**Tutorial 03 –** Build WordPress CMS on Multiple Platforms and using Best Practices, extend with images, menus, themes, pages, posts, categories, tags, and plugins.

Students are required to have access to at least two Web Host accounts that they use to create and manage multiple CMS websites.   
Additionally, we will do a few PHP problems, to test your understanding of the PHP material we covered in class.   
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 and change the* ***red text to black text.*** *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 answer all questions.***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: [STUDENT NAME – Aamir, Patel] [A01]**

**2. Student Contact Information:   
[Student Preferred Email: - Aamir8567@gmail.com] [A02]   
[Student Phone Number - Aamir8567@gmail.com- Aamir8567@gmail.com] [A03]**

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

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

1. Enter the following shared web host WordPress site information **[B.1]**
   1. Shared Web Host Company Name: [ NameCheap ]
      1. Shared Web Host URL: [ **namecheap.com**]
      2. Shared Web Host account username: [ aamir8567 ]
      3. Shared Web Host password: [ Kamboli1!]
   2. WordPress Website Info:
      1. Website Admin Username: [ aapadcpate]
      2. Website Admin Password: [ 9aIwR@LZB%#uk9Fgh3 ]
      3. Database Name: [wp322]
      4. Table Prefix: [ wppj\_ ]
   3. WordPress website Frontend URL: [ http://aamirpatel.tech/mod3tutorial3/]

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

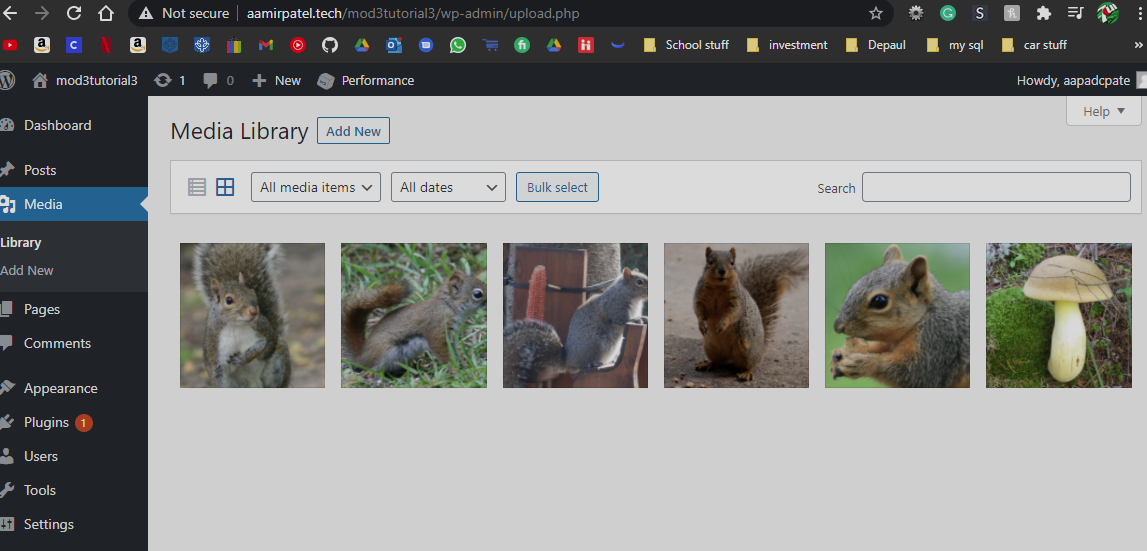
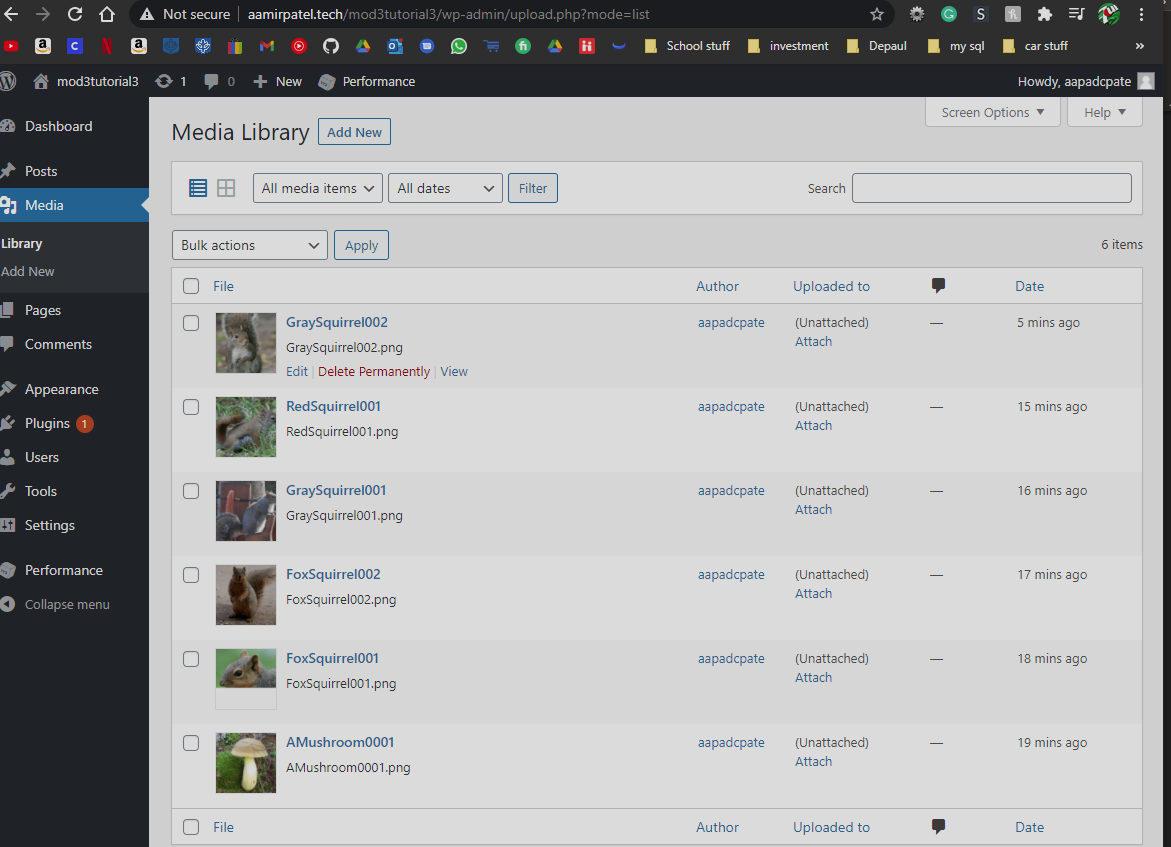
**Part B.2 – Create a WordPress site on Pantheon (PaaS)**

1. Enter the following shared web host WordPress site information **[B.2]**
   1. PaaS WordPress Front end site URL: [ **N/A**]
   2. PaaS Web Host account username: [ **N/A**]]
   3. PaaS Web Host password: [**N/A**]]
   4. PaaS WordPress site Admin Username: [ **N/A**]
   5. PaaS WordPress site Admin Password: [ **N/A**]]

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

**Part C – Extend WordPress with Images**

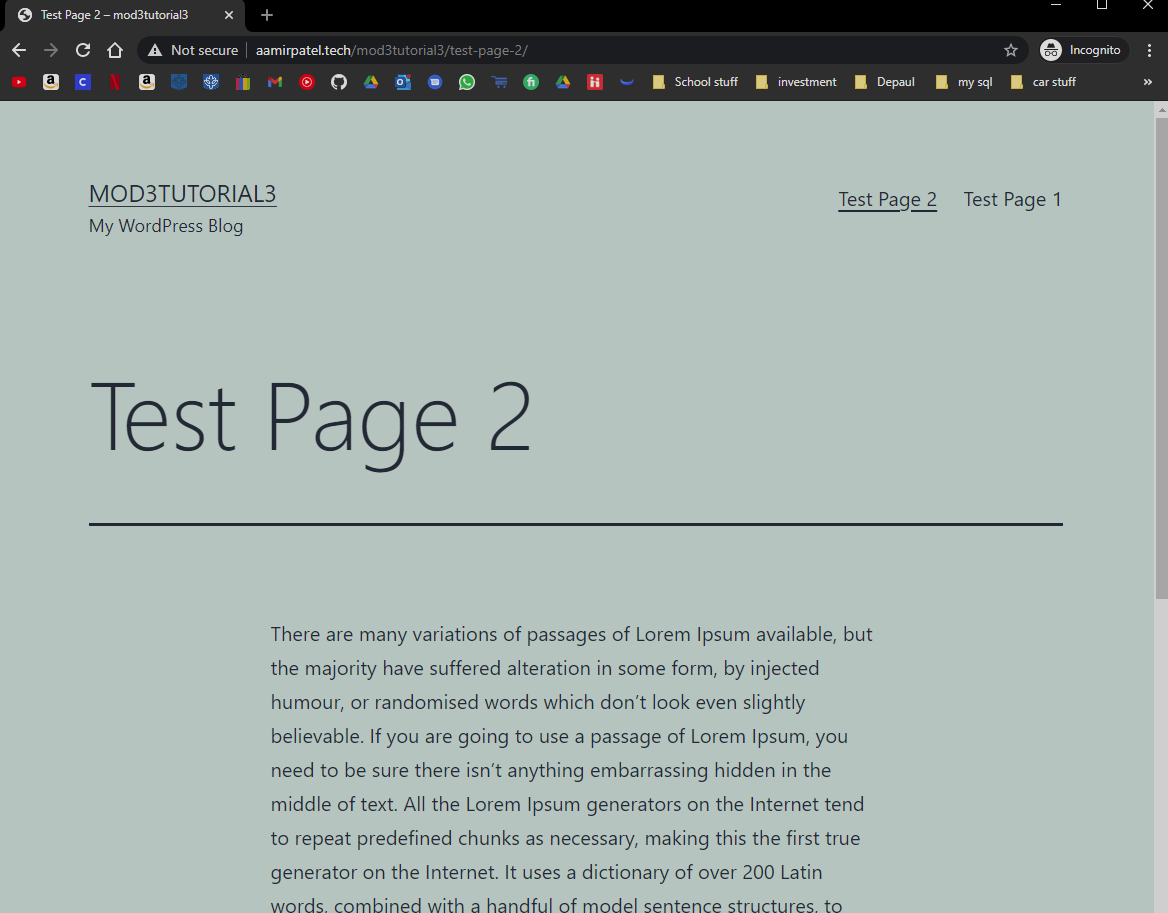
**Please make sure all Part C screen capture include the web browser address bar. [C]**

1. Take screen captures of the backend administration WordPress site Dashboard > Appearance > Media page – with the media library in GRID mode **[C.1]  
   [ Grid - ]**
2. Take screen captures of the backend administration WordPress site Dashboard > Appearance > Media page – with the media library in LIST mode **[C.2]  
   [ List: - **

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

**Part D – Extend WordPress with Menus**

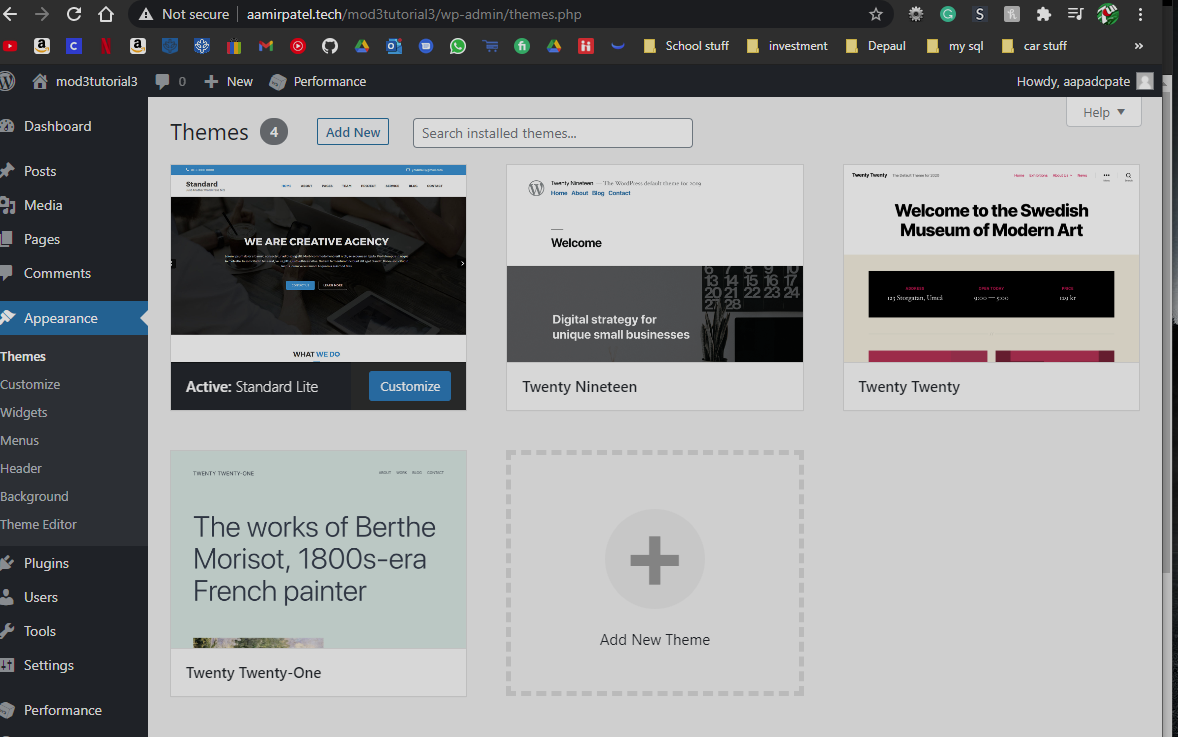
**Please make sure all Part D screen capture include the web browser address bar. [D]**

1. Take screen captures of the backend administration WordPress site Dashboard > Appearance > Menu page – please make sure the menu Structure and Menu Settings sections are captured **[D.1]  
   [ ]**
2. Create 2 test pages – see tutorial video for details
3. Take screen captures of the front-end client facing WordPress site showing the 2 test page menu items on the home page**[D.2]  
   ]**

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

**Part E – Extend WordPress with Themes**

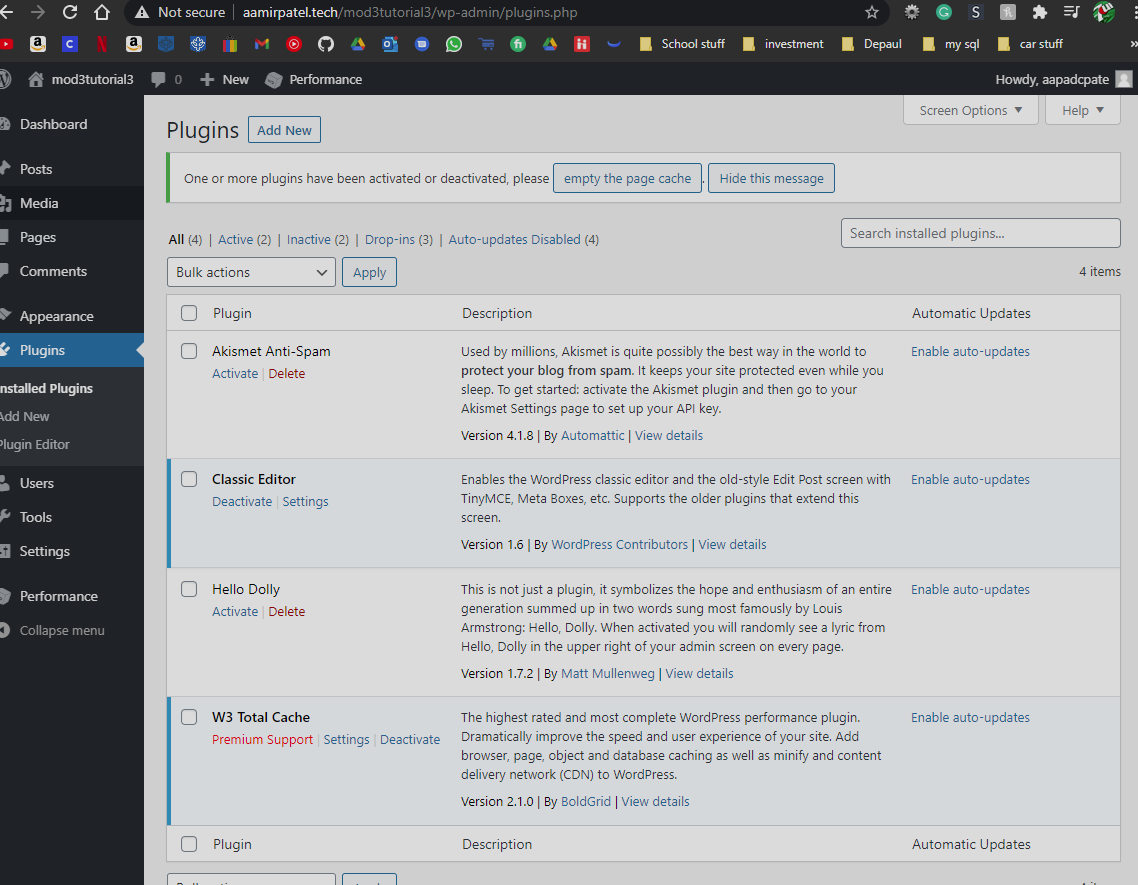
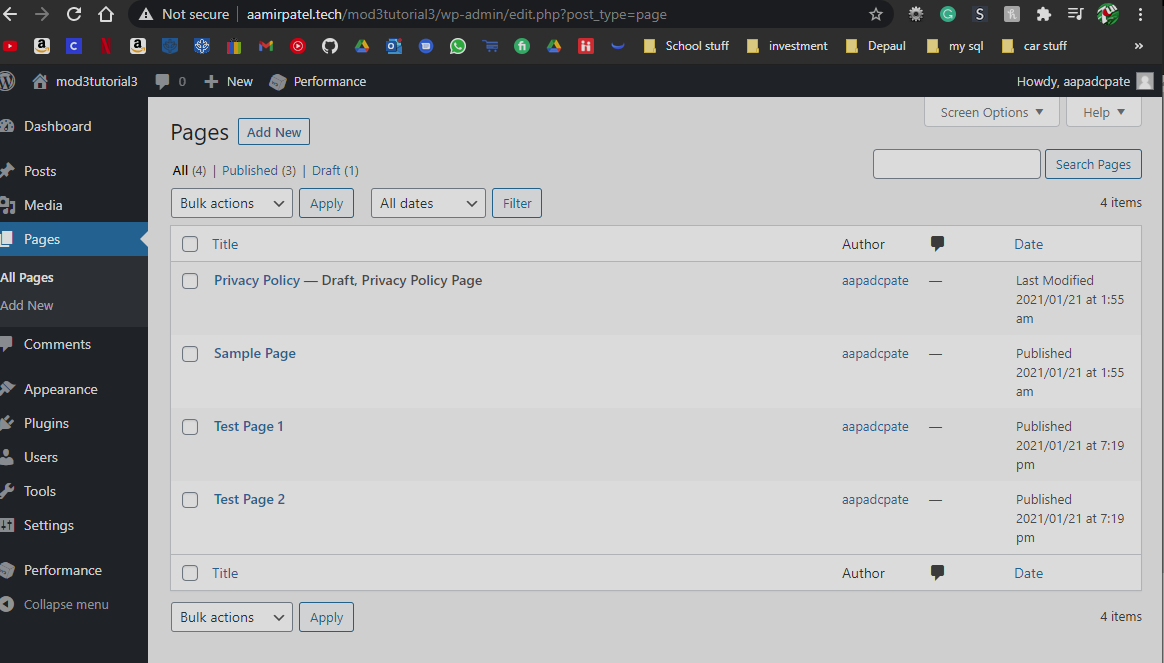
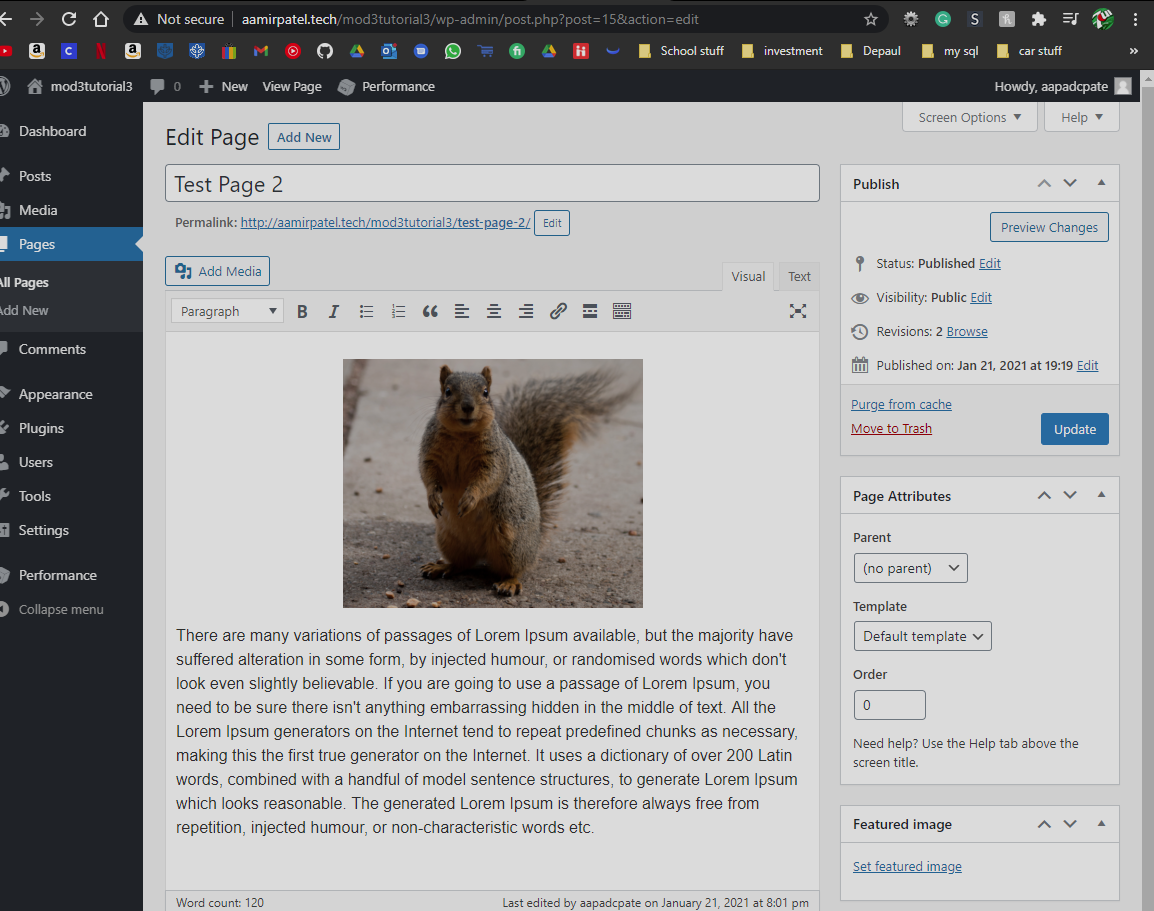
**Please make sure all Part E screen capture include the web browser address bar. [E]**

1. Take screen captures of the backend administration WordPress site Dashboard > Appearance > page – **[E.1]  
   [ ]**

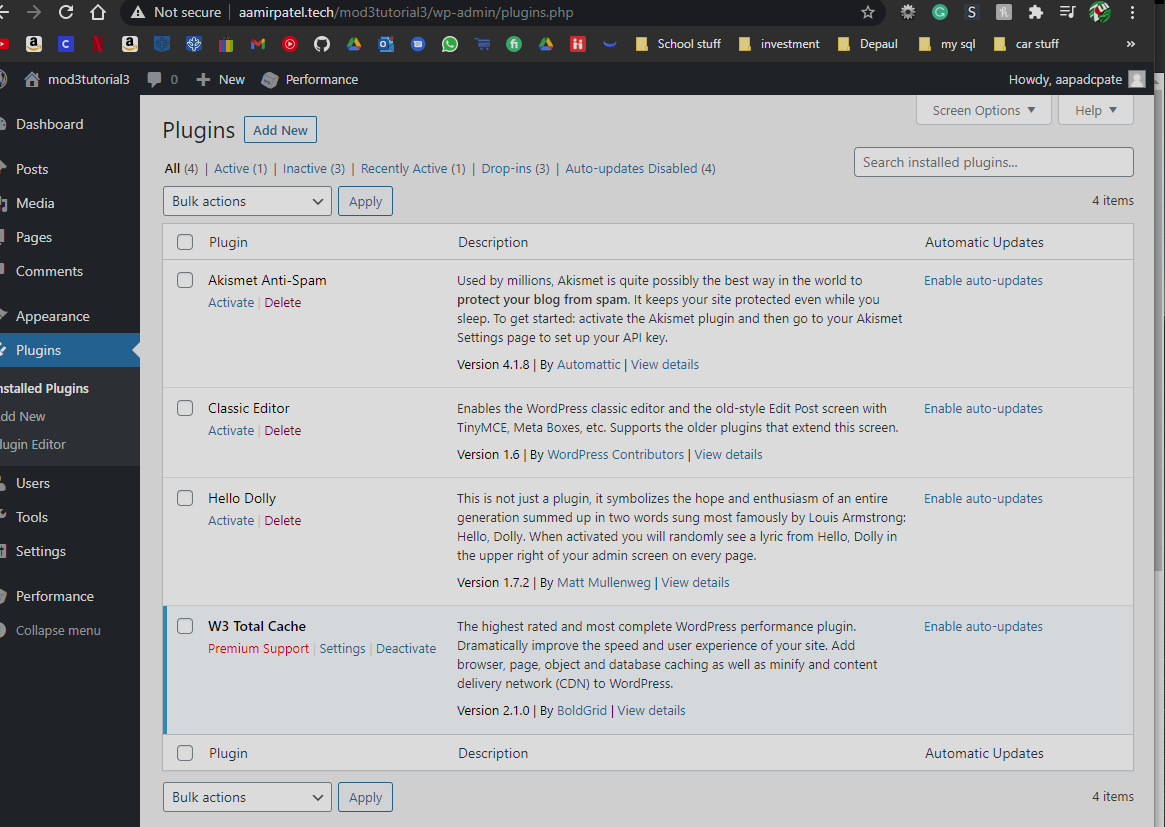
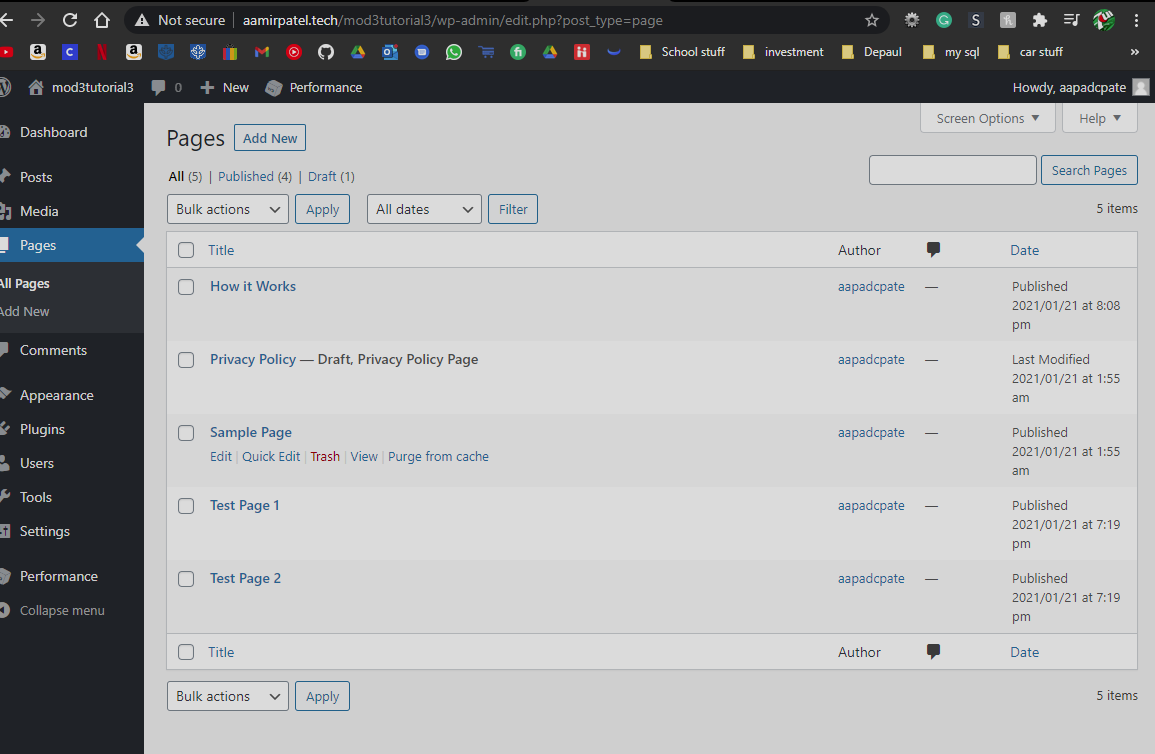
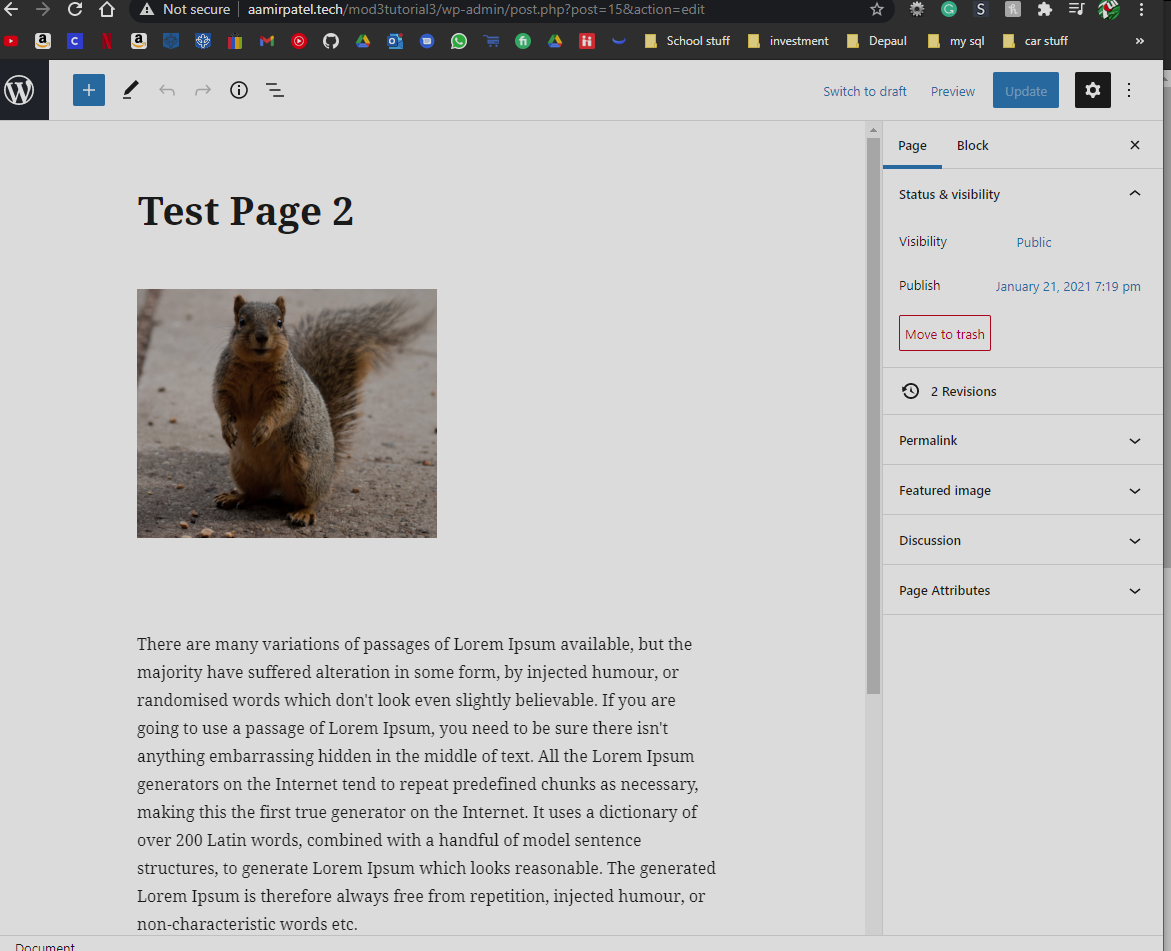
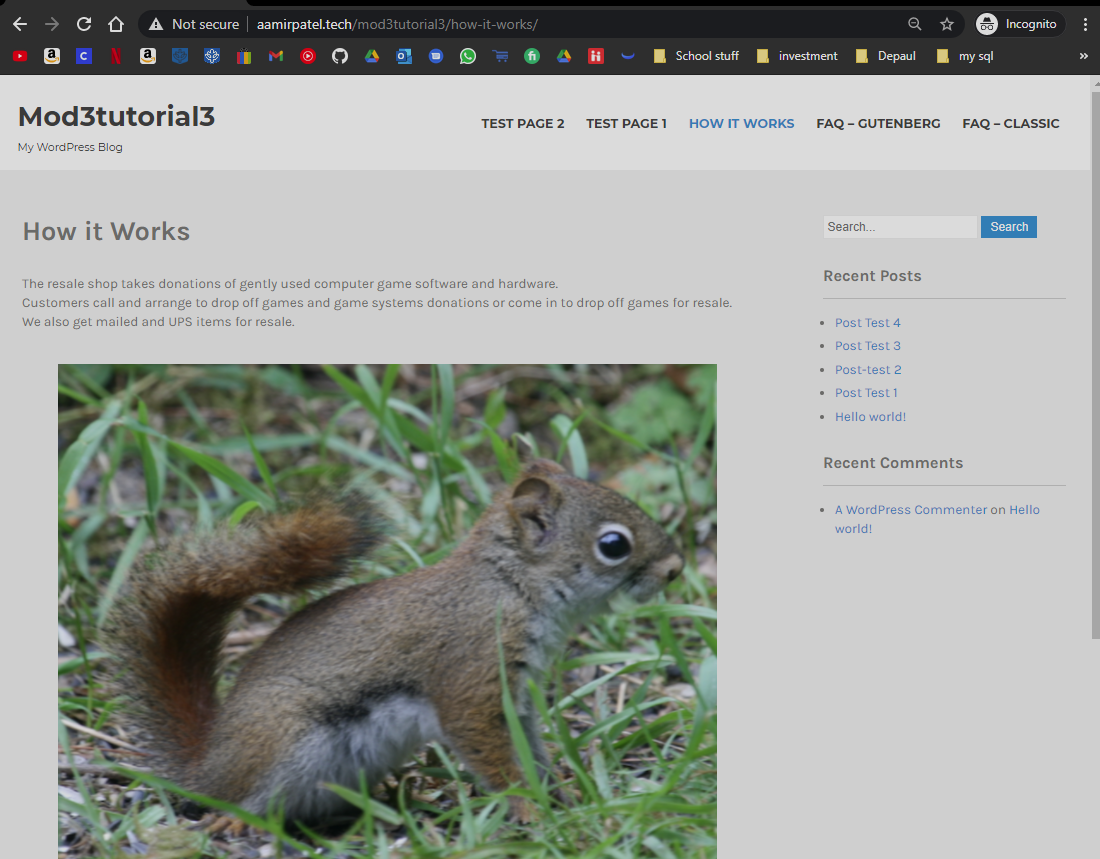
**Go back to the instructions and continue at Part F.**

**Part F – Extend WordPress with Page Content – Classic vs. Gutenberg**

**Please make sure all Part F screen capture include the web browser address bar.   
Pages Created with Classic Editor [F.1]**

1. Take screen captures of the backend administration WordPress site Dashboard > Plugins page showing the Classic Editor Plugin Active **[F.1.1]  
   [ ]**
2. Take screen captures of the backend administration WordPress site Dashboard > Pages showing the list of Pages **[F.1.2]  
   [ ]**
3. From the Dashboard, select Pages. Select Edit for “Test Page 1” and take a screen capture of the Test Page Template **[F.1.3]  
   [ ]**
4. Take screen captures of the frontend client facing WordPress site showing the menu and   
   “Test Page 2” menu item selected **[F.1.4]  
   [ ]**

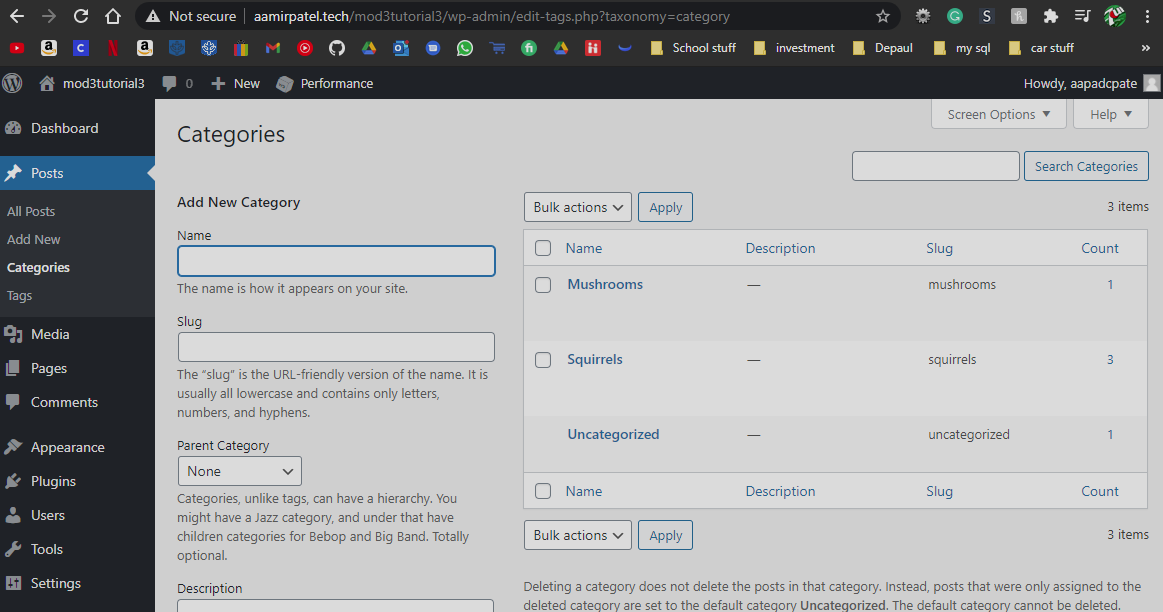
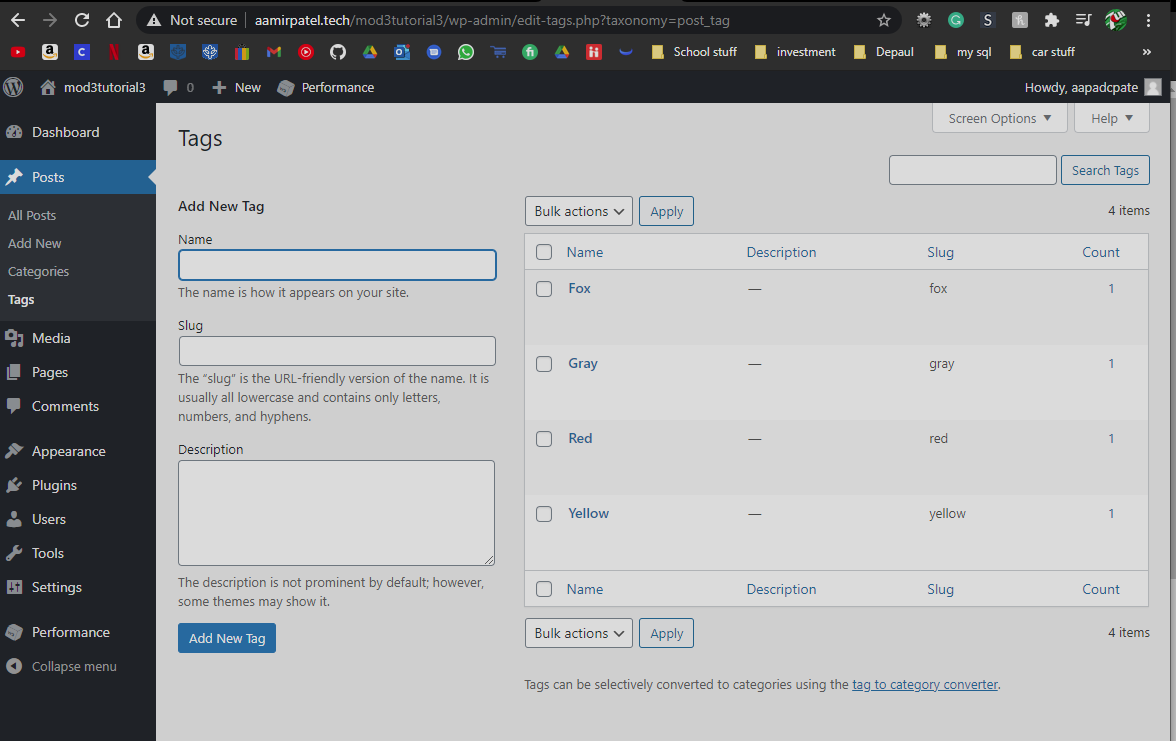
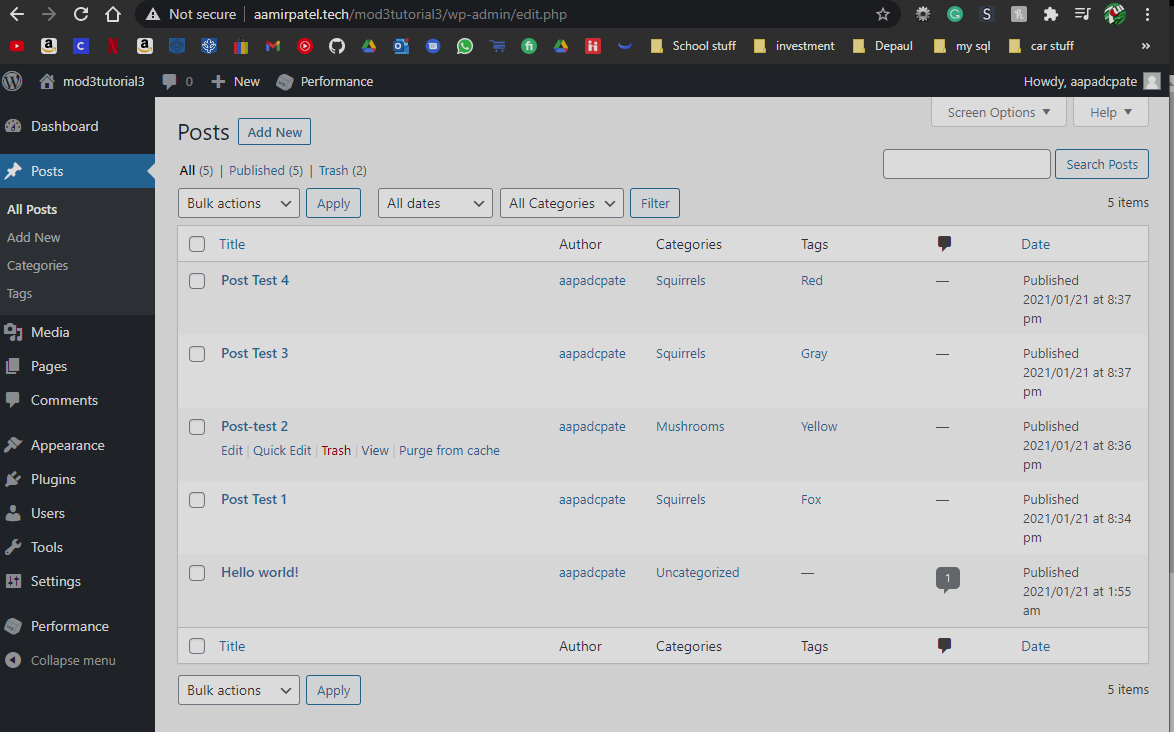
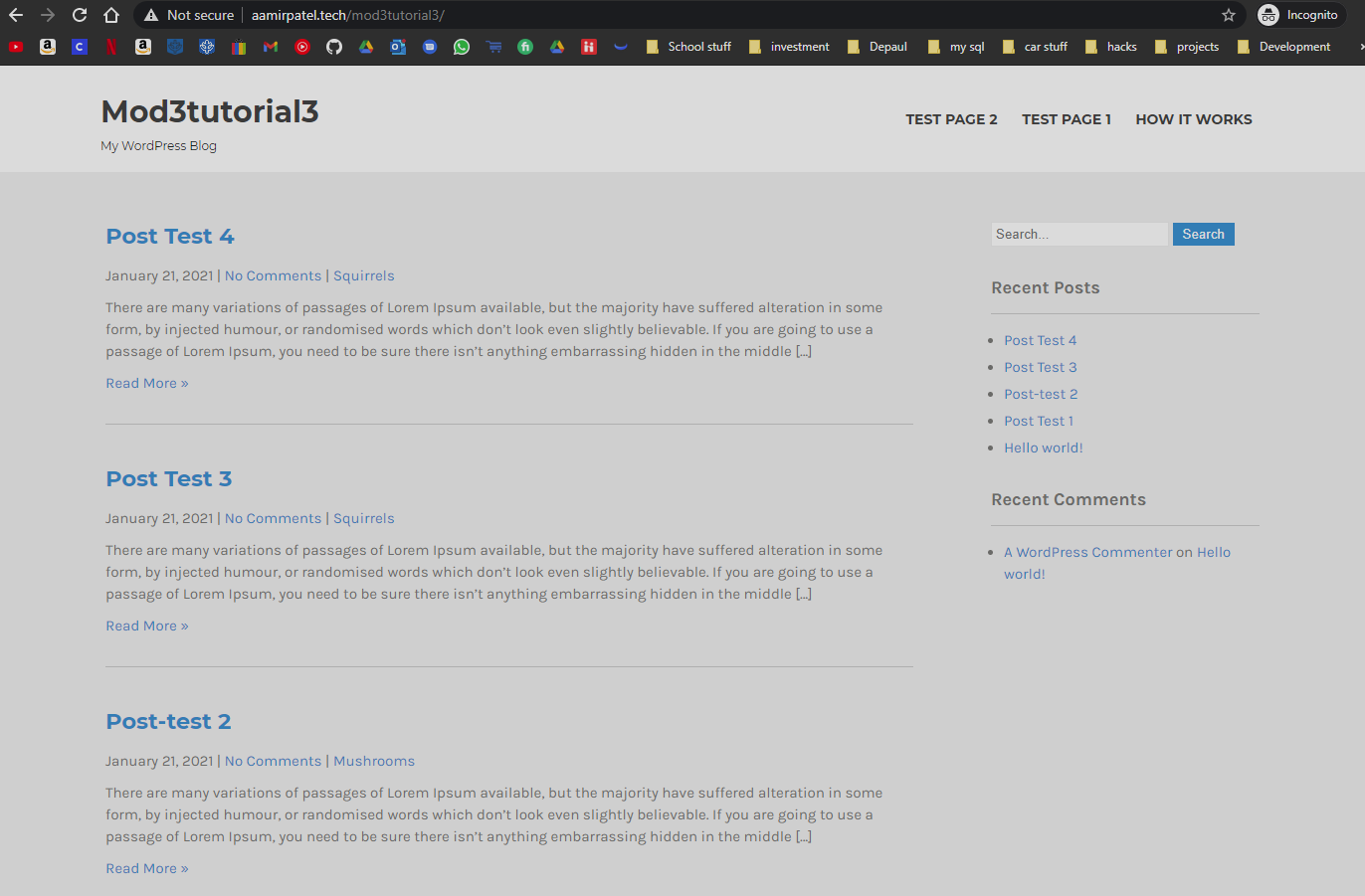
**Pages Created with Gutenberg Editor [F.2]**

1. Take screen captures of the backend administration WordPress site Dashboard > Plugins page showing the Classic Editor Plugin Deactivated and Deleted. **[F.2.1]  
   [ ]**
2. Take screen captures of the backend administration WordPress site Dashboard > Pages showing the list of Pages **[F.2.2]  
   [ ]**
3. From the Dashboard, select Pages. Select Edit for “Test Page 2” and take a screen capture of the Test Page Template **[F.2.3]  
   [ ]**
4. Take screen captures of the frontend client facing WordPress site showing the menu and   
   “How it works” menu item selected **[F.2.4]  
   [ ]**

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

**Part G – Extend WordPress with Post, Category, and Tag Content**

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

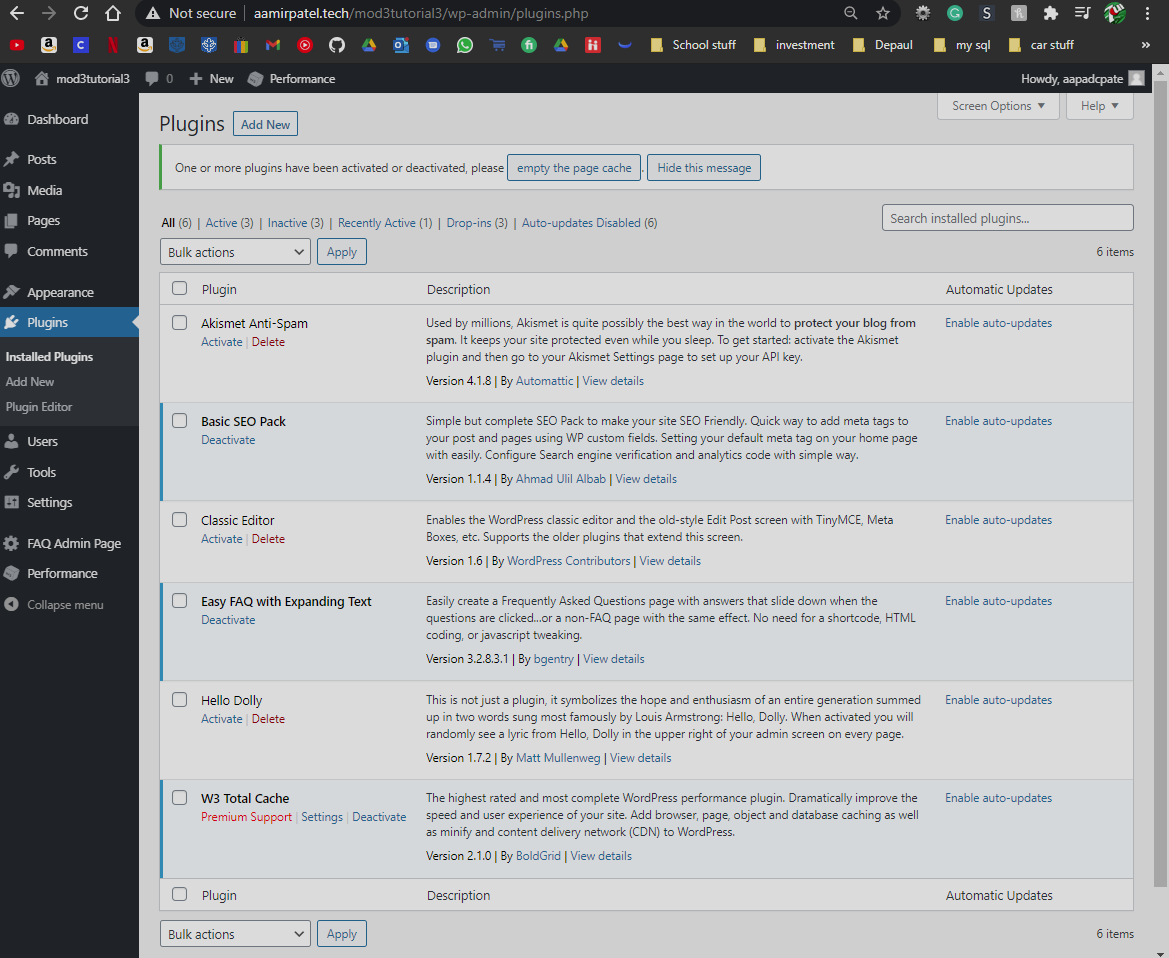
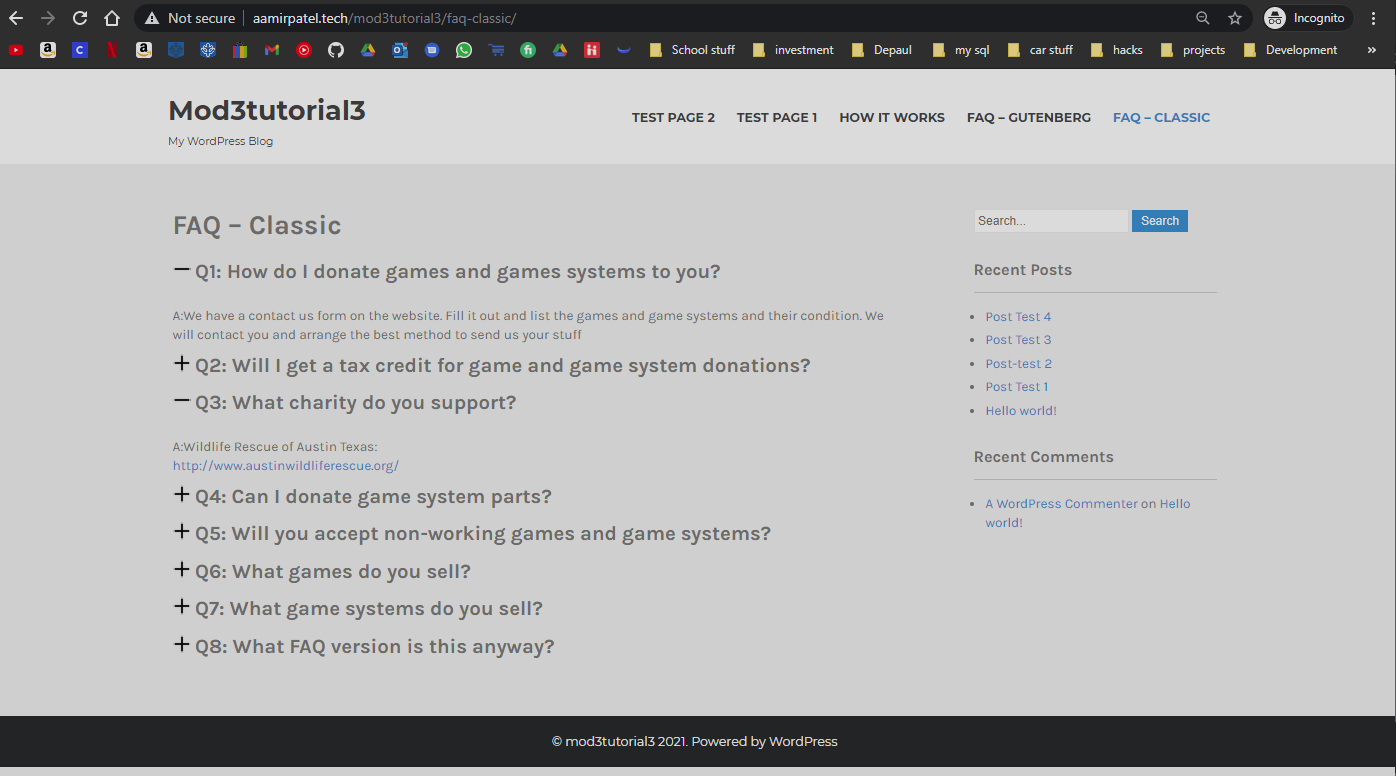
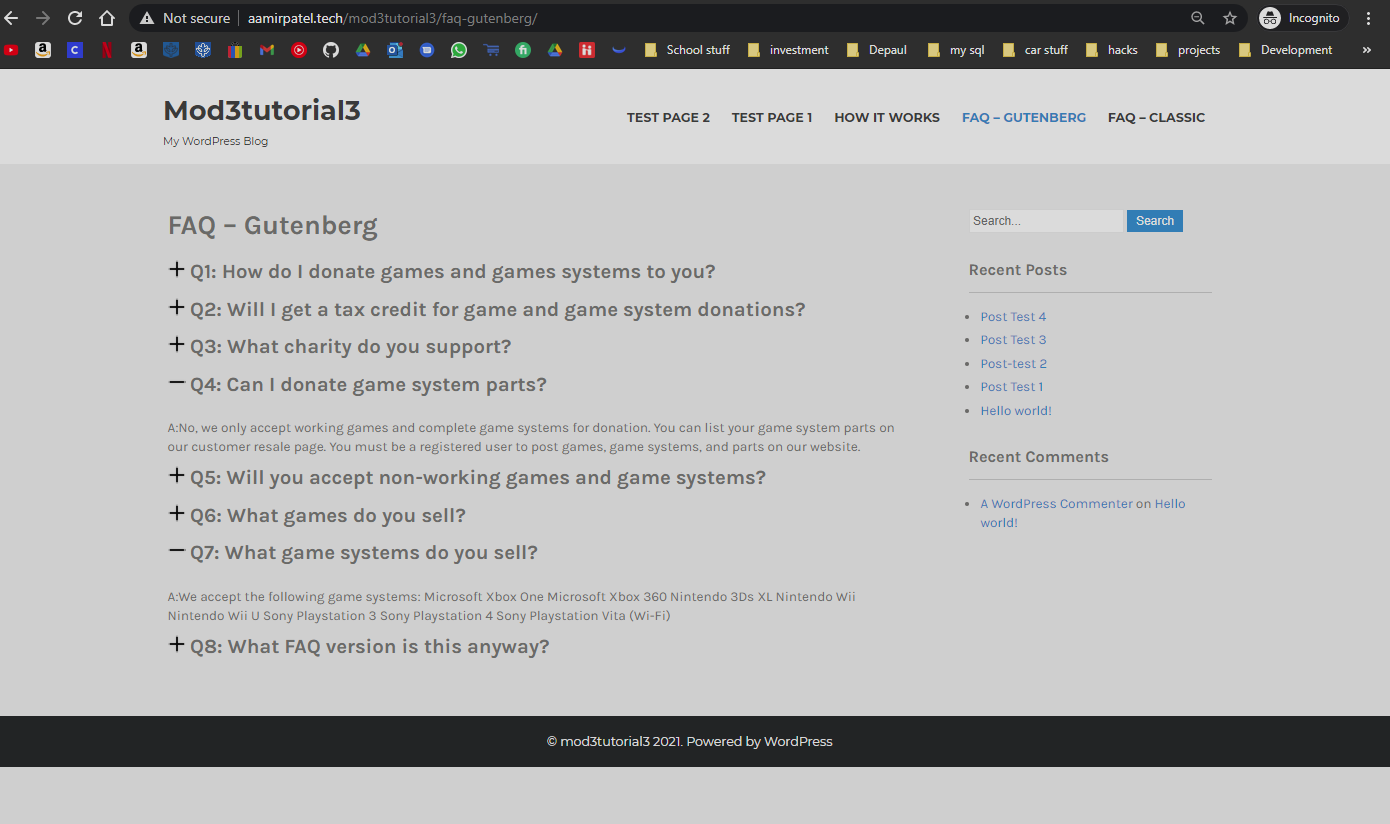
1. Take a **backend** administration screen captures of the Posts > Categories **[G.1]  
   [ ]**
2. Take a **backend** administration screen captures of the Posts > Tags **[G.2]  
   [ ]**
3. Take a **backend** administration screen captures of the Dashboard > All Posts **[G.3]  
   [ ]**
4. Take a screen capture of the **front-end** client facing Home page showing the Posts **[G.4]  
   [ ]**

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

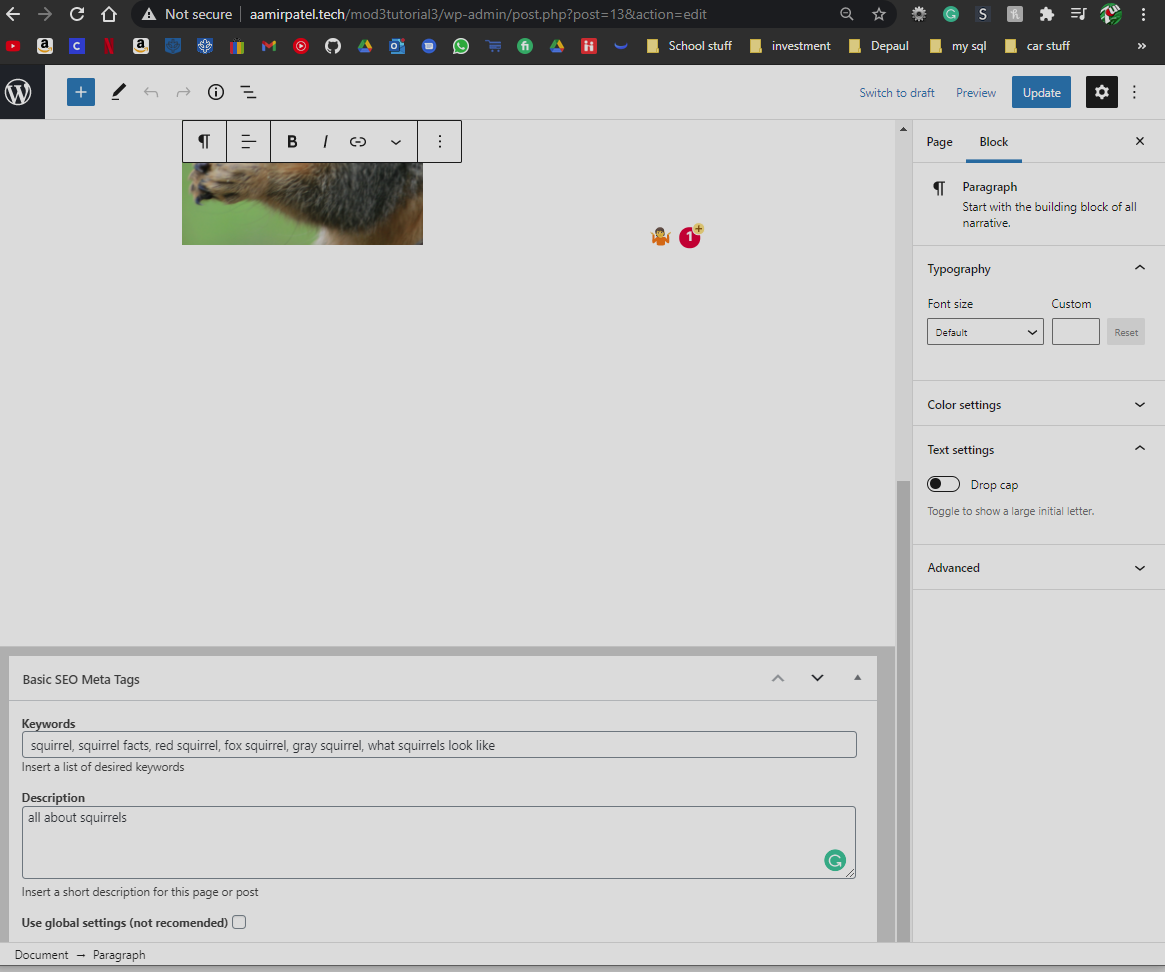
**Part H – Extend WordPress with Plugins**

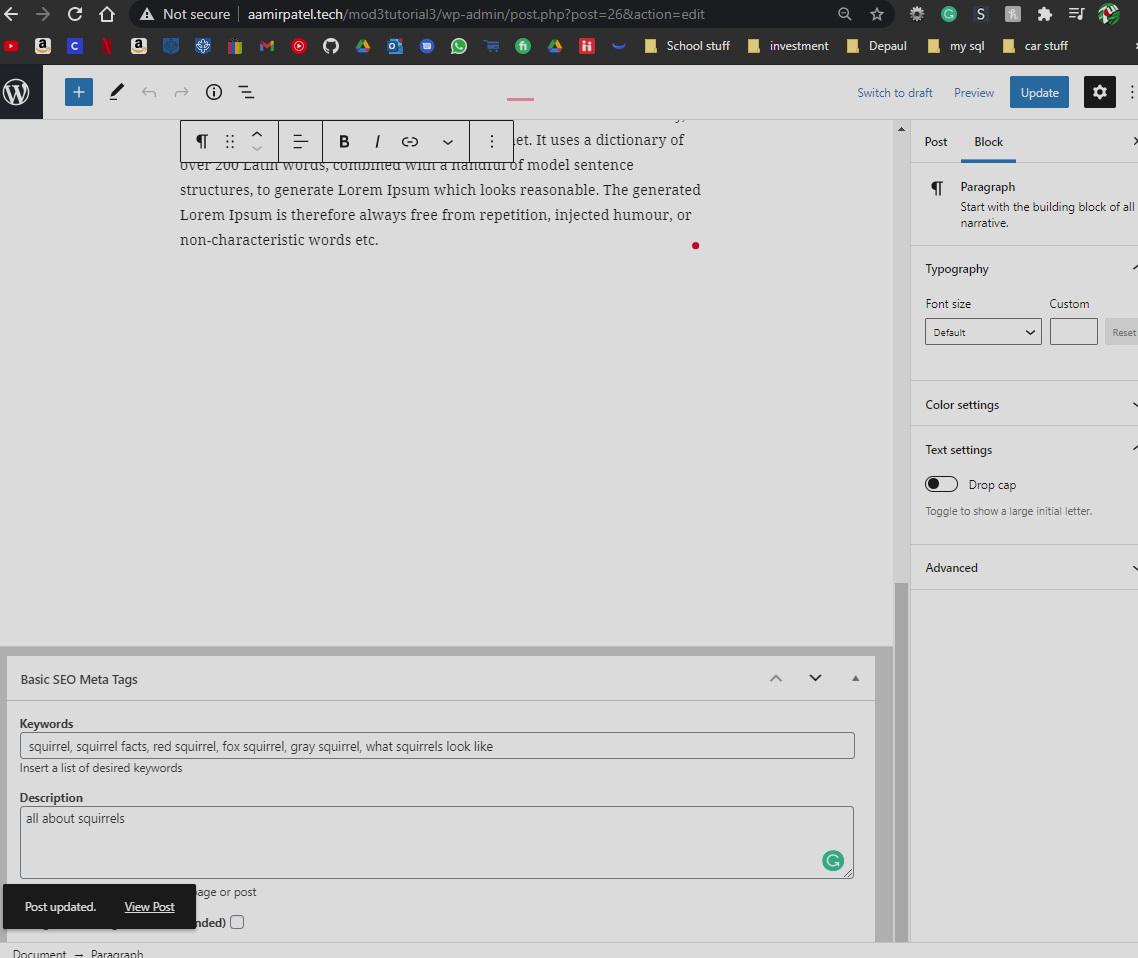
**Part H.1 – There are no screen captures for this part.**

**Part H.2 – FAQ Plugin on Classic and Gutenberg Editor  
Please make sure all Part H.2 screen capture include the web browser address bar. [H.2]**

1. Take a screen captures of the backend Plugins page with the Dashboard visible **[H.2.1]  
   [ ]**
2. Front-end screen capture of the FAQ-Classic page **[H.2.2]  
   [ ]**
3. Front-end screen capture of the FAQ-Gutenberg page **[H.2.3]  
   [ ]**

**Part H.3 – SEO on Gutenberg Editor**

1. On the WordPress Backend, edit one of the Pages and Posts that you added SEO keywords and descriptions to and take a screen capture of its Template where the SEO Keywords and descriptions are visible **[H.3.1]  
   [Page - **

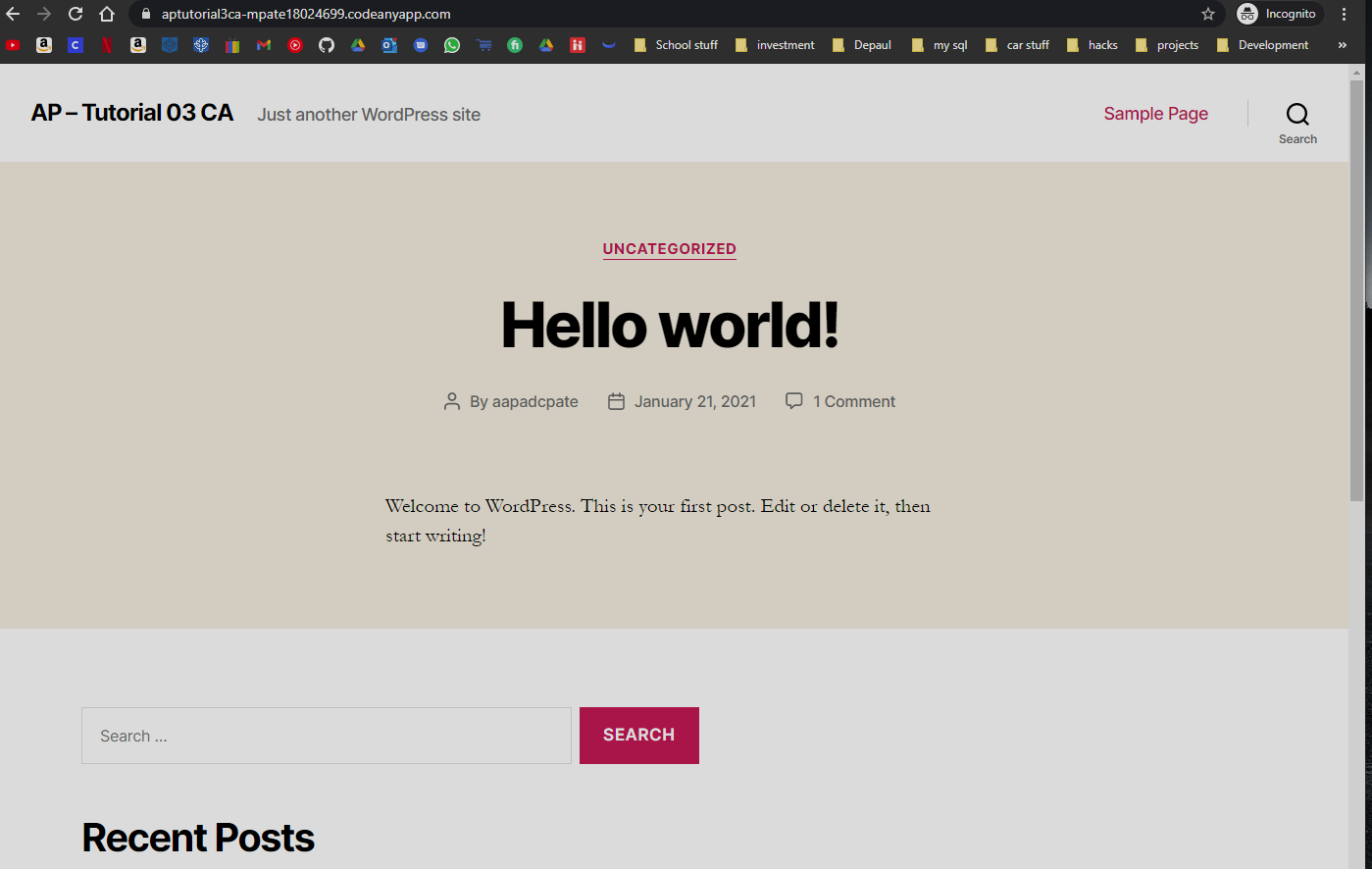
Post - **]**

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

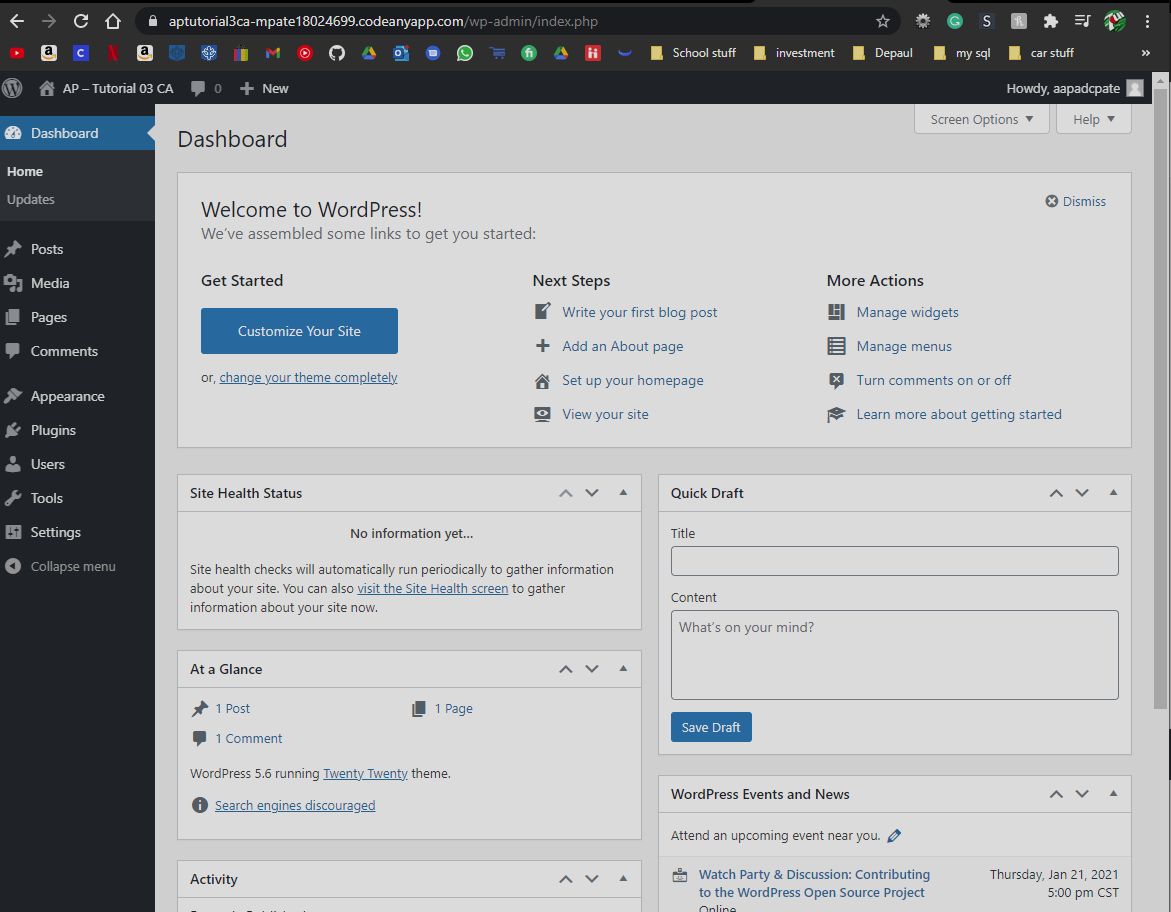
**Part I – Create a WordPress site on Codeanywhere**

1. Enter the following Codeanywhere web host WordPress site information **[I.1]**
   1. Codeanywhere account username(email): [mpate180@depaul.edu]
   2. Codeanywhere account password: [Kamboli1!]
   3. Site Title: [ AP-Tutorial 03 CA]
   4. Username: [ aapadcpate]
   5. Password: [ 9aIwR@LZB%#uk9Fgh3]

**Please make sure all I.2 screen capture includes the web browser address bar. [I.2]**

1. Take screen captures of the **front-end** client facing WordPress site **[I.2.1]  
   [ ]**
2. Take screen captures of the **backend** administrative WordPress site

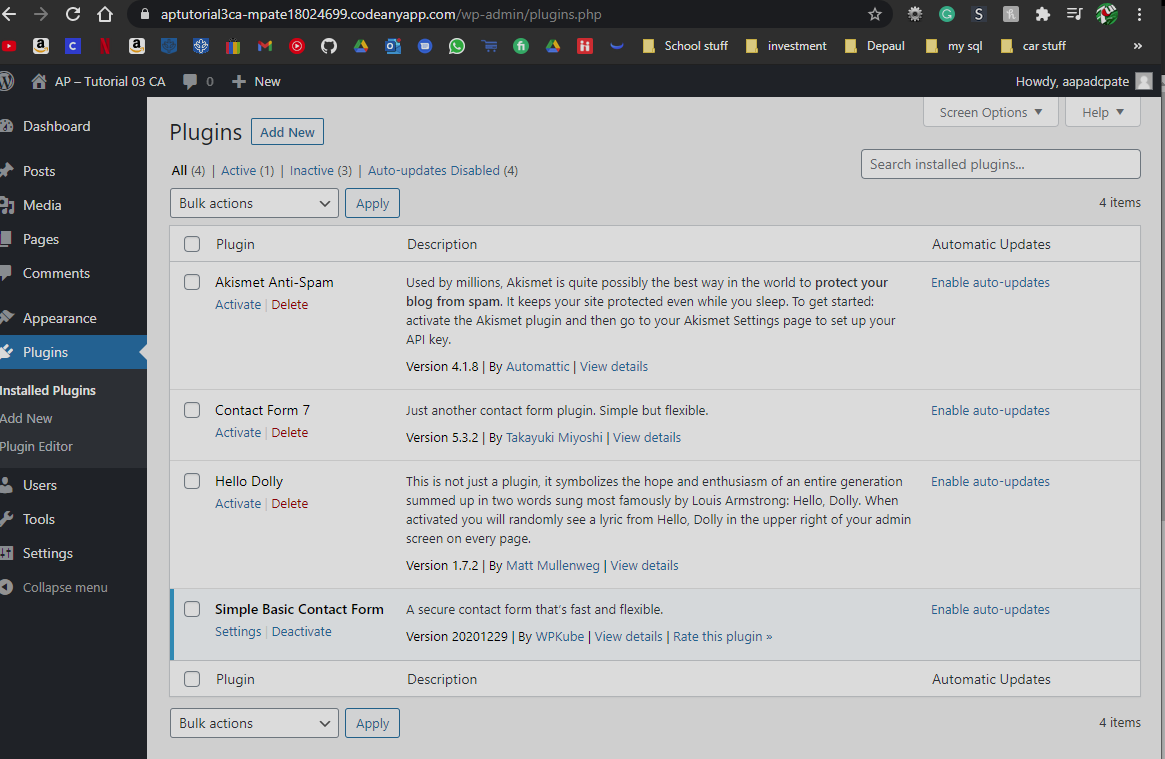
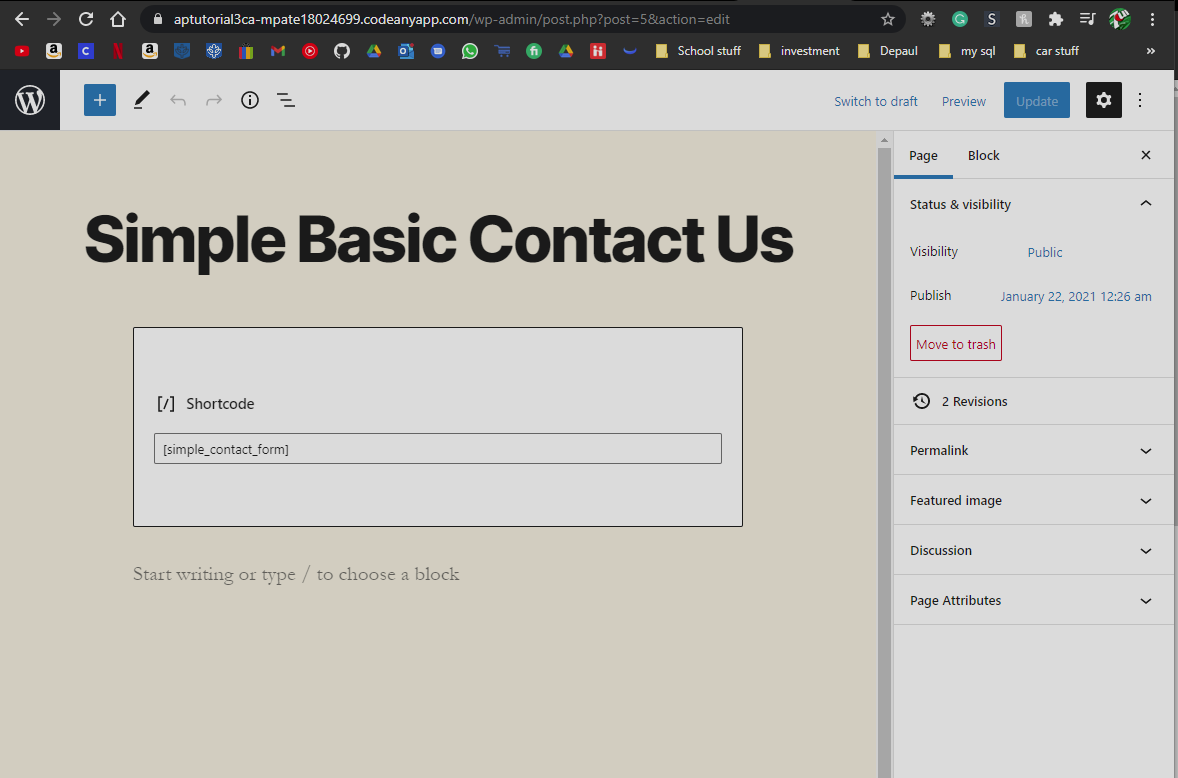
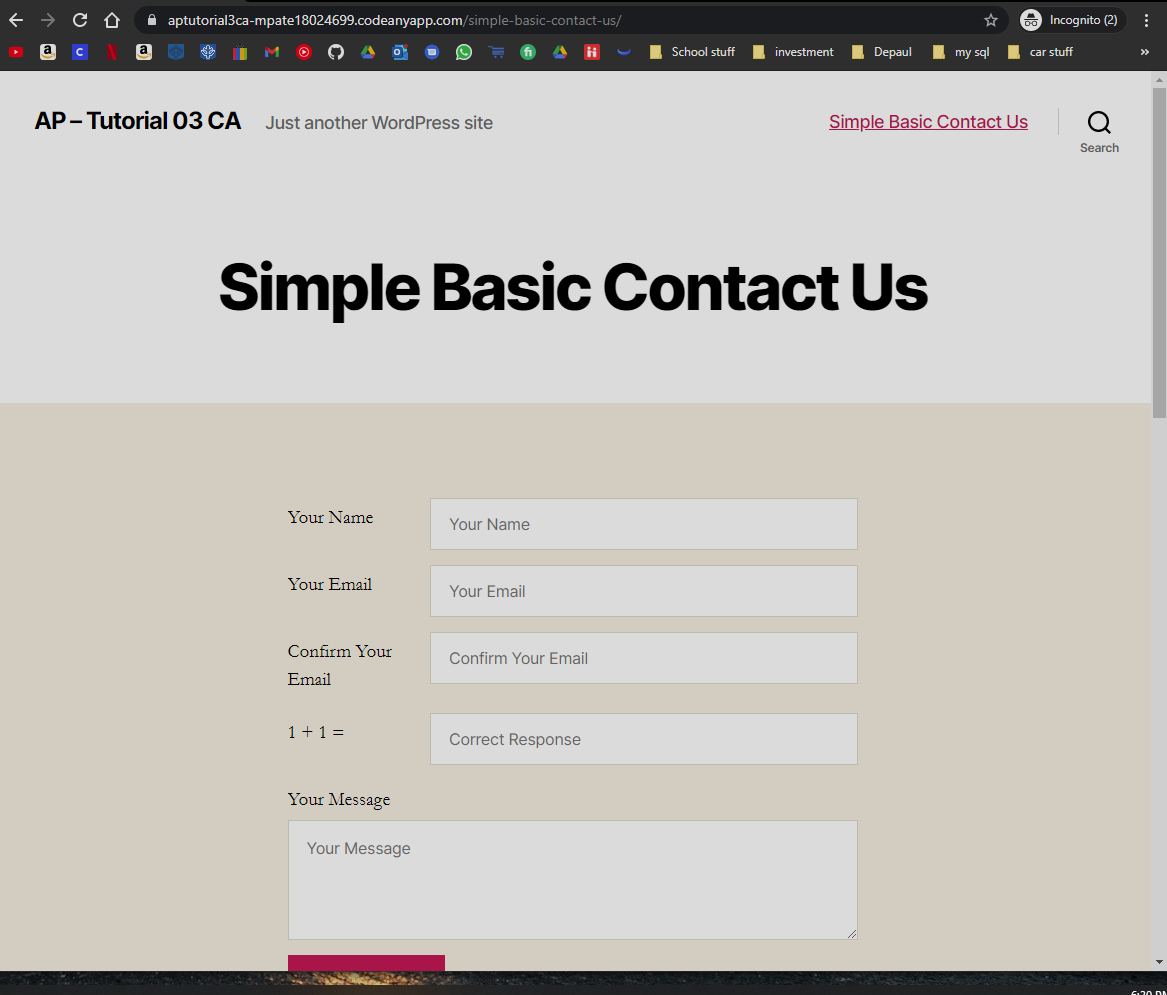
Dashboard > Home

With the web address bar visible and the “At a Glance” Home page section visible **[I.2.2]  
[ ]**

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

**Part J – Extend the Codeanywhere WordPress site with Forms**

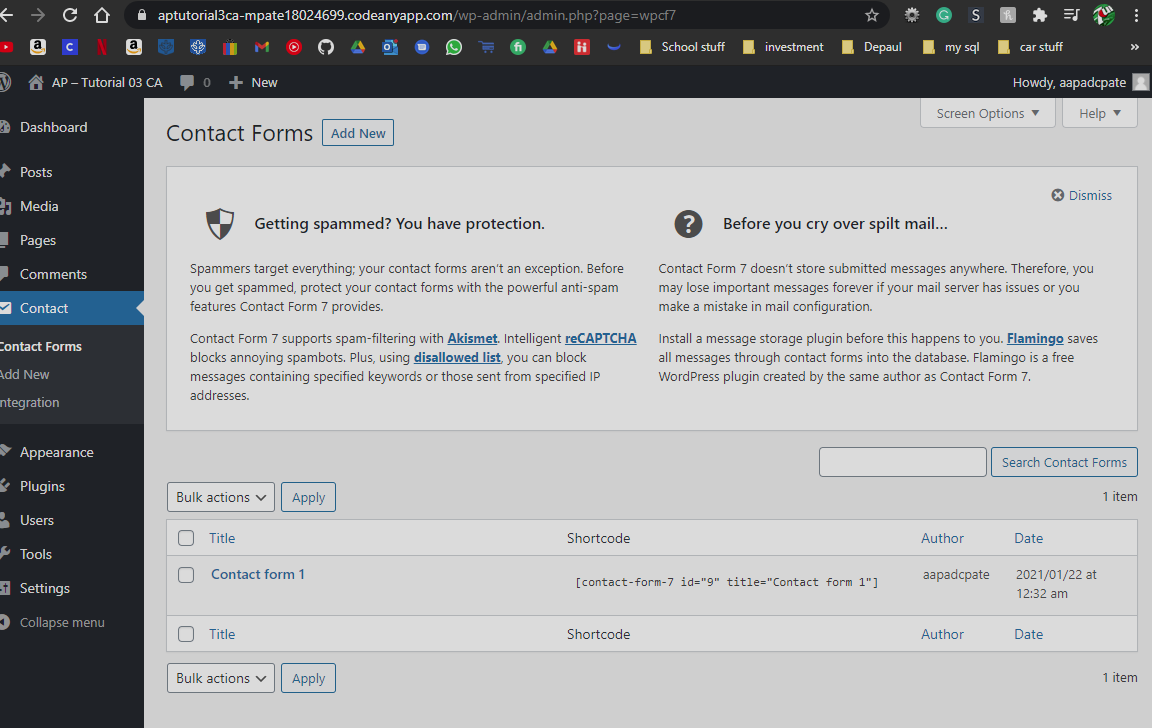
J.1 – Simple Basic Contact Form   
**Please make sure all J.1 screen capture includes the web browser address bar. [J.1]**

1. Take a backend administration screen captures of the Plugins **[J.1.1]  
   [ ]**
2. Take a backend administration screen captures of the Simple Basic Contact Page Template **[J.1.2]  
   [ ]**
3. Take a screen capture of the front-end client facing Home page showing the Contact Us form displayed **[J.1.3]  
   [ ]**

**Go back to the instructions and continue at Part J.2.**

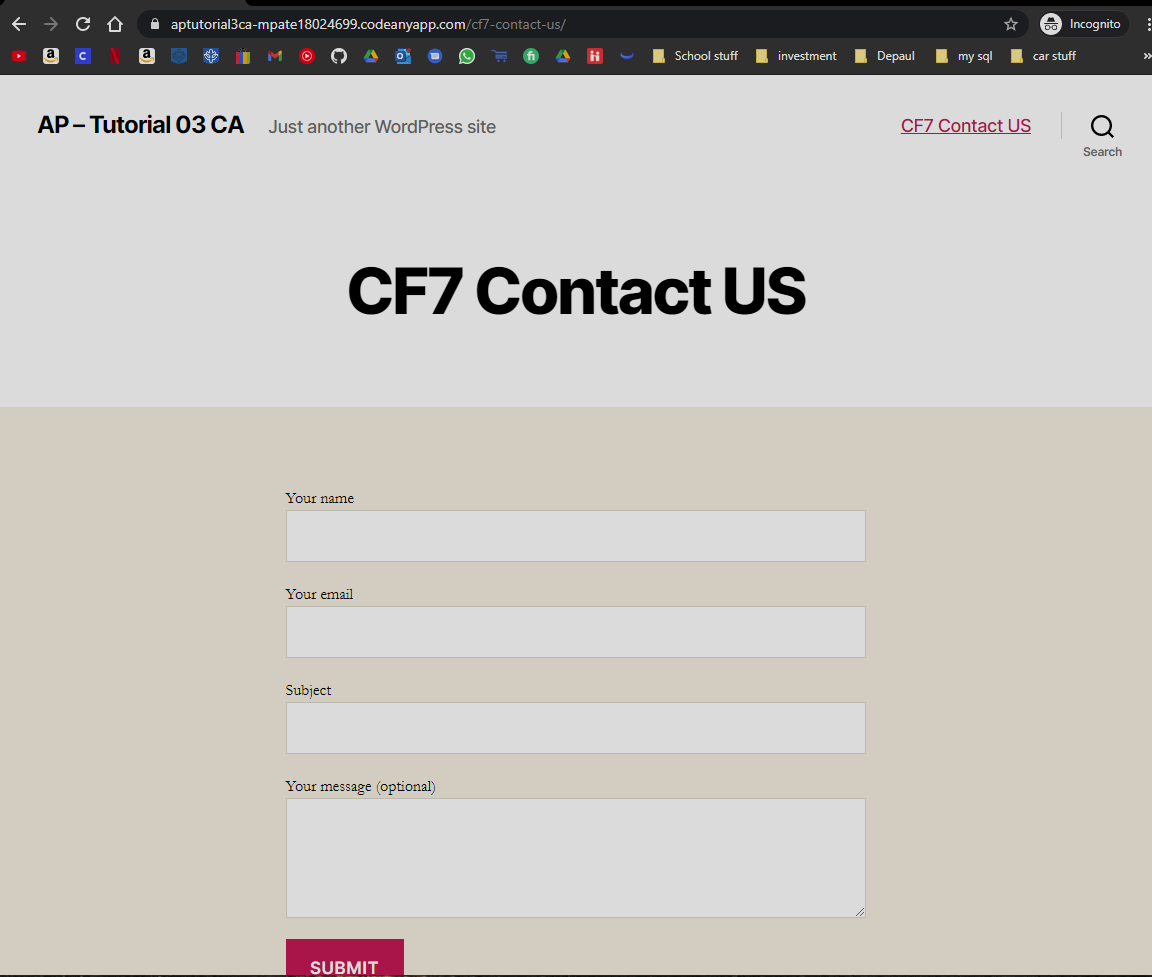
J.2 – Contact Form 7 + Honeypot Plugins **Please make sure all J.2 screen capture includes the web browser address bar. [J.2]**

1. Take a backend administration screen captures of the Plugins page list **[J.2.1]  
   [ Paste screen capture here]**
2. Take a backend administration screen captures of the Contact Form 7 list page, by selecting [Dashboard > Contact > Contact Forms] **[J.2.2]  
   [ ]**
3. Take a backend administration screen captures of the Contact Form 7 list page, by selecting [Dashboard > Contact, then Select Edit] **[J.2.3]  
   [**

****

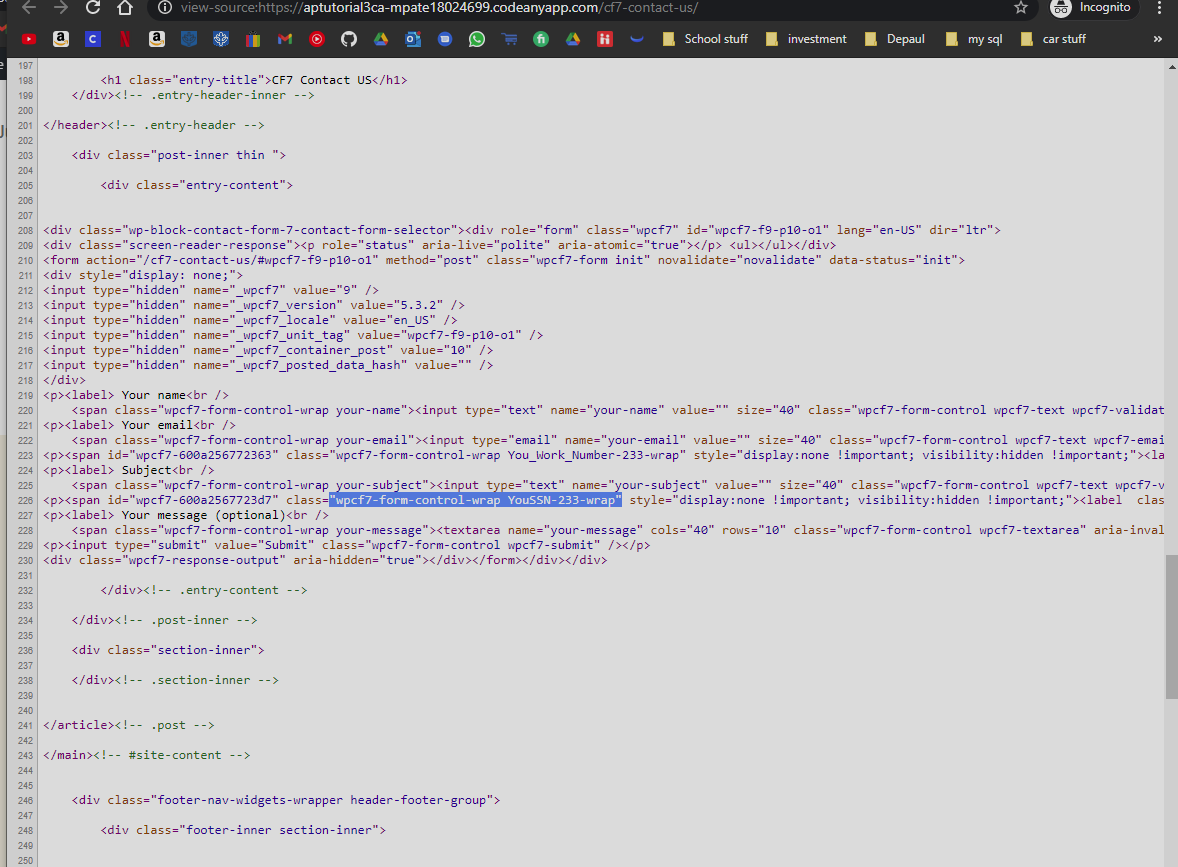
**]**

1. Take a screen capture of the front-end client facing Home page showing the Contact Form 7 form displayed. **[J.2.4]  
   [**

****

**]**

1. Take a screen capture of the front-end client facing Contact Form 7 form “view source” – where the honeypot fields you added are visible **[J.2.5]  
   [**

**]**

J.3 – **Contact Form Plugin Feature Analysis**There are no screen captures for this part.   
Please return to the instructions and continue with Part K.

**Part K – PHP QUESTIONS**

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

* Online PHP test sites:
  + <http://phptester.net>
  + <http://www.writephponline.com/>
  + <http://sandbox.onlinephpfunctions.com/>
  + <https://wtools.io/php-sandbox>
* 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 does the following:
   1. Creates 2 variables $a and $b, with values 1 and 5
   2. Write an if statement that compares the two values using a less than comparison operator
   3. If $a is less than $b, echo “that the first variable is less than the second variable”
   4. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste 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. |
|  |

1. Write and test PHP code that does the following:
2. Creates 2 variables $x and $y, with values 1 and 5
3. Write an if statement that compares the two values to check if they are equal
4. If $x is equal to $y, echo “these two variables are equal” else, if they are not equal, echo “They are not equal”.
5. **Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy 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. |
|  |

1. Write and test PHP code that does the following:
   1. Creates 2 variables $x and $y, with values 1 and 5
   2. Write a PHP function named **compareTwoVars** that has two entry arguments $a and $b
   3. In the function body, write PHP code that compares $a and $b to check if they are equal
   4. If $a is equal to $b, echo “These two variables are equal”, else if they are not equal, then echo “They are not equal”
   5. Call the function with the two variables $x and $y

**Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste 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. |
|  |

1. Write and test PHP code that does the following:
2. Creates 2 variables, with values 10, 12.
3. Write 5 individual PHP statements that compares the 2 variables using the following comparison operators
   * 1. Less Than, <
     2. Less than or Equal to, <=
     3. Greater than, >
     4. Greater than or Equal to, >=
     5. Not Equal to, !=
4. If the comparison value is TRUE, echo a string statement that describes the results of the comparison.
5. Display the $result value using an echo statement.

**Test your code using an online PHP emulator and paste the code and results in the table cell below.**

|  |
| --- |
| Copy and paste 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. |
|  |

1. Write and execute a PHP function, without entry arguments that echo’s a string value “Hi” in its function body. Write PHP code to call/execute the function. Please test your code in a PHP web emulator.

|  |
| --- |
| Copy and paste 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. |
|  |

1. Create an array variable named $mydogs with the following dogs names as strings “spot”, “rover”, “max”, “dot”, “bandit”. After creating the array, write code to iterate the array using a **for** statement and echo out each array item. Test your code in a PHP web emulator.

|  |
| --- |
| Copy and paste 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. |
|  |

1. Create an array variable named $mycats with the following cats names as strings “midnight”, “fritz”, “bill”, “amber”, “happy”. After creating the array, write code to iterate the array using a **foreach** statement and echo out each array item. Test your code in a PHP web emulator.

|  |
| --- |
| Copy and paste 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. |
|  |

1. Create an array variable named $mymixedarray with 10 total items of both integer, and string values. After creating the array, write code to iterate the array using a **foreach** statement and echo out each array item. Test your code in a PHP web emulator.

|  |
| --- |
| Copy and paste 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. |
|  |

1. Create an associative array named $fruitarray with 3 total array items with key and values of 1 for Apple, 2 for Orange, 3 for Grape. After creating the array, write code to iterate the array using a **foreach** statement and echo out each array item. Test your code in a PHP web emulator.

|  |
| --- |
| Copy and paste the code block and record the results 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. |
|  |

**Go back to the instructions and work on Part L.**