Tutorial 05 Template – Version 8.3 – 40 Items **Tutorial 05 –** Build WordPress CMS on Multiple Platforms and using Best Practices, extend with images, menus, themes, pages, posts, categories, tags, and plugins.

If you have difficulties completing this assignment, please inform the instructor and consider working with DePaul tutors.

**Instructions for completing template:***Replace* **[bracketed red text]** *with the requested information.  
Do not include full screen shot images, or full screen shots that are reduced in size. Use a snippet tool to capture parts of the screen and insert those images into the template.***Please do not alter template numbering.  
Each answer and screen capture are worth equal points.**

**PART A – Getting started - Student Contact Information**

1. **Student Name:** Wesley Wu **[A01]**
2. **Student Email Information:** wwu36@depaul.edu **[A02]**
3. **Student Phone:** 3124792868 **[A03] (optional)**

**Go back to the tutorial instructions and begin working on part B.**

**Part B – Create a WordPress sites on All Platforms**

Part B.1 – Create a WordPress site on a shared web host

1. Enter the following shared web host + cPanel WordPress site information **[**B.1.1**]**
   1. Shared Web Host Company Name: [ namecheap ]
   2. Website Admin Username: [ cmsadmin ]
   3. Website Admin Password: [ CMSPassword9! ]
   4. Database Name: [ wp805 ]
   5. Table Prefix: [ wppj\_ ]
   6. WP website URL: [ <https://it320-domain.online/tutorial05> ]

**- OR Pantheon-**

Part B.2 – Create a WordPress site on Pantheon (PaaS)

1. Enter the following shared web host WordPress site information **[**B.2.1**]**
   1. PaaS WordPress site Admin Username: [ website admin username here ]
   2. PaaS WordPress site Admin Password: [ website admin password here ]
   3. PaaS WordPress Front end site URL: [ website URL here ]

**-AND-**Part B.3 - Create a WordPress site on Codeanywhere

1. Enter the following Codeanywhere web host WordPress site information **[**B.3.1**]**
   1. Site Title: wwu36-tutorial-05-ca
   2. Username: cmsadmin
   3. Password: CMSPassword9!

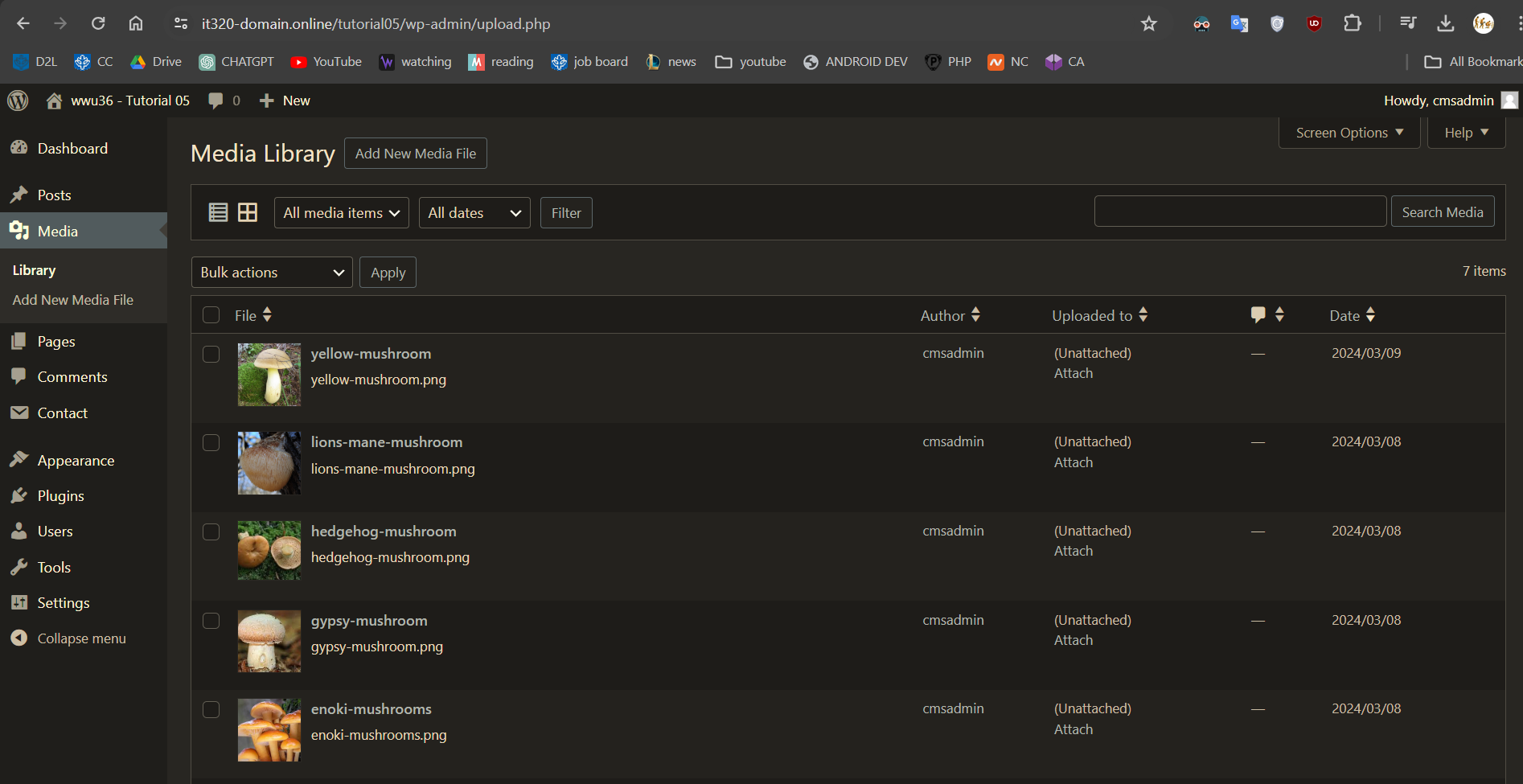
Go back to the tutorial instructions and begin working on Part C.

**Part C –WordPress XML Import**

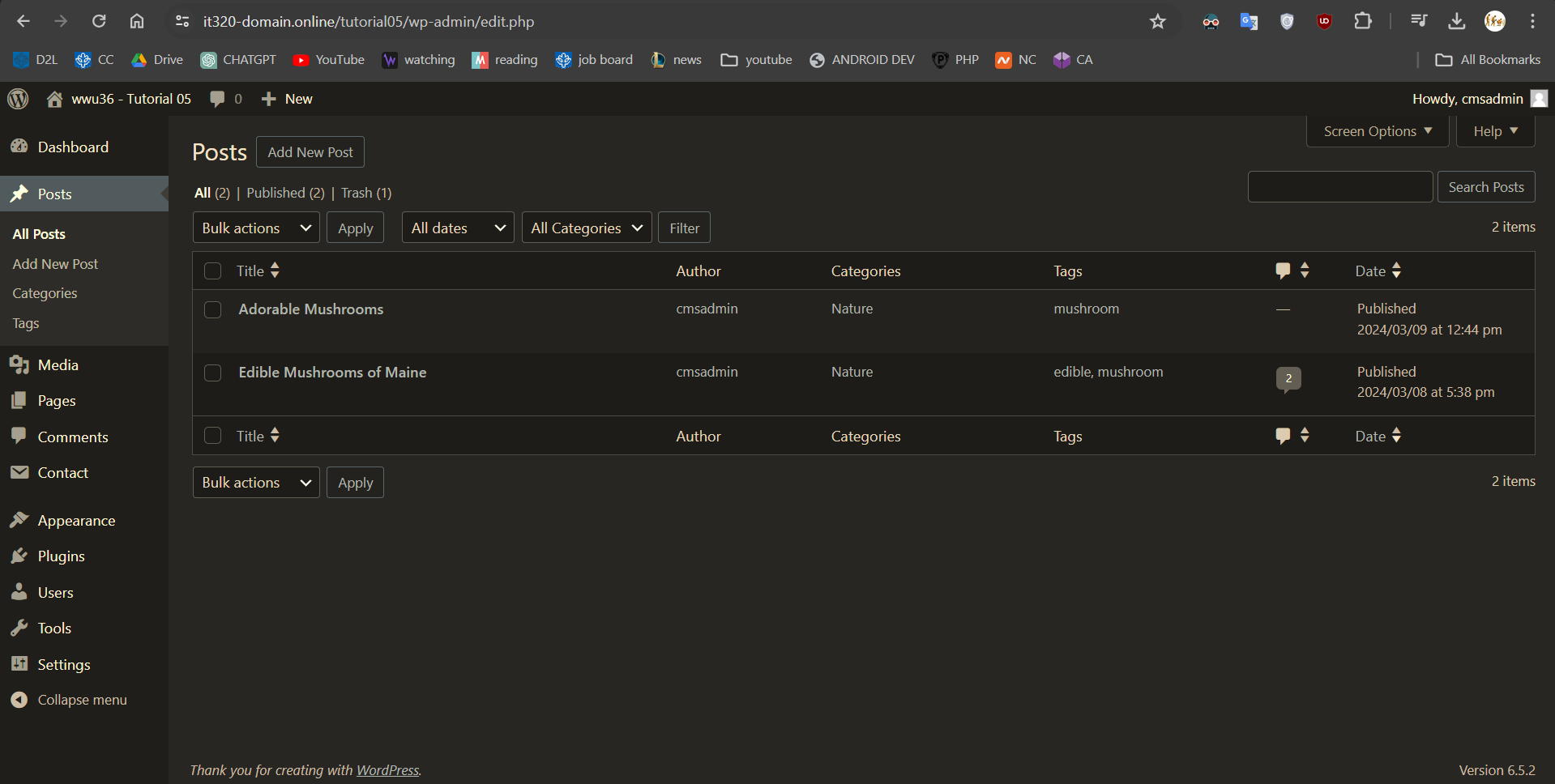
**Please make sure all Part C screen capture include the web browser address bar.**

**Part C.1 –** Imported WordPress site on a shared web host or Patheon

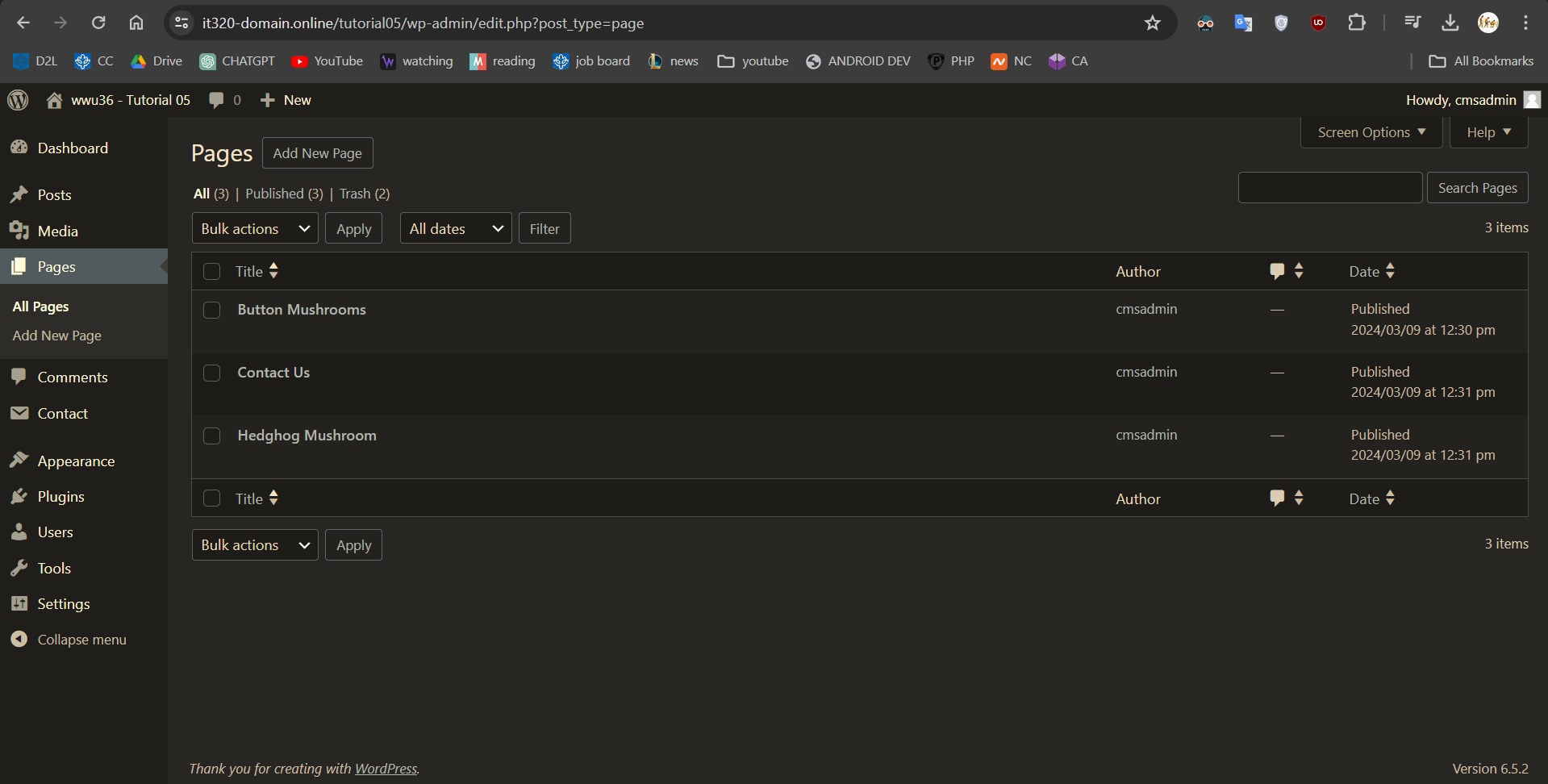
1. Take screen captures of the backend administration WordPress site Dashboard > Media > library in List mode **[**C.1.1**]**



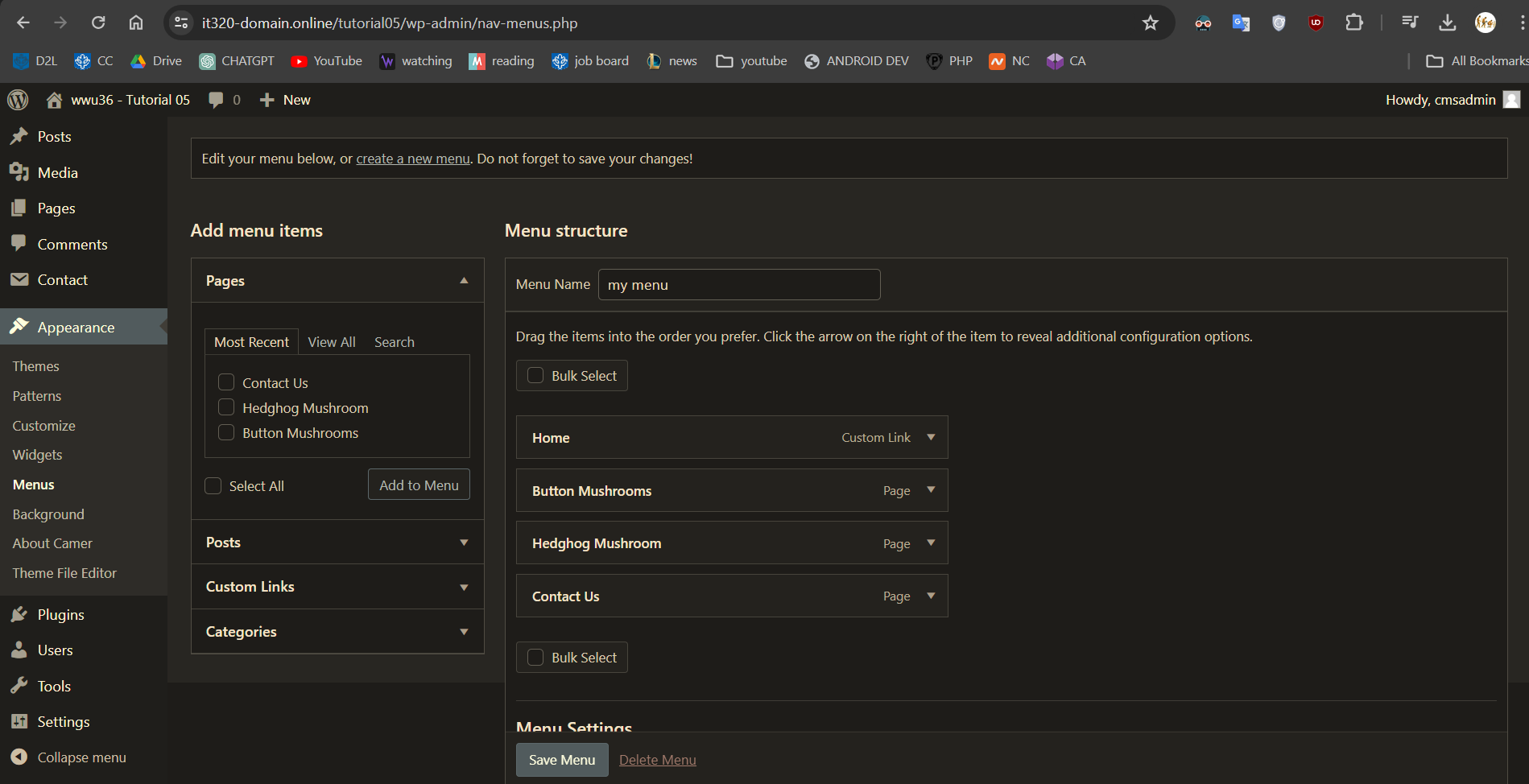
1. Take screen captures of the backend administration WordPress site Dashboard > Posts **[**C.1.2**]**



1. Take screen captures of the backend administration WordPress site Dashboard > Pages **[**C.1.3**]**

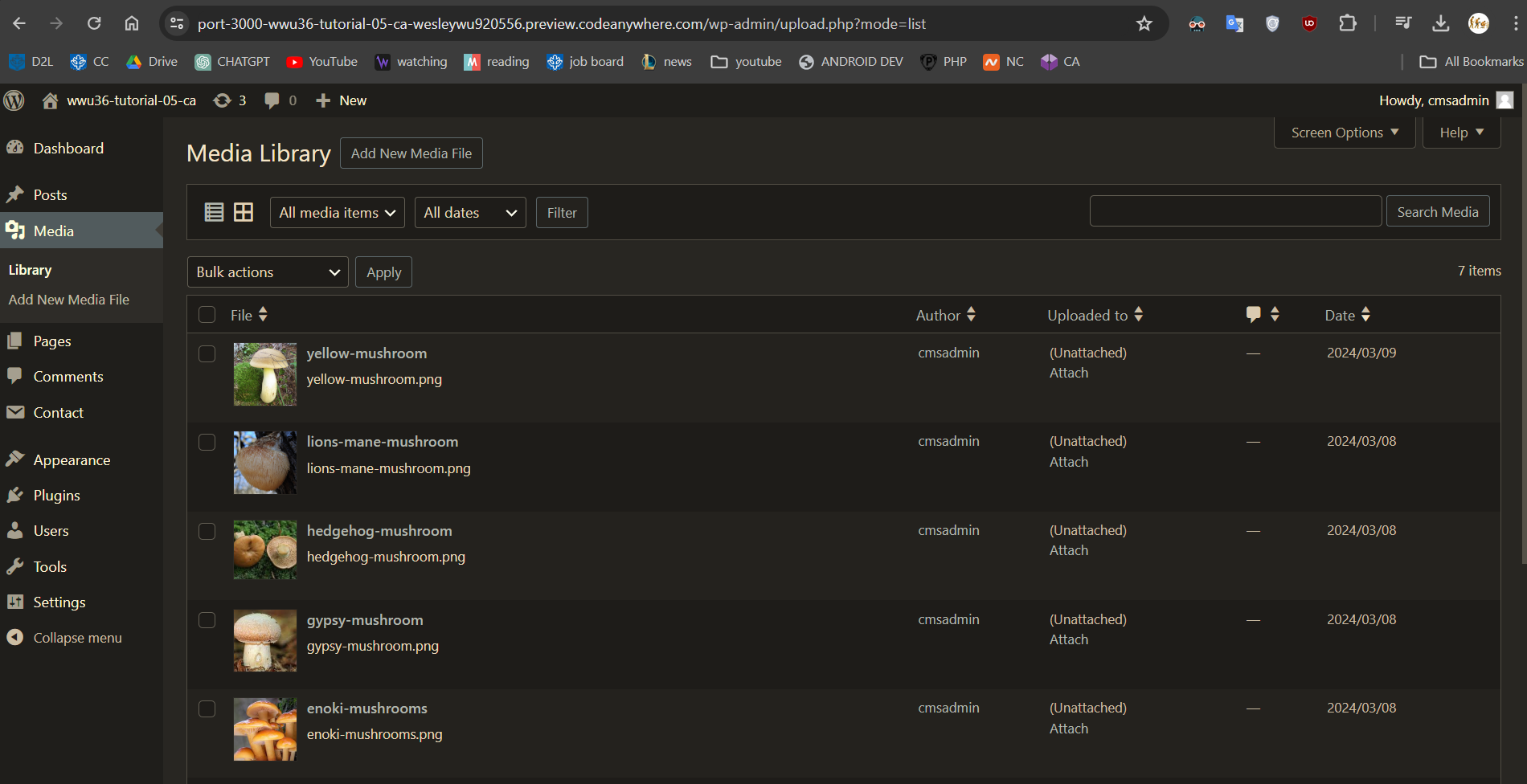


1. Take screen captures of the backend administration WordPress site Dashboard > Appearance > Meus **[**C.1.4**]**

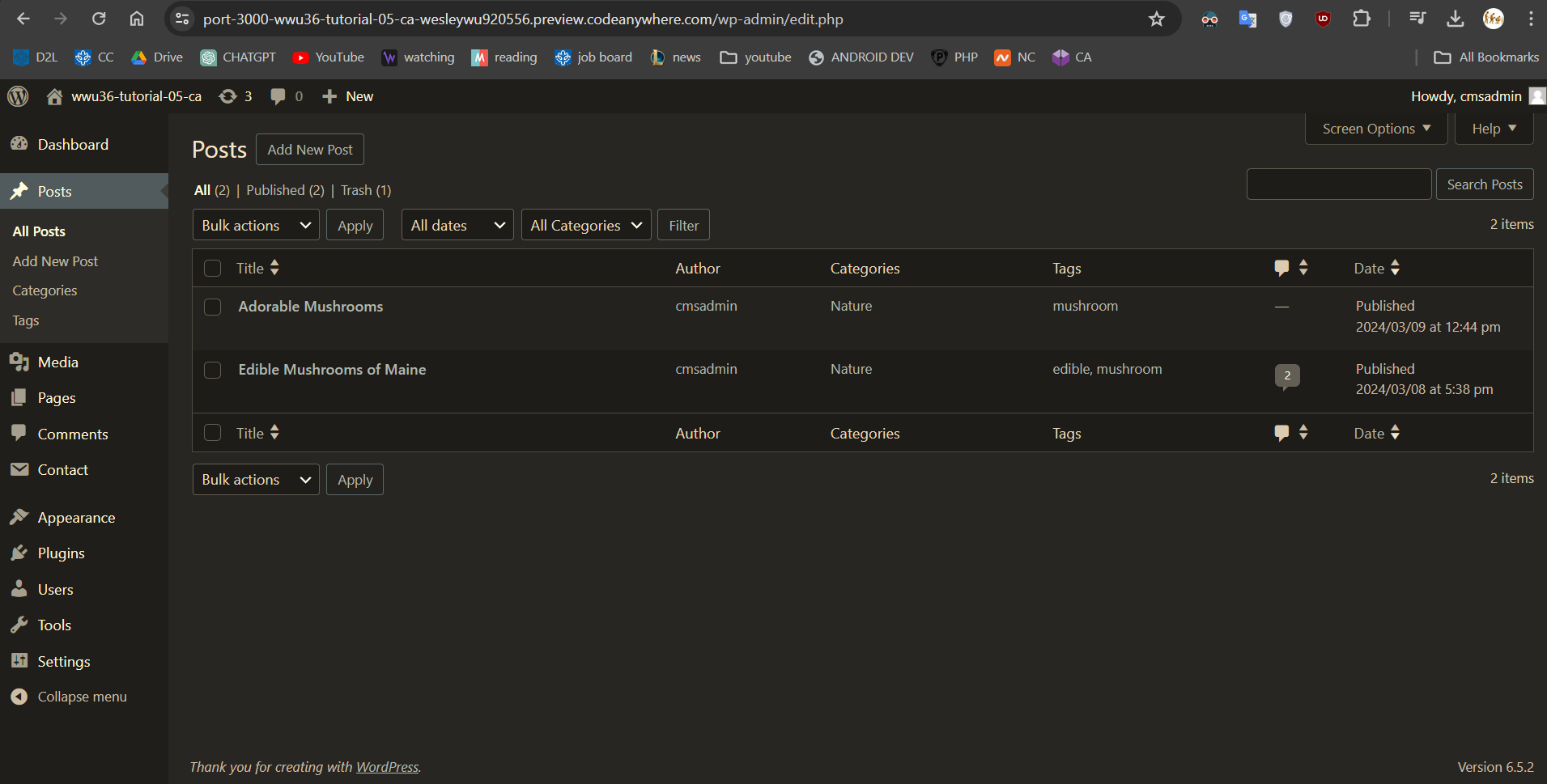
****

**Part C.2 –** Imported WordPress site on a Codeanywhere

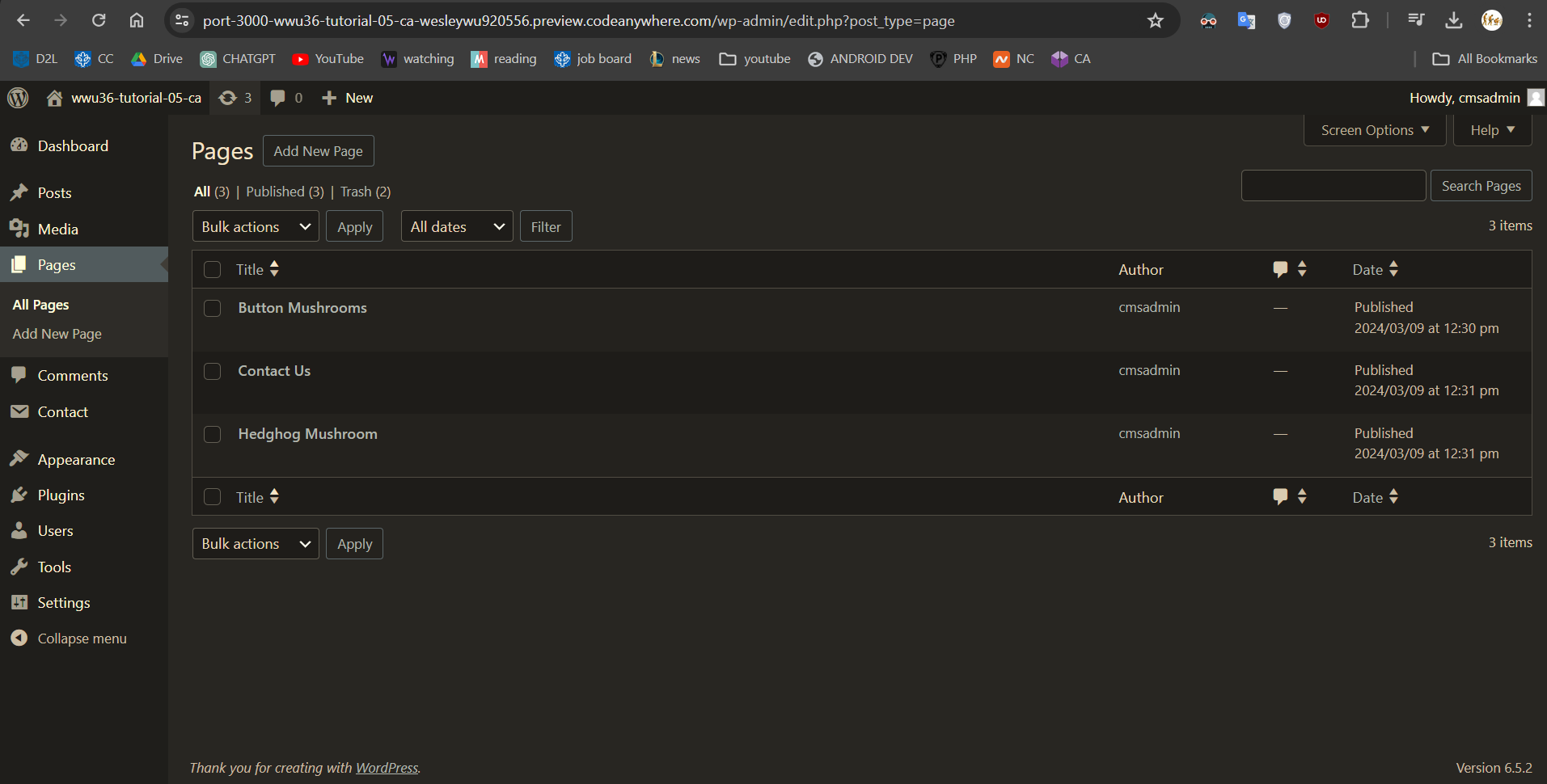
1. Take screen captures of the backend administration WordPress site Dashboard > Media > library in List mode **[**C.2.1**]**



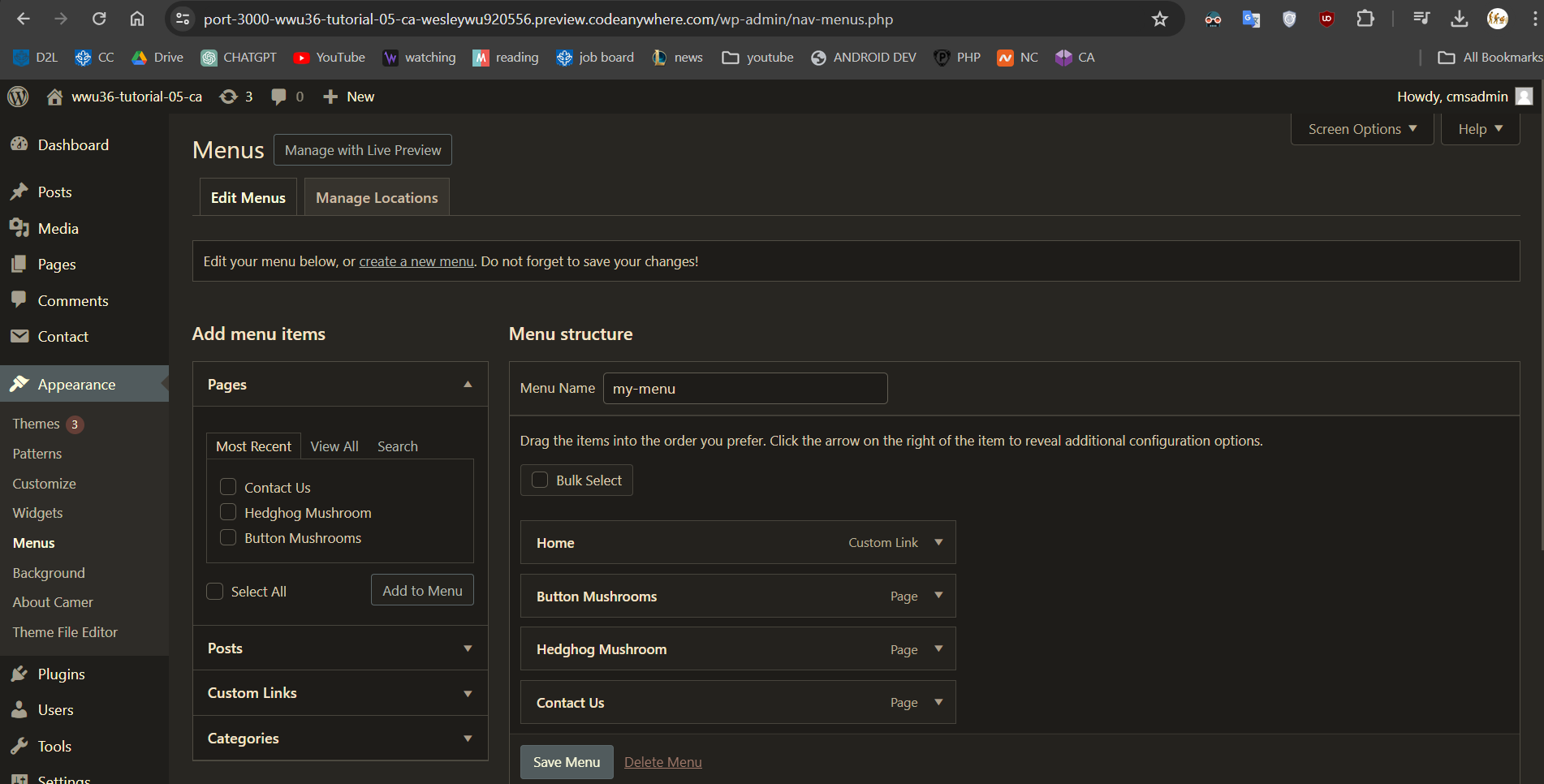
1. Take screen captures of the backend administration WordPress site Dashboard > Posts **[**C.2.2**]**



1. Take screen captures of the backend administration WordPress site Dashboard > Pages **[**C.2.3**]**



1. Take screen captures of the backend administration WordPress site Dashboard > Appearance > Meus **[**C.2.4**]**

****

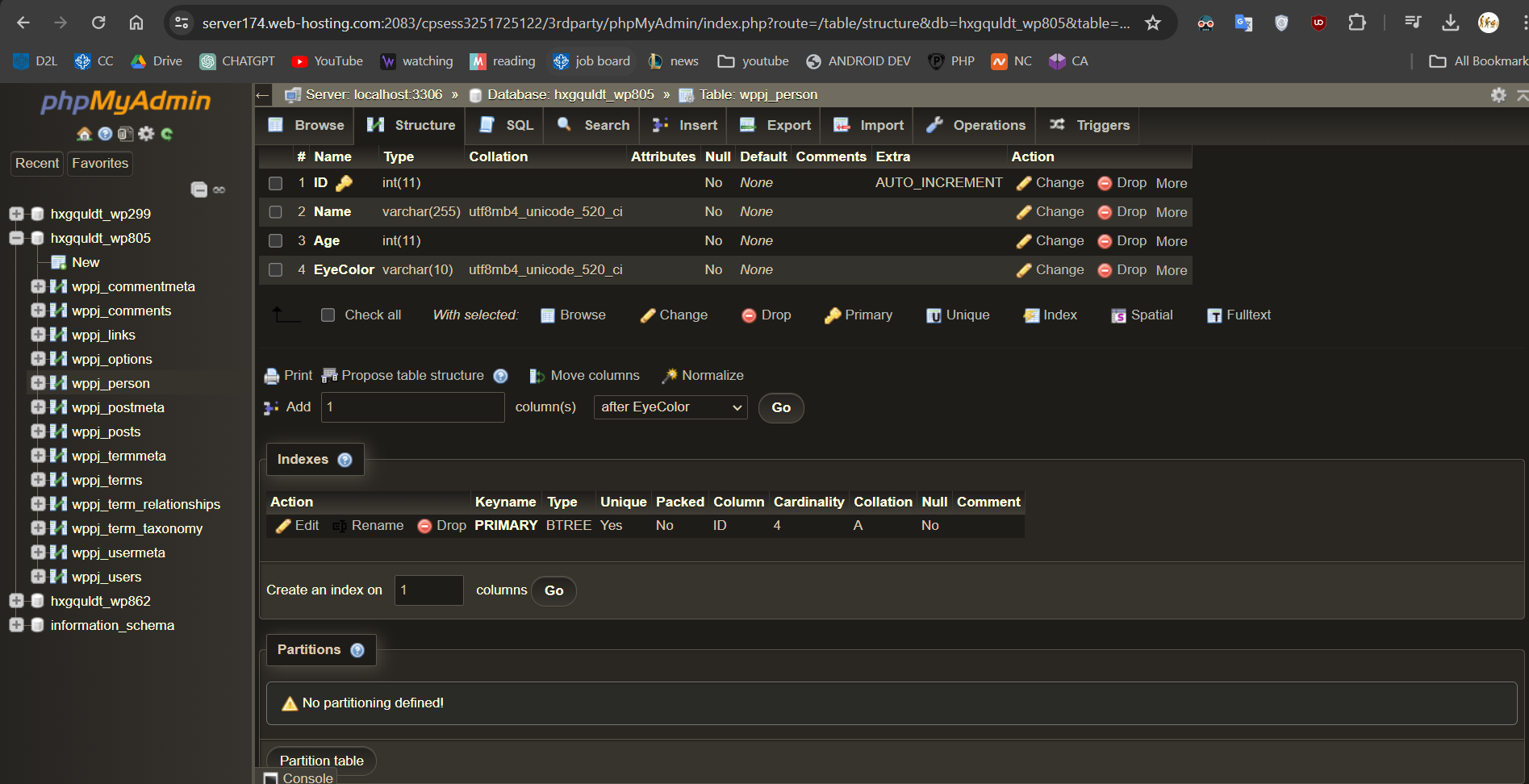
**Go back to the instructions and continue at Part D.**

**Part D – Introduction to phpMyAdmin**

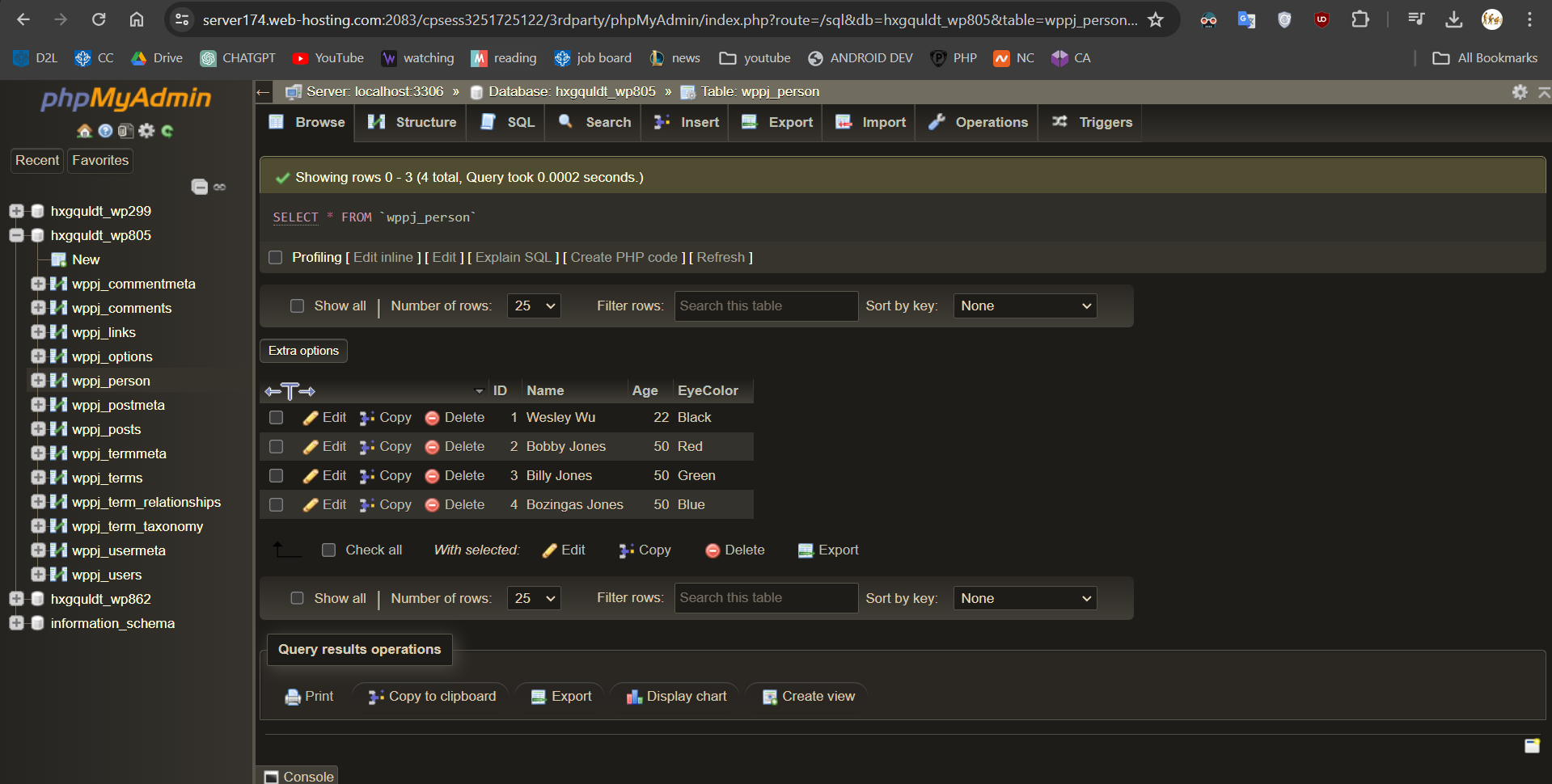
**Please make sure all Part D screen capture include the web browser address bar.**

D.1 Shared Web Host - Access phpMyAdmin on your shared web host   
(Note: this work is for Namecheap shared web host cPanel account users)

1. Select the current WordPress site’s MySQL Database
2. Select the Person table and take a screen capture of its structure **[**D.1.1**]**



1. Select the Person table and take a screen capture of its data (records) **[**D.1.2**]**

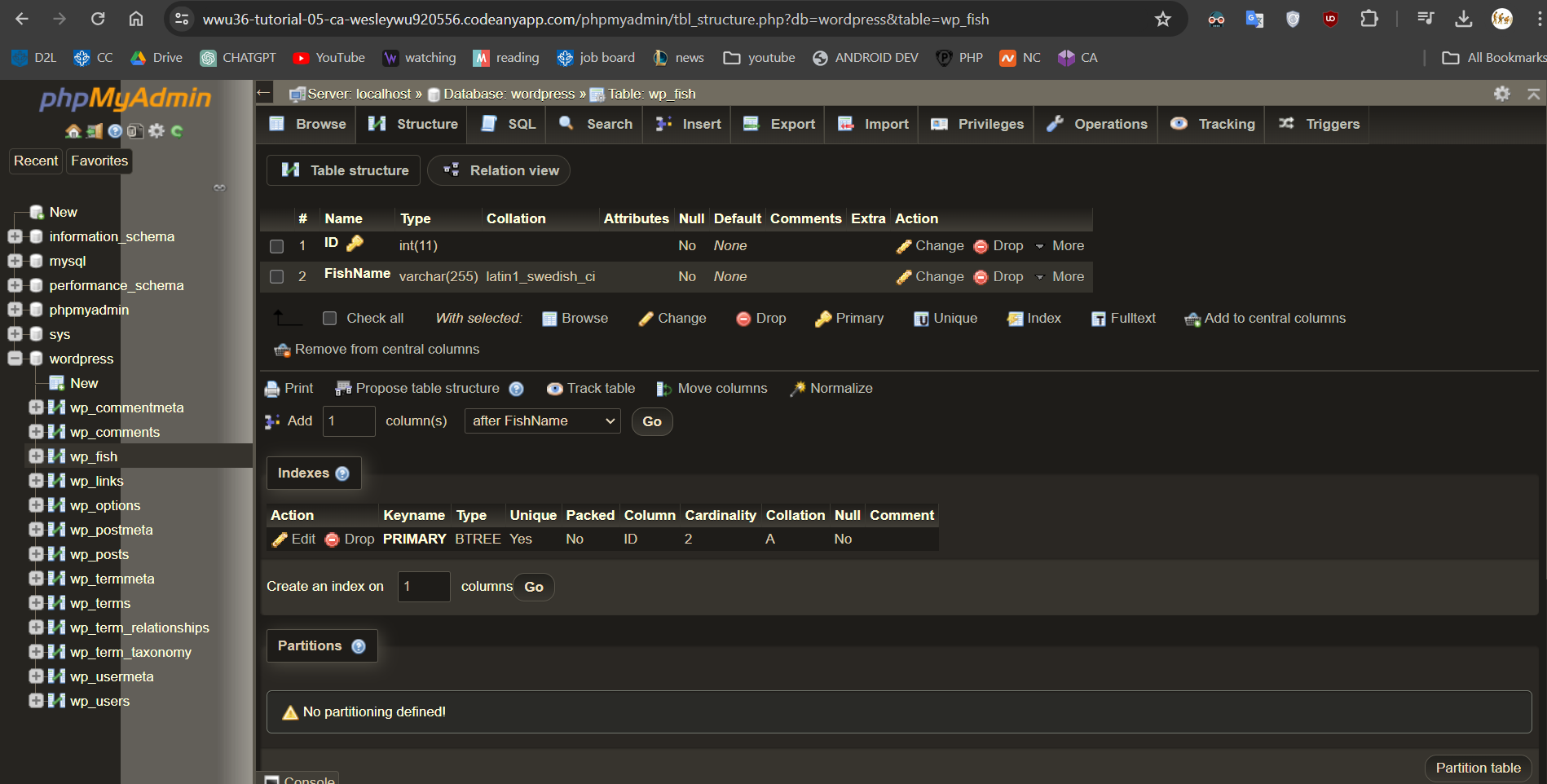
****

D.2 Pantheon - Access MySQL Workbench on your local computer   
(Note: this work is for Pantheon account users. If you have Namecheap, don’t do this part)

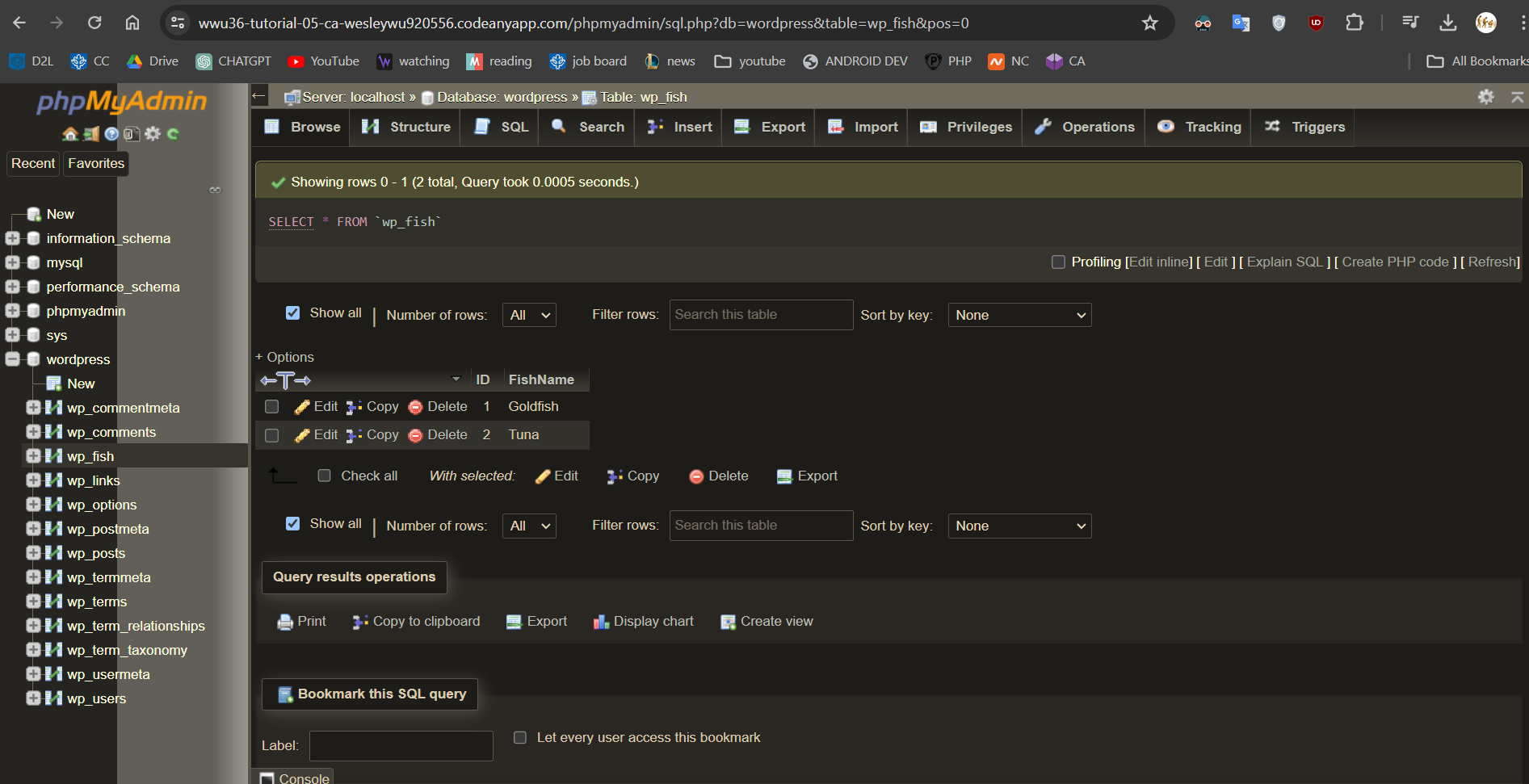
1. Connect to your Pantheon WP site’s MySQL database
2. Select the Person table and take a screen capture of its structure **[**D.2.1**]  
   [ Paste screen capture here ]**
3. Select the Person table and take a screen capture of its data (records) **[**D.2.2**]  
   [ Paste screen capture here ]**

D.3 Codeanywhere - Access phpMyAdmin on your Codeanywhere account  
(Note: Everyone does this part)

1. Access and login to phpMyAdmin using the passwordless root account
2. Select the wordpress MySQL Database
3. Select the wp-fish table and take a screen capture of its structure   
   **[**D.3.1**]**



1. Select the fish table and take a screen capture of its data (records)   
   **[**D.1.2**]**

****

Go back to the instructions and continue at Part E.

**Part E – Improve WordPress site security**

**Please make sure all Part E screen capture include the web browser address bar.**

1. What 5 steps should you take to enhance your WordPress site security   
   **[**E.1**]**1. Keep site and systems up to date, 2. salt keys, 3. use strong passwords, 4. limit admins, 5. scan for malware

1. What exactly does it mean (what do you do) to keep your WordPress environment up to date?   
   **[**E.2]

**[**Answer here**]**

Core files, plugins, and themes can be updated in the dashboard or FTP.

1. Why is it so important to keep your WordPress environment up to date? **[**E.3**]**

**[**Answer here**]**

Updates to old software does not only add new features but may also target security issues.

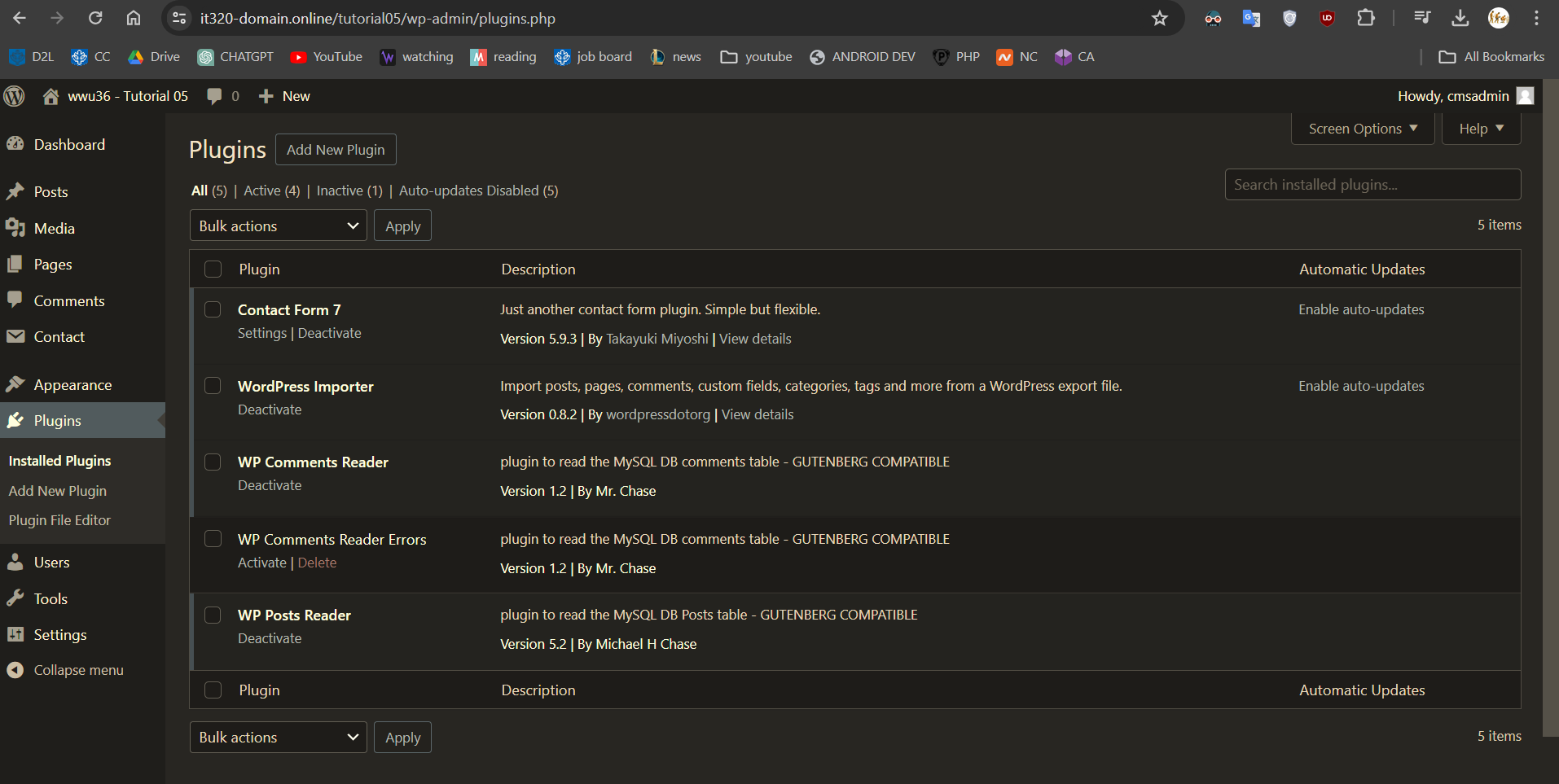
Go back to the instructions and continue at Part F.

**Part F – Extend WordPress custom Plugins**

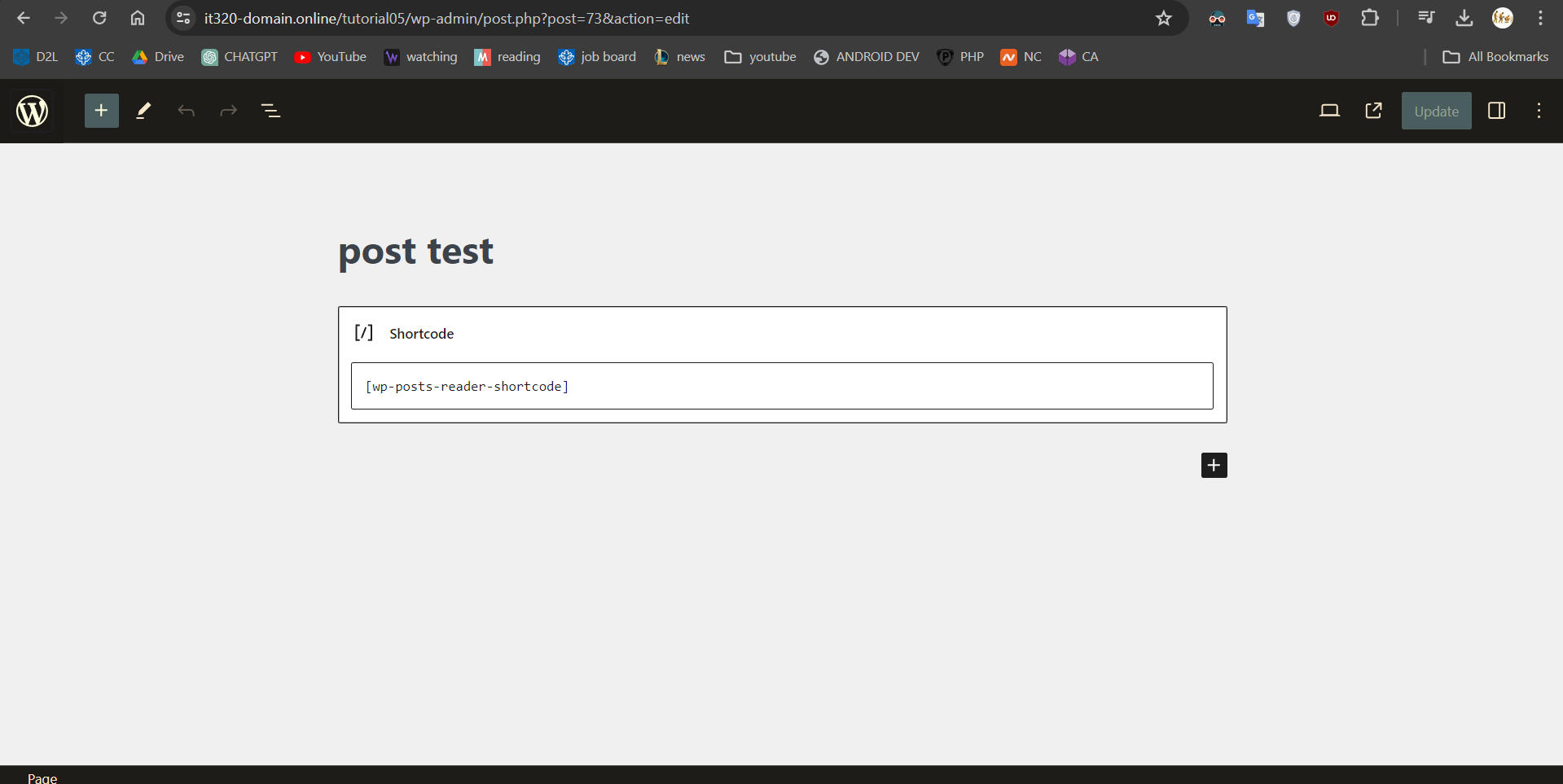
**Please make sure all Part F screen capture include the web browser address bar.   
The Classic Editor should have been included with the WP site import.   
It should be in a disabled state. If it is not disabled, please disable and delete it now.  
Do not enable the Classic Editor plugin.  
We do not want the Classic Editor enabled as we want to create Gutenberg compatible WP plugins.   
Cache plugins should not have been included in the WP site import.   
If cache plugins exist on your imported WP site, please deactivate and delete them now.**

F.1-F.2 Review the Module 05 D2L Presentation  
 **F.3 Shared Web Host Custom Plugin – Posts Reader**(Note: this work is for Namecheap shared web host cPanel users, not Pantheon users)

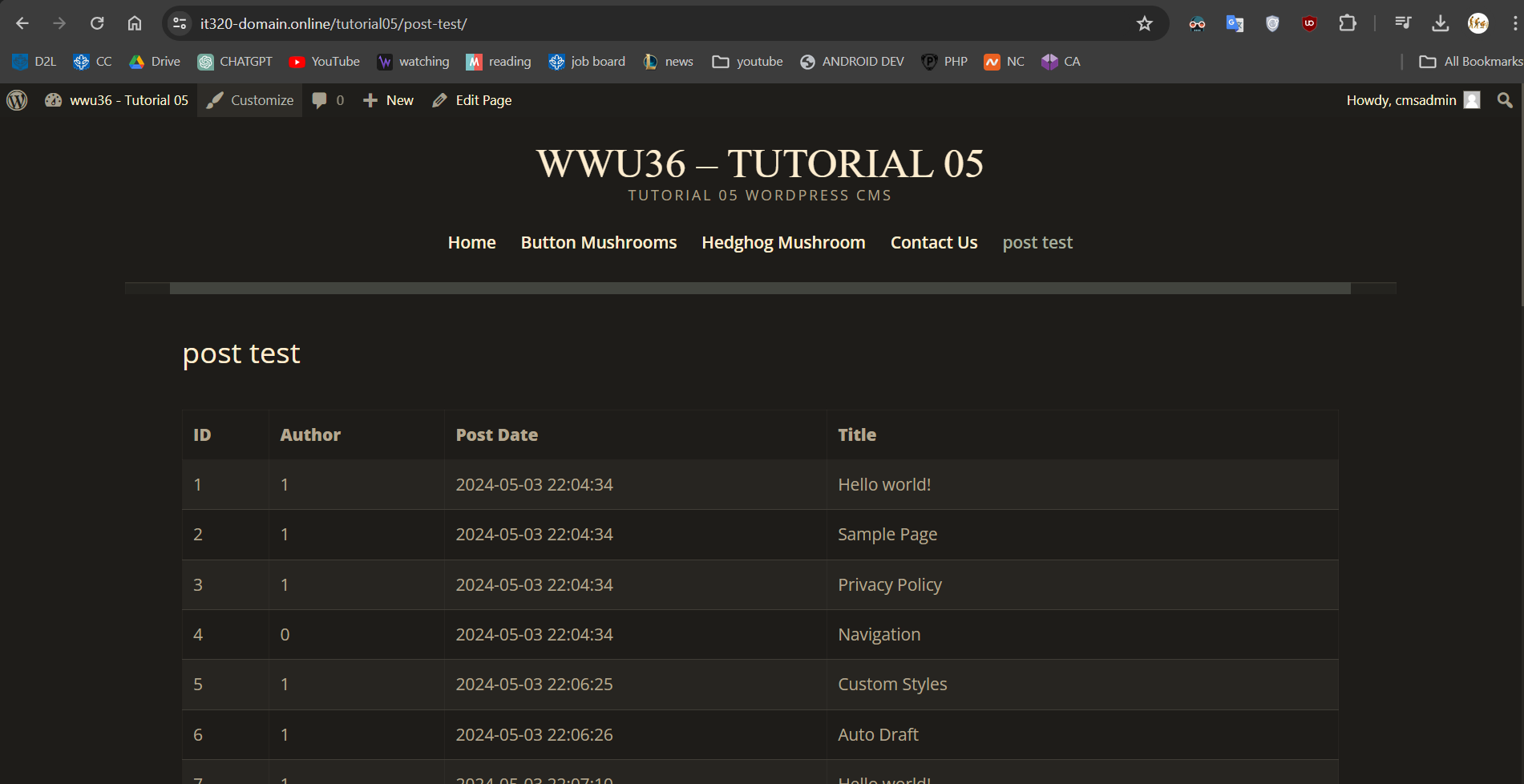
1. Take screen captures of the backend administration WordPress site Dashboard > Plugins page showing the Classic Editor Plugin & Cache plugins are not listed (not installed)   
   **[**F.3.1**]**



1. From the Dashboard, select Pages. Select Edit for Post Test Page and take a screen capture of the Test Page Template   
   **[**F.3.2**]**



1. Take screen captures of the frontend client facing WordPress site showing the menu and   
   Post Test Page menu item selected   
   **[**F.3.3**]**

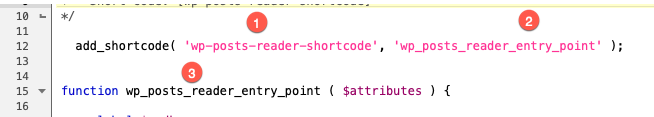


**F.4 Pantheon Custom Plugin – Posts Reader**(Note: this work is only Pantheon users, not cPanel account users)

1. Take screen captures of the backend administration WordPress site Dashboard > Plugins page showing the Classic Editor & Cache Plugins are not installed  
    **[**F.4.1**]  
   [**Paste screen capture here**]**
2. From the Dashboard, select Pages. Select Edit for Post Test Page and take a screen capture of the Test Page Template   
   **[**F.4.2] **[**Paste screen capture here**]**
3. Take screen captures of the frontend client facing WordPress site showing the menu and   
   Post Test Page menu item selected  
   **[**F.4.3**]  
   [**Paste screen capture here**]**

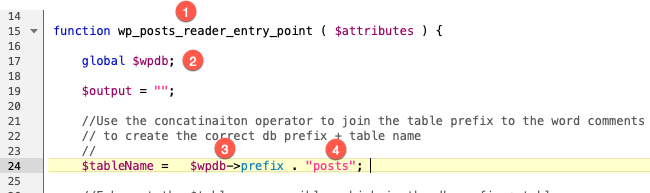
**F.5 There is no F.5**

**F.6 Creating a WP Custom Plugin by copying, renaming and editing an existing WP custom module file**

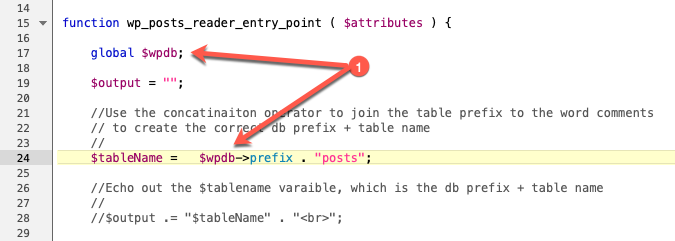
1. In this scenario, I already have the posts reader plugin installed on my WP site and it is ACTIVE. I want to make a copy of the posts reader plugin and then install it the same WP site. The table I want to read and display its content is named “users”. If I have copied the posts reader custom plugin file and want to edit the file to read and display information from the MySQL database “users” table, what changes must I make to the file name and 1,2,3 below?   
   **[F.6.1]**

Answer: Change the file name **wp-posts-reader.php** to wp-users-reader.php  
  
Answer: Changes to (1) (2) & (3)  
(1) wp-users-reader-shortcode  
(2) wp\_users\_reader\_entry\_point  
(3) wp\_users\_reader\_entry\_point

1. You have copied and renamed the posts reader custom plugin file and want to edit the new file to read and display information from the MySQL database “**users**” table. What changes must you make to the following items 1-4 in the picture below?

**[**F.6.2**]**  


Answer: 1 → wp\_users\_reader\_entry\_point, 4 → “users”

1. In a few sentences, describe the purpose of the $wpdb variable?  
   **[**F.6.3**]**  
   

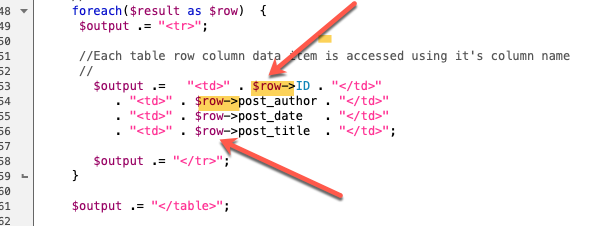
Answer: It is an object which represents the database, so queries can be made from within the plugin by first fetching a table name.

1. Are you allowed to change the $wpdb variable name (line 17 & 24 above) and if not, why not? **[**F.6.4**]**

Answer: No, because then you couldn’t interact with the database.

1. What is the purpose of the PHP statement $wpdb->prefix (line 24 in picture above) and why are we using it in our custom plugin code? **[**F.6.5**]**

Answer: Since multiple installations share the same database distinguished by their prefixes, the statement wpdb→prefix allows the code to fetch the current installation’s tables.

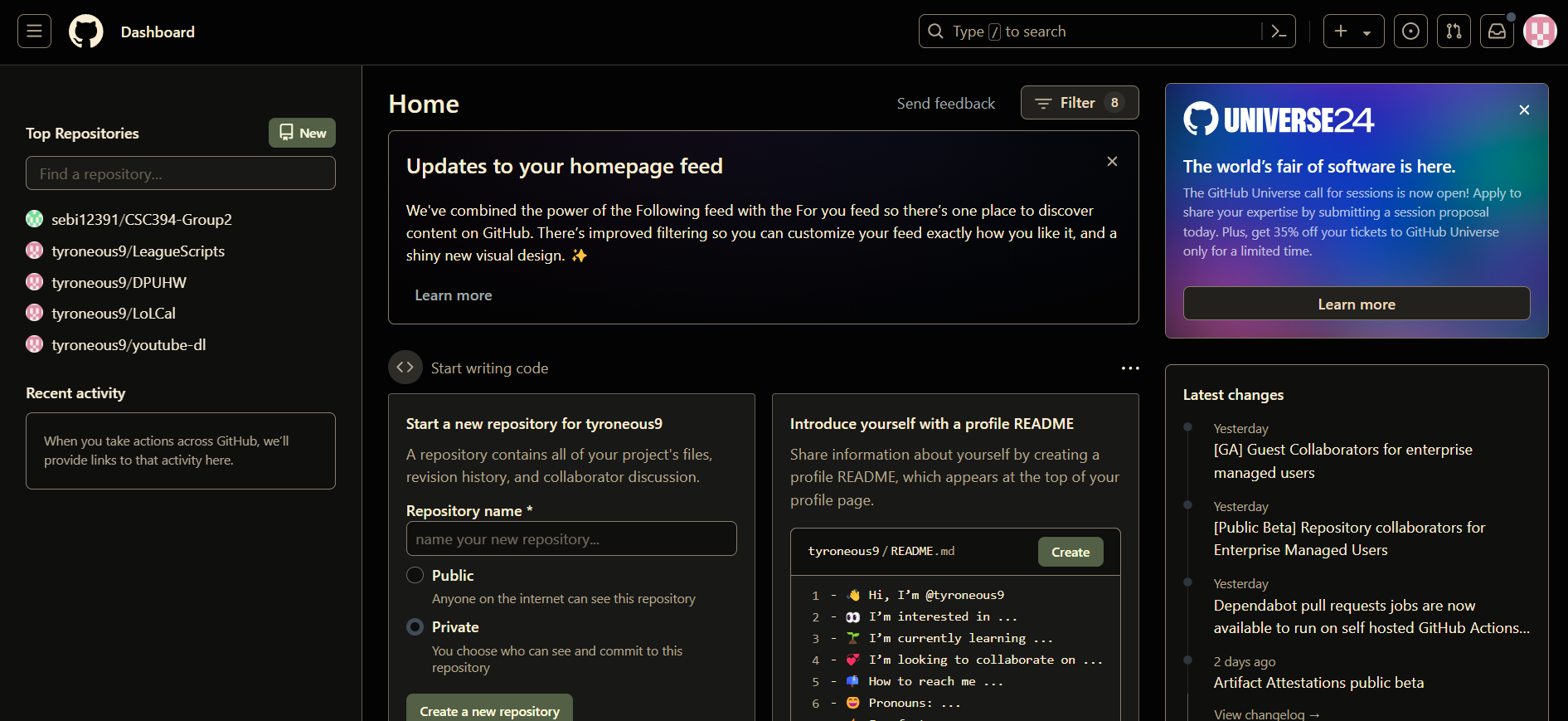
1. You are assigned the task of creating a WP custom plugin to read the MySQL “users” table. The only issue is, you do not know the exact table name. Is the table name “user” singular or “users”plural? What steps should you take, what tools should you use to determine the correct table name?   
   Hint: the answer is not email or ask someone. **[**F.6.6**]**  
   Answer: Either look for the table in myphpadmin or run a test query with the custom plugin.
2. In the picture below, there are 4 **$row->** operators. Each $row-> operator is followed by a variable (ID, post\_author, post\_date, post\_title). What are these variables? Where can you go to find their exact spelling and case, so you want to make changes to the PHP code that uses them, like when you copy and paste PHP code form one custom plugin, that queries table AA and in your custom plugin, you want to query table ZZ, a different MySQL DB table.  
   **[**F.6.7**]**  
     
   Answer: Each variable corresponds to a column in the table. You can check through myphpadmin by finding the table and checking its structure.

Go back to the instructions and continue at Part G.

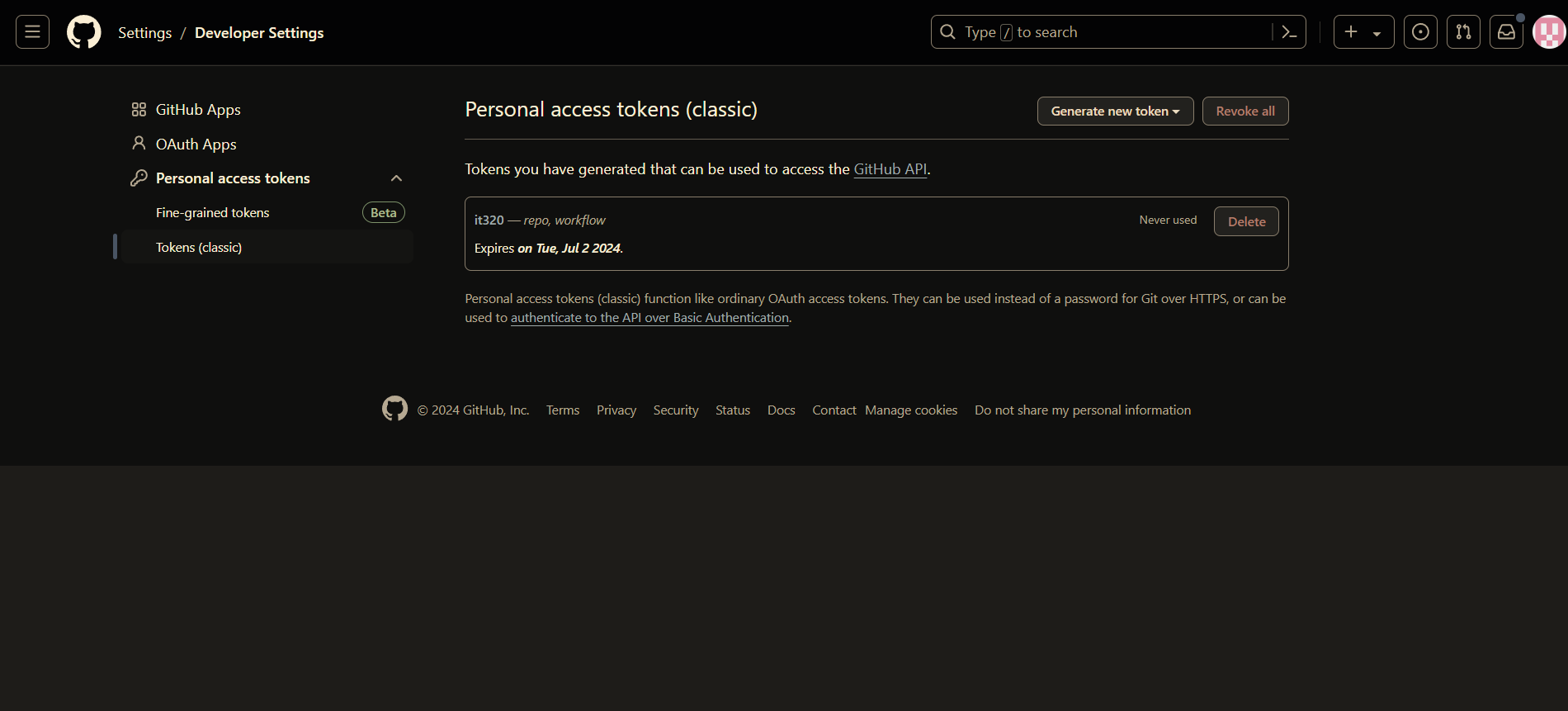
**Part G – GitHub Workflow**

**Please make sure all Part G screen capture include the web browser address bar. [G]**

1. Take a screen capture of your GitHub.com Account main page   
   **[**G.1**]  
   [**Paste screen capture here**]**



1. Proof of GitHub Personal Access Token - **WORTH 10 POINTS**  
   **[**G.2**]  
   [**Screen capture for proof of Personal access token**]  
   This can be the Personal Access Token page on your GitHub account, or a screen capture of the email form GitHub about your Personal access token being created.**



1. Write a few sentences about the GitHub workflow discussed on the tutorial video. - **WORTH 10 POINTS**  
   Why are we encrypting the WordPress database?   
   Why are we creating and using a .env file?   
   What are the dangers of pushing a collection of unprotected WordPress files to a Public REPO?   
   **[**G.3**]  
   [**Paste screen capture here**]**

We encrypt the database because it would be exposed in a public repo. The .env file is what encrypts the db. Pushing wp files to a public repo can lead sensitive info to be exposed, such as login info in the wp-config file.

Go back to the instructions and continue at Part H.

**Part H – PHP QUESTIONS**

**Testing PHP Programs using PHP Emulation Websites:**

* Online PHP test sites:
  + [http://phptester.net](http://phptester.net/)
  + <http://sandbox.onlinephpfunctions.com/>
* How you test your script depends on which test site you use.
* **Each PHP emulation test site works slightly different**.
* Please read the site instructions before testing your PHP code.

1. Write and test PHP code that uses the sizeof prewritten function to get the size of a PHP array or integers and then echo out the size of the array:
   1. Create a PHP integer array of 7 integers.
   2. Write a PHP statement that uses the sizeof function to get the size of the array and assigns the size to a variable named $sizeOfAarray.
   3. Echo out the $sizeOfAarray value.
   4. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

1. Write and test PHP code that uses the gettype prewritten function to get the variable data types of 6 different variable data types:
2. Create 6 variables with the flowing names and values:
   1. $cars = array(‘Ford’, ‘Mazda’, ‘Telsa’);
   2. $name = “Bill”;
   3. $age = 35;
   4. $weight = 189.35;
   5. $isTrue = false;
   6. $carsAssocArray = array('a' => 'Alpha', 'B' => 'Bentley');
3. Write a PHP statement that uses the gettype function to get the data types of each variable and on the same line, echo out the value of the return value of the call to the gettype function call.
   1. Example: echo gettype( $variableNameHere) . “<br>”;
4. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

1. Write and test PHP code that uses the strlen prewritten function to get the length of a PHP string variable then echo out the length of the string:
   1. Create a PHP variable with a string value of at least 10 characters.
   2. Write a PHP statement that uses the strlen function to get the length of the string and echo out the value.
   3. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

1. Write and test PHP code that uses the strlen prewritten function to get the length of a PHP string variable then echo out the length of the string:
   1. Create a PHP variable with a string value of at least 5 characters with 5 spaces at the front of the string and 5 spaces at the end of the string.
      1. Example: $a = “ cccccc “;
   2. Write a PHP statement that uses the strlen and trim functions to get the length of the string before and after calling the trim function and echo out the values.
   3. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

1. Write and test PHP code that uses the print\_r prewritten function to print out the values of an integer variable, a string variable, and integer array variable and an associative array variable. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

1. Write and test PHP code that uses the empty prewritten function to evaluate string variables with and without length and only print out the values of stings that are not empty.   
   **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste the code block and record the results from the web emulator on the next row. A screen shot from the PHP test site will be the best way to not affect template format. Please try not to change the template numbering or format. |
| Paste screen capture from PHP test website here |

Please go back to the tutorial instructions and start work on Part I.