

report.html

Report generated on 31-Oct-2020 at 18:45:00 by [pytest-html](#) v2.1.1

Environment

Packages {"pluggy": "0.13.1", "py": "1.9.0", "pytest": "6.1.1"}
Platform Windows-10-10.0.19041-SP0
Plugins {"flask": "1.0.0", "html": "2.1.1", "metadata": "1.10.0"}
Python 3.8.2

Summary

15 tests ran in 2.88 seconds.

15 passed, 0 skipped, 0 failed, 0 errors, 0 expected failures, 0 unexpected passes

Results

[Show all details](#) / [Hide all details](#)

vvv Result	vvv Test	vvv Duration	Links
Passed	tests.py::test_pytest_functional	0.00	
-----Captured log call-----			
INFO root:tests.py:19 Initializing PyTest module and ensuring its functionality			
Passed	tests.py::test_db_initialization	0.17	
-----Captured log call-----			
INFO root:tests.py:60 Test DB Initialization: INFO root:tests.py:61 This test ensures the Database Library can create a new database and that it can load an existing one INFO root:tests.py:62 Expecting active sqlite3 connection and cursor handlers are returned by this test INFO root:tests.py:23 Creating empty testing database			
Passed	tests.py::test_user_insertion_and_search	0.22	
-----Captured log call-----			
INFO root:tests.py:69 Test User Insertion And Search: INFO root:tests.py:70 This test ensures that users can be inserted into and searched for in a database INFO root:tests.py:71 Expecting successful function return from creating user (testuser, test_password) INFO root:tests.py:72 Expecting a list cnotaining a correctly initialized single user and None intialized pacemaker parameters from search for user (testuser, test_password) INFO root:tests.py:23 Creating empty testing database			
Passed	tests.py::test_get_user	0.22	
-----Captured log call-----			
INFO root:tests.py:83 Test Get User: INFO root:tests.py:84 This test ensure that users can be found in the database when searched for by their known unique ID INFO root:tests.py:85 Expecting a list containing the same user as previously created (1, testuser, test_password) when searching for users with unique ID of 1 INFO root:tests.py:23 Creating empty testing database			
Passed	tests.py::test_get_rows	0.61	
-----Captured log call-----			
INFO root:tests.py:96 Test Get Rows: INFO root:tests.py:97 This test ensures the database has the correct number of rows and is capable of returning that number INFO root:tests.py:98 Expecting 5 rows returned after creating a new db and inserting exactly 5 users INFO root:tests.py:29 Creating partially populated testing database			
Passed	tests.py::test_parameter_update	0.72	
-----Captured log call-----			
INFO root:tests.py:104 Test Parameter Update: INFO root:tests.py:105 This test ensures previously created users can have their parameters updated successfully INFO root:tests.py:106 Expecting a user with updated parameters (67, 43, 44, 45, 87, 56, 89, 90) after creating new user and changing their parameters to (67, 43, 44, 45, 87, 56, 89, 90) INFO root:tests.py:29 Creating partially populated testing database			
Passed	tests.py::test_config_initializes	0.00	
-----Captured log call-----			
INFO root:tests.py:115 Test Config Initializes: INFO root:tests.py:116 This test ensures the config module can create a config handler containing the correct information when given a config file with known contents INFO root:tests.py:117 Expecting logger.level == DEBUG && db.local-uri == db_test_name_variable INFO root:tests.py:52 Overriding default config with test values			
Passed	tests.py::test_user_initializes	0.14	

vvv Result	vvv Test	vvv Duration	Links
-----Captured log call-----			
INFO root:tests.py:125 Test User Initializes: INFO root:tests.py:126 This test ensures a user can be created, and that the created user properly initializes their dependencies (database) INFO root:tests.py:127 Expecting a user containing active connection and cursor handlers to a database matching the test_config specifications INFO root:tests.py:52 Overriding default config with test values INFO root:tests.py:23 Creating empty testing database			
Passed	tests.py::test_login_acessable	0.60	
-----Captured log call-----			
INFO root:tests.py:139 Test Login Accessable: INFO root:tests.py:140 This test ensures that the login state is acessable when the DCM is first created INFO root:tests.py:141 Expecting a rendered login page INFO root:tests.py:52 Overriding default config with test values INFO root:tests.py:29 Creating partially populated testing database			
Passed	tests.py::test_login_populated	0.00	
-----Captured log call-----			
INFO root:tests.py:156 Test Login Populated: INFO root:tests.py:157 This test checks that the rendered login state has rendered the expected content INFO root:tests.py:158 Expecting a rendered login page matching the home directory index.html			
Passed	tests.py::test_login_correct	0.01	
-----Captured log call-----			
INFO root:tests.py:165 Test Login Correct: INFO root:tests.py:166 This test checks that when a login attempt is made using the correct username and password that DCM access is granted INFO root:tests.py:167 Expecting state change to logged in and the user specific page is rendered and matching that of user.html			
Passed	tests.py::test_login_incorrect	0.00	
-----Captured log call-----			
INFO root:tests.py:174 Test Login Incorrect: INFO root:tests.py:175 This test checks that when a login attempt is made using incorrect parameters that login is denied and the user is informed INFO root:tests.py:176 Expecting no state change and the user is flashed with an Invalid Credentials message			
Passed	tests.py::test_logout	0.00	
-----Captured log call-----			
INFO root:tests.py:183 Test Logout: INFO root:tests.py:184 This test ensures that if a logged in user attempts to logout that the logout is handled correctly and they are actually logged out INFO root:tests.py:185 Expecting a state change to home and the home page is rendered and matching that of index.html INFO root:tests.py:186 Expecting the user is flashed with the correct logout message			
Passed	tests.py::test_user_route_blocked	0.00	
-----Captured log call-----			
INFO root:tests.py:194 Test User Route Blocked: INFO root:tests.py:195 This test checks that when no user is logged in that all user specific pages are blocked and access to a users data is denied INFO root:tests.py:196 Expecting no state change and the user is flashed with a You Need To Login First message			
Passed	tests.py::test_user_pages_acessable	0.01	
-----Captured log call-----			
INFO root:tests.py:203 Test User Pages Accessible: INFO root:tests.py:204 This test ensures that when a user is logged in they are able to access every user specific page and their parameters INFO root:tests.py:205 Expecting all state change attempts accepted and all user specific pages rendered on request			