Module - 3

1. Describe Server Side Scripting
2. List Server side scripting languages
3. PHP(Hypertext Preprocessor)
4. State advantages of PHP
5. Software used for Server Side Programming
6. How Apache, MySQL, and PHP are installed and configured
7. How a PHP script is embedded in a webpage and executed
8. Describe PHP language elements
    1) Variable
    2) Data Types
    3) Operators
    4) Defining a Constant
    5) Incrementing and Decrementing Variable
    6) Conditional Statements
    7) Loops
    8) Function
    9) Arrays
        a) Associative Array
        b) Getting the Size of an Array

c) Manipulating Arrays

9. Describe Form Handling

a) Global and Environment variables

b) A Script to Acquire User Input-GET and POST Method

c) Redirecting the User

10. Cookie Handling in PHP

11. Session & Cookie Handling in PHP

12. File Operations in PHP

a) Create a file

b) Write into a file

c) Read from a file

d) Close a file

e) File operation modes

f) Create and Delete a file

g) Inbuilt functions

13. File Upload in PHP

1. Describe Server Side Scripting

- The server-side environment that runs a scripting language is a web server.
- A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages.
- This HTML is then sent to the client browser.
- It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

2. List Server side scripting languages

- ASP.NET (Active Server Pages dot network enabled technology)
- ColdFusion Markup Language
- Perl
- Ruby
- PHP(Hypertext Preprocessor)
- JSP(Java Server Page)
- C# (C Sharp)

3. PHP(Hypertext Preprocessor)

- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"
- PHP was running on more than 1 million hosts
- PHP code are executed on the server
- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can encrypt data

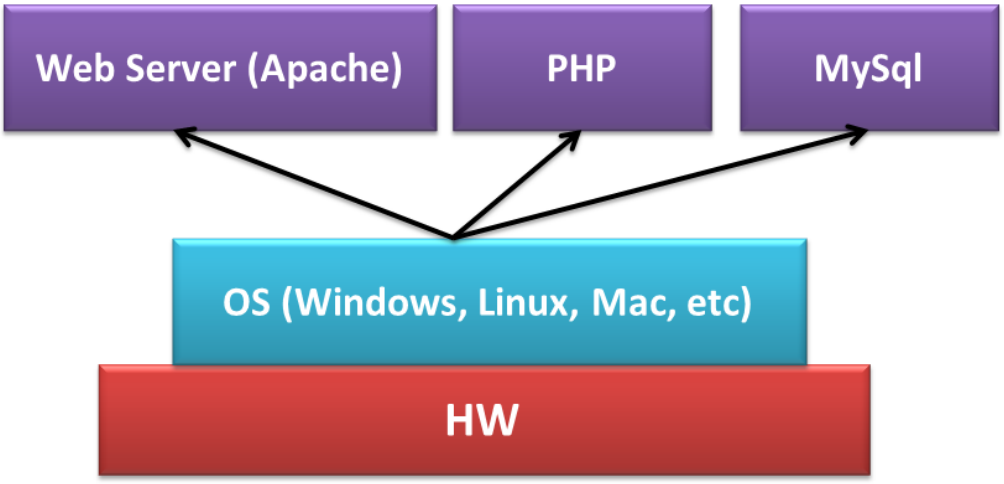
- PHP Current Version PHP7
- PHP runs on various platforms (Windows, Linux, Unix, Mac OS, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is easy to learn and runs efficiently on the server side

4. State advantages of PHP

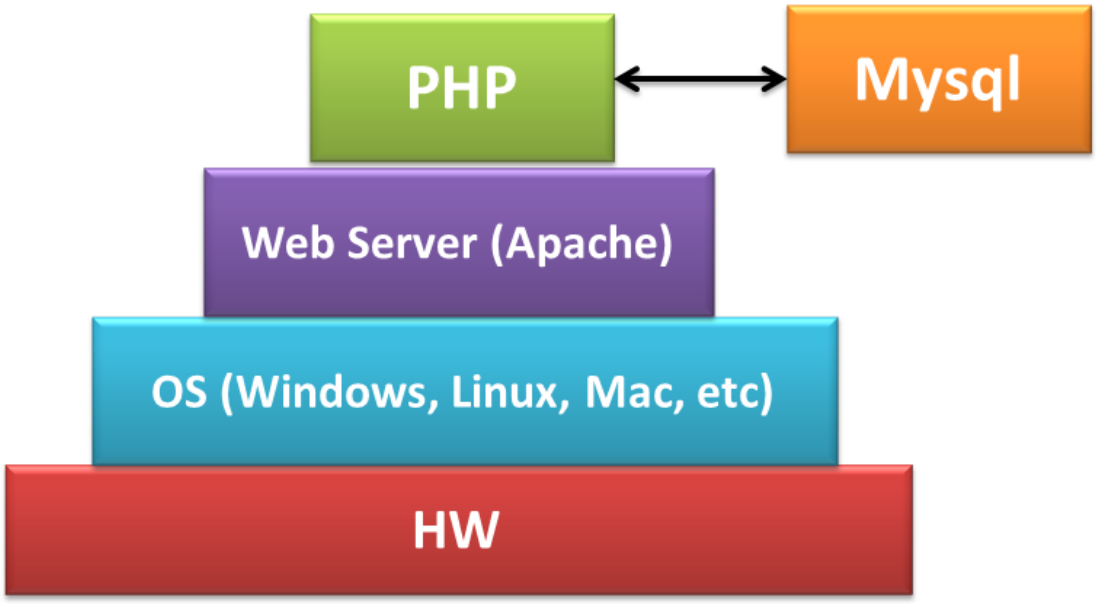
- **Speed of Development**
  - The syntax of PHP is based on other programming languages, primarily C and Perl.
  - If you already know C or Perl, or a C-like language such as C++ or Java, you will be productive using PHP almost immediately.
- **PHP Is Open Source**
- **Performance**
  - PHP is very fast.
- **Database Integration**
  - PHP has native connections available to many database systems.
  - PHP supports a wide range of databases
- **Scalability**
  - Support large no of servers and users
- **Object-Oriented Support**
  - PHP version 7 has well-designed object-oriented features.
- **Powerful library support**
  - Because PHP was designed for use on the Web, it has many built-in functions for performing many useful web-related tasks.
- **PHP is free.**
  - PHP is free of cost.
  - Free to download and use anytime anywhere.
- **Portability**
  - PHP is available for many different operating systems.
  - Well-written code will usually work without modification on a different system running PHP.
  - PHP runs on various platforms (Windows, Linux, Unix, Mac OS, etc.)
- **Availability of source code**
  - You have access to PHP's source code
  - If you want to modify something or add to the language, you are free to do so.

- **Availability of support**
  - The PHP documentation and community are mature and rich resources with a wealth of information to share.
  - PHP development by offering support and related software on a commercial basis.

5. Software used for Server Side Programming



Software Running on Server Machine



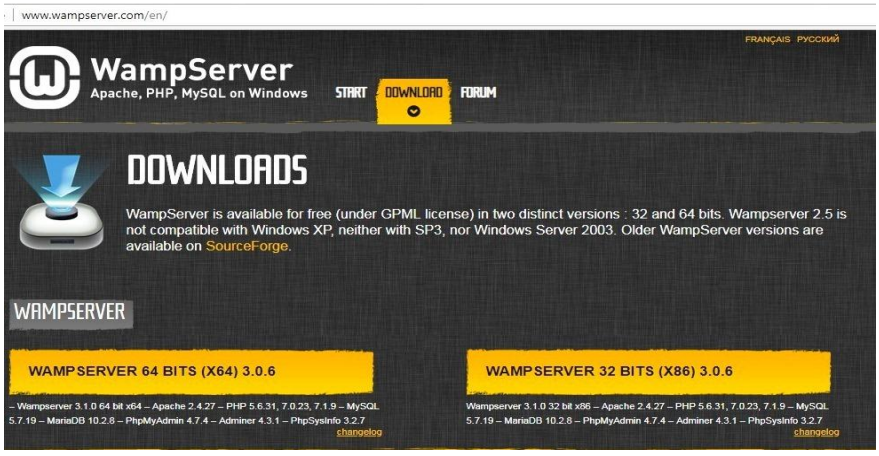
6. How Apache, MySQL, and PHP are installed and configured

- We can use WAMP server for configuring Apache MySQL and PHP

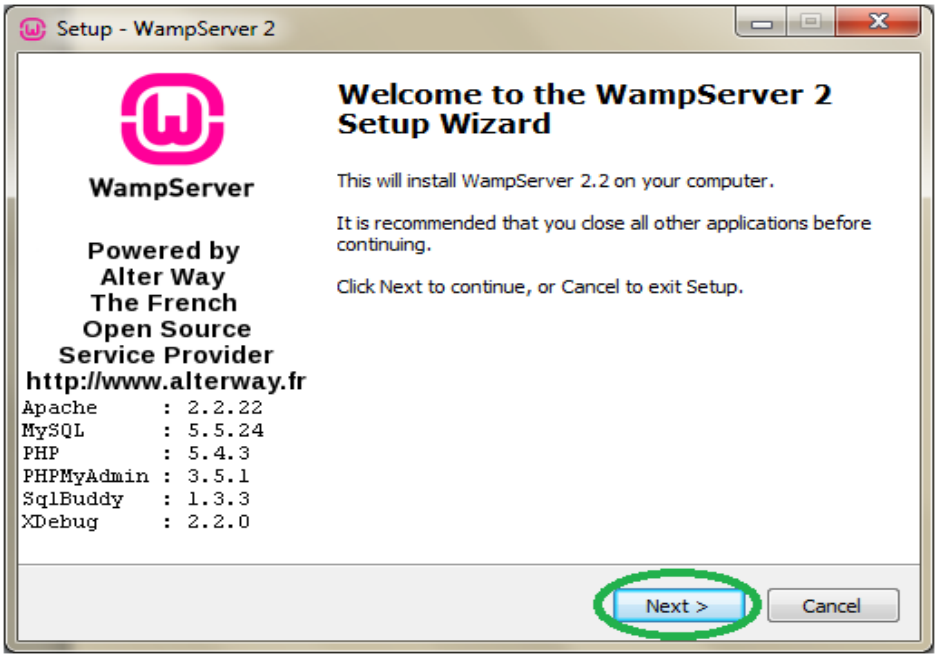
- It is often used for web development.
- The most important part of the WAMP package is Apache which is used run the web server within Windows.
- MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database.
- By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.

WAMP Server installation Steps

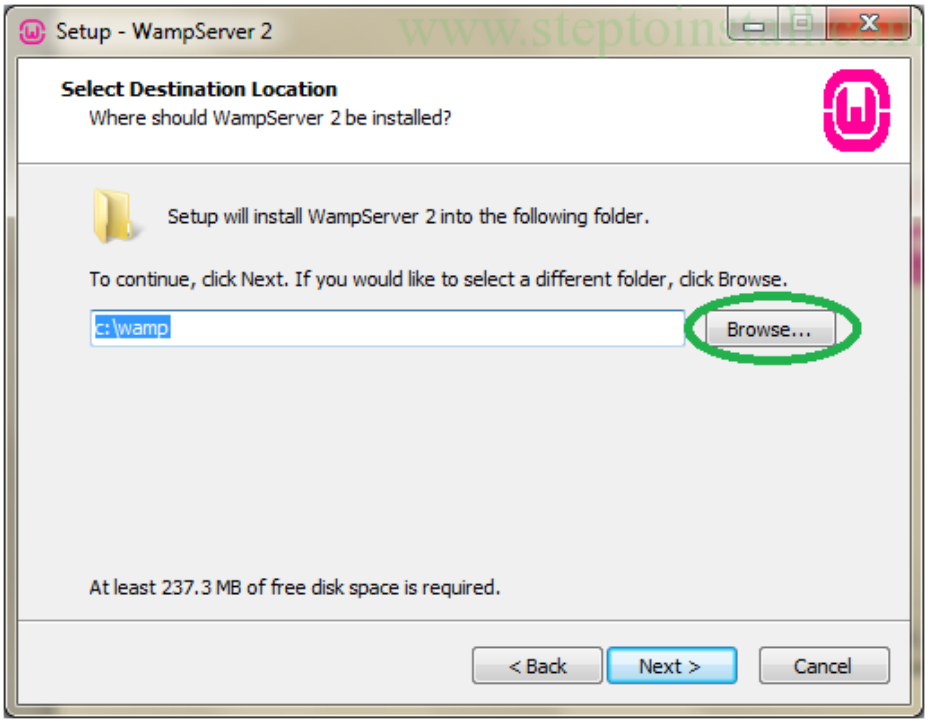
- Download the WAMP server Package ".exe" file from official WAMP server page. http://www.wampserver.com/en/

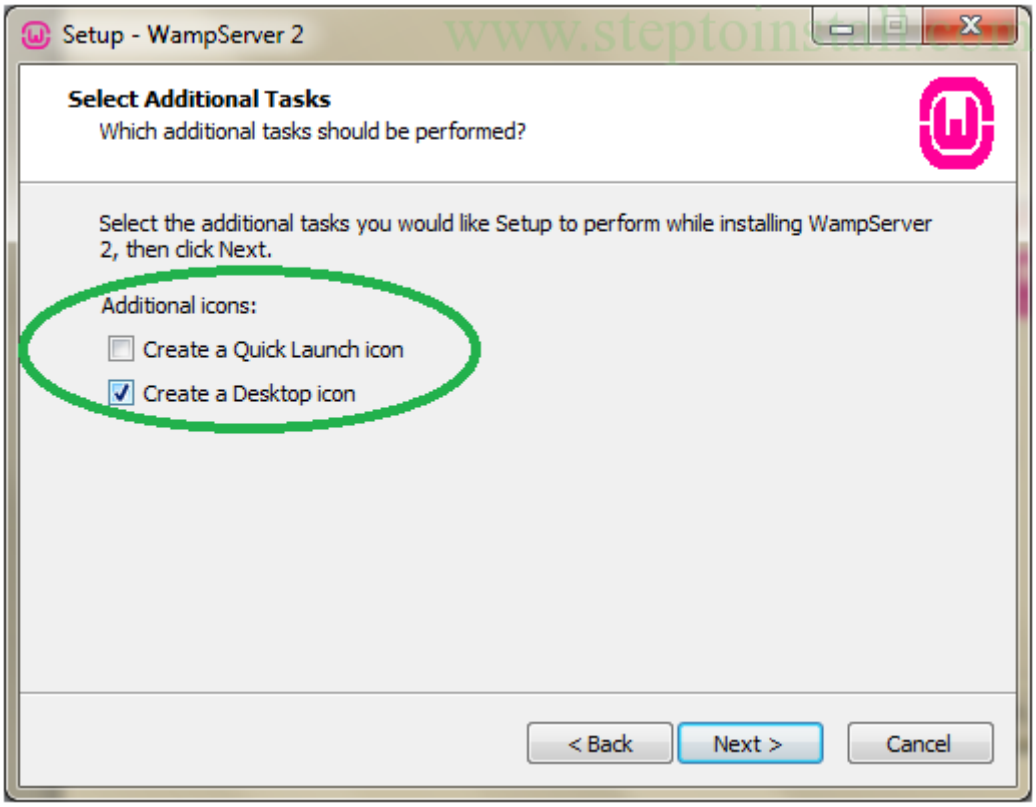


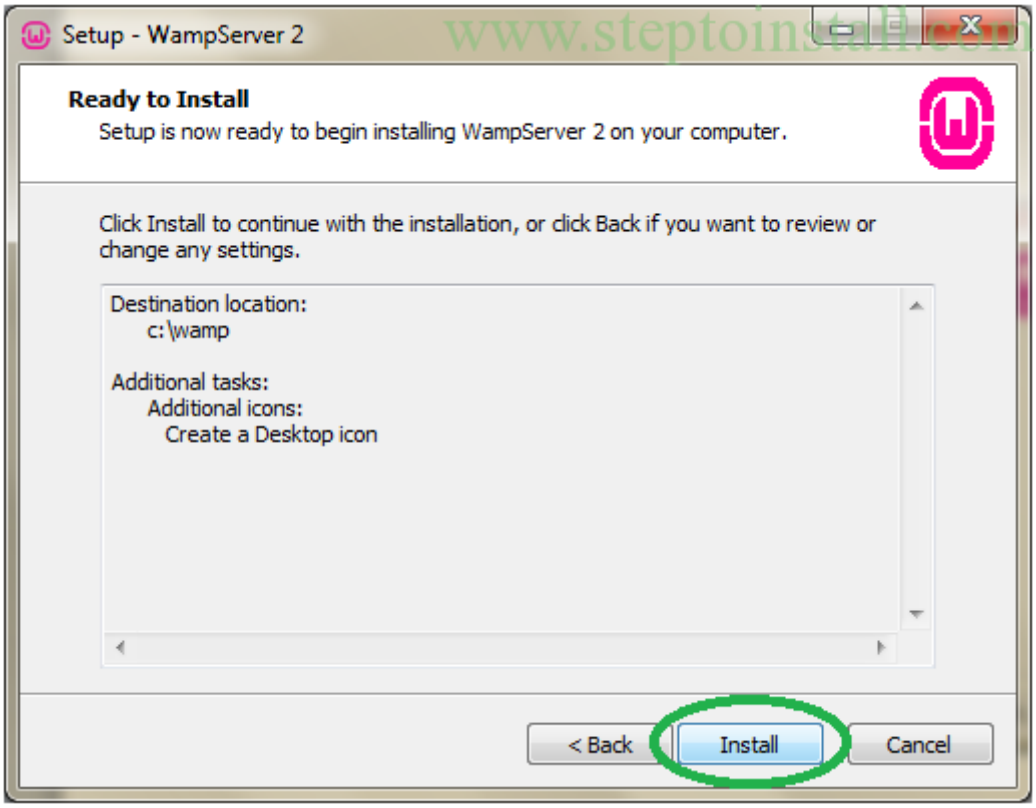
- You'll get below Setup Wizard window, When you Run WAMP server .exe file. It shows config of WAMP. Click 'Next' button to continue.

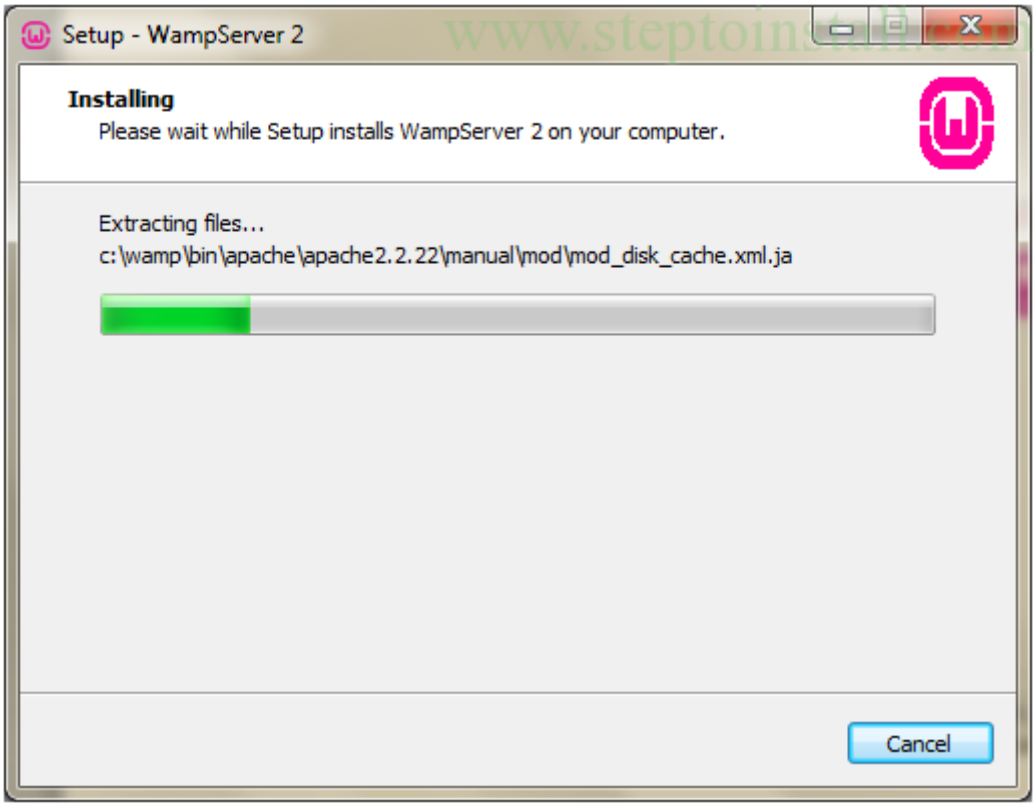


- You should select the Server Location. Most of the server installing into C drive

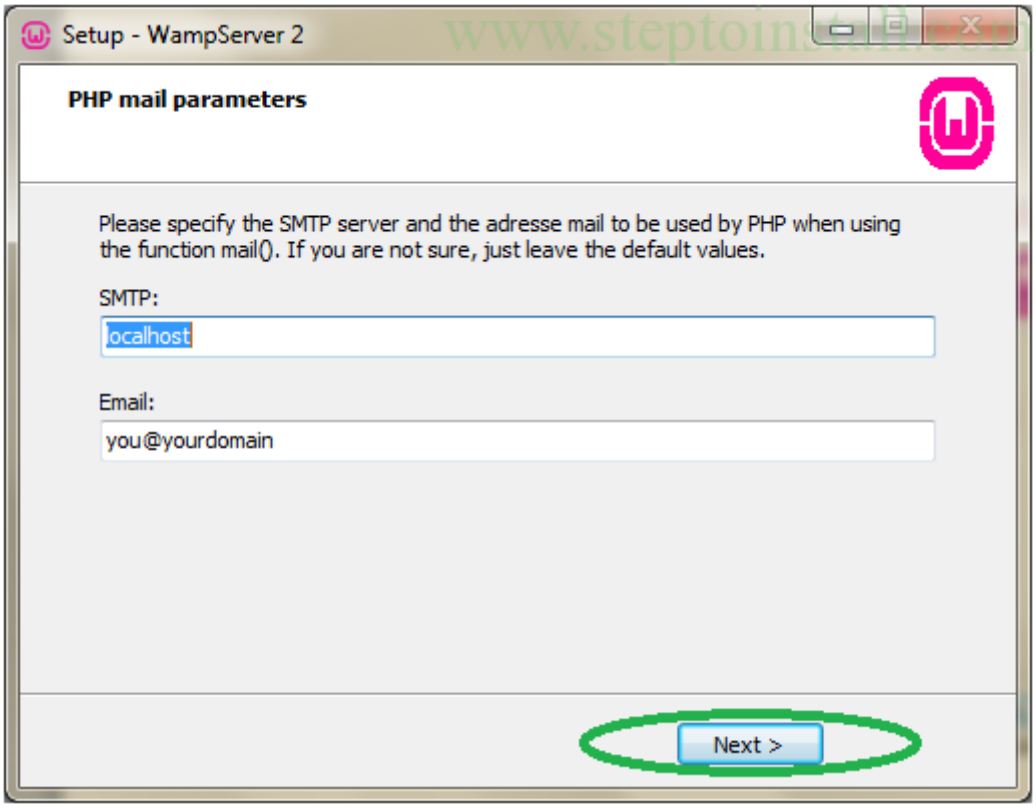


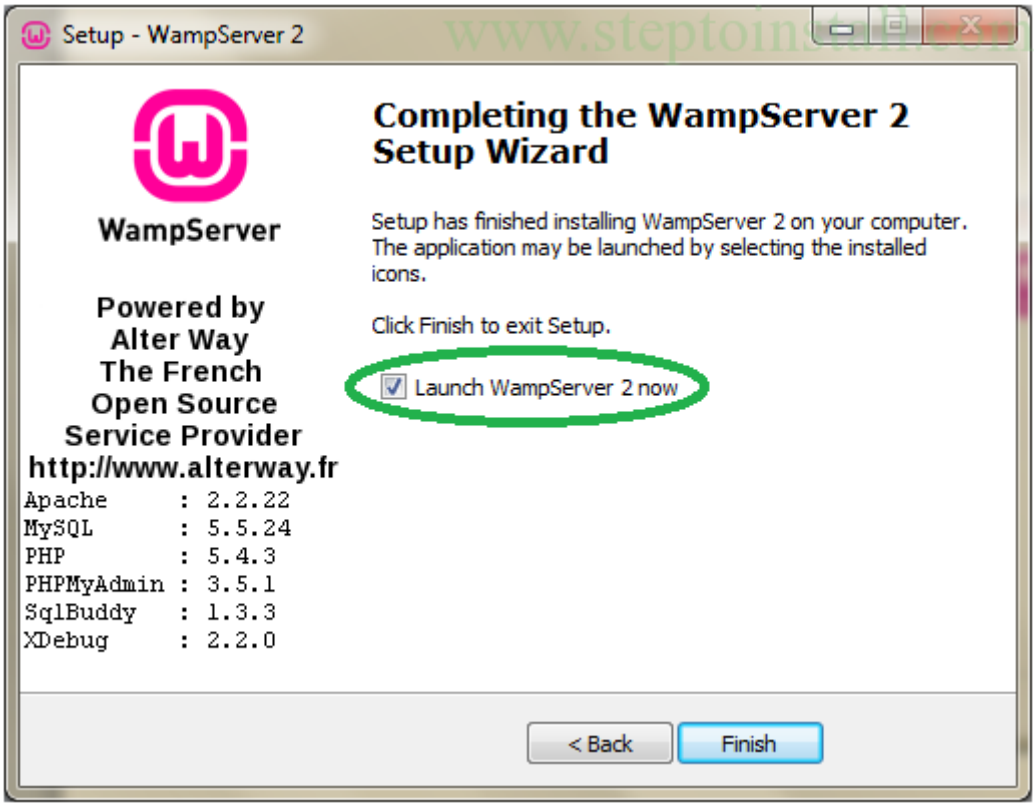

Setup - WampServer 2
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing WampServer 2, then click Next.
Additional icons:
Create a Quick Launch icon
Create a Desktop icon
< Back
Next >
Cancel


Setup - WampServer 2
Ready to Install
Setup is now ready to begin installing WampServer 2 on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
c:\wamp
Additional tasks:
Additional icons:
Create a Desktop icon
< Back
Install
Cancel



- Set SMTP server and mail address to default and click next




Setup - WampServer 2
WampServer
Completing the WampServer 2 Setup Wizard
Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch WampServer 2 now
Powered by Alter Way The French Open Source Service Provider http://www.alterway.fr
Apache : 2.2.22
MySQL : 5.5.24
PHP : 5.4.3
PHPMyAdmin : 3.5.1
SqlBuddy : 1.3.3
XDebug : 2.2.0
< Back
Finish


WampServer
Version 2.2 Version Française
Server Configuration
Apache Version : 2.2.22
PHP Version : 5.4.3
Loaded Extensions : Core, date, iconv, pcre, tokenizer, PDO, xmlreader, mysql, xdebug, bcmath, ereg, json, Reflection, zip, Phar, xmlwriter, mysqli, calendar, filter, mcrypt, session, zlib, SimpleXML, apache2handler, pdo_mysql, com_dotnet, ftp, SPL, standard, libxml, wddx, mbstring, pdo_sqlite, ctype, hash, odbc, mysqlnd, dom, xml, gd, mhash
MySQL Version : 5.5.24
Tools
phpinfo()
phpmyadmin
Your Projects

7. How a PHP script is embedded in a webpage and executed

Tag Style	Start Tag	End Tag
Standard tags	<?php	?>
Short tags	<?	?>
Script tags	<SCRIPT LANGUAGE="php">	</SCRIPT>

- **Sample Programs**

```
<?
print("Hello Web!");
?>
```

```
<?php
print("Hello Web!");
?>
```

```
<SCRIPT LANGUAGE="php">
print("Hello Web!");
</SCRIPT>
```

➢ **Combining HTML and PHP**

```
<html>
<head>
<title>PHP script including
HTML</title>
</head>
<body>
<b>
<?php
print "hello world";
?>
</b>
</body>
</html>
```

8. Describe PHP language elements

1) Variable

- A variable consists of a name that you can choose, preceded by a dollar ($) sign.
- The variable name can include letters, numbers, and the underscore character (_).
- Variable names cannot include spaces or characters that are not alphanumeric
- Eg: $name,$n5989

a) Variable Scope

i. Local

ii. Global

i. Local

```
<?php
function test()
{
$testvariable = "this is a
test variable";
}
print "test variable:
$testvariable<br>";
?>
```

ii. Global

```
<?php
$life=42;
function meaningOfLife()
{
global $life;
print "The meaning of life is
$life<br>";
}
meaningOfLife();
?>
```

2) Data Types

Type	*Example*	*Description*
Integer	5	A whole number
Double	3.234	A floating-point number
String	"hello"	A collection of characters
Boolean	true	One of the special values true or false

```
<body>
<?php
$testing = 5;
print gettype( $testing );
print "<br>";
$testing = "five";
print gettype( $testing );
print("<br>");
$testing = 5.0;
print gettype( $testing );
print("<br>");
$testing = true;
print gettype( $testing );
print "<br>";
?>
</body>
```

Output:

```
integer
string
double
boolean
```

3) Operators

a) Arithmetic Operators

Operator	*Name*	*Example*	*Example Result*
+	Addition	10+3	13
-	Subtraction	10-3	7
/	Division	10/3	3.3333333333333
*	Multiplication	10*3	30
%	Modulus	10%3	1

b) Assignment Operator

Operator	*Example*	*Equivalent to*
+=	$x += 5	$x = $x + 5
−=	$x −= 5	$x = $x − 5
/=	$x /= 5	$x = $x / 5
*=	$x *= 5	$x = $x * 5
%=	$x%=5	$x = $x % 5
.=	$x .= "test"	$x = $x" test"

c) Concatenation Operator

```
$a="hello"."
world";
Print $a;
```

Output:

```
hello world
```

d) Comparison Operators

Operator	*Name*	*Returns True if*	*Example*	*Result*
==	Equivalence	Left is equivalent to right	$x == 5	false
!=	Non-equivalence	Left is not equivalent to right	$x != 5	true
===	Identical	Left is equivalent to right and they are the same type	$x === 5	false

>	Greater	Left is than greater than right	$x >4	false
>=	Greater than or equal to	Left is greater than or equal to right	$x >= 4	true
<	Less than	Left is less than right	$x < 4	false
<=	Less than or equal to	Left is less than or equal to right	$x <= 4	true

e) Logical Operators

Operator	*Name*	*Returns True if...*	*Example*	*Result*
||	Or	Left or right is true	true || false	true
or	Or	Left or right is true	true or false	true
xor	Xor	Left or right is true but not both	true xor true	false
&&	And	Left and right are true	true && false	false
and	And	Left and right are true	true and false	false
!	Not	The single operand is not true	! true	false

4) Defining a Constant

Syntax

```
define( "CONSTANT_NAME", 42 );
```

```
<body>
<?php
define ( "USER", "Gerald" );
print "Welcome ".USER;
?>
</body>
```

Output

```
Welcome Gerald
```

5) Incrementing and Decrementing Variable

```
$x++; // $x is incremented
$x- - ; // $x is decremented
```

6) Conditional Statements

a) if Statements
b) if . . . else Statements
c) else if statements
d) A switch Statement
e) ? Operator

a) if Statements

```
<?php
if ( $mood == "happy" )
{
print "Hooray, I'm in a good mood";
 }
?>
```

b) if . . . else Statements

```
<?php
if ( $mood == "happy" )
{
print "Hooray, I'm in a good mood";
}
else
{
print "Not happy but $mood";
}
?>
```

c) else if statements

```
<?php
$mood = "sad";
if ( $mood == "happy" )
{
print "Hooray, I'm in a good mood";
}
elseif ( $mood == "sad" )
{
print "Awww. Don't be down!";
}
else
{
print "Neither happy nor sad but
$mood";
}
?>
```

d) The switch Statement

```
<?php
$mood = "sad";
```

```
switch ( $mood )
{
case "happy":
print "Hooray, I'm in a good mood";
break;
case "sad":
print "Awww. Don't be down!";
break;
default:
print "Neither happy nor sad but
$mood";
}
?>
```

e) ? Operator

Syntax

```
(expression)?returned_if_expression_is_true:returned_if_
expression_is_false;
```

```
<?php
$mood = "sad";
( $mood=="happy" )?"Hooray, I'm in a good
mood":"Not happy but $mood";
?>
```

7) Loops

a) while Statement

b) do..while Statement

c) for Statement

d) foreach

a) while Statement

```
<?php
$counter = 1;
while ( $counter <= 12 )
{
print "$counter times 2 is".$counter."<br>";
$counter++;
}
?>
```

b) do..while Statement

```
<?php
$num = 1;
do
{
print "Execution number: ".$num;
$num++;
}
while ( $num > 10);
?>
```

c) for Statement

```
<?php
for ( $counter=1; $counter<=12; $counter++ )
{
print "$counter times 2 is ".$counter*2;
}
?>
```

d) foreach

```
Syntax
        foreach( $array as $temp )
        {
        //...
        }
```

```
$users = array ("Bert", "Sharon", "Betty", "Harry" );
foreach ( $users as $val )
{
print "$val<br>";
}
```

8) Function

```
<?php
function bighello()
{
print "<h1>HELLO!</h1>";
 }
bighello();
?>
```

a) Function With Arguments

```
<?php
function printBR( $txt )
{
print ("$txt<br>");
}
printBR("This is a line");
printBR("This is a new line");
```

```
printBR("This is yet another line");
?>
```

```
<?php
function sayHello()
{
print "hello<br>";
}
$function_holder = "sayHello";
$function_holder();
?>
```

9) Arrays

```
$users = array ("Bert", "Sharon", "Betty", "Harry" );
print "$users[2]";

$users[0] = " Bert";
$users[200] = "Sharon";
```

a) Associative Array

- An associative array is indexed with strings between the square brackets rather than numbers.

Eg: Address

```
$address = array (
name=>"bob",
hname=>"Bob Vila",
place=>"USA",
pin=> 100567
);
Print $character[age];
```

Looping Through an Array

Syntax

```
foreach( $array as $temp )
{
//...
}
```

```
$users = array ("Bert", "Sharon", "Betty", "Harry" );
foreach ( $users as $val )
{
print "$val<br>";
}
```

Looping Through an Associative Array

```
$address = array (
name=>"bob",
hname=>"Bob Vila",
place=>"USA",
pin=> 100567
);
foreach ( $character as $key=>$val )
{
print "$key = $val<br>";
}
```

b) Getting the Size of an Array

```
$users = array ("Bert", "Sharon", "Betty", "Harry" );
print $users[count($users)- 1];
```

c) Manipulating Arrays

i. Merge Arrays
ii. Adding Multiple Variables to an Array
iii. Removing the First Element of an Array
iv. Slicing Arrays
v. Sorting Numerically Indexed Array
vi. Sorting an Associative Array

i. Merge Arrays

```
$first = array("a", "b", "c");
$second = array(1,2,3);
$third = array_merge( $first, $second );
foreach ( $third as $val )
{
print "$val<BR>";
}
```

ii. Adding Multiple Variables to an Array

```
$first = array("a", "b", "c");
$total = array_push( $first, 1, 2, 3,"kl",2 );
print "There are $total elements";
foreach ( $first as $val )
{
print "$val<BR>";
}
```

iii. Removing the First Element of an Array

```
$an = array("a", "b", "c");
$val = array_shift( $an);
foreach ($an as $va)
{
print "$va <br>";
```

```
}
```

iv. Slicing Arrays

```
$first = array("a", "b", "c", "d", "e", "f");
$second = array_slice($first, 2, 3);
foreach ( $second as $var )
{
print "$var<br>";
}
```

v. Sorting Numerically Indexed Array

```
$an_array = array(5,9,2,1,6,7);
sort( $an_array);
foreach ( $an_array as $var )
{
print "$var<BR>";
}
```

vi. Sorting an Associative Array asort

```
$first = array("first"=>5,"second"=>2,"third"=>1);
asort( $first );
foreach ( $first as $key => $val )
{
print "$key = $val<BR>";
}
```

vii. Sorting an Associative Array ksort

```
$first = array("x"=>5,"a"=>2,"f"=>1);
ksort( $first );
foreach ( $first as $key => $val )
```

```
{
print "$key = $val<BR>";
}
```

9. Describe Form Handling

- For acquiring and working with user input.
- On the World Wide Web, HTML forms are the principal means by which substantial amounts of information can pass from the user to the server.
- PHP is designed to acquire and work with information submitted via HTML forms.

a) Global and Environment variables

b) A Script to Acquire User Input-GET and POST Method

c) Redirecting the User

a) Global and Environment Variables

i. Global

- $GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).
- PHP stores all global variables in an array called $GLOBALS[index].
- The index holds the name of the variable.

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

<?php
$x = 75;
$y = 25;

function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>

</body>
</html>
```

```
100
```

ii. Environment Variables

Environment Variables	How To use	Contains	Output
'REMOTE_ADDR'	$_SERVER['REMOTE_ADDR']	The IP address of the client	158.152.55.35
HTTP_USER_AGENT	$_SERVER['HTTP_USER_AGENT']	The name and version of the client	Mozilla/4.6 (X11;I;Linux2.2.6-15apmac ppc)
REQUEST_METHOD	$_SERVER['REQUEST_METHOD']	Whether the request was GET or POST	POST
REQUEST_URI	$_SERVER['REQUEST_URI']	The full address	Project/index.php

b) A Script to Acquire User Input-GET and POST Method

i. POST Method

- The POST method transfers information via HTTP headers.
- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol.
- By using Secure HTTP you can make sure that your information is secure.
- The PHP provides $_POST associative array to access all the sent information using POST method.

ii. A Script to Acquire User Input-POST Method

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

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Welcome USER
Your email address is: user@gmail.com

```
<?php
print "Welcome";
Print $_POST("name");
print $_POST("name");
print "<br>";
$em=$_POST("email");
print "Your email address is".$em;
?>
```

iii. GET Method

- The GET method sends the encoded user information appended to the page request.
- The page and the encoded information are separated by the ? character.
- The GET method produces a long string
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.

- The PHP provides $_GET associative array to access all the sent information using GET method.

iv. A Script to Acquire User Input-GET Method

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

<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Welcome user
Your email address is: user@gmail.com

```
<?php
print "Welcome";
print $_GET("name");
print "<br>";
$em=$_GET("email");
print "Your email address is".$em;
?>
```

c) Redirecting the User

```
header( "Location: http://www.google.com" );
```

Example-Login

```
<?php
if(isset($_GET['log'])){
$u=$_GET['user'];
```

```
$p=$_GET['pass'];
if($u=="user" && $p=="pass")
{
print "login Successful";
header( "Location: http://www.google.com" );
}else{
print "login not Successful";
}
}?>
```

10. Cookie Handling in PHP

- A cookie is a small amount of data stored by the user's browser
- Compliance with a request from a server
- Each cookie consists of a name, value, and expiry date, as well as host and path Information.
- A Host can request that up to 20 cookies be stored by a user's browser.
- Each cookie consists of a name, value, and expiry date, as well as host and path information.
- An individual cookie is limited to 4KB.
- A cookie is often used to identify a user.
- After a cookie is set, only the originating host can read the data
- Ensuring that the user's privacy is respected.
- Cookies can be an excellent way of saving small amounts of information about a user from page to page or even from visit to visit.
- Cookies are usually set in an HTTP header

a) Setting a Cookie with PHP

- You can set a cookie in a PHP script using setcookie function.

i. setcookie()

Syntax:

```
setcookie(name,value,expire,path,domain,secure)
```

```
<!DOCTYPE html>
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 =
1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
```

```
Cookie 'user' is set!
Value is: John Doe
```

```php
<!DOCTYPE html>
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
    echo "Cookies are enabled.";
} else {
    echo "Cookies are disabled.";
}
?>

</body>
</html>
```

Cookies are enabled.

Parameter	Description
name	Specifies the name of the cookie
value	Specifies the value of the cookie
Expire	Specifies when the cookie expires. If this parameter is omitted or set to 0, the cookie will expire at the end when the browser closes.
Path	Specifies the server path of the cookie. If set to "/", the cookie will be available within the entire domain.
Domain	Specifies the domain name of the cookie. set domain to "example.com"
Secure	Specifies whether or not the cookie should only be transmitted over a secure HTTPS connection

11. Session & Cookie Handling in PHP

- A session is a way to store information (in variable) to be used multiple pages
- Unlike cookie, the information is not stored on the users computer
- Computer knows who you are, but internet not because HTTP is stateless protocol.

- Session variables solve this problem by storing information to be used across multiple pages

```php
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

Session variables are set.

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

Favorite color is green.
Favorite animal is cat.

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();

echo "All session variables are now
removed, and the session is destroyed."
?>

</body>
</html>
```

All session variables are now removed, and the session is destroyed.

12. File Operations in PHP

h) Create a file

i) Write into a file

j) Read from a file

k) Close a file

l) File operation modes

m) Create and Delete a file

n) Inbuilt functions

a) Create a file

```
<?php
$fname="newfile.txt";
$f1=fopen($fname, "w") or die("Unable to open file!");
// f1 is a file pointer
print $f1;
fclose($f1);
?>
```

b) Write into a file

```
<?php
$f1=fopen($fname, "w") or die("Unable to open file!");
$txt="hello this is my page";
fwrite($f1, $txt);
fclose($f1);
?>
```

c) Read from a file

```
<?php
$f1=fopen($fname, "r") or die("Unable to open file!");
```

```
$txt=fread($f1,filesize("newfile.txt"));
print $txt;
fclose($f1);
?>
```

d) File Modes

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	Creates a new file for write only. Returns FALSE and an error if file already exists
r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	Creates a new file for read/write. Returns FALSE and an error if file already exists

e) Create and Deleting Files

Create :-

```
touch("myfile.txt");
```

Delete :-

```
unlink("myfile.txt");
```

f) Inbuild Functions – FILE

- filesize()
- file_exists()
- feof()
- fgets()
- fgetc()

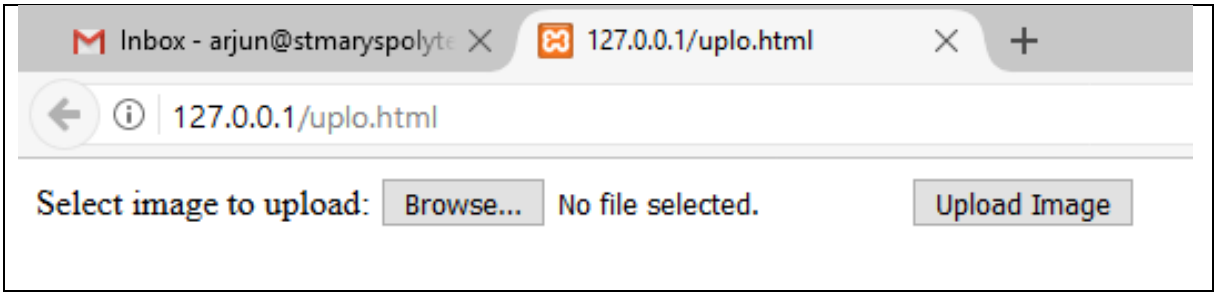
13. File Upload in PHP

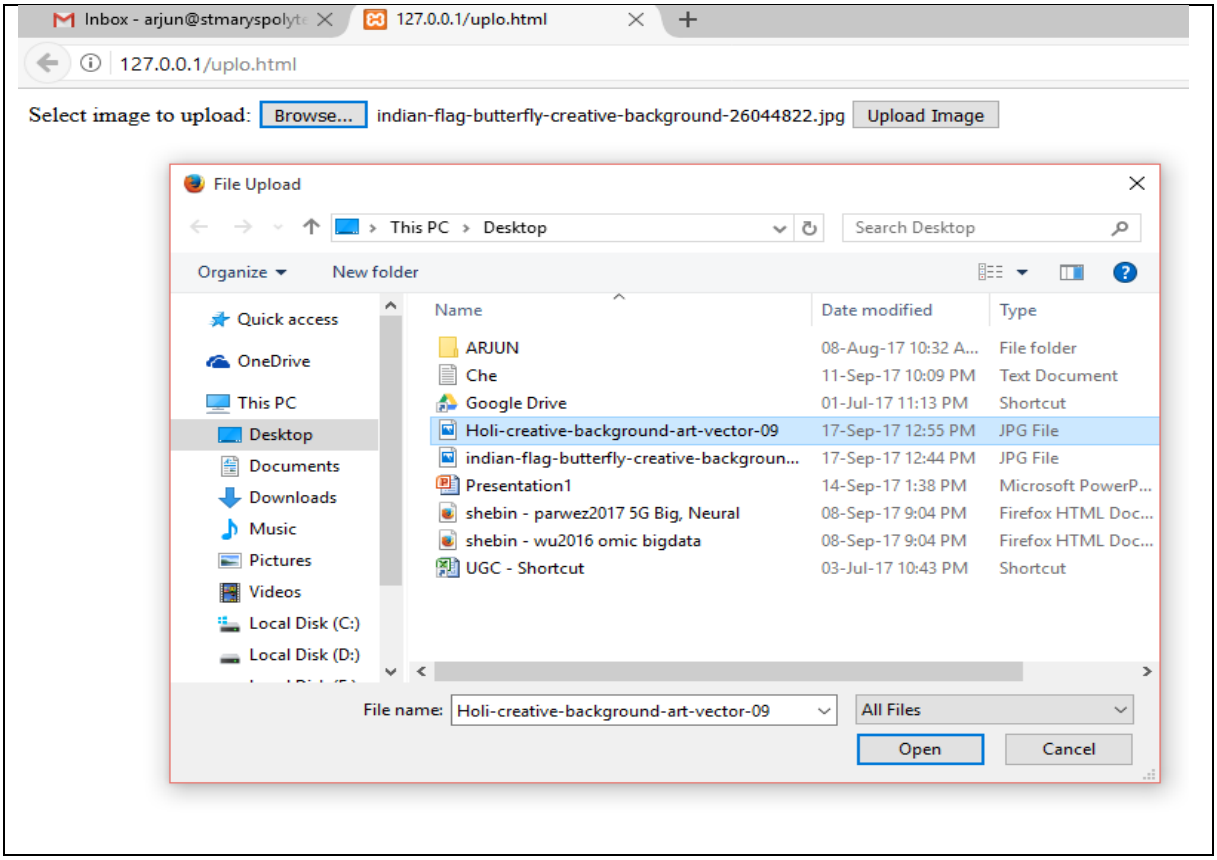
```
<form action="upload.php" method="post"
enctype="multipart/form-data">
Select image to upload:
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="Upload Image" name="submit">
</form>
```

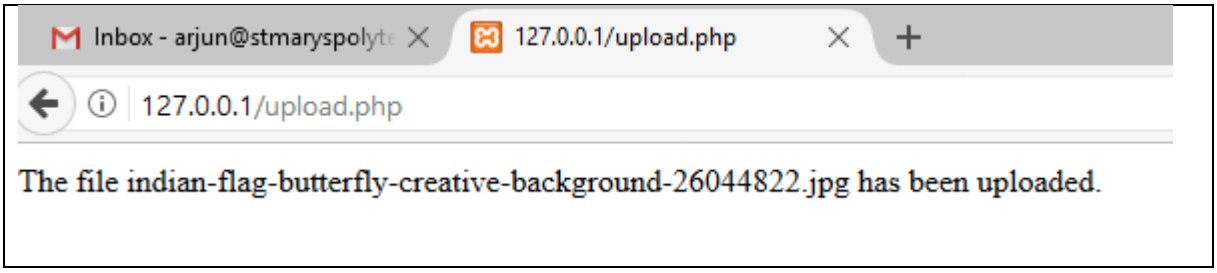
```
<?php
$target_dir = "uploads/";
$target_file = $target_dir .
basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
```

```
// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}

// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}

// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" &&
$imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are
allowed.";
    $uploadOk = 0;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
}else {
    if
(move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
$target_file)) {
        echo "The file ". basename(
$_FILES["fileToUpload"]["name"]). " has been uploaded.";
    } else {
        echo "Sorry, there was an error uploading your file.";
```

```
        }
}
?>
```


Inbox - arjun@stmaryspolyte
127.0.0.1/uplo.html
127.0.0.1/uplo.html
Select image to upload:
Browse...
No file selected.
Upload Image


Inbox - arjun@stmaryspolyte
127.0.0.1/uplo.html
127.0.0.1/uplo.html
Select image to upload:
Browse...
indian-flag-butterfly-creative-background-26044822.jpg
Upload Image
File Upload
This PC › Desktop
Search Desktop
Organize
New folder
Quick access
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Name
Date modified
Type
ARJUN 08-Aug-17 10:32 A... File folder
Che 11-Sep-17 10:09 PM Text Document
Google Drive 01-Jul-17 11:13 PM Shortcut
Holi-creative-background-art-vector-09 17-Sep-17 12:55 PM JPG File
indian-flag-butterfly-creative-backgroun... 17-Sep-17 12:44 PM JPG File
Presentation1 14-Sep-17 1:38 PM Microsoft PowerP...
shebin - parwez2017 5G Big, Neural 08-Sep-17 9:04 PM Firefox HTML Doc...
shebin - wu2016 omic bigdata 08-Sep-17 9:04 PM Firefox HTML Doc...
UGC - Shortcut 03-Jul-17 10:43 PM Shortcut
File name: Holi-creative-background-art-vector-09
All Files
Open
Cancel


Inbox - arjun@stmaryspolyte
127.0.0.1/upload.php
127.0.0.1/upload.php
The file indian-flag-butterfly-creative-background-26044822.jpg has been uploaded.