

Milestone 5: Unit Testing

ConnectCU

Tyler Moore, Braden Holmes, Tanya Leung, Javier Ramirez, Nestor Vargas

Vision Statement: To provide a better learning experience for students and give them a place to get help that is free and reliable.

Automated Test Cases. We used PHPUnit to test the backend of our website.

Feature	Data Requirement	Expected Result	Pass/Fail	Comments
1. Connection established between user interface and website database (versus stored locally)	None. .php and .sql files.	Set up database functionality on the web page to start recording data from the users. When the connection is correctly established, we will be able to save the questions and answers for all subjects. Account creation for students is also important, to keep record of users and passwords.	Pass	We were able to check that the database is hooked up to the website.
2. User question input is functional and correctly places the content into the database.	User input i.e. question, question subject, and username Feature 1 has to be ready and passed with expected outputs.	Correct recording of any question being asked on the different subjects by any user. Save the input given and display it on the website for every other student using it. Interaction between users will be possible.	Pass	Testing was successful. We were able to input questions into different subjects. They were inserted correctly into the database.
3. Question should be able to be deleted by the administrators	.php and .sql files. Feature 1 has to be ready and passed with expected outputs.	Functionality to erase any record saved on the database and being able to stop the display of that record in the webpage for every student.	Pass	We were able to see the database results being deleted when we wanted them to.

```
tyler@tyler-VirtualBox: ~/Dropbox/software-dev/project3/web/php
tyler@tyler-VirtualBox:~/Dropbox/software-dev/project3/web/php$ phpunit testfunc
tions.php
PHPUnit 5.1.3 by Sebastian Bergmann and contributors.

...                                                                    3 / 3 (100%)

Time: 34 ms, Memory: 4.00Mb

OK (3 tests, 3 assertions)
tyler@tyler-VirtualBox:~/Dropbox/software-dev/project3/web/php$
```

Our test cases' code

	testfunctions.php	functions.php
1	<code><?php</code>	
2		
3	<code>require_once('functions.php');</code>	
4	<code>//require_once('delete.php');</code>	
5	<code>//require_once('questionHandler.php');</code>	
6	<code>require_once('PHPUnit/Framework/TestCase.php');</code>	
7		
8	<code>class TestFunctions extends PHPUnit_Framework_TestCase</code>	
9	<code>{</code>	
10	<code>function test1()</code>	
11	<code>{</code>	
12	<code> \$found = sqlConnect();</code>	
13	<code> \$this->assertTrue(\$found != null);</code>	
14	<code>}</code>	
15		
16	<code>function test_delete()</code>	
17	<code>{</code>	
18	<code> \$this->assertTrue(delete());</code>	
19	<code>}</code>	
20		
21	<code>function test_add()</code>	
22	<code>{</code>	
23	<code> \$this->assertTrue(add('Math', 'Whats 2 + 2?', 'Bob'));</code>	
24	<code>}</code>	
25	<code>}</code>	
26		
27	<code>?></code>	
28		

User Acceptance Test Plan

Feature	Data Requirement	Expected Result	Pass/Fail	Comments
1. User input adds to the website interface correctly	User input i.e. question, question subject, comments/answers.	That all the questions and answers added by the users can be recorded into the database in order to be seen by every other student using the web page on the correct subject category, displaying the name of the input author.	Pass	User input is correctly inputted into database and is added correctly to website interface
2. Users are able to sign up/log-in and post questions & comments with their accounts	UAT Feature 1 correct functionality. .php and .sql files connection to obtain data. Profile data, creation of a username and a password.	Connection with Feature 1. Correct recording of profiles (username and password) into database.	Fail	The requirement of feature 1 is already working. Database setup, functionality still in progress
3. Users will be able to vote on the questions shown on the website. The questions will be sorted from highest to lowest on the page.	.php and .sql files to sort the voting system. Users clicking the up/down arrow	The highest voted question will be at the top of the page. Clicking the up or down arrow will change the score stored in the database and update appropriately.	Fail	We are still working on the voting algorithm to make it work.