment
- Master the basic syntax of server-side scripts
- Effectively handle forms and user input
- Implement database connectivity
- Implement security features in web development

2.1. Development Environment

PHP is a commanding language widely utilized in web development. To fully harness its potential, it's essential to set up a robust, reliable PHP development environment on Windows. This process provides the necessary PHP development tools for Windows it can help in efficient coding, testing & debugging applications, and ensuring streamlined efficiency during the transition from development to production.

To set up PHP on Windows, XAMPP proves to be an excellent solution. This free, open-source software offers an easy-to-navigate platform for PHP development. Bundling essential core components like PHP, Apache, and MySQL XAMPP's accessibility makes it ideal for beginners, yet its comprehensive toolset caters to the needs of experienced PHP developers. This guide outlines the necessary steps to set up a PHP development environment in Windows using XAMPP, aiming to simplify your coding journey.

Configuring XAMPP for PHP Development

Whether you're an established developer or a beginner venturing into PHP, setting up an efficient development environment is critical One of the most recommended and popular solutions is XAMPP - a free and open-source software that stands as a beacon for setting up a local web server for developers worldwide

In the process of configuration, the next steps are opening XAMPP Control Panel and setting-up local development server.

Apache: Core of Server-Side Processing

Apache HTTP Server, commonly referred to as Apache, is a highly customizable and robust open-source web server software It interprets and executes the PHP code embedded in your HTML and sends the resulting data to the client in other words, Apache in the bridge that connects your PHP scripts to the user's browser.

Apache is on indispensable tool in a PHP development environment, providing sever-side processing capabilities that empower developers to build dynamic web application.

MySQL: Core of Data Management

MySQL, on the other hand, is a relational database management system. It allows you to store, retrieve, and manipulate databases with which your PHP scripts interact. Whether building a simple website or a complex web application, data management is a core function, making MySQL an essential part of the PHP development process.

2.2. Basic syntax of Server-side scripts

When we talk about popular programming languages, usually at the top of the list is PHP, a favorite among developers and software engineers who use frameworks based on this language to build modern and multifunctional web pages and applications.

PHP is one of the most popular scripting languages and works on the server side. Its acronym means “Hypertext Preprocessor” in Spanish and it is embedded in HTML. It allows: creating personalized web content, sending and receiving cookies, evaluating form data sent from a browser, etc.

In addition to its features, it has integration with several popular databases like Postgre SQL, Oracle, Sybase, SQL, and MySQL. It also handles forms, saves data to files, and collects data from files.

2.1.1. Basic PHP Syntax

A PHP script can be placed anywhere in a document. This script starts with <?php and ends with?>. We present an example:

```
1
2  <?php
3  // PHP code goes here
4  ?>
5
```

The default extension of PHP files is: .php. A PHP file usually contains HTML tags and some PHP script code.

The following is an example of a simple PHP file, with a script that uses a built-in echo function to output the text "Hello world!" On a website:

```
1  <!DOCTYPEhtml>
2  <html>
3  <body>
4  <h1>My first PHP page</h1>
5  <?php
6  echo "Hello World!";
7  ?>
8  </body>
9  </html>
10
```

2.1.2. PHP Variables

In PHP, a variable is declared using a **$ sign** followed by the variable name. Here, some important points to know about variables:

- As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyzes the values and makes conversions to its correct datatype.
- After declaring a variable, it can be reused throughout the code.
- Assignment Operator (=) is used to assign the value to a variable.

Syntax of declaring a variable in PHP is given below:

```
$variablename=value;
```

Rules for declaring PHP variable:

- A variable must start with a dollar ($) sign, followed by the variable name.
- It can only contain alpha-numeric character and underscore (A-z, 0-9, _).
- A variable name must start with a letter or underscore (_) character.
- A PHP variable name cannot contain spaces.
- One thing to be kept in mind that the variable name cannot start with a number or special symbols.
- PHP variables are case-sensitive, so $name and $NAME both are treated as different variable.

A. PHP Variable: Declaring string, integer, and float

Let's see the example to store string, integer, and float values in PHP variables.

File: variable1.php

```php
<?php
$str="hello string";
$x=200;
$y=44.6;
echo "string is: $str <br/>";
echo "integer is: $x <br/>";
echo "float is: $y <br/>";
?>
```

Output:

```
string is: hello string
integer is: 200
float is: 44.6
```

B. PHP Variable: Sum of two variables

File: variable2.php

```php
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;
?>

```

Output:

```
11
```

C. PHP Variable: case sensitive

In PHP, variable names are case sensitive. So variable name "color" is different from Color, COLOR, COLor etc.

File: variable3.php

```php
<?php
$color="red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>
```

Output:

```
My car is red
Notice: Undefined variable: COLOR in C:\wamp\www\variable.php on line 4
My house is
Notice: Undefined variable: coLOR in C:\wamp\www\variable.php on line 5
My boat is
```

D. PHP Variable: Rules

PHP variables must start with letter or underscore only.

PHP variable can't be start with numbers and special symbols.

File: variablevalid.php

```php
<?php
$a="hello";//letter (valid)
$_b="hello";//underscore (valid)

echo "$a <br/> $_b";
?>
```

Output:

```
hello
```

hello

File: variableinvalid.php

```
1  <?php
2  $4c="hello";//number (invalid)
3  $*d="hello";//special symbol (invalid)
4
5  echo "$4c <br/> $*d";
6  ?>
7
```

Output:

Parse error: syntax error, unexpected '4' (T_LNUMBER), expecting variable (T_VARIABLE) or '$' in C:\wamp\www\variableinvalid.php on line 2

2.1.3. PHP Data Types

PHP data types are used to hold different types of data or values. PHP supports 8 primitive data types that can be categorized further in 3 types:

i. Scalar Types (predefined)
ii. Compound Types (user-defined)
iii. Special Types

PHP Data Types: Scalar Types

It holds only single value. There are 4 scalar data types in PHP.

i. Boolean
ii. integer
iii. float
iv. string

PHP Data Types: Compound Types

It can hold multiple values. There are 2 compound data types in PHP.

i. array
ii. object

PHP Data Types: Special Types

There are 2 special data types in PHP.

i. resource
ii. NULL

2.1.4. PHP Operators

Operators are used to performing operations on some values. In other words, we can describe operators as something that takes some values, performs some operation on them, and gives a result. From example, "1 + 2 = 3" in this expression '+' is an operator. It takes two values 1 and 2, performs an addition operation on them to give 3.

Just like any other programming language, PHP also supports various types of operations like arithmetic operations (addition, subtraction, etc), logical operations (AND, OR etc), Increment/Decrement Operations, etc. Thus, PHP provides us with many operators to perform such operations on various operands or variables, or values. These operators are nothing but symbols needed to perform operations of various types. Given below are the various groups of operators:

- Arithmetic Operators
- Logical or Relational Operators
- Comparison Operators
- Conditional or Ternary Operators
- Assignment Operators
- Array Operators
- Increment/Decrement Operators
- String Operators

i. Arithmetic Operators

The arithmetic operators are used to perform simple mathematical operations like addition, subtraction, multiplication, etc. Below is the list of arithmetic operators along with their syntax and operations in PHP.

Operator	Name	Syntax	Operation
+	Addition	$x + $y	Sum the operands
–	Subtraction	$x – $y	Differences the operands

*	Multiplication	$x * $y	Product of the operands
/	Division	$x / $y	The quotient of the operands
**	Exponentiation	$x ** $y	$x raised to the power $y
%	Modulus	$x % $y	The remainder of the operands

ii. Logical or Relational Operators

These are basically used to operate with conditional statements and expressions. Conditional statements are based on conditions. Also, a condition can either be met or cannot be met so the result of a conditional statement can either be true or false. Here are the logical operators along with their syntax and operations in PHP.

Operator	Name	Syntax	Operation
and	Logical AND	$x and $y	True if both the operands are true else false
or	Logical OR	$x or $y	True if either of the operands is true else false
xor	Logical XOR	$x xor $y	True if either of the operands is true and false if both are true
&&	Logical AND	$x && $y	True if both the operands are true else false
||	Logical OR	$x || $y	True if either of the operands is true else false
!	Logical NOT	!$x	True if $x is false

iii. Comparison Operators

These operators are used to compare two elements and outputs the result in Boolean form. Here are the comparison operators along with their syntax and operations in PHP.

Operator	Name	Syntax	Operation
==	Equal To	$x == $y	Returns True if both the operands are equal
!=	Not Equal To	$x != $y	Returns True if both the operands are not

			equal
<>	Not Equal To	$x <> $y	Returns True if both the operands are unequal
===	Identical	$x === $y	Returns True if both the operands are equal and are of the same type
!==	Not Identical	$x == $y	Returns True if both the operands are unequal and are of different types
<	Less Than	$x < $y	Returns True if $x is less than $y
>	Greater Than	$x > $y	Returns True if $x is greater than $y
<=	Less Than or Equal To	$x <= $y	Returns True if $x is less than or equal to $y
>=	Greater Than or Equal To	$x >= $y	Returns True if $x is greater than or equal to $y

iv. Conditional or Ternary Operators

These operators are used to compare two values and take either of the results simultaneously, depending on whether the outcome is TRUE or FALSE. These are also used as a shorthand notation for if…else statement that we will read in the article on decision making.

Syntax:

$var = (condition)? value1 : value2;

Here, the condition will either evaluate as true or false. If the condition evaluates to True, then value1 will be assigned to the variable $var otherwise value2 will be assigned to it.

Operator	Name	Operation
?:	Ternary	If the condition is true? then $x : or else $y. This means that if the condition is true then the left result of the colon is accepted otherwise the result is on right.

v. Assignment Operators

These operators are used to assign values to different variables, with or without mid-operations. Here are the assignment operators along with their syntax and operations, that PHP provides for the operations.

Operator	Name	Syntax	Operation
=	Assign	$x = $y	Operand on the left obtains the value of the operand on the right
+=	Add then Assign	$x += $y	Simple Addition same as $x = $x + $y
-=	Subtract then Assign	$x -= $y	Simple subtraction same as $x = $x – $y
*=	Multiply then Assign	$x *= $y	Simple product same as $x = $x * $y
/=	Divide then Assign (quotient)	$x /= $y	Simple division same as $x = $x / $y
%=	Divide then Assign (remainder)	$x %= $y	Simple division same as $x = $x % $y

vi. Array Operators

These operators are used in the case of arrays. Here are the array operators along with their syntax and operations, that PHP provides for the array operation.

Operator	Name	Syntax	Operation
+	Union	$x + $y	Union of both i.e., $x and $y
==	Equality	$x == $y	Returns true if both has same key-value pair
!=	Inequality	$x != $y	Returns True if both are unequal
===	Identity	$x === $y	Returns True if both have the same key-value pair in the same order and of the same type
!==	Non-Identity	$x !== $y	Returns True if both are not identical to each other
<>	Inequality	$x <> $y	Returns True if both are unequal

vii. Increment/Decrement Operators

These are called the unary operators as they work on single operands. These are used to increment or decrement values.

Operator	Name	Syntax	Operation

++	Pre-Increment	++$x	First increments $x by one, then return $x
—	Pre-Decrement	–$x	First decrements $x by one, then return $x
++	Post-Increment	$x++	First returns $x, then increment it by one
—	Post-Decrement	$x–	First returns $x, then decrement it by one

viii. String Operators

This operator is used for the concatenation of 2 or more strings using the concatenation operator (‘.’). We can also use the concatenating assignment operator (‘.=’) to append the argument on the right side to the argument on the left side.

Operator	Name	Syntax	Operation
.	Concatenation	$x.$y	Concatenated $x and $y
.=	Concatenation and assignment	$x.=$y	First concatenates then assigns, same as $x = $x.$y

2.1.5. PHP Comments

PHP comments can be used to describe any line of code so that other developer can understand the code easily. It can also be used to hide any code.

PHP supports single line and multi line comments. These comments are similar to C/C++ and Perl style (Unix shell style) comments.

A. PHP Single Line Comments

There are two ways to use single line comments in PHP.

- // (C++ style single line comment)
- # (Unix Shell style single line comment)

```
1  <?php
2  // this is C++ style single line comment
3  # this is Unix Shell style single line comment
4  echo "Welcome to PHP single line comments";
5  ?>
6
```

Output:

```
Welcome to PHP single line comments
```

B. PHP Multi Line Comments

In PHP, we can comment multiple lines also. To do so, we need to enclose all lines within /* */. Let's see a simple example of PHP multiple line comment.

2.1.6. Control Structures in PHP

Control structures are an essential part of any programming language and PHP is no exception. They provide the ability to control the flow of execution in a program based on certain conditions. In other words, they allow you to make decisions in your code and execute different blocks of code based on those decisions. This helps to simplify complex tasks, making it easier to write and maintain the code.

These statements are mainly categorized into following:

- Conditional Statements
- Loop Statements
- Jump Statements

I. Conditional Statements

Conditional Statements performs different computations or actions depending on conditions. In PHP, the following are conditional statements

- **if** statement
- **if - else** statement
- **if - elseif - else** statement
- **switch** statement
- **if statement**

The if statement is used to test a specific condition. If the condition is true, a block of code (if-block) will be executed.

Syntax:

```
if (condition)
{
        statements
}
```

- **if - else statement**

The if-else statement provides an else block combined with the if statement which is executed in the false case of the condition.

Syntax:

```
if (condition)
{
        statements
}
else
{
        statements
}
```

- **if - elseif - else statement**

The elseif statement enables us to check multiple conditions and execute the specific block of statements depending upon the true condition among them.

Syntax:

```
if (condition1)
{
	statements
}
else if (condition2)
{
	statements
}
else if (condition3)
{
	statements
}
.
.
else
{
	statements
}
```

- **switch statement**

The switch statement enables us to execute a block of code from multiple conditions depending upon the expression.

Syntax:

```
switch (expression)
{
case 1: statements
		break;
case 2: statements
		break;
case 3: statements
		break;
.
.
default: statements
}
```

II. Loop Statements

Sometimes we may need to alter the flow of the program. If the execution of a specific code may need to be repeated several numbers of times then we can go for loop statements.

In PHP, the following are loop statements

- **while** loop
- **do - while** loop
- **for** loop

A. while loop statement

With the while loop we can execute a set of statements as long as a condition is true. The while loop is mostly used in the case where the number of iterations is not known in advance.

Syntax:

```
while (condition)
{
       statements
}
```

B. do - while loop statement

The do-while loop will always execute a set of statements at least once and then execute a set of statements as long as a condition is true.

Syntax:

```
do
{
  statements
} while (condition);
```

C. for loop statement

With the for loop, we can execute a set of statements specified number of times. The for loop is mostly used in the case where the number of iterations is known in advance.

Syntax:

```
for (initialization; condition; increment/decrement)
```

```
{
  statements
}
```

III. Jump Statements

Jump statements in PHP are used to alter the flow of a loop like you want to skip a part of a loop or terminate a loop.

In PHP, the following are jump statements

- **break** statement
- **continue** statement

A. break statement

The break is a keyword in PHP which is used to bring the program control out of the loop. i.e. when a break statement is encountered inside a loop, the loop is terminated and program control resumes at the next statement following the loop.

Syntax:

```
break;
```

B. continue statement

The continue statement in PHP is used to bring the program control to the beginning of the loop. i.e. when a continue statement is encountered inside the loop, remaining statements are skipped and loop proceeds with the next iteration.

The continue statement skips the remaining lines of code inside the loop and start with the next iteration. It is mainly used for a particular condition inside the loop so that we can skip some specific code for a particular condition.

Syntax:

```
continue;
```

2.1.7. Differences between PHP and JavaScript

JavaScript is, like PHP, one of the most popular programming languages. It can be defined as a high-level, dynamic, interpreted language used with HTML web applications. It is also used for non-web documents such as PDFs and desktop widgets.

Among the main differences between both programming languages are:

- PHP is a server-side scripting language while JavaScript is a client-side scripting language.
- PHP does not run inside the browser, while JavaScript runs inside the browser.
- PHP supports databases while JavaScript does not support databases.
- PHP accepts variables in upper and lower case, while JavaScript does not.
- When we compare PHP and JavaScript, PHP does not support swapping objects and arrays, while JavaScript supports swapping objects and arrays.

It is up to the developer to choose which programming language best suits the demands of the project in question: JavaScript or PHP.

2.3. Forms and user input

When a user submits a form, PHP takes charge of processing the data. This involves accessing the form elements using super global variables like $_POST and $_GET. Understanding the nuances of form submission and super global variables is fundamental to effective user input handling.

To create a form, you use the <form> element as follows:

```
<form action="form.php" method="post">
</form>
```

The <form> element has two important attributes:

- **action:** specifies the URL that processes the form submission. In this example, the form.php will process the form.
- **method:** specifies the HTTP method for submitting the form. The most commonly used form methods are POST and GET. In this example, the form method is post.

The form method is case-insensitive. It means that you can use either post or POST. If you don't specify the method attribute, the form element will use the get method by default.

Typically, a form has one or more input elements including text, password, **checkbox**, **radio button**, **select**, **file upload**, etc. The input elements are often called form fields. An input element has the following important attributes name, type, and value. The name attribute will be used for accessing the value in PHP.

PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Let's see a simple example to receive data from get request in PHP.

File: form1.html

```
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
<input type="submit" value="visit"/>
</form>
```

File: welcome.php

```
<?php
$name=$_GET["name"];//receiving name field value in $name variable
echo "Welcome, $name";
?>
```

PHP Post Form

Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc. The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

Let's see a simple example to receive data from post request in PHP.

File: form1.html

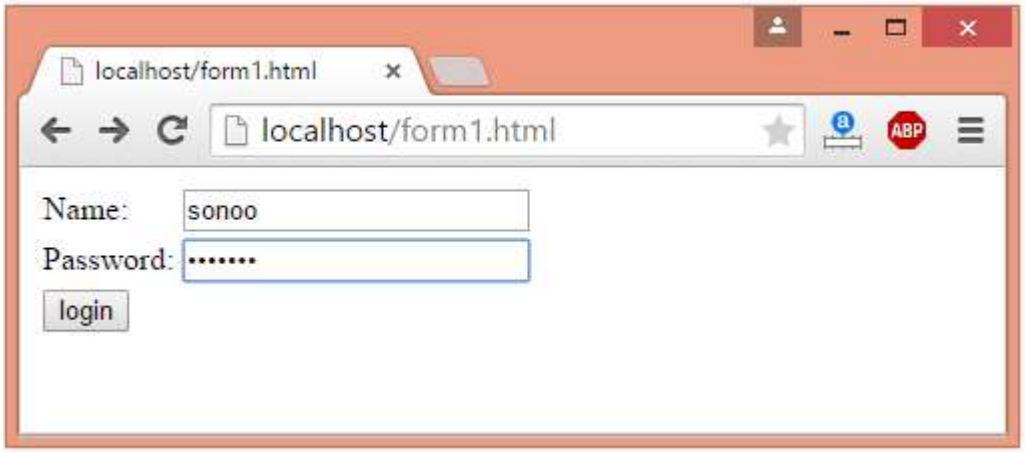
```
<form action="login.php" method="post">
<table>
```

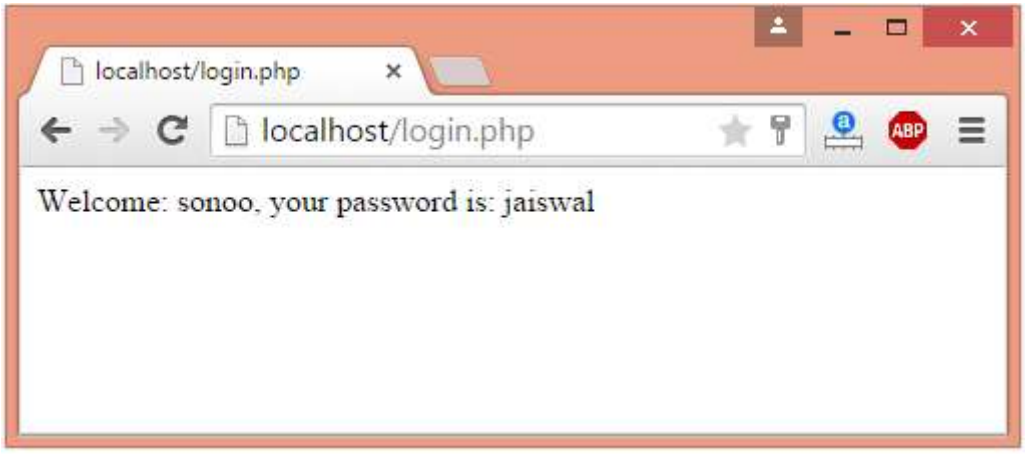
```
<tr><td>Name:</td><td> <input type="text" name="name"/></td></tr>
<tr><td>Password:</td><td> <input type="password" name="password"/></td></tr>
<tr><td colspan="2"><input type="submit" value="login"/>  </td></tr>
</table>
</form>
```

File: login.php

```
<?php
$name=$_POST["name"];//receiving name field value in $name variable
$password=$_POST["password"];//receiving password field value in $password variable
echo "Welcome: $name, your password is: $password";
?>
```

Output:





2.4. Working with Database

2.4.1. Brief Overview of MySQL

MySQL is a widely popular open-source relational database management system (RDBMS) that excels in storing, managing, and retrieving data. It is renowned for its efficiency, versatility, and widespread adoption.

MySQL, when combined with PHP cloud, employs a structured approach to data organization. It arranges data into tables consisting of rows and columns with defined data types. Relationships between tables are established through primary and foreign keys, ensuring data integrity and enabling complex data querying.

MySQL's adaptability is demonstrated by its cross-platform compatibility, operating seamlessly across various operating systems. Its extensive community and ecosystem contribute to its ongoing refinement and integration with other tools and technologies.

2.4.2. System Requirements of MySQL

The Enterprise Service Manager has certain system requirements that are recommended for optimal performance. These requirements are outlined below.

Platform	Service Manager Minimum Disk Space	Agent Minimum Disk Space
Linux x86 64-bit	1.3 GB	800 MB
macOS	1.2 GB	700 MB
Windows x86 64-bit	800 MB	500 MB

There are 3 types of methods in PHP to connect MySQL database through the backend:

I. MySQL
II. MySQLi
III. PDO

mysql() is now obsolete because of security issues like SQL injection etc., but the other two are being actively used.

I. MySQLi

MySQLi is an API that serves as a connector function, linking the backend of PHP applications to MySQL databases. It is an improvement over its predecessor, offering enhanced safety, speed, and a more extensive set of functions and extensions.MySQLi was introduced with PHP 5.0.0,

and its drivers were installed in version 5.3.0. The API was designed to support MySQL versions 4.1.13 and newer.

II. PDO

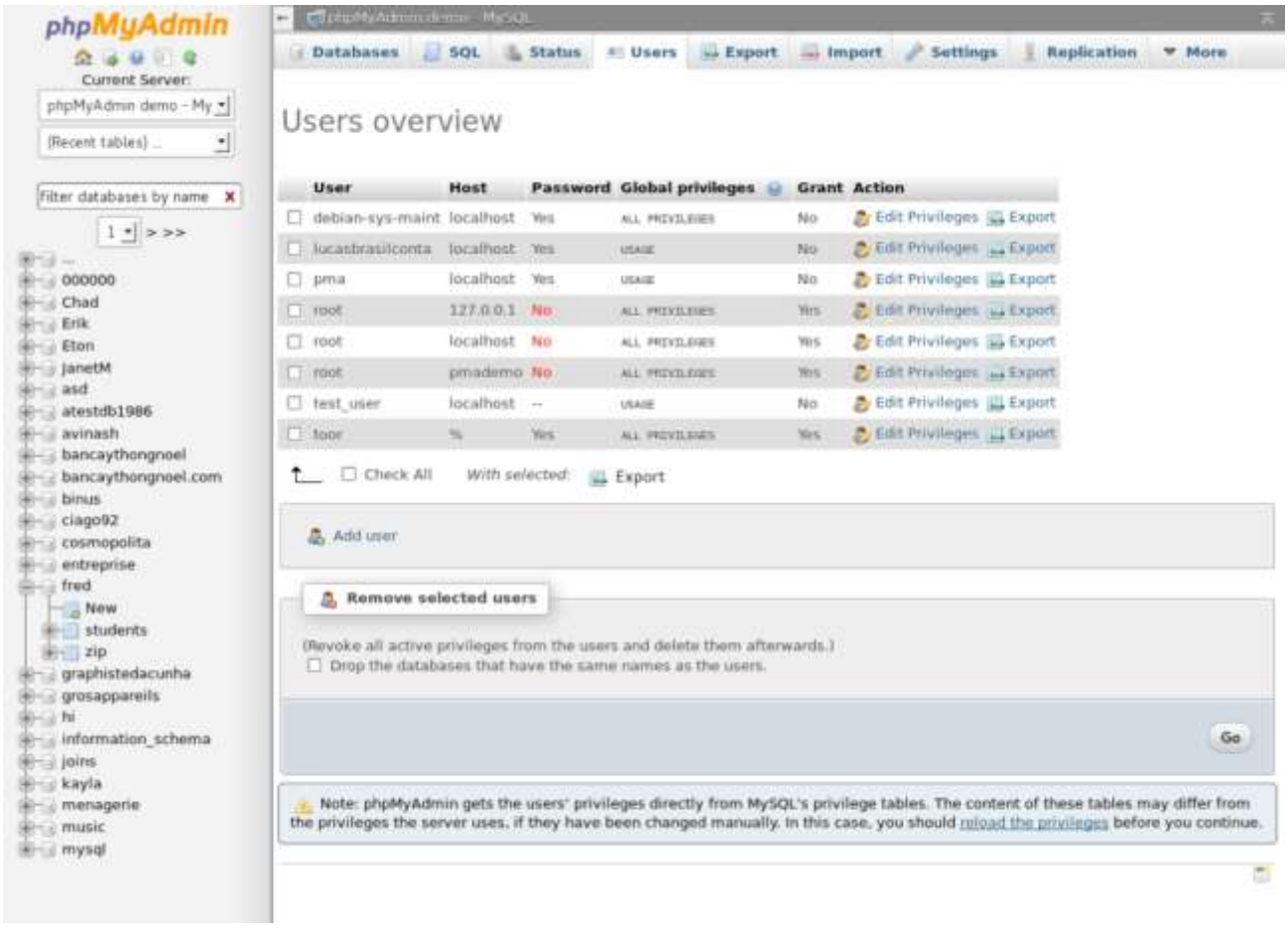
The PHP Data Objects (PDO) extension is a Database Abstraction Layer that serves as an interface for the backend to interact with MySQL databases. It allows for changes to be made to the database without altering the PHP code and provides the flexibility to work with multiple databases. One of the significant advantages of using PDO is that it keeps your code simple and portable.

2.4.3. PHPMyAdmin

phpMyAdmin is an open-source tool for MySQL that's available for free. It is fundamentally written in PHP as a versatile web application and has become one of the foremost popular MySQL administration tools over the internet. It supports a wide extent of operations on MySQL and MariaDB.

phpMyAdmin is a popular application for MySQL database administration. Developers can create, update, drop, alter, delete, import, and export MySQL database tables using this phpMyAdmin. Moreover, phpMyAdmin supports many different operations that can be performed via the user interface.

The operations like managing databases, tables, relations, columns, indexes, permissions, users, etc., on MySQL and MariaDB, phpMyAdmin is also used in performing administrative-level tasks such as database creation and query execution.



phpMyAdmin supports RTL and LTR languages, so many users can easily use this software. It can help you run MySQL queries, repair, optimize, check tables, and execute other database management commands.

2.5. Security Features in web development

Website security is a primary consideration in web development but it is often not taken seriously by a lot of website owners. If you have built a secure website, then you must have sought out the services of a security expert who spots areas of weaknesses in your system and also carry out routine maintenance checks for new flaws and vulnerabilities. These are the minimum requirements for any safe website and I have grouped them into 10 must-have features for your website.

a. Registry lock

A skilled hacker can take control of an unsecured domain, alter the configurations and redirect the site elsewhere. Apart from the embarrassment that such a breach

might bring, there could also be potential legal consequences. In 2003, the New York Times had to deal with the consequences of such a breach.

With a registry lock feature, it becomes very difficult for a domain to be hijacked or its DNS configurations altered by a third party without rigorous procedures. A registry lock demands multiple party authorization from registrar and registry before alterations to the domain can be made. This is a fundamental requirement for bigger organizations especially. Installing this feature requires a bit of manual effort and companies typically charge for this service.

b. Hotlink protection

Some sites can take images and hyperlinks from your website and display them on their pages, essentially stealing your data. This process is called hotlinking. Hotlinking also affects your bandwidth and disk space of your site so preventing this is crucial. You can protect your website from this theft of your data by making use of special preventive tools available.

c. Spam stop feature

If you are a frequent user of the internet, it is impossible not to encounter ads and commercials and these are increasingly using up a lot of online space. A little pop-up here or there might be benign and users often want to support their favorite brands. Sometimes, these pop-ups are not so benign and can cause your site to become infected with spam and this could be bad for user experience. As a business owner, you want your audience to get a pleasant experience and one way of doing this is by investing in spam stop feature.

d. DDOs attack protection

Distributed denial of service attacks is a common nuisance most sites have to deal with. But you want these attacks to be as infrequent as possible as they can cause your site to become spammed if they originate from multiple points. These attacks can also cause your site to run slower than normal. Your web hosting company can also provide protection against these kinds of attacks.

e. Secure sockets layer (SSL protection)

This feature provides privacy and security of communication done over the web. This is especially important if you want you want to sell products or services on your site. SSL protects the integrity of your website in two basic ways:

i. It creates a secure network between users and tracks every message that is exchanged over the internet. Some web hosting companies use an encryption service called a secure shell host (SSH). SSH reduces the need for additional security installations.

SSL employs “optional session caching” in optimizing the connections between networks.

ii. It employs a mechanism called symmetric cryptography to maintain complete privacy during web communication between parties. It is also particularly useful when the communication involves transactions of a financial nature.

f. Two-Step verification

This is also known as two-factor authentication (2FA). It is a security feature that requires owners of online accounts to produce two authentication factors rather than one. Accounts that require just one factor for their authentication are known as single-factor authentication (SFA) accounts. An example of an SFA account would be one that requires a password for access while an example of a 2FA account would be your bank account as you would require your debit card and a pin before withdrawing money from an ATM.

The dual-factor authentication is a very powerful security measure and as a rule, you should never patronize a domain company that does not provide a two-step verification process for all its user accounts.

g. Secure administrative passwords

It goes without saying that a strong access password is crucial. A website with a weak password is an easy target for hackers, more so when it is hosted on an open content management platform like WordPress. Make sure you select a password that is lengthy and does not look like a word. It should also not be information about you or your business that can be easily researched or even guessed. Get a reliable password generator. The most secure websites will only accept strong passwords and will also require that the admins change their passwords quarterly. A tedious process no doubt but way less tedious than the problems you’ll have to deal with if your weak password is breached.

h. Bot Blocking

Search engines employ bots to go through websites to help them index and rank efficiently. Non-friendly spider bots can also get information the same way and this can be sold or used for malicious purposes. Bot can overrun your websites giving you skewed analytics and inflated traffic results. Distributed denial of service attack (DDOS) can also be caused by bots. They overload your website by attacking your network from multiple systems causing it to become overloaded, this slows down your website or can cause it to crash altogether.

A security-conscious web developer will take measures using available web tools to prevent malicious bot attacks.

i. Protection from cross-site scripting (XSS) attacks

JavaScript can be maliciously introduced into a website. This can cause unwanted effects like changing the content of your web pages or worse still lead to data theft from your website to the point of origin.

Your web security provider would want to focus on how user data can be obtained and manipulated by an external party causing it to be misinterpreted by the browser.

j. Data backup

Website data breaches and loss are always a possibility. As undesirable as they may be, it is wise to have a content backup plan in the event of a breach. Hosting companies offer data backup services that ensure you do not lose your data even if you are compromised and you will be able to get your website up and running within a few hours even if you come under damaging attack.

Self-Check 2

Part-I: Choose the correct answer

1. What is XAMPP used for in PHP development on Windows?
   A) Graphic design C) Development environment setup
   B) Database management D) Network security
2. Which of the following is a core component of XAMPP for PHP development?
   A) Python C) Java
   B) Apache D) C#
3. What is the purpose of MySQL in PHP development?
   A) Front-end scripting
   B) Server-side processing
   C) Database management
   D) Graphic design
4. How do you declare a variable in PHP for storing an integer value?
   A) var integer = 10;
   B) $number = "10";
   C) int $number = 10;
   D) $number = 10;
5. Which PHP feature is used for preventing hotlinking of website content?
   A) Spam stop
   B) Two-Step Verification
   C) Hotlink protection
   D) DDoS protection

Part-II: Answer the following questions accordingly

1. Describe the differences between PHP and JavaScript.
2. Discuss the significance of secure sockets layer (SSL) protection in web development.
3. What is the importance of data backup in website security.

Operation sheet 2.1: Setup PHP development on windows

Operation Title: Windows PHP development setup

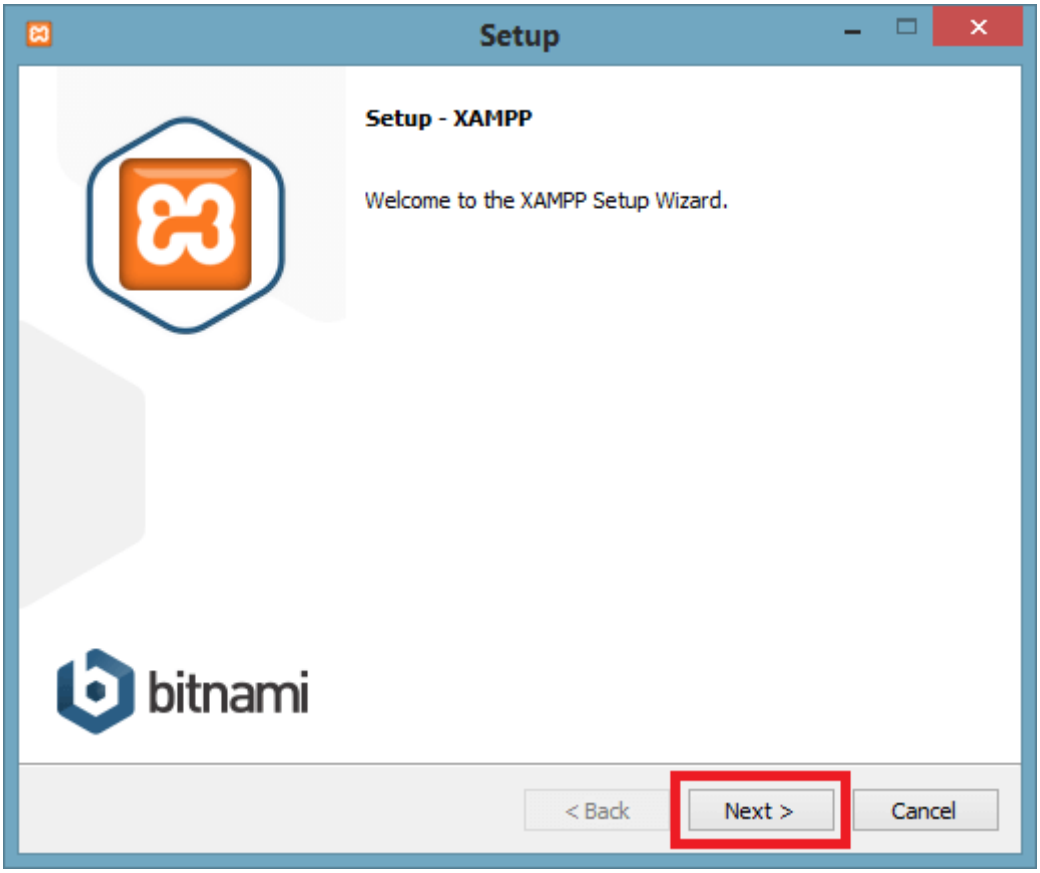
Purpose: Establish a Local PHP development environment on windows.

Equipment Tools and Materials:

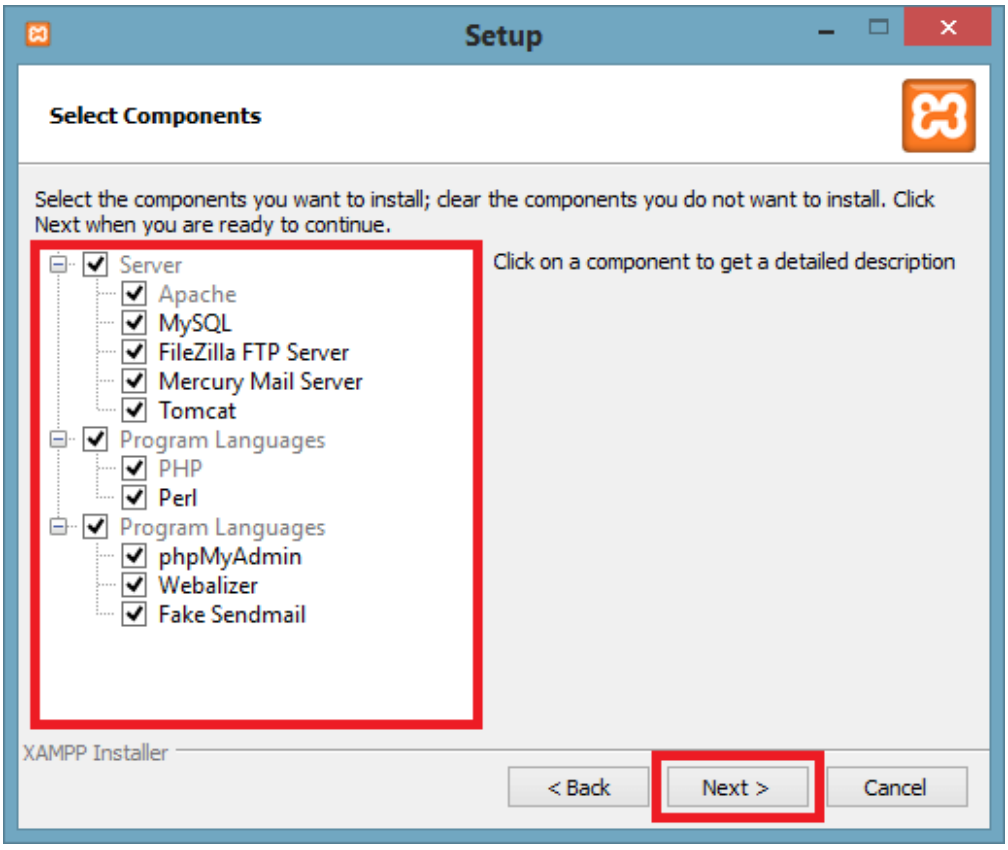
- Computer
- Internet

Steps in doing the task

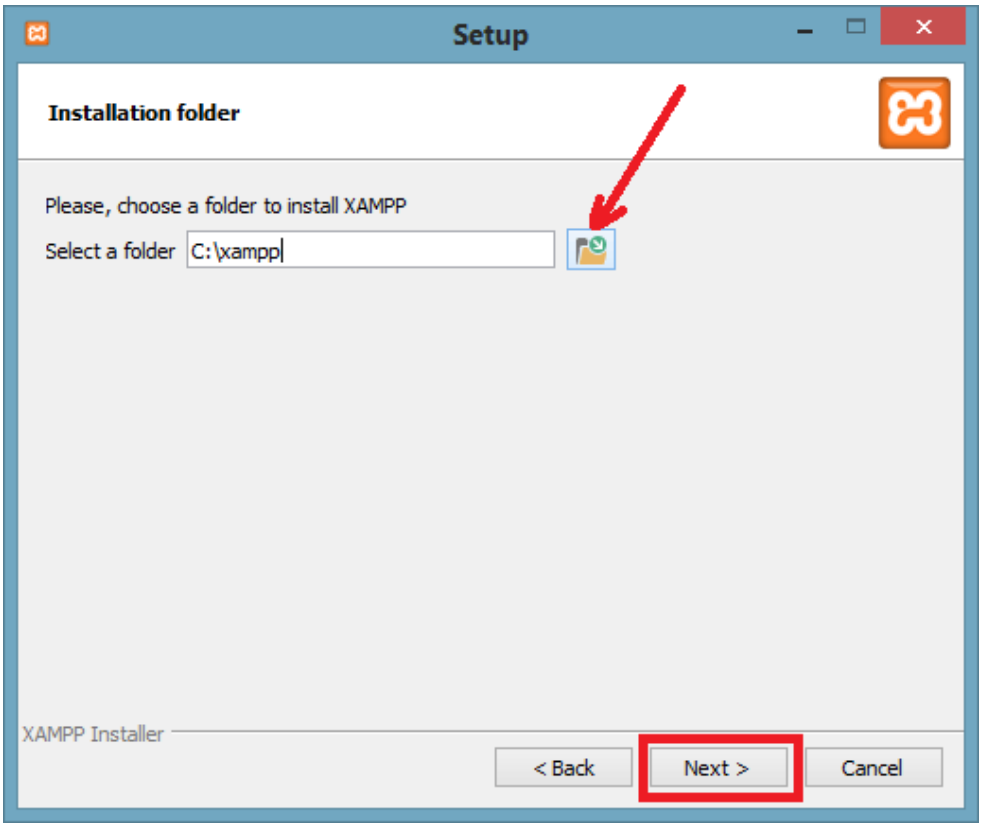
1. Downloading XAMPP – Begin by visiting the official XAMPP websites at https://www.apachefriends.org/index.html. Select the version that suits your requirements. Click on the download button
2. Once the download is complete you need to open the installer. Click Next



3. Selecting components – For PHP development select ‘phpMyAdmin’ and ‘MySQL’. Once you have selected the desired components click ‘Next’.

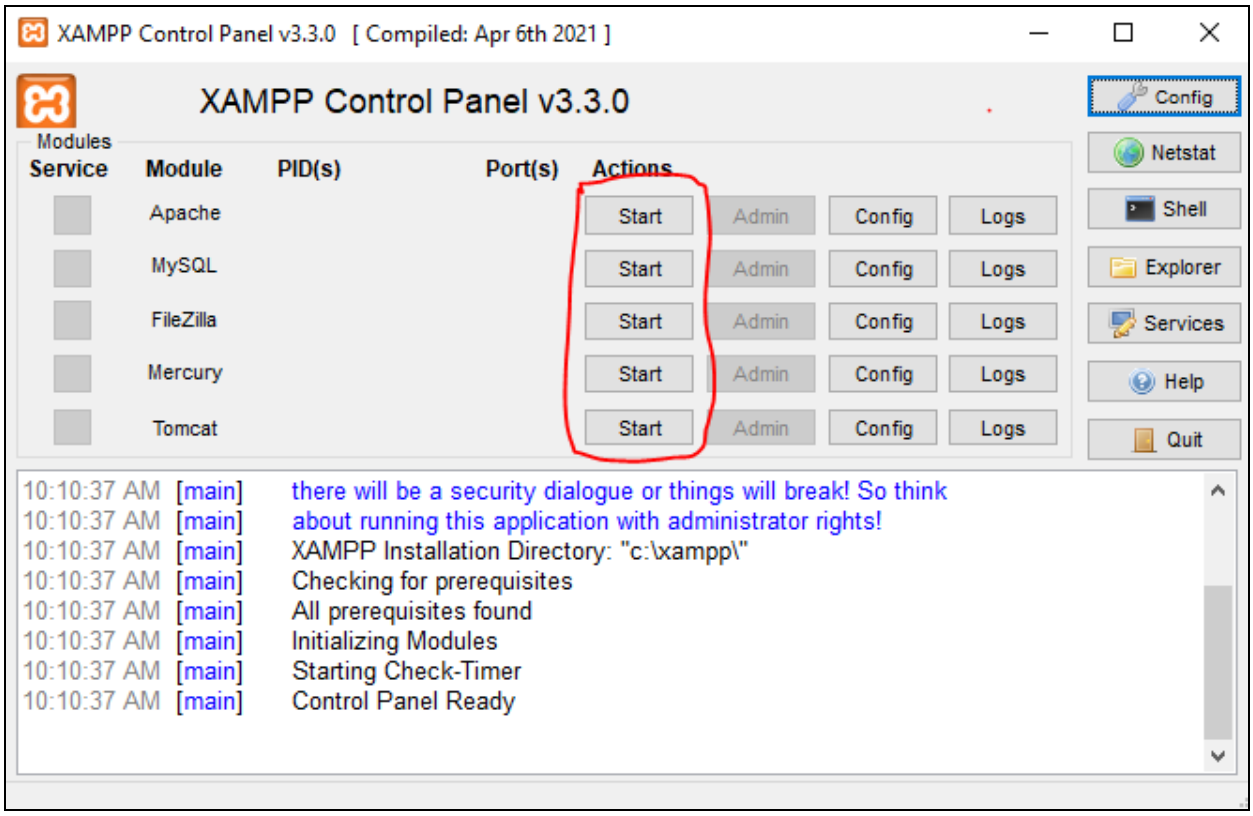


4. Select a directory where XAMPP will be installed. Click 'Next' to continue



5. The installation process will begin installing the components.

6. Once you have installed XAMPP on your system locate the XAMPP folder and open XAMPP Control panel.
7. Look for ‘Apache’ and ‘MySQL’ and click on the start button.



Once these services are running, they will be highlighted in green indicating that your PHP development environment is now set up and ready for action.

Quality Criteria: Well configured XAMPP setup

Operation sheet 2.2: Create and run a new PHP file

Operation Title: PHP file creation and execution

Purpose: Generate and execute a new PHP file for testing and development.

Equipment Tools and Materials:

- Computer
- Text editors (Notepad/Notepad ++/Sublime/VS code)
- XAMPP/WAMPP

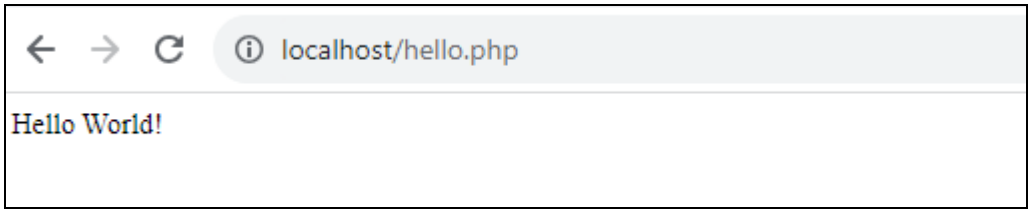
Steps in doing the task

3. Navigate to the directory where you have installed XAMPP (usually it's C:/xampp) and open the htdocs folder. Inside this folder, create a new PHP file. Let's name it hello.php for now.
4. Open hello.php file in your preferred text editor and write a simple PHP script.

```
1  <?php
2      echo "Hello World!";
3  ?>
4
```

5. Run the XAMPP server and start Apache and MySQL
6. Now open the web browser and type: http://localhost/hello.php on your browser window. Hello World! will be shown in your browse as an output.

Quality Criteria: The web page has to show information like the following:



Operation sheet 2.3: Create and Run a PHP file for Basic Syntax

Operation Title: PHP basic syntax test

Purpose: Generate and execute a new PHP file to test and understand basic PHP syntax.

Equipment Tools and Materials:

- Computer
- Text editors (Notepad/Notepad ++/Sublime)
- XAMPP/WAMPP Server

Steps in doing the task

1. Open the htdocs folder. Inside this folder, create a new PHP file and name it basics.php.
2. Open basics.php file in your preferred text editor and write a simple PHP script with basic syntax elements.

```php
<?php
    // Basic variable and output
    $greeting = "Hello, PHP!";
    echo $greeting;

    // Basic loop
    for ($i = 1; $i <= 5; $i++) {
        echo "<p>This is paragraph $i.</p>";
    }

    // Basic if statement
    $number = 10;
    if ($number > 0) {
        echo "The number is positive.";
    } elseif ($number < 0) {
        echo "The number is negative.";
    } else {
        echo "The number is zero.";
    }
?>
```

3. To run this PHP file, go back to your web browser and enter the following URL: http://localhost/basics.php. If everything is set up correctly, you should see a web page displaying the output of your PHP script.

Quality Criteria: The web page should display the output of your basic PHP script.

Operation Sheet 2.4: Variables and Data Types

Operation Title: PHP Variables and Data Types

Purpose: Demonstrate the use of variables and different data types in PHP.

Equipment, Tools, and Materials:

- Computer, Text editor, Web browser, XAMPP

Steps in Doing the Task:

1. Create and save the following file with Variables_DataTypes.php in the root directory.

```php
<html>
<head>
    <title>Variables and Data Types Demo</title>
</head>
<body>
    <h1>Variables and Data Types Demo</h1>
    <?php
    // Integer
    $integerVar = 42;
    echo "<p>Integer Variable: $integerVar</p>";

    // Float
    $floatVar = 3.14;
    echo "<p>Float Variable: $floatVar</p>";

    // String
    $stringVar = "Hello, PHP!";
    echo "<p>String Variable: $stringVar</p>";

    // Boolean
    $boolVar = true;
    echo "<p>Boolean Variable: " . ($boolVar ? "true" : "false") . "</p>";

    // Array
    $arrayVar = array(1, 2, 3);
    echo "<p>Array Variable: ";
    print_r($arrayVar);
    echo "</p>";
    ?>
</body>
</html>
```

2. Open your web browser and Search http://localhost/Variables_DataTypes.php.

Quality Criteria:

- The web page should display a heading "Variables and Data Types Demo" and information about variables of different data types.

Operation Sheet 2.5: PHP Operators

Operation Title: PHP Operators Demonstration

Purpose: Understand and demonstrate the use of different operators in PHP.

Equipment, Tools, and Materials:

- Computer
- Text editor (Notepad/Notepad++/Sublime/VS Code)
- Web browser
- Local server environment (XAMPP/WAMPP or equivalent)

Steps in Doing the Task:

1. Create the following Arithmetic operators code on Operators.php.

```php
<html>
<head>
    <title>PHP Operators Demo</title>
</head>
<body>
    <h1>PHP Operators Demo</h1>
    <?php
    // Arithmetic Operators
    $num1 = 10;
    $num2 = 5;
    $sum = $num1 + $num2;
    $difference = $num1 - $num2;
    $product = $num1 * $num2;
    $quotient = $num1 / $num2;
    $remainder = $num1 % $num2;

    echo "<p>Arithmetic Operators:</p>";
    echo "<p>Sum: $sum</p>";
    echo "<p>Difference: $difference</p>";
    echo "<p>Product: $product</p>";
    echo "<p>Quotient: $quotient</p>";
    echo "<p>Remainder: $remainder</p>";
```

2. Write PHP code for Assignment Operators code to file

```php
    $x = 5;
    $y = 3;
    $x += $y; // Equivalent to $x = $x + $y;
    echo "<p>Assignment Operator: x = $x</p>";
```

3. Write PHP code for Comparison Operators code to the file

```php
    $a = 7;
    $b = 7;
    $isEqual = ($a == $b);
    $isIdentical = ($a === $b);
    $isNotEqual = ($a != $b);
    $isNotIdentical = ($a !== $b);

    echo "<p>Comparison Operators:</p>";
    echo "<p>Equal: " . ($isEqual ? "true" : "false") . "</p>";
    echo "<p>Identical: " . ($isIdentical ? "true" : "false") . "</p>";
    echo "<p>Not Equal: " . ($isNotEqual ? "true" : "false") . "</p>";
    echo "<p>Not Identical: " . ($isNotIdentical ? "true" : "false") . "</p>";

```

4. Write PHP code for Logical Operators code to the Operators.php

```php
    $p = true;
    $q = false;
    $logicalAnd = ($p && $q);
    $logicalOr = ($p || $q);
    $logicalNotP = !$p;
    $logicalNotQ = !$q;

    echo "<p>Logical Operators:</p>";
    echo "<p>Logical AND: " . ($logicalAnd ? "true" : "false") . "</p>";
    echo "<p>Logical OR: " . ($logicalOr ? "true" : "false") . "</p>";
    echo "<p>Logical NOT P: " . ($logicalNotP ? "true" : "false") . "</p>";
    echo "<p>Logical NOT Q: " . ($logicalNotQ ? "true" : "false") . "</p>";
    ?>
</body>
</html>
```

5. Run all the code

Quality Criteria:

- The web page should display a heading "PHP Operators Demo" and information about various operators.
- Arithmetic operators should perform calculations and display results.
- Assignment operators should modify variable values and display the changes.
- Comparison operators should evaluate and display true/false results.
- Logical operators should perform logical operations and display true/false results.

Operation Sheet 2.6: Conditional statements in PHP

Operation Title: PHP Conditional statement

Purpose: Understand and demonstrate the use of conditional statements in PHP to compare two variables.

Equipment, Tools, and Materials:

- Computer
- Text editor (Notepad/Notepad++/Sublime/VS Code)
- Web browser
- Local server environment (XAMPP/WAMPP or equivalent)

Steps in Doing the Task:

1. Create and save the following file with Conditional.php in the root directory

```php
<html>
<head>
<title>Conditional Demo</title>
</head>
<body>
<?php
$x=15;
$y=5;
if ($x > $y)
{
    echo "$x is greater than $y";
}
else if ($x < $y)
{
    echo "$x is lessthan $y";
}
else
{
    echo "Both are Equal";
}
?>
</body>
</html>
```

2. Start your local server environment and open your web browser and type http://localhost/Conditional.php.

Quality Criteria: The web page should display the result of the conditional comparison between variables

Operation Sheet 2.7: For Loop in PHP

Operation Title: PHP For Loop

Purpose: Understand and demonstrate the use of a for loop in PHP to generate and display a sequence of numbers.

Equipment, Tools, and Materials:

- Computer
- Text editor (Notepad/Notepad++/Sublime/VS Code)
- Web browser
- Local server environment (XAMPP/WAMPP or equivalent)

Steps in Doing the Task:

1. Create and save the following file with For_loop.php in the root directory

```
<html>
<head>
<title>For Loop</title>
</head>
<body>
<h1>For Loop</h1>
<?php
for($i=1;$i<=5;$i++)
{
    echo "$i <br/>";
}
?>
</body>
</html>
```

2. Start your local server environment and open your web browser and type http://localhost/For_loop.php.

Quality Criteria:

- The web page should display a heading "For Demo" and a sequence of numbers from 1 to 5 generated using a for loop.
- The numbers should be displayed vertically, each on a new line.

Operation sheet 2.8: PHP Forms and user input

Operation Title: PHP form creation and user input.

Purpose: Develop a PHP-based form to collect user input and create a script to handle and display the submitted data.

Equipment Tools and Materials:

- Computer
- Text editors (Notepad/Notepad++/Sublime/VS Code)
- Web browser
- XAMPP/WAMPP (or any other web server setup)

Steps in doing the task

1. Create a new file with the name **user_form.php** in **htdocs** folder.
2. Open the file with preferred text editor and write the following HTML code to create a form with fields to collect user information. Include fields for name, email and age.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Information Form</title>
</head>
<body>
    <h2>User Information Form</h2>
    <form action="process_form.php" method="post">
        <label for="name">Name:</label>
        <input type="text" name="name" required>

        <label for="email">Email:</label>
        <input type="email" name="email" required>

        <label for="age">Age:</label>
        <input type="number" name="age" required>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

3. Create a new PHP file named "**process_form.php**" to handle the form submission.

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Collect user input from the form
    $name = htmlspecialchars($_POST["name"]);
    $email = htmlspecialchars($_POST["email"]);
    $age = htmlspecialchars($_POST["age"]);

    // Display the collected information
    echo "<h2>Information Summary:</h2>";
    echo "<p><strong>Name:</strong> $name</p>";
    echo "<p><strong>Email:</strong> $email</p>";
    echo "<p><strong>Age:</strong> $age</p>";
} else {
    // Redirect to the form if accessed directly without submission
    header("Location: user_form.php");
    exit();
}
?>
```

2. Open your web browser and go to http://localhost/user_form.php. Fill out the form and submit it.

Quality Criteria:

- The web page should display a form with fields for name, email, and age.
- After submission, the PHP script should process the form data and display the collected information on a new page.

Operation Sheet 2.9: Adding Password to XAMPP Account for PHPMyAdmin

Operation Title: Adding XAMPP Account Password for PHPMyAdmin

Purpose: Enhance security by setting a password for the XAMPP account used to access PHPMyAdmin.

Equipment, Tools, and Materials:

- Computer
- XAMPP or similar local server environment

Steps in Doing the Task:

1. Open PHPMyAdmin: Open your web browser and type localhost/PHPMyAdmin in the address bar or click on "Admin" in the XAMPP UI.
2. Navigate to the User Account section and locate the username shown in the image below. From there, you can set a password for your account.

3. To set a password for your XAMPP account, click on the Edit Privileges button and navigate to the Change Admin Password section. Enter your password and click Save to update your account. Be sure to remember this password, as you'll need it to connect to your database.

Quality Criteria:

- A password is successfully set for the XAMPP account used to access PHPMyAdmin and the password is strong and follows good security practices.

Operation sheet 2.10: Creating MySQL Database at Localhost with PHP Connection

Operation Title: MySQL Database Setup and PHP Connection

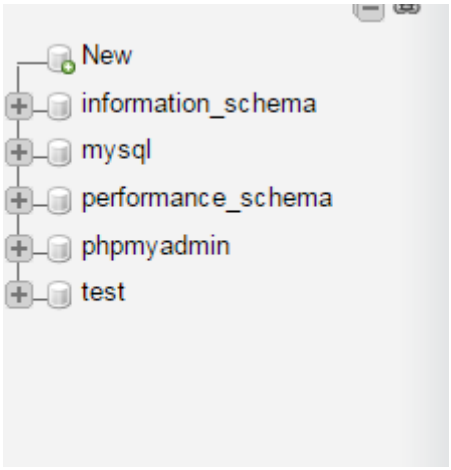
Purpose: Set up a MySQL database on localhost using PHPMyAdmin and create a PHP connection file for database access.

Equipment Tools and Materials:

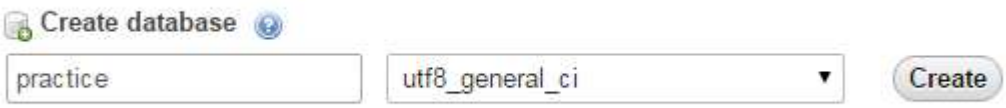
- Computer
- XAMPP or similar local server environment
- Text editor (Notepad/Notepad++/Sublime/VS Code)
- Web browser

Steps in doing the task

1. Create a New Database: open your web browser and type localhost/PHPMyAdmin in the address bar or click on "Admin" in the XAMPP UI. Click the "New" button to create a new database.



2. Enter a name for the database (e.g., "practice"). Choose utf8_general_ci as the collation. Click "Create" to create the new database.



3. Create a Folder in htdocs: Navigate to the XAMPP installation folder and open the htdocs subfolder (usually located at C:\xampp). Inside htdocs, create a new folder named "practice" to store web files.

4. Create Database Connection File in PHP (db_connection.php): create a new file named db_connection.php inside the "practice" folder. Copy and paste the following PHP code into db_connection.php:

```php
<?php
function OpenCon()
{
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "1234";
$db = "example";
$conn = new mysqli($dbhost, $dbuser, $dbpass,$dbname) or die("Connect failed: %s\n". $conn -> error);
return $conn;
}
function CloseCon($conn)
{
$conn -> close();
}
?>
```

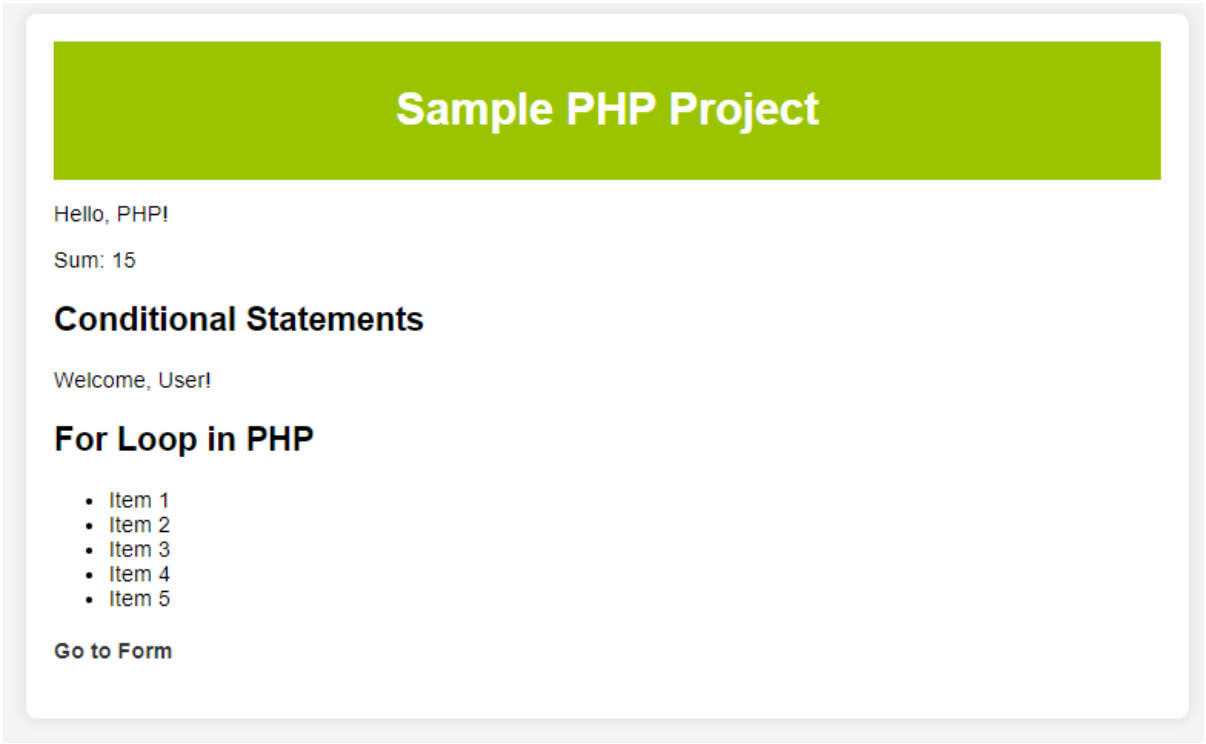
5. Update the $dbpass variable with the password you set for your XAMPP account. Save the file.
6. Check Database Connection (index.php): Create a new file named index.php inside the "practice" folder. Copy and paste the following PHP code into index.php:
7. Test the Connection: Start your local server environment (XAMPP). Open your web browser and go to localhost/practice/index.php. Confirm that you see the message "Connected Successfully."

Quality Criteria:

- A new database named "practice" is created in PHPMyAdmin.
- The "practice" folder is created inside the htdocs directory.
- The db_connection.php file is created with the correct connection details.
- The index.php file successfully connects to the database and displays the "Connected Successfully" message.

Lap Tests

Instructions: Given necessary templates, tools and materials you are required to perform the following tasks accordingly.



Create the following tasks based on the above Sample Project

Task 1: Install and Run XAMPP to your Computer.

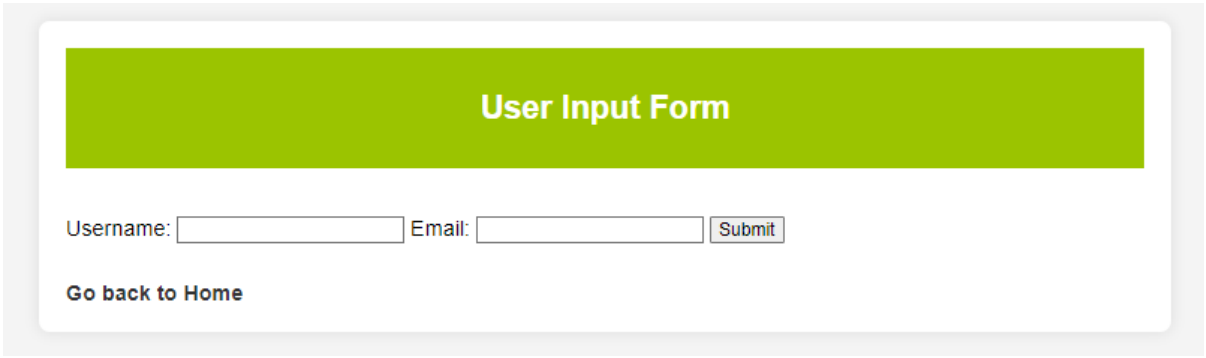
Task 2: Create a basic syntax to display 'Hello, PHP!' by creating message variable.

Task 3: Use arithmetic operators to display the sum of two numbers which is 10+5.

Task 4: Create If statement to show welcome message when the user is logged in.

Task 5: Create a simple for loop is used to generate unordered list with five list items.

Task 6: Create a link to Go to Form and create the following user input form to accept Username and Email.


User Input Form
Username:
Email:
Submit
Go back to Home

Unit Three: Produce Web Documents

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Introduction to XHTML
- Write Accessible XHTML
- Server-Side Scripts to XHTML Standards

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Introduce to XHTML
- Implement XHTML syntax and structure
- Translate server-side scripts to XHTML standards

3.1. Introduction to XHTML

XHTML is HTML markup that follows a more rigorous XML-style formatting and is the standard for HTML development going forward. Today, most browsers still display old or poorly applied HTML. Because it has stricter rules for applying markup, XHTML will, in the long run, provide better performance than HTML, especially for a wider variety of Web-connected devices, including cell phones and handheld devices. Although browser support for HTML will probably not disappear overnight, with a few modifications, you can start applying the XHTML standards in your Web pages right away.

A. The use of XHTML

- XHTML documents are validated with standard XML tools.
- It is easily to maintain, convert, edit document in the long run.
- It is used to define the quality standard of web pages.
- XHTML is an official standard of the W3C, your website becomes more compatible and accurate with many browsers.

B. Difference Between HTML and XHTML

HTML	**XHTML**
HTML or HyperText Markup Language is the main markup language for creating web pages	XHTML (Extensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML)
Flexible framework requiring lenient HTML specific parser	Restrictive subset of XML which needs to be parsed with standard XML parsers
Proposed by Tim Berners-Lee in 1987	World Wide Web Consortium Recommendation in 2000.
Application of Standard Generalized Markup Language (SGML).	Application of XML
Extended from SGML.	Extended from XML, HTML

3.2. Write Accessible XHTML

The XHTML documents contains three parts, which are discussed below:

- **DOCTYPE:** It is used to declare a DTD
- **head:** The head section is used to declare the title and other attributes.
- **body:** The body tag contains the content of web pages. It consists many tags.

Creating a XHTML web page, it is necessary to include DTD (Document Type Definition) declaration. There are three types of DTD which are discussed below:

- Transitional DTD
- Strict DTD
- Frameset DTD

i. Transitional DTD

It is supported by the older browsers which does not have inbuilt cascading style sheets supports. There are several attributes enclosing the body tag which are not allowed in strict DTD.

Syntax:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">
<html xmlns=" https://www.w3schools.com" xml:lang="en" lang="en">
```

ii. Strict DTD

Strict DTD is used when XHTML page contains only markup language. Strict DTD is used together with cascading style sheets, because this attribute does not allow CSS property in body tag.

Syntax:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"DTD/xhtml1-strict.dtd">
<html xmlns=" https://www.w3schools.com" xml:lang="en" lang="en">
```

iii. Frameset DTD

The frameset DTD is used when XHTML page contains frames.

Syntax:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

3.3. Server-Side Scripts to XHTML Standards

The reference to "Server-Side Scripts to XHTML Standards" likely suggests using server-side scripting languages (e.g., PHP, Python, Ruby) to generate XHTML-compliant markup on the server before sending it to the client's browser. XHTML (eXtensible HyperText Markup Language) is a stricter and more XML-compliant version of HTML. Here are some considerations for adhering to XHTML standards when using server-side scripts:

i. Proper Document Structure:

Ensure that the generated markup follows the proper XHTML document structure, including a DOCTYPE declaration, html, head, and body elements.

ii. Well-Formed XML:

XHTML requires well-formed XML syntax. Ensure that all tags are properly nested, closed, and that attribute values are enclosed in double quotes.

iii. Content-Type Header:

Set the correct Content-Type header in the HTTP response to indicate that the content is XHTML. For example:

```
bashCopy code
Content-Type: application/xhtml+xml
```

iv. Character Encoding:

Specify the character encoding using the **meta** tag or the HTTP Content-Type header to ensure proper rendering of special characters.

v. Separation of Concerns:

Encourage a separation of concerns by keeping server-side logic (e.g., PHP code) separate from XHTML markup. Use templates or a similar approach to organize code.

vi. Conditional Comments:

Use conditional comments or other techniques to serve different versions of XHTML or HTML based on the browser's capabilities. Older browsers may not fully support XHTML.

vii. Error Handling:

Implement proper error handling in your server-side scripts to catch any issues that may cause malformed XHTML and provide appropriate feedback or logging.

viii. CDATA Sections:

If including script or style content within XHTML, use CDATA sections to avoid conflicts with XML special characters.

```
xmlCopy code
<![CDATA[ Your script or style content here ]]>
```

ix. Client-Side Scripting:

If including client-side scripts, ensure that they are compatible with XHTML. Some scripts may need adjustments to work seamlessly with XHTML.

x. Validation:

Regularly validate your XHTML markup using tools like the W3C Markup Validation Service to catch any issues and ensure compliance with standards.

xi. Progressive Enhancement:

Consider using a progressive enhancement approach. Serve a baseline of functionality to all users, and then enhance the experience for those with modern browsers that fully support XHTML.

Remember that the choice of XHTML over HTML depends on specific project requirements and browser support considerations. In many modern web development scenarios, HTML5 is widely adopted and provides a more flexible and forgiving syntax compared to XHTML.

Self-Check 3

Part-I: Say True or False

1. XHTML is a more rigorous XML-style markup language.
2. HTML and XHTML have the same level of flexibility, requiring lenient HTML-specific parsers.
3. Transitional DTD is supported by older browsers and allows CSS properties in the body tag.
4. The use of CDATA sections in XHTML is essential to prevent conflicts with XML special characters when including script or style content.

Part-II: Answer the following questions accordingly

1. Explain the significance of using a DOCTYPE declaration in an XHTML document.
2. What are the advantages of XHTML over HTML in terms of document validation and maintenance?
3. Discuss the role of server-side scripts in adhering to XHTML standards.

Operation sheet 3.1: Create XHTML Page

Operation Title: Creating XHTML Page

Purpose: Develop an XHTML page with a dropdown menu for selecting months.

Equipment Tools and Materials:

- Computer
- Text editors (Notepad/Notepad ++/Sublime)

Steps in doing the task

1. Launch your preferred text editor and create a new file with the extension .**xhtml.** For Example**: dropdown_page.xhtml.**
2. Write the following code to your XHTML File:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
   <head>
      <title>Transitional DTD XHTML</title>
   </head>
   <body bgcolor="#dae1ed">
      <div style="color:#090;font-size:40px;
            font-weight:bold;text-align:center;
            margin-bottom:-25px;">XHTML Basics</div>
      <p style="text-align:center;font-size:20px;">
            Web Development and Database Administration portal</p>
      <p style="text-align:center;font-size:20px;">
      Option to choose month:
      <select name="month">
         <option selected="selected">January</option>
         <option>February</option>
         <option>March</option>
         <option>April</option>
```

```
        <option>May</option>
        <option>June</option>
        <option>July</option>
        <option>August</option>
        <option>September</option>
        <option>October</option>
        <option>November</option>
        <option>December</option>
      </select>
      </p>
   </body>
</html>
```

3. Save the file in a directory of your choice.

Quality Criteria: The output should be look like this.

Operation sheet 3.2: Create accessible XHTML Document

Operation Title: XHTML document with accessibility considerations

Purpose: Develop an XHTML document with a focus on accessibility to ensure a more inclusive web experience.

Equipment Tools and Materials:

- Computer
- Text editors (Notepad/Notepad ++/Sublime/VS Code)
- Web browser
- Knowledge of XHTML and accessibility best practices

Steps in doing the task

4. Open your preferred text editor. Create a new file and save it with an XHTML extension, for example, "AccessibleDocument.xhtml."
5. Start with the basic XHTML document structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<!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="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Accessible XHTML Document</title>
</head>
<body>
    <!-- Content Goes Here -->
</body>
</html>
```

6. Use semantic HTML elements to give meaning to the content.

```
<body>
    <header>
        <h1>Accessible XHTML Document</h1>
    </header>
    <nav>
        <!-- Navigation Links Go Here -->
    </nav>
    <main>
        <!-- Main Content Goes Here -->
    </main>
    <footer>
        <!-- Footer Content Goes Here -->
    </footer>
</body>
```

7. Provide text alternatives for non-text content using the alt attribute.

```
<img src="image.jpg" alt="Description of the Image" />
```

8. Use proper heading structure to create a logical document outline.

```
<main>
    <h2>Section 1</h2>
    <p>Content of Section 1.</p>

    <h2>Section 2</h2>
    <p>Content of Section 2.</p>
</main>
```

9. Ensure links are descriptive and provide context. Use semantic elements for navigation.

```
<nav>
    <ul>
        <li><a href="#section1">Section 1</a></li>
        <li><a href="#section2">Section 2</a></li>
    </ul>
</nav>
```

10. Use accessible forms with proper labels for input fields.

```
<form action="submit.php" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" />
</form>
```

11. If using tables, include captions for clarity.

```
<table>
    <caption>Table Title</caption>
    <!-- Table Content Goes Here -->
</table>
```

12. Use online tools or browser extensions to check the document's accessibility. Manually test with keyboard navigation and screen readers.

Quality Criteria:

Adjust the content and structure based on your specific needs and content.

Lap Tests

Instructions: Given necessary templates, tools and materials you are required to perform the following tasks accordingly.

Task 1: Create XHTML document with Strict DTD.

Task 2: Create accessible XHTML Document

Unit Four: Test scripts and debug

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Iterative Testing for Functionality
- Documentation and Submission for Approval

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Implement iterative testing for functionality
- Create comprehensive documentation.
- Prepare and submit for approval

4.1. Iterative Testing for Functionality

Iterative testing for functionality in PHP is a crucial practice that involves continuous validation of code at different stages of development. This approach ensures that the software is reliable and functional as it evolves. Let's explore key considerations for implementing iterative testing for functionality in PHP through various testing levels.

I. Unit Testing:

Unit testing is the foundation of iterative testing in PHP, focusing on verifying the correctness of individual functions or methods. Using tools like PHPUnit, developers write test cases for each function, targeting different scenarios and edge cases. Executing these unit tests frequently during development provides immediate feedback on the accuracy of specific code components.

II. Integration Testing:

Integration testing is essential for ensuring that different components or modules of the application work seamlessly together. Automation of integration tests helps maintain consistency and identifies integration issues early in the development process.

III. Functional Testing:

Functional testing takes a broader view, validating the functionality of larger parts of the application, such as modules or features. PHPUnit and Behat, a behavior-driven development tool, can be used for writing tests that simulate user interactions. These tests cover end-to-end functionality, ensuring that the application meets specifications and performs as expected under various use cases.

IV. Regression Testing:

Regression testing is critical for detecting and preventing the introduction of new bugs as code evolves. This practice ensures that new changes do not break existing functionality, and version control systems help identify changes that may impact testing.

V. Continuous Integration (CI):

Continuous Integration (CI) automates the process of running tests whenever code changes are committed. Using CI tools such as Jenkins, Travis CI, or GitHub Actions, developers set up pipelines to execute tests automatically on each code push. Immediate feedback on the impact of

code changes facilitates early bug detection and promotes a reliable and continuously validated codebase.

VI. Test-Driven Development (TDD):

Test-Driven Development (TDD) encourages developers to write tests before implementing the actual code. By defining tests based on expected functionality and incrementally building the code to pass these tests, developers ensure test coverage from the outset. TDD promotes a robust testing culture and facilitates iterative improvements.

VII. Data Driven Testing:

Data-driven testing involves testing a function or method with multiple sets of input data to ensure comprehensive coverage. Developers define test cases with different input values and execute the same test logic with each set of data. This approach helps validate the function's behavior under various conditions, contributing to a more resilient and adaptable codebase.

VIII. Monitoring and Logging:

Implementing monitoring and logging mechanisms is crucial for gaining insights into the application's behavior in a live environment. By recording relevant information during runtime, developers can identify issues and track performance. Logs and monitoring data contribute to an iterative improvement process, allowing developers to refine functionality based on real-world usage.

Incorporating these testing strategies into PHP development ensures that each iteration is thoroughly tested for functionality. This iterative testing approach reduces the risk of introducing bugs, enhances software quality, and promotes a more reliable and maintainable codebase.

4.2. Documentation and Submission for Approval

Documentation and submission for approval are essential steps in the software development process, ensuring transparency, maintainability, and stakeholder alignment. Here's a guide on how to handle documentation and submission for approval in a PHP project:

I. Code Documentation

Use PHPDoc comments to document functions, methods, classes, and variables. Provide details on parameters, return types, and a brief description of functionality.

Example:

```php
/**
 * Calculate the sum of two numbers.
 *
 * @param int $a The first number.
 * @param int $b The second number.
 * @return int The sum of $a and $b.
 */
function add($a, $b) {
    return $a + $b;
}
```

README File:

Create a README file to provide an overview of the project. Include information on installation, configuration, usage, and any dependencies.

II. Version Control:

Use version control (e.g., Git) to track changes and collaborate with a team. Commit regularly with meaningful commit messages. Utilize branches for feature development and bug fixes.

III. Testing Documentation:

Write PHPUnit tests for functions, methods, and classes. Document test cases with clear descriptions and expected outcomes.

IV. Deployment Documentation:

Document the deployment process, including server requirements and configurations. Specify any environment variables or settings needed for deployment.

V. API Documentation:

If your project involves APIs, use Swagger or OpenAPI to document API endpoints. Include details on request and response formats, authentication, and usage examples.

VI. Submission for Approval:

Address any issues or concerns raised during the review process. Clearly define the approval workflow within your team or organization. Specify who needs to review and approve changes before deployment.

VII. Continuous Integration (CI):

Integrate CI tools to automate checks for coding standards, tests, and other quality measures. Ensure that CI passes before submitting code for approval.

VIII. User Manuals (if applicable):

If your project has end-users, create user manuals or guides. Provide clear instructions on how to use the application.

IX. Communication:

Communicate changes and updates to stakeholders, particularly if they involve user-facing features. Provide demonstrations or walkthroughs if necessary.

By following these steps, you can ensure that your PHP project is well-documented, follows coding standards, undergoes thorough testing, and is ready for submission and approval. Effective documentation and a streamlined submission process contribute to a successful and maintainable software project.

Self-Check 4

Part-I: Say True or False

1. Unit Testing validates individual functions or methods in isolation.
2. Documentation is optional and does not significantly contribute to project success.
3. Version control tools like Git are essential for effective collaboration.
4. Continuous Integration (CI) ensures code meets quality standards before submission.

Part-II: Choose the correct answer

13. What is the primary focus of Unit Testing in PHP?
    A) Validating integration between components
    B) Verifying correctness of individual functions or methods
    C) Testing end-to-end functionality
    D) Detecting and preventing regression issues
14. Which testing level is crucial for identifying integration issues early in the development process?
    A) Unit Testing C) Functional Testing
    B) Integration Testing D) Regression Testing
15. Continuous Integration (CI) tools automate the process of running tests whenever:
    A) Code is written
    B) Code changes are committed
    C) Code is deployed
    D) Bugs are reported
16. Data Driven Testing in PHP involves:
    A) Testing a function with only one set of input data
    B) Ignoring input data variations
    C) Testing a function with multiple sets of input data
    D) Testing only for specific edge cases

Unit Five: Set up Security

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Permission management for error prevention
- Server security for database attack prevention

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Determine Necessary Permissions to prevent error message
- Configure server software to minimize potential database attacks

5.1. Permission management for error prevention

Handling permissions properly in PHP is crucial for preventing error messages and ensuring the security and stability of your web application. Permissions control access to files, directories, and resources on the server, and improper settings can lead to security vulnerabilities and expose sensitive information. Let's see the key aspects of managing permissions in PHP to avoid error messages.

- **File and Directory Permissions**

When your PHP application interacts with files or directories, it's essential to set appropriate permissions. The common permissions are read (‘**r**’), write (‘**w**’), and execute (‘**x**’). These permissions can be assigned to the owner of the file, the group associated with the file, and others. For instance, if PHP needs to write to a file or directory, the web server user (e.g., **‘www-data’** for Apache) should have write permissions.

Improper permissions can result in errors such as "Permission Denied" or "Access Forbidden." To mitigate this, execute the following command to grant write permissions to the web server user:

```
chmod o+w /path/to/directory
```

- **Database Connection Permissions**

PHP applications often connect to databases, and managing database permissions is crucial. Incorrect permissions may lead to connection errors or expose sensitive information in error messages. When configuring database users for your PHP application, follow the principle of least privilege. Only grant the necessary permissions for the application to function correctly, avoiding overly permissive settings.

Ensure that your database credentials, especially passwords, are securely stored and not directly exposed in code. Use environment variables or configuration files outside the web root to safeguard sensitive information.

- **Error Handling and Display**

PHP provides settings to control how errors are handled and displayed. While developing, it's common to display errors to aid debugging, but this should be disabled in a production environment to prevent sensitive information leakage.

In your **'php.ini'** configuration file or within your PHP code, set the following directives:

```
display_errors = Off
log_errors = On
```

By turning off **'display_errors'**, you prevent error messages from being shown to users, while **'log_errors'** ensures that errors are logged for later analysis.

- **Secure File Inclusions:**

If your PHP application uses file inclusion functions like **include** or **require**, be cautious about user input. Only include files from trusted sources, and avoid dynamically constructing file paths based on user input to prevent directory traversal attacks.

Verify that the files you include have appropriate permissions to be read by the web server user. Insufficient permissions can lead to "File Not Found" errors or unauthorized access.

- **Web Server Configuration:**

The web server itself plays a role in enforcing permissions. Ensure that the web server runs with the minimum necessary privileges.

Check and update your virtual host or server block configurations to set the appropriate permissions. For Apache, this might involve using the **Allow** and **Deny** directives, while Nginx relies on the **'location'** block and **'try_files'** directive.

- **Regular Security Audits:**

Regularly audit your application's security, including permissions. Conduct security reviews and penetration testing to identify and address any vulnerabilities. Automated tools and manual inspection can help ensure that permissions are correctly configured and that your PHP application is robust against potential exploits.

Managing permissions in PHP involves a combination of setting the correct file and database permissions, configuring error handling, securing file inclusions, and maintaining a secure web server configuration. Adhering to these best practices helps prevent error messages, protects sensitive information, and enhances the overall security of your PHP application.

5.2. Server security for database attack prevention

Securing your server software is a critical step in minimizing the potential for database attacks. Database vulnerabilities often exploit weaknesses in the server environment, so it's crucial to configure your server securely. Here are key practices to follow in order to configure your server software and reduce the risk of database attacks:

- **Update and Patch**

Regularly updating your server software, including the operating system, web server (e.g., Apache, Nginx), and database server (e.g., MySQL, PostgreSQL), is a foundational practice in security. Ensure that security patches are applied promptly to address known vulnerabilities and enhance overall system resilience.

- **Firewall Configuration**

Configure a firewall to control incoming and outgoing traffic. Limit access to only necessary ports, blocking unused ports to reduce the attack surface. Implement rules to restrict access to the database server to trusted IP addresses, enhancing network security.

- **Secure Database Configuration**

Enhance the security of your database by changing default credentials and usernames. Disable unnecessary database services and features, reducing potential attack vectors. Implement strong password policies for database users to mitigate the risk of unauthorized access.

- **Encryption**

Enable encryption for data in transit by implementing protocols like TLS/SSL. Ensure that database connections are encrypted to protect sensitive information during transmission, making it more challenging for attackers to intercept or manipulate data.

- **Access Controls**

Implement robust access controls and permissions for database users. Adhere to the principle of least privilege, granting users only the minimum necessary permissions for their tasks. Regularly review and update user permissions to maintain a secure environment.

- **Database Auditing**

Enable database auditing features to log and monitor activities within the database. Regularly review these logs to detect and respond to suspicious or unauthorized activities. Set up alerts for anomalous events to enhance proactive threat detection.

- **Backup and Recovery**

Establish regular backup procedures for your databases. Store backups securely and regularly test the restoration process to ensure data recovery in the event of a successful attack or data loss.

- **Security Headers**

Configure security headers in your web server to bolster overall security. Implement headers such as HTTP Strict Transport Security (HSTS) to enforce secure connections and protect against certain types of attacks.

- **Limiting SQL Injection Risks**

Protect against SQL injection attacks by using parameterized queries or prepared statements. Avoid constructing dynamic SQL queries based on user input, as this can expose vulnerabilities that attackers may exploit.

- **Web Application Firewall (WAF)**

Consider implementing a Web Application Firewall (WAF) to filter and monitor HTTP traffic between your web application and the internet. WAFs can detect and prevent common web application attacks, including those targeting databases.

- **Regular Security Audits**

Conduct regular security audits and vulnerability assessments to identify potential weaknesses. Utilize security scanning tools to assess the system's integrity and promptly address any identified vulnerabilities.

- **Secure File Permissions**

Set appropriate file permissions for configuration files and sensitive directories. Limit access to critical files, ensuring that only authorized users have read and write permissions. Proper file permissions are crucial for preventing unauthorized access to sensitive information.

- **Server Hardening**

Follow server hardening best practices, such as disabling unnecessary services and limiting the use of root or administrator accounts. Server hardening measures contribute to a more secure server environment and reduce the risk of unauthorized access.

- **Monitor and Respond**

Establish monitoring systems to detect unusual or suspicious activities on your server. Implement an incident response plan to respond swiftly to security incidents, minimizing the potential impact of an attack.

- **Educate and Train Users**

Educate database users and administrators on security best practices. Raise awareness about social engineering threats and emphasize the importance of safeguarding credentials. A well-informed user base contributes to the overall security posture of your server environment.

Self-Check 5

Part-I: Choose the correct answer

1. Proper PHP file permissions are crucial to prevent "Permission Denied" errors.
2. Displaying PHP errors in a production environment is recommended for effective debugging.
3. Regularly updating server software is foundational for server security.
4. Encryption using TLS/SSL protects sensitive information during data transmission.
5. Web Application Firewalls (WAFs) specifically prevent SQL injection attacks in PHP.

Part-II: Answer the following questions accordingly

1. Explain the role of firewall configuration in minimizing the risk of database attacks.
2. Discuss the importance of the principle of least privilege in database access controls.
3. Describe the purpose of server hardening and provide examples of server hardening measures.
4. Explain how educating and training users enhances overall server security.

Reference

Books

PHP and MySQL Web Development: Second Edition By Luke Welling, Laura Thomson

Learning PHP 7 By Antonio Lopez · 2016

Modern PHP New Features and Good Practices By Josh Lockhart · 2015

Learn PHP 7 By Steve Prettyman 2015

Learning PHP, MySQL, JavaScript, and CSS By Robin Nixon · 2012

URL

https://www.w3schools.com

https://blotter.com/dynamic-functionality-of-a-website/

https://rockcontent.com/blog/scripting-languages/

https://www.geeksforgeeks.org/web-scripting-and-its-types/

https://www.mindspun.com/blog/website-requirements-document-a-complete-guide/

https://thesquareagency.com/blog/website-requirements-document-template

https://wpwebinfotech.com/blog/setup-php-development-environment-on-windows/

https://www.c-sharpcorner.com/article/how-to-install-and-configure-xampp-in-windows-10/

https://rootstack.com/en/blog/php-what-basic-syntax-programming-language

https://studyglance.in/php/display.php?tno=7&topic=Operators%20in%20php

https://www.developer.com/languages/php/php-control-structures/

https://www.javatpoint.com/php-form

https://www.cloudways.com/blog/adminer-vs-phpmyadmin/

https://www.cloudways.com/blog/connect-mysql-with-php/

https://hackernoon.com/10-features-of-a-secure-website-m16s36y5

https://www.javatpoint.com/what-is-xhtml

https://www.geeksforgeeks.org/xhtml-introduction/

Developer's Profile

NO	Name	Qualification	Field of Study	Organization/ Institution	Mobile number	E-mail
1	Frew Atkilt	M-Tech	Network & Information Security	Bishoftu Polytechnic College	0911787374	frew.frikii@gmail.com
2	Gari Lencha	MSc	ICT Management	Gimbi Polytechnic	0917819599	Garilencha12@gmail.com
3	Kalkidan Daniel	BSc	Computer Science	Entoto Polytechnic College	0978336988	kalkidaniel08@gmail.com
4	Solomon Melese	M-Tech	Computer Engineering	M/G/M/Polytechnic College	0918578631	solomonmelese6@gmail.com
5	Tewodros Girma	MSc	Information system	Sheno Polytechnic College	0912068479	girmatewodiros @gmail.com
6	Yohannes Gebeyehu	BSc	Computer Science	Entoto Polytechnic College	0923221273	yohannesgebeyehu73@gmail.com