SENG1050 - Laboratory 6

The objectives of this lab are:

1. learn how to configure the Apache web server and create custom behaviour using PHP, that was covered in Week 6 lectures.
   - Apache
   - Basic PHP
   - PHP with form data
   - Advanced PHP

Task 1 - Task 2 - Task 3 - Task 4

Task 1 - Configure Apache

In this task, you are going to learn how to configure the Apache web server to host a website and serve content securely. You do not need to be familiar with every possible configuration setting in Apache, the main aim of this task is for you to become familiar with how to host a website, and how to adjust commonly used settings. In case you are stuck or are curious about the reasoning behind these settings, you may refer to the supplementary video on Canvas (under the manual).

1. Start your virtual machine from Azure Labs (use your regular student email to login) and wait until the VM has started.
   If you cannot see the SENG1050 VM in the list, check your email for an invite link and register.
2. Click the small computer icon to download a connection link (SENG1050.rdp). For Windows users, double-click on the rdp file and login to the VM using the password 'WebTech.2024' (without the quotes). Ignore any certificate error message and click 'Yes' to proceed.
   For macOS users, you have to download and install Microsoft Remote Desktop as your remote desktop client before double-clicking the rdp.
   For Linux users, you have to download and install Remmina as your remote desktop client before double-clicking the rdp. You may have to set the Username as SENG1050 (without '~\') and Domain as WORKGROUP.
3. Once you have logged into the VM, start the WampServer application, either from the start menu, or from the desktop icon.
   Wait until the icon in the system tray turns green.
4. Navigate to the default document root of WampServer, which is C:\wamp64\www. In this folder, create a new directory for your website. Name the directory anything you like (e.g., mysite).
5. Create a copy of your *template.html* file from Lab 2, and place it in the new directory. Rename the copy to 'index.html'.
6. Once WampServer has started, open a web browser and navigate to http://localhost , you should see the WampServer web interface.
7. Select "Add a Virtual Host", you will see a screen prompting you for information to create the new virtual host.
8. Give your virtual host a name (this is generally the name of your website, e.g., mysite), note that it cannot contain underscores '_' or spaces.
9. For the path, copy the absolute path to the directory you created in step 4 (e.g. C:/wamp64/www/mysite).
   You'll need to change the backslashes '\' used in Windows paths to forward slashes '/'.
10. Leave the bottom field empty, and start the creation of the virtual host.
11. Once the process has completed, restart the DNS service of the computer so that you can use the new virtual host.
    You can do this from the WampServer menu, right-click on the tray icon, then navigate to 'Tools' -> 'Restart DNS'.
12. In the web browser, navigate to your new website using the name you gave it in Step 8 (e.g., http://mysite). You should see your 'index.html' file in the browser.

Now with the virtual host up and running, we will look at some commonly used Apache configuration settings.

1. Open the main Apache configuration file in a text editor (such as Notepad++). You'll find it at: C:\wamp64\bin\apache\apacheX.X.X\conf\httpd.conf
2. First, we will remove the OS and version information Apache sends in every HTTP response.
   In the browser, open the Inspector and select the Network tab. Refresh the page and look at the headers of the response. Notice the field 'Server' that exposes the version of Apache and our OS? We want to remove as much information from this as possible.
3. In the configuration file, find the line for 'ServerSignature' and change the 'On' to 'Off'.
4. Just under this, change the line for 'ServerTokens' from 'Full' to 'Prod'.
5. Save the file and apply the new settings by right-clicking on the WampServer tray icon, and selecting 'Refresh'.
6. Refresh the page in the browser and view the response headers in the inspector again. If you have set the configuration correctly, only the name of the web server, 'Apache' should be shown in the 'Server' field.
7. Second, rename the 'index.html' file to 'index_temp.html' and refresh the page. Without finding the default document, Apache shows a listing of all files in the directory, which can expose information we want to keep secret. Let's fix this now:
8. The main configuration file references several other configuration files, one of which is for all of the virtual hosts that Apache manages. This is located in a subdirectory of WampServer called 'extra'. The full path is: 'C:\wamp64\bin\apache\apacheX.X.X\conf\extra\httpd-vhosts.conf'. Open this file in a text editor.
9. In this virtual hosts configuration file, find the VirtualHost block that contains the ServerName matching the one you created. This block contains settings specific to your Virtual Host (remember, Apache can manage multiple websites, each on a different virtual host).
10. In the configuration block for your virtual host, locate the 'Options' line, and change '+Indexes' to '-Indexes'. Save the file and refresh Apache as you did in Step 5.
11. Refresh your browser, you should now see a 403 Forbidden error message instead of a full directory listing.
12. Finally, in your browser, try and access a file that doesn't exist, such as 'http://newsite/notthere.html'. You should see the default 'Not Found' message that Apache returns with a 404 response. Let's change this:
13. In the configuration file for the virtual host, in the VirtualHost block for your site, add the following line: "ErrorDocument 404 /index_temp.html". Save and refresh Apache.
14. Try once again in your browser to access a file that doesn't exist. This time you should find that your HTML file is loaded in the browser as the default 404 page.
    You can verify this in the inspector by looking in the Network tab at the response code.

If you're interested, more information on securing and configuring Apache can be found here.

Task 2 - Basic PHP

In this task, you are going to learn how to use PHP to create and send HTML documents to the client.

1. Create a new file in your website, call it 'index.php'. Copy the following code into the file:

   ```
   <!DOCTYPE html>
   <html>
       <head>
           <meta charset="utf-8">
           <title>Hello World</title>
       </head>
       <body>
           <?php
               echo "<p>Hello World</p>";
           ?>
       </body>
   </html>
   ```

2. Now, open your website in the browser (e.g., http://newsite). As 'index.php' is one of the default documents in Apache, you should see the file load, with "Hello World" displayed on the page.
3. Next, we will edit the 'index.php' file to generate a countdown from 10 to 0 using a while loop:

4. Replace the echo statement in the PHP script with the following:

   ```
   $counter = 10;
   while ($counter >= 0) {
       echo "<p>" . $counter-- . "</p>";
   }
   ```

5. Save and run the file in your browser. You should see a countdown from 10 to 0.
6. Now, we will edit the file to display only even numbers in the countdown. In your PHP file, add the following code just before the echo statement within the 'while' loop:

   ```
   if ($counter % 2 != 0) {
       $counter--;
       continue;
   }
   ```

7. Save the file and reload the page in your browser. You should see a countdown from 10 to 0 with only the even numbers displayed.
8. Now, it's time for you to try some PHP tasks on your own. Edit your PHP file to perform the following tasks:
   - Declare the variable `$the_string = "The cat sat on the mat";` and calculate the length of this string, as well as the number of times the letter 't' appears in it.
   - Display the basic times-tables for 1x1 through to 12x12.
   - Display all the leap years from 1900 to 2022 (the leap years are the years that are either: evenly divisible by 4 AND not evenly divisible by 100; OR: evenly divisible by 4 AND 400.)

If you run into problems with your PHP code, try outputting the value of variables with echo statements, or checking for syntax errors with a PHP tester.

Task 3 - PHP with form data

PHP becomes very powerful when used with dynamic data collected from the user via a form. In this task, we will learn to use form data with PHP.

1. Create a new file in your website called 'form.html'. Copy the following code into the file:

   ```
   <!DOCTYPE html>
   <html>
       <head>
           <meta charset="utf-8">
           <title>My Form</title>
       </head>
       <body>
           <form method="POST" action="index.php">
               <label for="name">Name</label>
               <input type="text" name="fname" id="name" /><br><br>
               <input type="submit">
           </form>
       </body>
   </html>
   ```

2. Remember, the value of the 'name' attribute is what we will use to get the value of the input in our PHP script.
3. Now, edit your 'index.php' file again to get the value of the text box, and display it back to the client. (Hint: You can use the super-global array `$_POST["fname"]` to retrieve the value of the input).
4. Edit your 'index.php' file again to perform the following tasks:
   - Display the length of the form input.
   - Display the number of times the letter 'e' appears in the input.
   - Display the form input in all upper-case letters.
5. Now, in 'form.html', add an additional input: `<input type="date" name="dob"/>` which will be used to collect the user's date of birth.
6. Back in your 'index.php file, use the value of the new date input to perform the following tasks:
   - Display the date of birth in year-month-day format.
   - Calculate and display the age of the user based on difference in years. Hint: retrieve the current date using `$today = date('Y-m-d');` and calculate the difference using `date_diff(date_create($dob), date_create($today))-> y;`

- If the user is under the age of 18, output the string: "Sorry, you need to be over the age of 18 to make this purchase".

Task 4 - Advanced PHP tasks

If you have extra time at the end, try out the following tasks (these are much more challenging, you may need to ask your demonstrator for assistance):

1. Edit your PHP script to output the Fibonacci sequence from 1 up to a number entered by the user via the form (you'll need add an input of type "number" to the form).
   The Fibonacci sequence sums the previous two numbers to get the next number in the sequence, e.g: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
2. Add four required number inputs to the form, edit your PHP script to display the second-largest number entered.
3. Edit your PHP script to operate as a simple calculator, taking two number inputs from the form, and using the value of the form's text input to determine which operation to perform ('+' for addition, '-' for subtraction, '*' for multiplication, and '/' for division). Display the result.
4. Edit your PHP script to take the text input, and display the string with reversed case (e.g. 'Hi There' should be output as 'hI tHERE')
5. Edit your PHP script to take the four number inputs from Step 2, and output them in ascending order.
6. Edit your PHP to take two text inputs from the form, and output the longest common ending between both inputs (for example, given the two strings "action" and "addition", the longest common ending would be "tion").