Tutorial/Laboratory 05

Developing Server-side Web Applications with PHP and NetBeans IDE

1. LEARNING OUTCOMES

Upon completion of these laboratory exercises, you should be able to:

- Build server-side web applications with PHP.
- Debug PHP pages using the Xdebug extension.
- Convert a HTML project into a PHP project.
- Organize and reuse content with PHP includes.

2. REQUIRED SOFTWARE

- Apache NetBeans 11.2 (or later): https://netbeans.apache.org/download/index.html
- FireFox (https://www.mozilla.org/en-US/firefox/new/) or Chrome (https://www.google.com/chrome/) web browser.
- PHP: https://www.php.net/downloads
- Xdebug: http://xdebug.org/docs/install

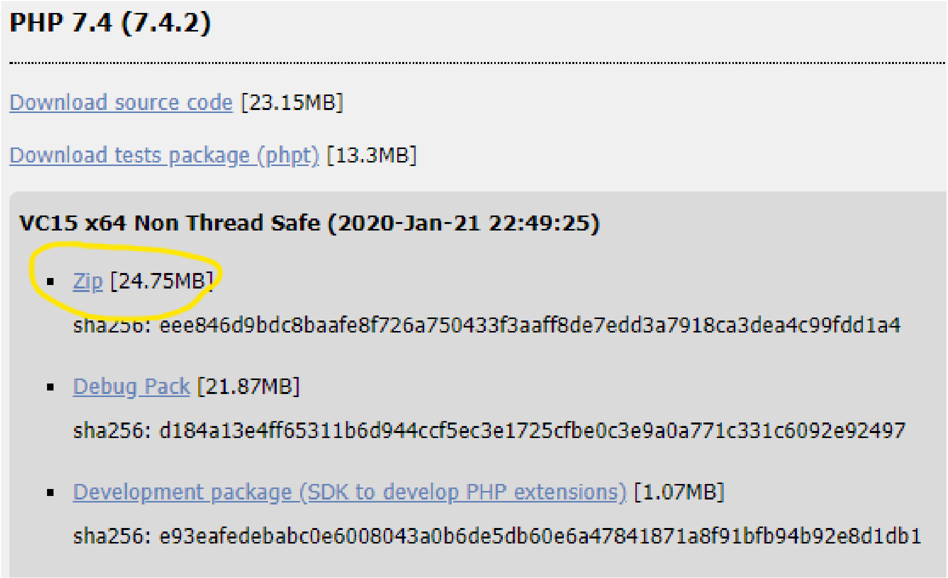
3. EXERCISE 1: INSTALLING PHP LOCALLY

3.1 Unlike pure HTML and JavaScript client-side applications, PHP files cannot be read/executed directly by a web browser. The website must be run on a web server, which can be local or remote.

In a production environment you would install PHP on a remote web server, typically a “LAMP” stack, which we’ll cover in a future Lab. But for development purposes, we can install PHP on our local machine and execute our applications using PHP’s built-in web server running on localhost.

3.2 To install PHP on Windows:

a. Download the latest version of PHP (currently 7.4.2) from the following site: https://windows.php.net/download. Since the built-in web server is single-threaded, you should select the “Non Thread Safe” version (**VC15 x64 Non Thread Safe**). Click on the **Zip** file link to download the file:



PHP 7.4 (7.4.2)

Download source code [23.15MB]

Download tests package (phpt) [13.3MB]

VC15 x64 Non Thread Safe (2020-Jan-21 22:49:25)

- Zip [24.75MB]

  sha256: eee846d9bdc8baafe8f726a750433f3aaff8de7edd3a7918ca3dea4c99fdd1a4

- Debug Pack [21.87MB]

  sha256: d184a13e4ff65311b6d944ccf5ec3e1725cfbe0c3e9a0a771c331c6092e92497

- Development package (SDK to develop PHP extensions) [1.07MB]

  sha256: e93eafedebabc0e6008043a0b6de5db60e6a47841871a8f91bfb94b92e8d1db1

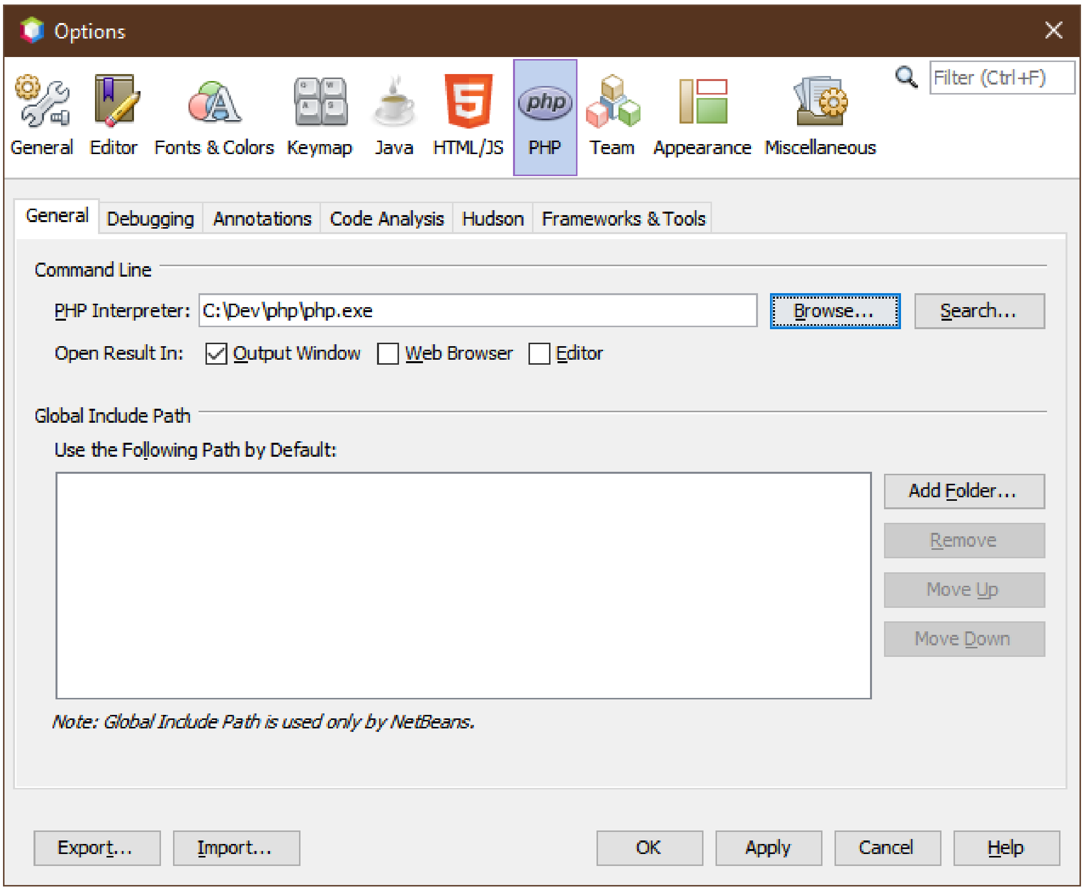
Note: the VC15 build requires the Visual C++ Redistributable for Visual Studio 2015-2019 x64, which most Windows computers should have. If yours does not, you can download and run the installer from here.

b. Unzip the file and copy the contents to an easily accessible location on your local drive, e.g. **C:\Dev\php**. You'll need to remember this location for subsequent tasks.

3.3 To install PHP on Mac OS, refer to the instructions at: https://www.php.net/manual/en/install.macosx.php. For other Linux-type operating systems, refer to https://www.php.net/downloads.

If you encounter any issues, you may refer to the PHP documentation at: https://www.php.net/manual/en/install.php.

3.4 Once PHP is installed, we need to configure NetBeans so that it can find the PHP executable files. In NetBeans, go to **Tools->Options** and select the **PHP** category. On the **General** tab, next to **PHP Interpreter**, click the **Browse…** button, navigate to the folder where you saved the PHP files earlier, and select **php.exe**:



Options

Command Line

PHP Interpreter: C:\Dev\php\php.exe [Browse...] [Search...]

Open Result In: ☑ Output Window □ Web Browser □ Editor

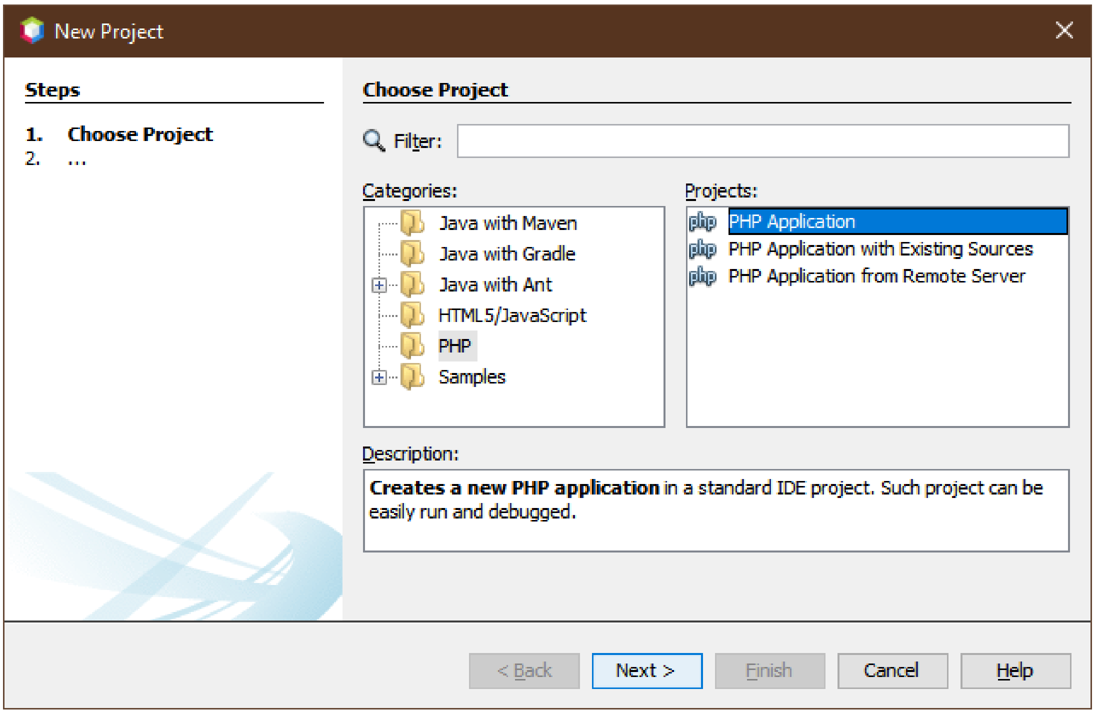
Global Include Path

Use the Following Path by Default:

Note: Global Include Path is used only by NetBeans.

Click **OK** to save the settings.

3.5 Finally, we'll create a PHP project in NetBeans to verify that everything is working. In NetBeans, choose **File->New Project** and select **PHP** under Categories, and **PHP Application** under Projects:



New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
- HTML5/JavaScript
- PHP
- Samples

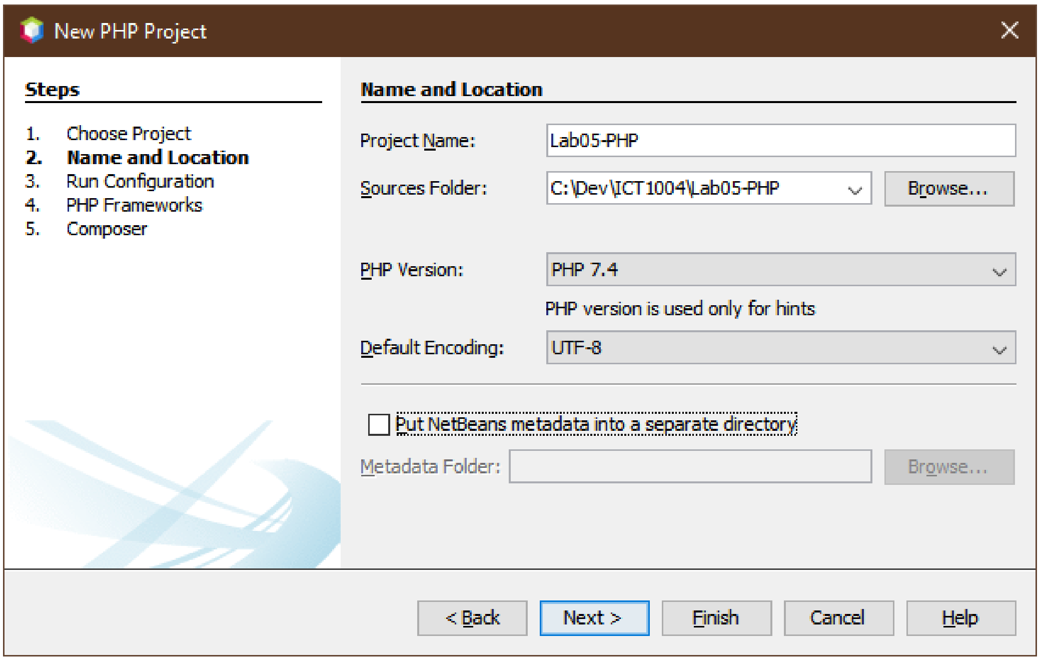
Projects:

- PHP Application
- PHP Application with Existing Sources
- PHP Application from Remote Server

Description:

Creates a new PHP application in a standard IDE project. Such project can be easily run and debugged.

Click **Next >** and enter the project name and location. Select **PHP 7.4** as the version (this only affects the hints provided in the IDE):



New PHP Project

Steps

1. Choose Project
2. **Name and Location**
3. Run Configuration
4. PHP Frameworks
5. Composer

Name and Location

Project Name: Lab05-PHP

Sources Folder: C:\Dev\ICT1004\Lab05-PHP

PHP Version: PHP 7.4

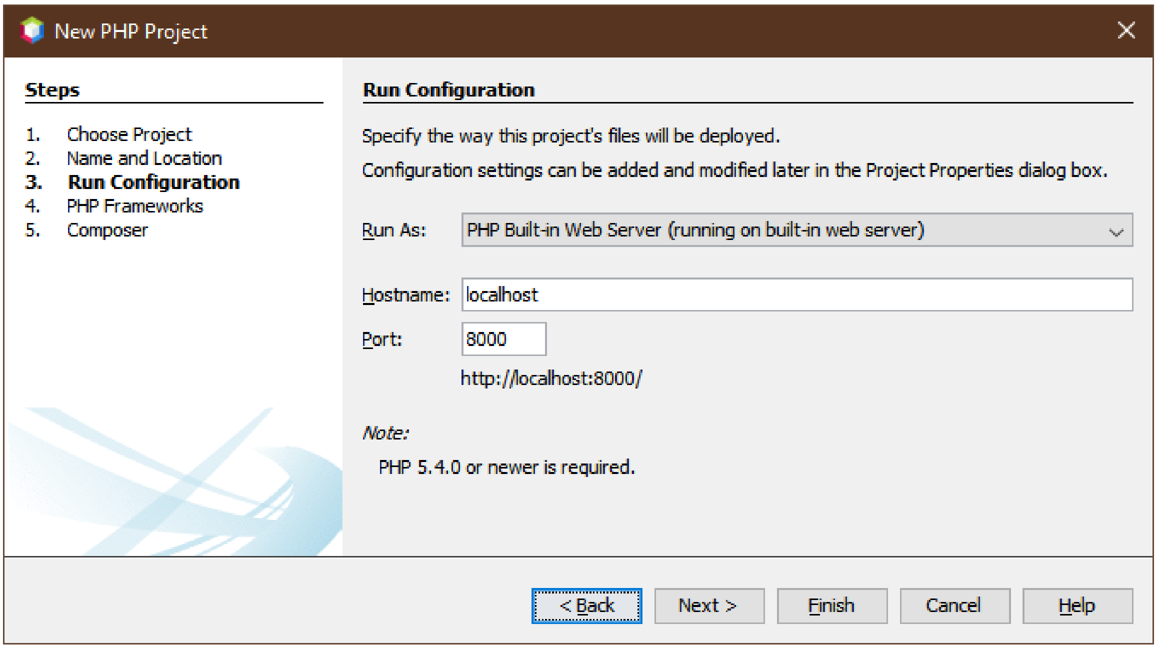
PHP version is used only for hints

Default Encoding: UTF-8

□ Put NetBeans metadata into a separate directory

Metadata Folder:

Click **Next >** and after Run As: select **PHP Built-in Web Server**:



New PHP Project

Steps

1. Choose Project
2. Name and Location
3. **Run Configuration**
4. PHP Frameworks
5. Composer

Run Configuration

Specify the way this project's files will be deployed.

Configuration settings can be added and modified later in the Project Properties dialog box.

Run As: PHP Built-in Web Server (running on built-in web server)

Hostname: localhost

Port: 8000

http://localhost:8000/

Note:

PHP 5.4.0 or newer is required.

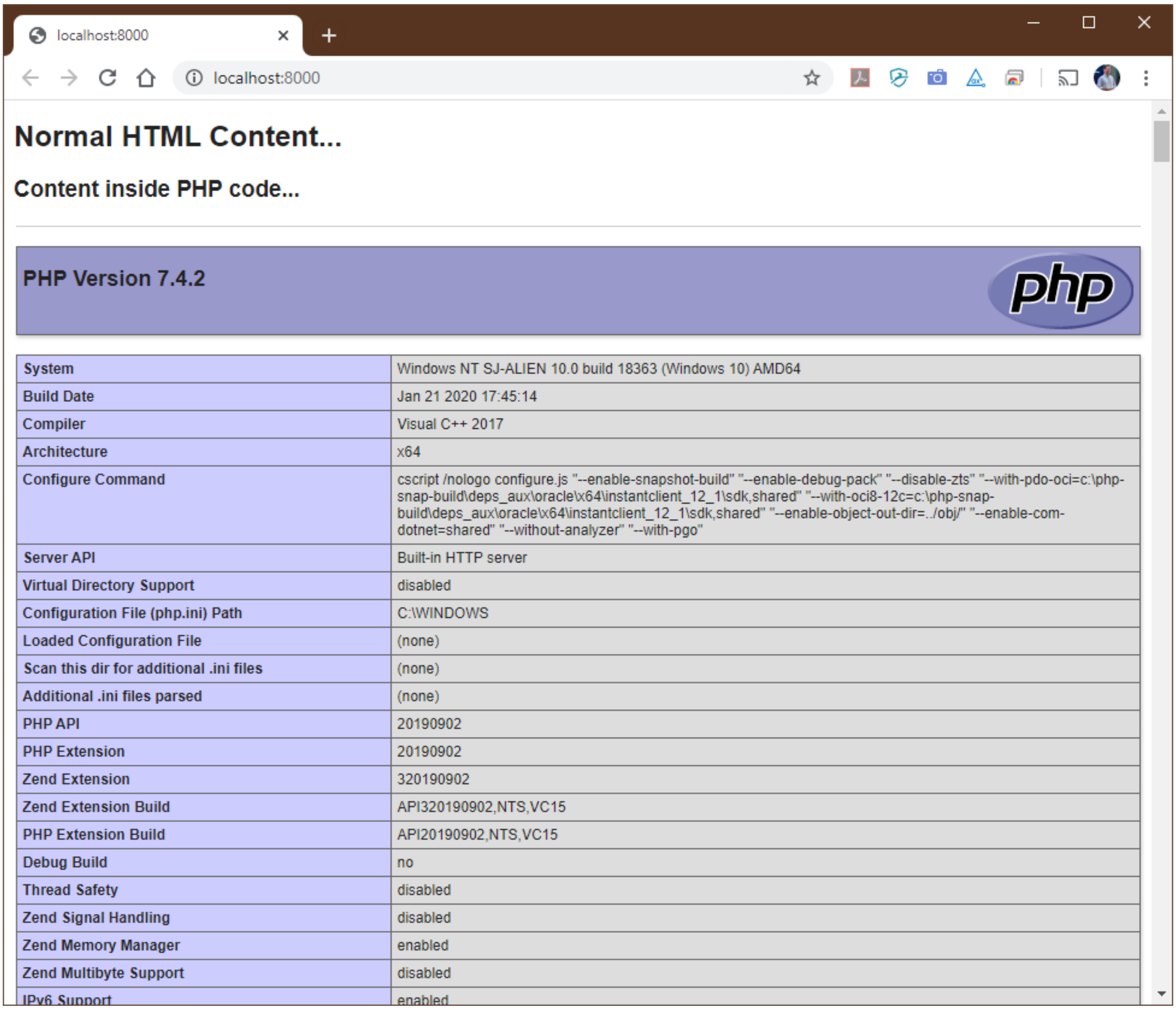
Click **Finish** to create the project (we are not using PHP Frameworks or Composer).

3.6 NetBeans will create the project with a single **index.php** file. This works like index.html in that it's the default file that the server will load if no explicit file is requested. Note that the structure is very similar to HTML, with the addition of the special `<?php ?>` element. Anything outside of the PHP tags will be treated as normal HTML, while any code written within the tags is executed by the PHP interpreter.

3.7 Replace the contents of **index.php** with the following code:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h1>Normal HTML Content...</h1>
        <?php
            echo "<h3>Content inside PHP code...</h3>";
            echo "<hr />";
            echo phpinfo();
        ?>
    </body>
</html>
```

Note how content within the PHP code element is displayed using the `echo` PHP command, and that statements must end in a semi-colon (PHP follows C-style syntax). Now click on the green Run Project button to run the website. You should see the content and PHP info displayed in the browser:



localhost:8000

Normal HTML Content...

Content inside PHP code...

PHP Version 7.4.2

System	Windows NT SJ-ALIEN 10.0 build 18363 (Windows 10) AMD64
Build Date	Jan 21 2020 17:45:14
Compiler	Visual C++ 2017
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Built-in HTTP server
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS,VC15
PHP Extension Build	API20190902,NTS,VC15
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

And that's it! You're now up and running with PHP development. In the next exercise, you'll learn more about the PHP language and how to debug PHP applications.

Discussion Questions:

a. What are the advantages and disadvantages of server-side development compared with client-side development?

b. What are some other popular server-side development tools and how do they compare with PHP?

4. EXERCISE 2: DEBUGGING PHP WITH XDEBUG

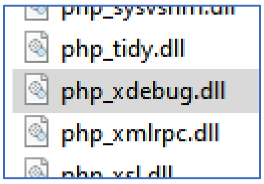
4.1 As we learned with JavaScript previously, knowing how to use debugging tools to find errors in your PHP code will save you lots of time and headaches, especially if you're just learning the language. In this exercise, you will learn how to use the Xdebug extension for PHP.

4.2 Follow these steps to install and configure Xdebug on Windows:

a. First, close NetBeans to ensure the built-in PHP web server is not running.

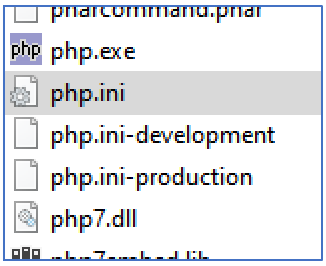
b. Go to http://xdebug.org/download and download the version of Xdebug matching your PHP installation. In our case, this is PHP 7.4 VC15 (64 bit).

c. Copy the .dll file to the **ext** subfolder under your PHP installation, e.g. C:\dev\php\ext. Then rename the file to "**php_xdebug.dll**" so that it's easier to reference.



php_sysvshm.dll
php_tidy.dll
php_xdebug.dll
php_xmlrpc.dll
php_xsl.dll

d. Navigate to the main PHP installation folder and look for a file named "**php.ini-development**". Make a copy of this file and rename the copy to "**php.ini**".



pharcommand.phar
php.exe
php.ini
php.ini-development
php.ini-production
php7.dll
php7embed.lib

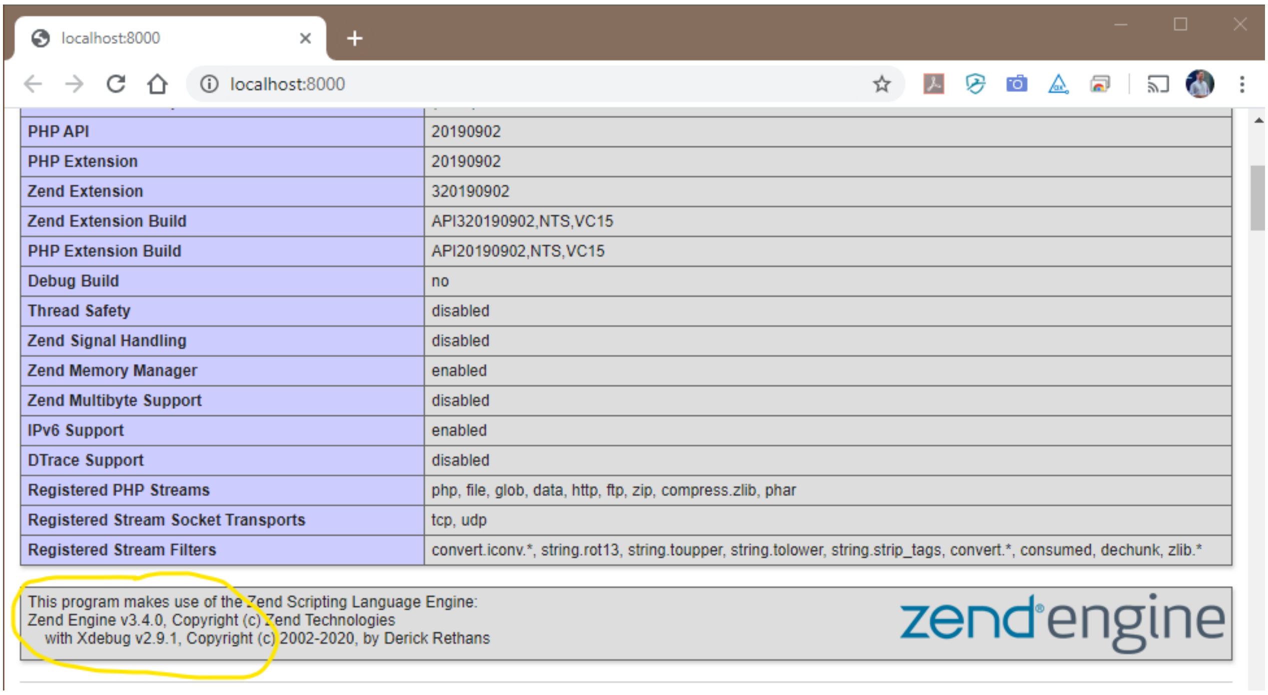
e. Open **php.ini** in Notepad or your desired text editor and add the following at the end of the file (if you installed PHP to a different location, change the filepath accordingly):

```
[Xdebug]
zend_extension="C:\Dev\php\ext\php_xdebug.dll"
xdebug.remote_enable=on
xdebug.remote_handler=dbgp
xdebug.remote_host=localhost
xdebug.remote_port=9000
```

f. Be sure to save and close the file.

4.3 For Mac OS and others, refer to: http://xdebug.org/docs/install, then follow the above steps to edit php.ini.

4.4 Now restart NetBeans and run the PHP project created in Exercise 1. Verify that Xdebug was installed properly by scrolling down and looking to the left of the Zend Engine logo - you should see Xdebug version information:



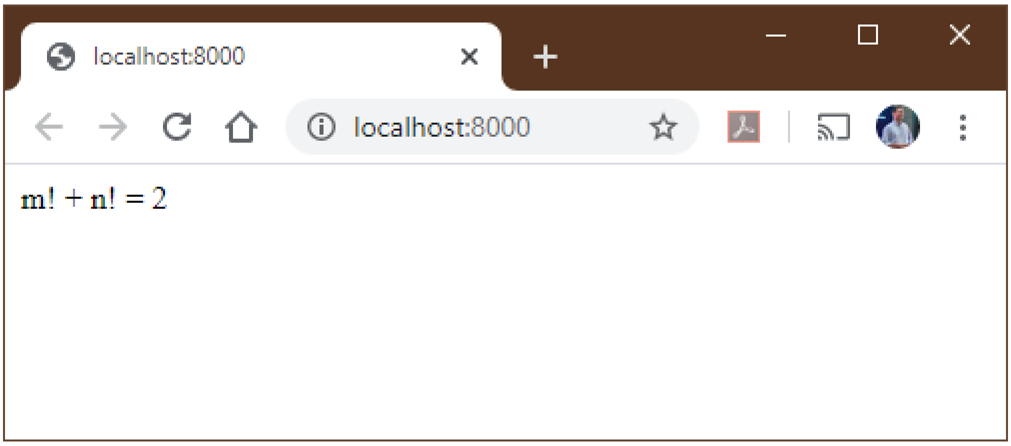
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS,VC15
PHP Extension Build	API20190902,NTS,VC15
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, phar
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Xdebug v2.9.1, Copyright (c) 2002-2020, by Derick Rethans

4.5 Now create a new PHP project (e.g. Lab05-Debugging) in NetBeans and add the following code to index.php, after the opening <body> tag:

```php
<?php
    $m = 5;
    $n = 10;
    $sum_of_factorials = calculate_sum_of_factorials($m, $n);
    echo "m! + n! = " . $sum_of_factorials;

    function calculate_sum_of_factorials($argument1, $argument2)
    {
        $factorial1 = calculate_factorial($argument1);
        $factorial2 = calculate_factorial($argument2);
        $result = calculate_sum($factorial1, $factorial2);
        return $result;
    }

    function calculate_factorial($argument)
    {
        $factorial_result = 1;
        for ($i = 1; $i < $argument; $i++)
        {
            $factorial_result = $factorial_result;
        }
        return $factorial_result;
    }

    function calculate_sum($argument1, $argument2)
    {
        return $argument1 + $argument2;
    }
?>
```

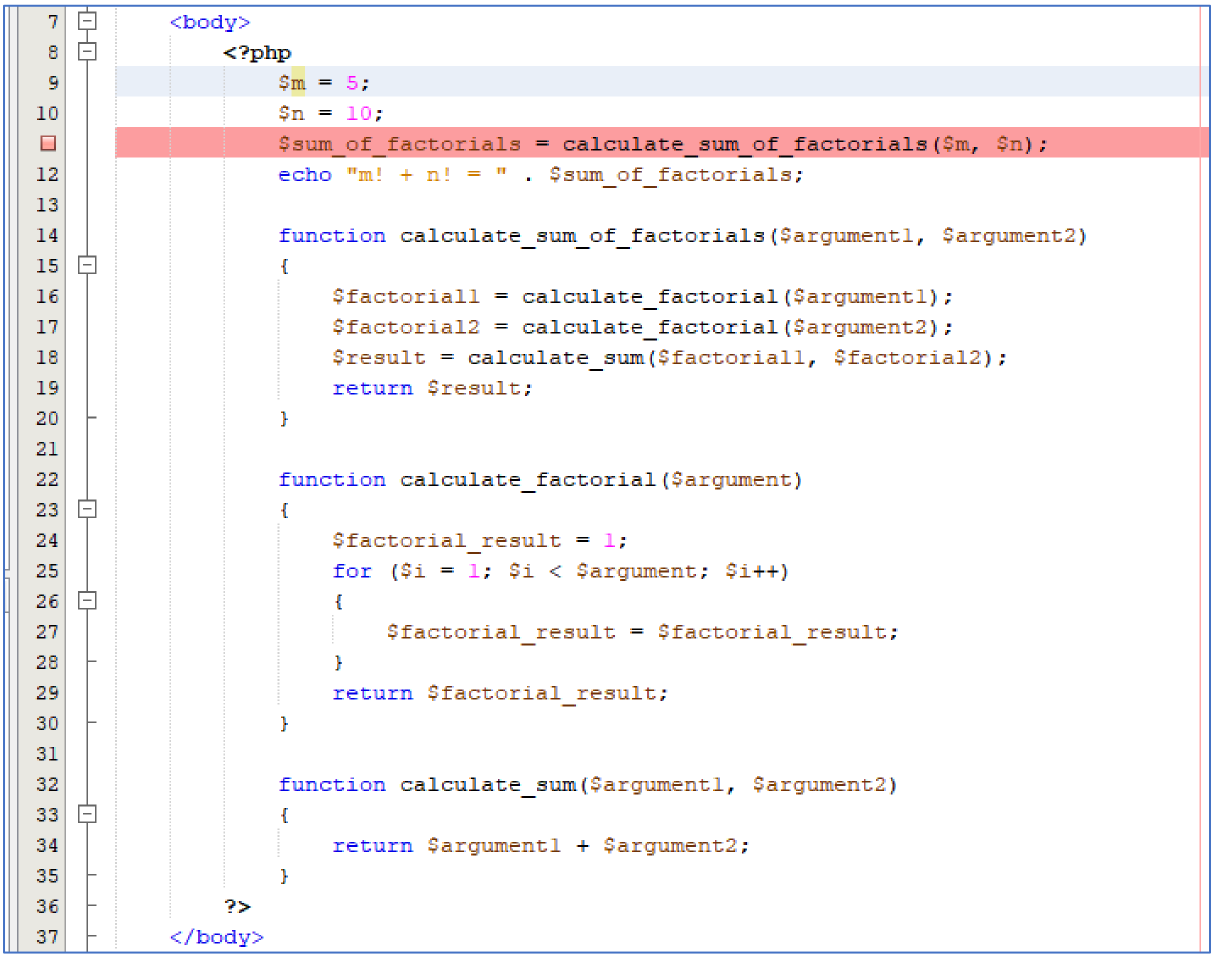
Run the application and view the web page in the browser - you should see the following rather suspicious output:



m! + n! = 2

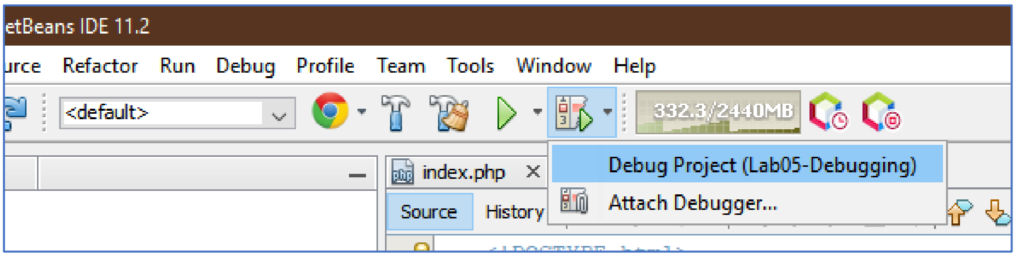
Clearly 5! + 10! does not equal 2, so we have errors. Let's use the debugger to find and fix them.

4.6 Start by setting one or more breakpoints in the PHP code. To set a breakpoint, click the line number corresponding to the PHP code that you want PHP execution to stop and for Xdebug to take over. The line should be highlighted and a square bullet will appear, indicating a breakpoint as shown below (to remove a breakpoint, click it again):



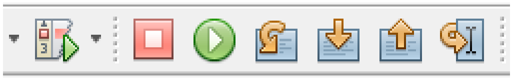
```
<body>
    <?php
        $m = 5;
        $n = 10;
        $sum_of_factorials = calculate_sum_of_factorials($m, $n);
        echo "m! + n! = " . $sum_of_factorials;

        function calculate_sum_of_factorials($argument1, $argument2)
        {
            $factorial1 = calculate_factorial($argument1);
            $factorial2 = calculate_factorial($argument2);
            $result = calculate_sum($factorial1, $factorial2);
            return $result;
        }

        function calculate_factorial($argument)
        {
            $factorial_result = 1;
            for ($i = 1; $i < $argument; $i++)
            {
                $factorial_result = $factorial_result;
            }
            return $factorial_result;
        }

        function calculate_sum($argument1, $argument2)
        {
            return $argument1 + $argument2;
        }
    ?>
</body>
```

4.7 Start debugging by selecting **Debug->Debug Project** from the menu or by clicking on the Debug icon in your NetBeans IDE toolbar:

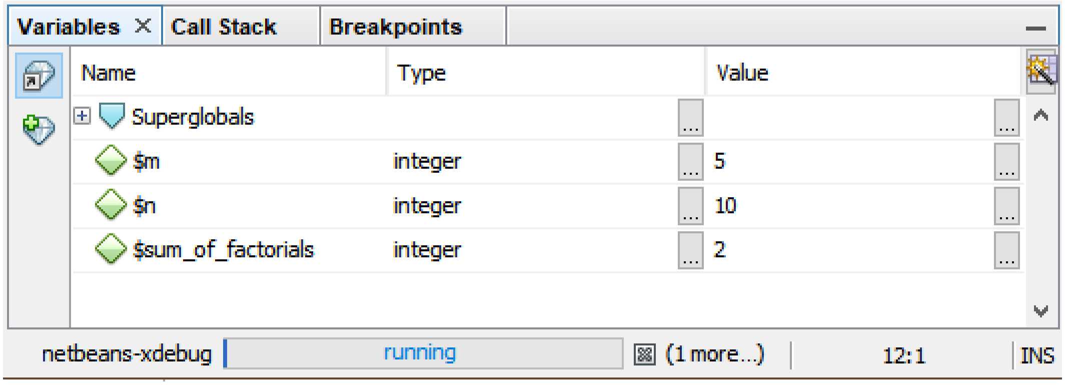


4.8 Once you have activated the debugger, additional icons for you to control the debugging processes will appear to the right of the Debug icon:

- Finish Debugger Session
- Continue
- Step Over
- Step Into
- Step Out
- Run to Cursor



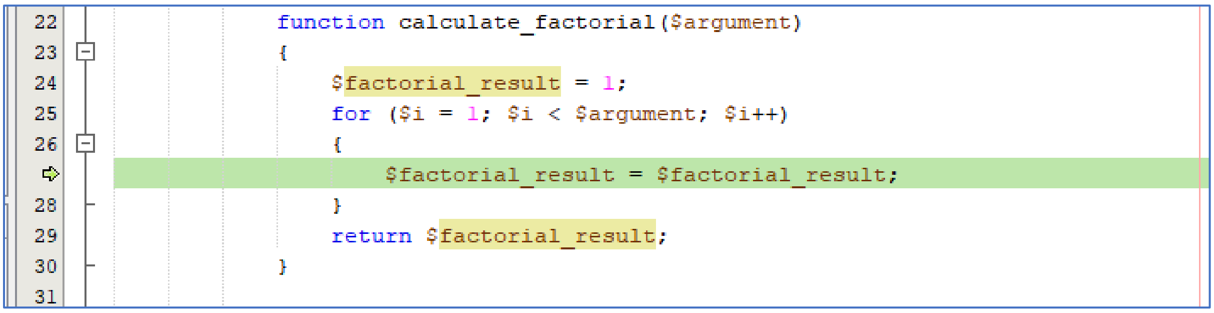
Additionally, a Debugger window will open at the bottom of your NetBeans IDE that allows you to inspect variable values as you step through the code (if your version of NetBeans does not display the Debugger window, click on **Window->Debugging->Variables** to monitor your variable values.):



Name	Type	Value
⊞ Superglobals		
$m	integer	5
$n	integer	10
$sum_of_factorials	integer	2

4.9 Use the **Step Into** operation to execute the code line by line, observing how the variables change. Notice that when you get to the `calculate_sum_of_factorials()` function, the green line (which indicates the next line to be executed) jumps down to that function. If you had used the **Step Over** operation, the function would execute and return instead.

4.10 Continue using Step Into until you reach the `calculate_factorial()` function. Step through the execution of the `for` loop and observe how the values of the variables change through each iteration. You should be able to immediately identify one of the problems.

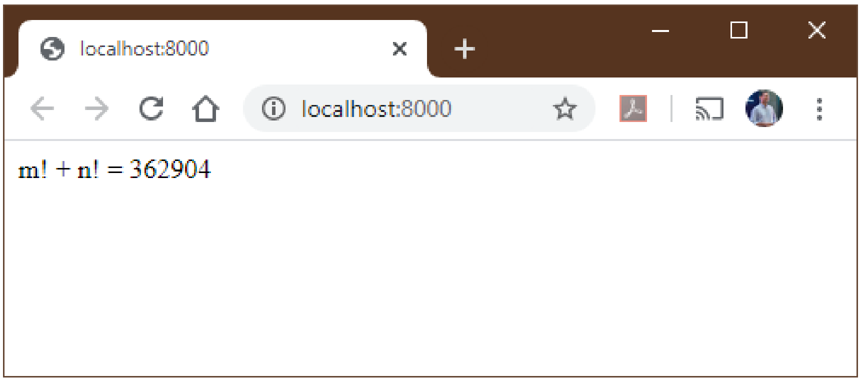


```
function calculate_factorial($argument)
{
    $factorial_result = 1;
    for ($i = 1; $i < $argument; $i++)
    {
        $factorial_result = $factorial_result;
    }
    return $factorial_result;
}
```

The variable `$factorial_result` is always 1, as it's being assigned to itself!

Stop the debugger by clicking the **Finish Debugger Session** icon, then fix that line of code as follows: `$factorial_result = $factorial_result * $i;`

4.11 Save the file and run the application again to observe the results after fixing this bug:



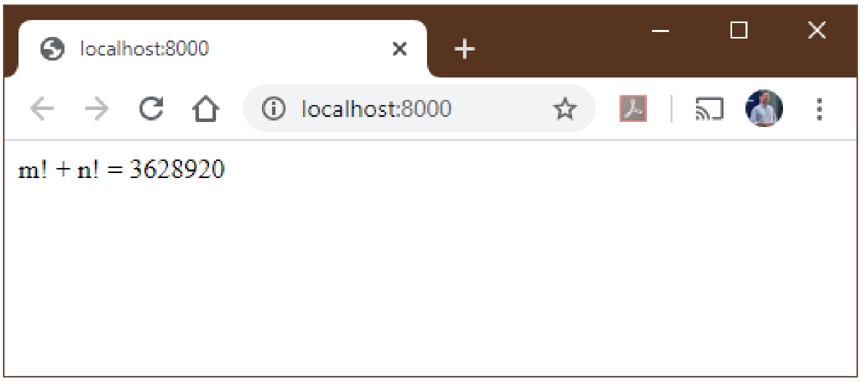
```
m! + n! = 362904
```

Looks more reasonable this time. But let's check the result using our calculator:

5! = 120
10! = 3,628,800
5! + 10! = **3,628,920**

Something still isn't right. Set a new breakpoint at the start of the `for` loop in the `calculate_factorial()` function (you can remove the breakpoint we set earlier). Run the project in the debugger and step through the loop once again, observing the variable values as they change. Can you find the second bug? **Hint:** this is what's known as a "off-by-one error" and is one of the most common programming mistakes.

After fixing this bug, your output should match with the calculator result:



```
m! + n! = 3628920
```

4.12 You're now ready to go on to more advanced PHP development. Always remember that the debugger is one of the most powerful tools in a developer's toolbox – use it often, it will save you many headaches!

5. EXERCISE 3: CONVERTING AN HTML PROJECT TO PHP

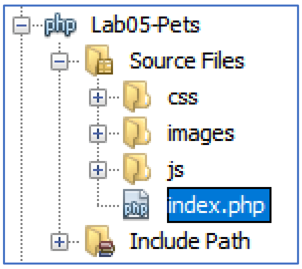
5.1 In this exercise we will "PHP'ify" the World of Pets project that we've been working on in prior labs - that is, we'll convert it from a HTML client-side application to a PHP powered server-side application.

5.2 Create a new PHP project (e.g. Lab05-Pets), then follow the steps below to migrate your World of Pets website into your new PHP application:

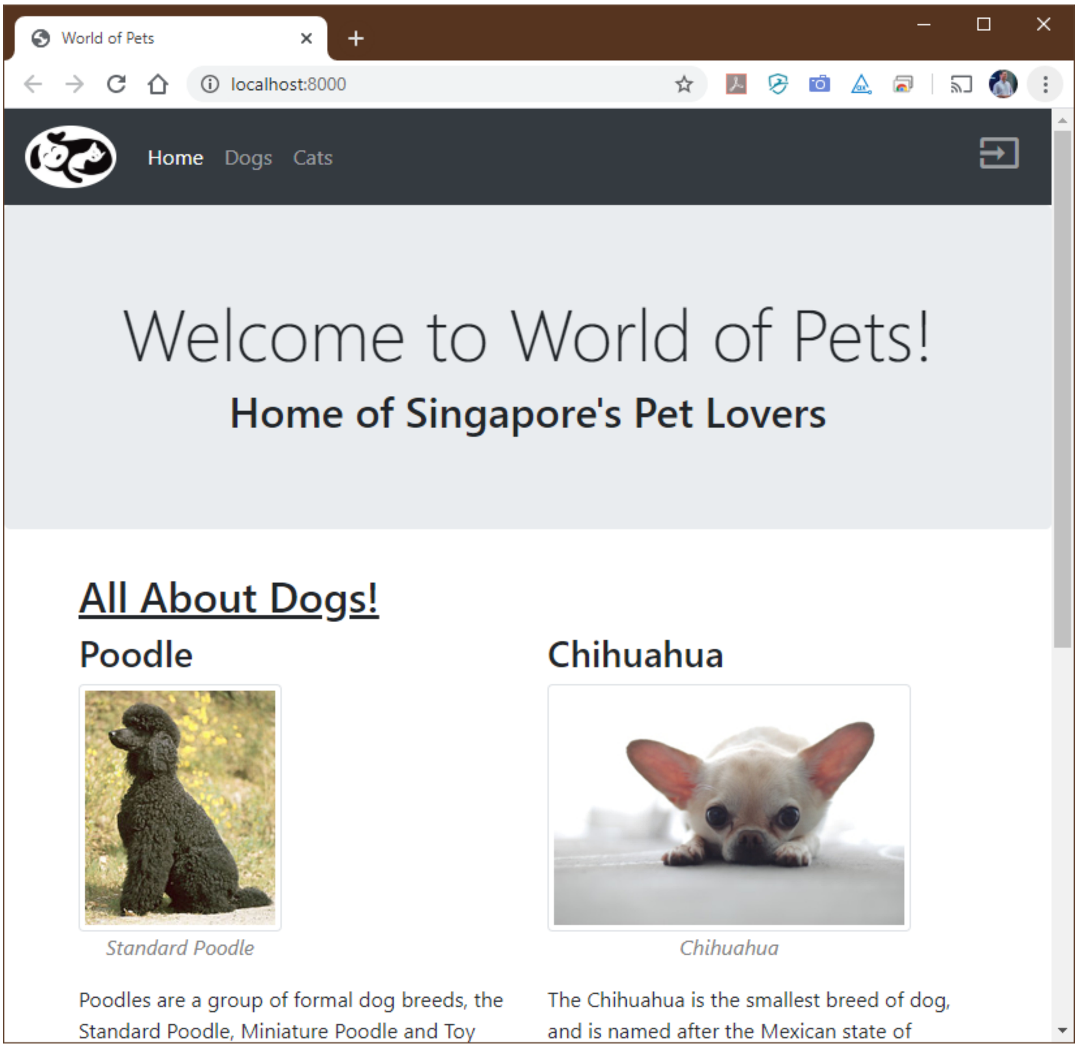
a. Open index.php and delete all contents.

b. Open index.html from the latest version of your World of Pets website and copy & paste the entire contents into index.php.

c. Copy the corresponding folders containing the resources (**css, images, js**) from the 'Site Root' folder in the old project into the 'Source Files' folder in the new PHP project. Be sure you have them in the same location relative to index.php:



d. Run your PHP application and verify that it looks and behaves the same as before:



Welcome to World of Pets!

Home of Singapore's Pet Lovers

All About Dogs!

Poodle

Standard Poodle

Poodles are a group of formal dog breeds, the Standard Poodle, Miniature Poodle and Toy

Chihuahua

Chihuahua

The Chihuahua is the smallest breed of dog, and is named after the Mexican state of

Note: you may be tempted to simply rename your existing .html files to .php, rather than migrate their content into the new PHP project. However, this won't work, as NetBeans treats HTML and PHP projects differently - there is no way to run an existing HTML project as a PHP application, even if you rename the files. You can, however, have a mixture of .html and .php within the same project, *provided you created it as a PHP project*.

6. EXERCISE 4: REUSING CONTENT WITH PHP INCLUDES

6.1 Now that our World of Pets website is PHP'ified, we can take advantage of PHP's page organization features to make it easier to extend and reuse common parts. In the following steps, we'll extract the common elements (the parts that we want to appear on every page in our website) and put them into separate PHP files so that they can easily be reused when we add more pages later. This also makes the site easier to maintain, since you can change common features, such as the navigation menu, in one place.

a. Add a new PHP file to the project (right-click on the project and select **New->PHP File…**) and name it **nav.inc.php**. Delete the default content from nav.inc.php, then move (cut & paste) all of the navigation related content from index.php into nav.inc.php. If you used the <nav> semantic tag previously, then it should be easy to select the entire <nav> element and move it to nav.inc.php.

   Note: you'll also want to replace "index.html" with "index.php" in any navigation links.

b. Similarly, add another PHP file to the project and name it **footer.inc.php**. Then move (cut & paste) all of the footer related content from index.php into footer.inc.php. Once again, if you used the <footer> semantic tag previously, then it should be easy to select the entire <footer> element and move it to footer.inc.php.

c. If you run your website now, you'll see that the menu and footer are gone, so let's use the PHP include function to add them back. Add the following PHP code to index.php at the location where you want your navigation menu to appear:

```
<?php
    include "nav.inc.php";
?>
```

d. Do the same for the footer (replacing 'nav.inc.php' with 'footer.inc.php'), then run your web application to make sure the menu and footer are appearing as before.

6.2 In our next Lab, we'll continue to enhance our website by adding forms and server-side processing. Meanwhile, do check out the Additional Practice section below to learn more about PHP.

7. SUBMISSION OF LAB ASSIGNMENT

7.1 In order to receive credit for this Lab assignment, you must submit your completed work to xSiTe LMS before the end of the Lab session. To submit your work:

a. Save all files and close NetBeans.

b. In File Explorer, navigate to the location where you saved your project and right-click on the folder name, then select 'Send to -> Compressed (zipped) folder' and ZIP up your entire NetBeans project(s). **Note: only .zip format is acceptable, do not use .rar, .7z, or any other format.**

c. In the ICT1004 module on xSiTe, go to **Assessments->DropBox** and locate the Dropbox folder corresponding to this Lab. Click the link to open the Dropbox then hit the **Add a File** button to submit your .zip file. You may also add comments if desired. Be sure to hit **Submit** to complete your submission.

d. Remember to save a copy of your work as we will be building upon this website in subsequent Lab assignments.

8. ADDITIONAL PRACTICE

8.1 Once you've completed this Lab assignment, you are encouraged to try out the following online tutorials:

a. PHP: https://www.w3schools.com/php/default.asp