

SENG1050 - Laboratory 7

The objectives of this lab is to learn how to configure the GetSimple Content Management System (CMS) and write templates using PHP, that was covered in the Week 7 lecture, through the following topics:

- [GetSimple CMS](#)
- [Simple Templates](#)
- [Templates with XML data](#)
- [Templates writing XML](#)

[Task 1](#) - [Task 2](#) - [Task 3](#) - [Task 4](#)

Task 1 - Configure GetSimple CMS

In this task, you are going to learn how to configure the GetSimple CMS to organise a website and serve content.

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. This task assumes that you have completed at least Task 1 of the previous lab, creating a virtual host to serve web content.
If you have not done this, please go back to the previous lab and complete Task 1.
5. Download the GetSimple CMS application from Canvas (in the Week 8 module) and copy it to your VM (e.g., Downloads directory).
Do not use the version from the official website as it does not work with PHP 8.
6. Unzip the GetSimple CMS application you downloaded from Canvas, and copy the contents into the root directory of your virtual host (e.g., C:\wamp64\www\mysite).
7. Open a web browser and start the installation of GetSimple by navigating to the Content Management Application (CMA).
To do this, type the address of the CMA ('/admin') located on your virtual host in the address bar.
For example, if your virtual host was named 'mysite', you would navigate to 'http://mysite/admin'.
8. You should see the initial setup screen, all items should be green except for the version of GetSimple at the top (you can ignore the Upgrade Check warning).
9. Click 'Continue with Setup' to go to the next screen. Enter the website name (this can be anything you like and may be different from the name of your virtual host), and a username (do not use the words 'admin', 'default', 'user', or 'root' as part of the username!). Enter an email address (this does not have to be a real email address). When finished, click 'Install Now!'.
10. Once the process has completed, you may notice some warnings. Ignore this and scroll down till you see a randomly generated password. Copy this as you will need it to login.

11. In the web browser, navigate to the CMA as you did previously (e.g., 'http://mysite/admin'). This time, you will be presented with a login screen. Use the username and password you received during the installation to login.
12. Once you log in, immediately set a new password which is memorable (e.g., Password123). Confirm the password again and click 'Save Settings'.

Now with the application up and running, we will have a look around, and become familiar with the system by performing some simple tasks.

1. First, we will learn to edit pages and insert images. Navigate to the 'Files' section of the CMA, and select 'Choose files' on the right-hand side.
Select any image to upload. Click 'Upload' to upload the image to the server (GetSimple will place it in the 'data/uploads' directory in your virtual host).
2. Once the image has been uploaded, navigate to the 'Pages' section of the CMA, and click on the default 'Welcome' page.
3. Add an image anywhere on the page by use the Insert Image button on the toolbar (the third button from the right, with a green '+' symbol in the lower-right corner). Select 'Browse Server' and choose the image you uploaded earlier. Once the image loads in the preview pane, click OK.
Debug: In case the image doesn't load in the preview pane and shows a square with cross, comment lines 31 and 32 (by adding '#' at the start of line) in the .htaccess file in data/uploads/.
4. There's a problem with inserting images this way, GetSimple uses absolute paths by default. Click on the 'Source' button on the toolbar (the right-most button) to view the HTML for the page. Find the 'img' tag and look at the 'src' attribute. We want to change this to a relative path. Remove the front of the path, such that it becomes src="data/uploads/myimage.jpg" (replace myimage.jpg with the actual name of your image).
5. Click 'Save Updates'. You can view the page in the Content Delivery Application (CDA) by navigating to your virtual host without the '/admin' part of the path.
6. Now, let's start to look at themes and templates. We will begin very simply, by learning from the ones that come with GetSimple.
7. Back in the CMA, navigate to the 'Theme' section. Change the theme to 'Innovation' from the drop-down menu and select 'Activate Theme'. Refresh the page in the CDA to see the results.
8. Back in the CMA, in the 'Theme' section, select the 'Edit Components' option on the side, and look at the default components that come with GetSimple.
Make a change to the Sidebar, such as changing a word or phrase, and click 'Save Components'. Refresh the page in the CDA to see your changes.
9. Using the file explorer, navigate to the 'theme/Innovation' directory of your virtual host. This is where the PHP theme files are stored. Open some of the PHP files and see if you can follow what is going on. Processing begins with the file 'template.php', this is the first file called in the theme.
10. You should observe that this theme has broken down the template into multiple files, each one containing a common part of the webpage, such as a header, footer, sidebar, etc.
11. Let's add the tagline component to this template. To do this, open the file 'footer.inc.php' and insert the following code directly after the line with the anchor tag containing the site name (line 21):

```
<?php get_component('tagline');?>
```

Save the file and refresh the page in the CDA to observe your changes.

12. Try removing the sidebar element entirely: in the file 'template.php' comment out the line that includes the sidebar template. Save and refresh to see your changes. Did the component disappear completely? If not, try commenting the line inside the PHP block instead of outside it.
13. Finally, try and access a page that doesn't exist (e.g., If you virtual host is called 'mysite', try accessing 'http://mysite/nothere').
Note the default 404 page from GetSimple. Where is this page coming from?
(Hint: try looking at the file 'index.php' in the document root of your virtual host to find this out). Without changing this script, how could we create our own 404 page?
The answer is in lines 63-82, we just need to create a page in the CMA and call it '404'. Look at the

code and make sure you understand why this would work.

Task 2 - Simple Templates

In this task, you are going to create your own theme, and apply it to your website.

1. Create a new custom theme directory in the 'theme' directory of your virtual host with any name you wish (e.g., MyTheme). Within this directory, create a new file called 'template.php', and copy the following code into it:

```
<?php if(!defined('IN_GS')){ die('you cannot load this page directly.')} ?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title><?php get_page_clean_title(); ?></title>
    <link href="<?php get_theme_url(); ?>/style.css" rel="stylesheet">
  </head>
  <body>
    <?php get_page_content(); ?>
    <strong>Hi!</strong>
  </body>
</html>
```

2. In the CMA, navigate to the 'Theme' section, change the theme to the one matching the name of the directory you create, and click 'Activate Theme'. Refresh the page in the CDA to see the effect.
3. Right click on the page and select 'View Page Source', copy the code into the [HTML validator](#) and see if the code is valid.
If you find any problems in the validator, you should fix them.
4. Create the file 'style.css' in the custom theme directory you created (i.e., theme/MyTheme). Set styles for a couple of tags on the homepage, save the file and refresh to see your changes applied.
5. Use some of the other template placeholder functions in GetSimple to complete the following tasks - a complete reference is available [here](#):
 - Add the date the page was last modified to the footer, after the main content.
 - Add the title of the page to a h1 element inserted before the main content.
 - Append the name of the website to the end of the 'title' element.
 - Add the pre-installed 'tagline' component at the bottom of the page, after the date.

Assessment 3 (Web-Based Assignment 2) Milestone 1: QUICK STOP HERE!

Once you have completed Tasks 1 and 2 above, you are able to complete milestone 1 of Assessment 3. Please refer to the Assessment 3 documentation and follow the steps listed carefully.

Upon completion of milestone 1, approach your lab instructor to claim your 1% score. Once you have been marked off for the milestone, continue the tasks below.

Task 3 - Templates with XML data

Combining templates with the SimpleXML class allows us to read XML data into our webpages.

1. Download the file 'airports.xml' from Canvas (in the Week 8 module). Place it in a new directory such that the relative path from the root directory of your virtual host is 'data/pages/content/airports.xml'.
2. Relative to the root directory of your virtual host, add any image you like to the 'data/uploads' directory and name it 'airport'.
3. Now create a new file in your custom theme directory and name it 'xml-template.php' Copy the following

code into the file and save it:

```
<?php if(!defined('IN_GS')){ die('you cannot load this page directly.')} ?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title><?php get_page_clean_title(); ?></title>
    <link href="<?php get_theme_url(); ?>/style.css" rel="stylesheet">
  </head>
  <body>
    <?php
      $xmlpath = GSDATAPAGESPATH."/content/".$_GET["name"].".xml";
      $root = simplexml_load_file($xmlpath);
      echo "<h1>" . $root->getName() . "</h1>";
    ?>
  </body>
</html>
```

4. In the CMA, create a new completely blank page with your preferred name (e.g., xml-test). Under 'Page Options', change the 'Page Template' from the default to the file 'xml-template.php'. Click 'Save Updates'.
5. Now, you will view the contents of 'airports.xml' according to the theme defined by 'xml-template.php' in the CDA.
In your browser, navigate to the URL 'http://mysite/xml-test?name=airports' (Note that you might need to replace 'mysite' and 'xml-test' with the names you have previously assigned to the virtual host and blank page respectively). You should see the name of the root tag in a h1 element on the page.
Before moving to the next step, try to think of the following:
 - In the URL, why did we set the value of the attribute 'name' to 'airports'?
 - How 'xml-template.php' defines the CDA output?
 - Why did we see the name of the root tag when the function 'getName()' is called?
6. Edit your 'xml-template.php' file again to retrieve the following information from 'airports.xml' (*Hint: Refer to slides 42-48, 56, 58, 60, 62, 63 and 67 of Lecture 7*):
 - The name and description of each airport should be displayed.
 - If an airport code exists, this should also be displayed.
 - If a URL element exists and is not blank, this should be inserted into the 'href' attribute of an anchor element on the page.
 - If an image element exists and is not blank, this should be inserted into the 'src' attribute of an img element on the page.

Task 4 - Writing to an XML file

If you have extra time at the end, try out the following tasks (useful for the bonus task of Assessment 3):

1. Create a new page containing a form with fields for 'name', 'description', and 'url'. Note that these fields correspond to the XML elements in 'airports.xml'. Have this form submit as a POST request to a PHP script called 'submit.php'.
2. Write the PHP script 'submit.php' which should be located in the root directory of your virtual host. Your script should check that the values for 'name', 'description' and 'url' are not empty, and add them to the file 'airports.xml' as a new 'airport' element (*Hint: Refer to slide 73 of Lecture 7*).

Additional Exercises

If you finish early, feel free to try out some of the extra tasks below:

- [PHP Exercises](#)

