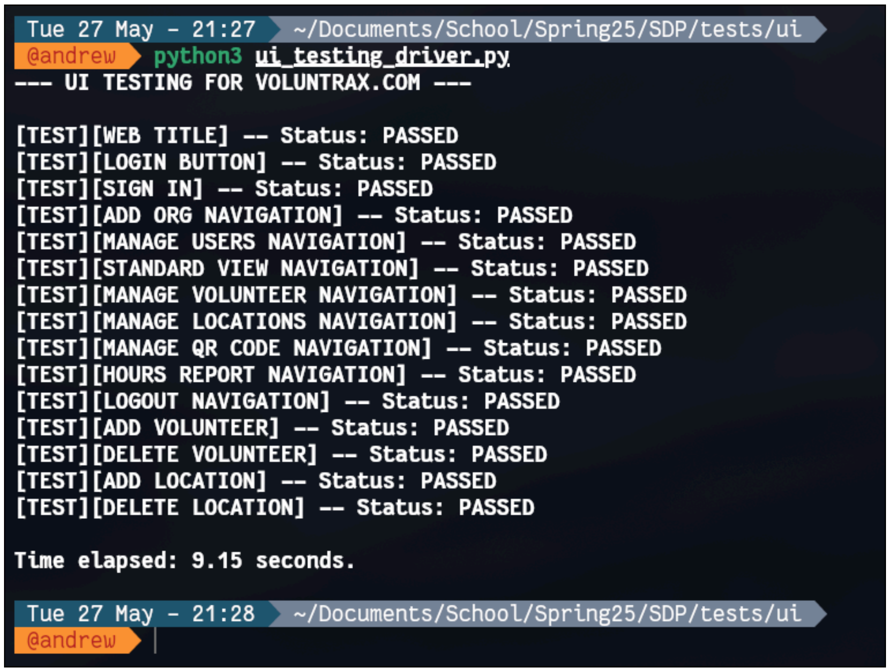
Code & Conquer
Software Development Principles
Spring 2025

Software Testing for Voluntrax

UI Testing

Selenium based UI testing results:

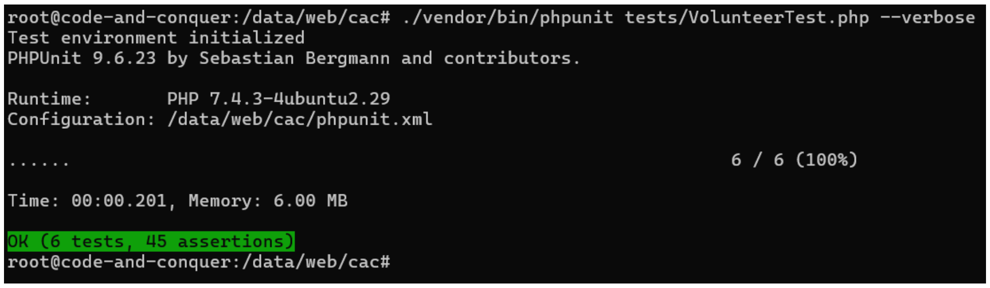


```
 Tue 27 May - 21:27  ~/Documents/School/Spring25/SDP/tests/ui
 @andrew  python3 ui_testing_driver.py
--- UI TESTING FOR VOLUNTRAX.COM ---

[TEST][WEB TITLE] -- Status: PASSED
[TEST][LOGIN BUTTON] -- Status: PASSED
[TEST][SIGN IN] -- Status: PASSED
[TEST][ADD ORG NAVIGATION] -- Status: PASSED
[TEST][MANAGE USERS NAVIGATION] -- Status: PASSED
[TEST][STANDARD VIEW NAVIGATION] -- Status: PASSED
[TEST][MANAGE VOLUNTEER NAVIGATION] -- Status: PASSED
[TEST][MANAGE LOCATIONS NAVIGATION] -- Status: PASSED
[TEST][MANAGE QR CODE NAVIGATION] -- Status: PASSED
[TEST][HOURS REPORT NAVIGATION] -- Status: PASSED
[TEST][LOGOUT NAVIGATION] -- Status: PASSED
[TEST][ADD VOLUNTEER] -- Status: PASSED
[TEST][DELETE VOLUNTEER] -- Status: PASSED
[TEST][ADD LOCATION] -- Status: PASSED
[TEST][DELETE LOCATION] -- Status: PASSED

Time elapsed: 9.15 seconds.

 Tue 27 May - 21:28  ~/Documents/School/Spring25/SDP/tests/ui
 @andrew 
```

The UI testing was done using a framework called Selenium. The tests were implemented using python. After importing the library, I could easily connect to our website and manipulate the frontend using code inorder to test its functionality. The tests include navigation through the website, login/logout, volunteer adding/deletion and location adding/deletion. While these tests are not completely extensive, they can be easily extended to test more features of the website.

Unit Testing

PHP-Unit testing results:

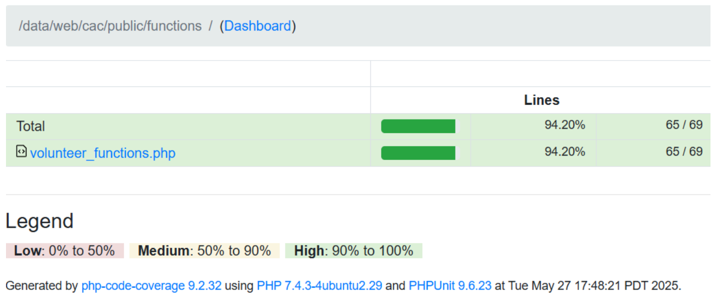


```
root@code-and-conquer:/data/web/cac# ./vendor/bin/phpunit tests/VolunteerTest.php --verbose
Test environment initialized
PHPUnit 9.6.23 by Sebastian Bergmann and contributors.

Runtime:       PHP 7.4.3-4ubuntu2.29
Configuration: /data/web/cac/phpunit.xml

......                                                          6 / 6 (100%)

Time: 00:00.201, Memory: 6.00 MB

OK (6 tests, 45 assertions)
root@code-and-conquer:/data/web/cac#
```

The unit testing was done utilizing the existing PHP-Unit testing tools available as packages on linux. Once installed and configured on the server, we were able to create a simple test file that runs a series of six tests. The tests include things like attempting to create records in the database, both formatted correctly and incorrectly, and making sure that the code handles things appropriately. There are a total of 45 assertions in the unit testing file, and currently all 45 are successful. (See screenshot above)

Code Coverage Testing

Code coverage test results from xdebug:



/data/web/cac/public/functions / (Dashboard)

	Lines		
Total	🟩	94.20%	65 / 69
volunteer_functions.php	🟩	94.20%	65 / 69

Legend

Low: 0% to 50% **Medium**: 50% to 90% **High**: 90% to 100%

Generated by php-code-coverage 9.2.32 using PHP 7.4.3-4ubuntu2.29 and PHPUnit 9.6.23 at Tue May 27 17:48:21 PDT 2025.

The code coverage testing was done leveraging the xdebug php module. We created a business functions file that simulates various system actions. The code and tests are reviewed by the xdebug plugin, and a score is given (see screenshot above.)

–*Both the Code Coverage and Unit Tests are automated and run hourly, posting reports to a web accessible link.*

Alpha Testing

Alpha testing results based on the required user stories: