|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** | **Outcome** |
| 1.0 | Selenium -Automate to run using driver.get() | driver.get('http://localhost:8000/projects/')  Using chrome drive | Pass | Pass –  (test\_testcase.py) |
| 1.1 | Selenium-Automate to both firefox and chrome at the same time | if request.param == "chrome":  web\_driver = webdriver.Chrome(executable\_path = chrome\_path)  #if request.param == "firefox":  # web\_driver = webdriver.Firefox(executable\_path = firefox\_path)  request.cls.driver = web\_driver  yield  web\_driver.close() | Pass | 1 fail, 1 pass |
| 1.2 | Selenium – Auto login to admin | Using fake email & Password  def admin\_login():  driver = webdriver.Chrome(executable\_path = chrome\_path)  driver.get("http://localhost:8000/admin/")  driver.find\_element\_by\_id("email").send\_keys("abc@hotmail.com")  driver.find\_element\_by\_id("pass").send\_keys("009090")  driver.find\_element\_by\_type("submit").click() | Fail – require to prompt for correct username and password | Fail. Will pass if correct details has been set |
| 1.3 | Selenium -Automate to run using driver.get()  Accesing 3 different pages at the same time | Automatically run 3 webpages consecutively – admin, projects, blog  driver = webdriver.Chrome(executable\_path = chrome\_path)  driver.get("http://localhost:8000/admin/")  driver2 = webdriver.Chrome(executable\_path = chrome\_path)  driver2.get("http://localhost:8000/projects/")  driver3 = webdriver.Chrome(executable\_path = chrome\_path)  driver3.get("http://localhost:8000/blog/") | Run consecutively without any interruption | Pass |
| 1.4 | Selenium -Automated with timer | driver.implicitly\_wait(10) # seconds  driver.get("http://localhost:8000/projects")  myDynamicElement = driver.find\_element\_by\_id("myDynamicElement") | Run aft 10 secs | Pass |
| 2.0 | Navigate from one page to another | Navigating  Localhost:8000/admin/  Localhost:8000/projects/  Localhost:/blog/ | Pass | Pass  -Prompt for login when accessing the admin site |
| 2.1 | Buttons | Clicking all functional buttons | Pass | Pass – navigate to the respective pages |
| 2.2 | Checkbox | Text and save “e-portfolio……” | Pass | Pass – Save and shown on blog |
| 2.3 | Submitting a comment | Leaving a comment at a blog -  Input: Name: No Name  Comment: 716969696E$^%$^$&% | Pass | Fail – Prompt to key in a name |
| 2.4 | Submitting a comment | Leaving a comment at a blog -  Input: Name: &&&  Comment: 716969696E$^%$^$&% | Fail | Pass – comment is submitted |
| 2.5 | Checking if every slices in the blog page is limit to 400 characters per post in a slice. | Input: Dummy 400 characters. | Pass | Pass, it is slice and separated properly |
| 3.0 | Pytest | python -m pytest -v --cov | Pass | 2 Fail test – on the assert Selenium. |