

Министерство образования и науки
федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по дисциплине: «Тестирование программного обеспечения»

Лабораторная работа 3

Выполнила: Полтавец
Елена Андреевна

Группа: К3322

Проверил: Кочубеев Николай
Сергеевич

Санкт-Петербург

2024

Цель: научиться проектировать, писать и применять end-to-end (E2E) тесты для проверки работы всей системы или ключевых пользовательских сценариев.

Задачи:

- Выбрать проект для тестирования;
- Проанализировать функциональность проекта и определите ключевые пользовательские пути;
- Написать минимум 2 E2E теста, которые охватывают основные пользовательские сценарии;
- Подготовить отчет о проделанной работе.

Ход работы

1. Выбор проекта для тестирования

Для написания тестов был выбран сайт saucedemo.com. Это одна из многих демонстрационных витрин, полезная для тестирования процессов входа в систему и корзины покупок. Ключевым отличием этого сайта является то, что он имеет 4 разных логина, которые вы можете использовать для разных операций на одном и том же сайте; стандартный, заблокированный, проблемный пользователь и пользователь с проблемами в производительности. Это веб-приложение имитирует интернет-магазин и предоставляет пользователям возможность выполнять ряд действий, включая авторизацию, добавление товаров в корзину и выполнение покупок. Поддерживается людьми из SauceLabs (Рисунки 1–3).

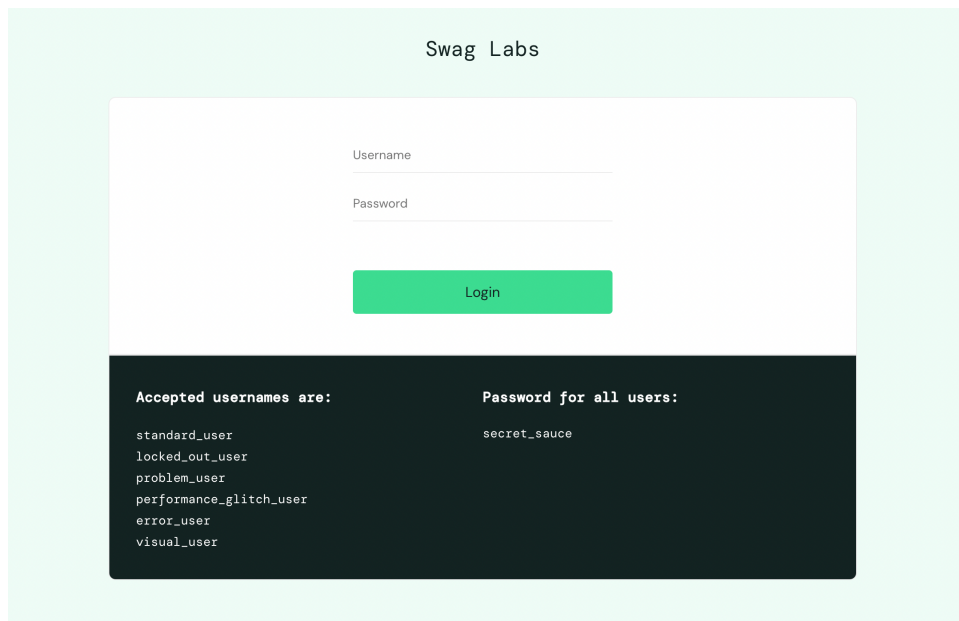


Рисунок 1 – Скриншот сайта [1]

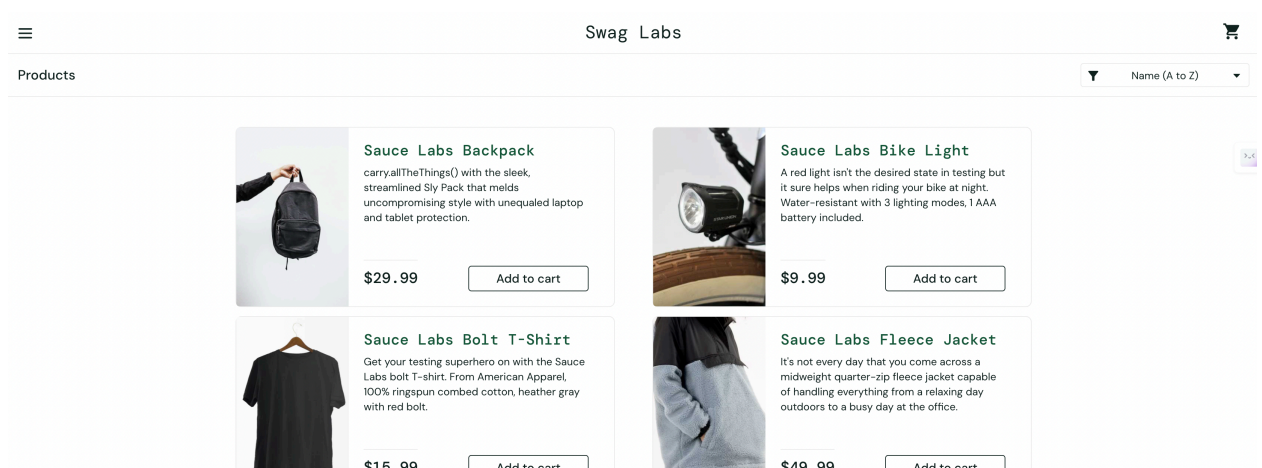


Рисунок 2 – Скриншот сайта [2]

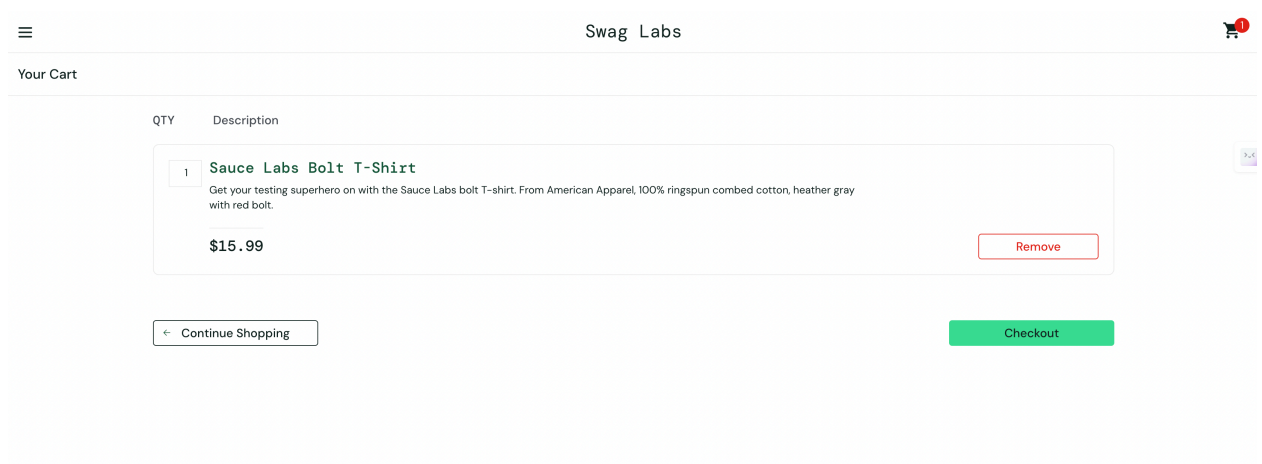


Рисунок 3 – Скриншот сайта [3]

2. Анализ пользовательских сценариев

На основании функциональности сайта можно выделить следующие ключевые пользовательские сценарии:

- Авторизация

Пользователь должен иметь возможность входа в систему с использованием корректных и некорректных данных для проверки функциональности.

- Просмотр товаров

Проверка, как пользователь может просматривать доступные товары, добавлять их в корзину и видеть их в корзине.

- Оформление покупки

Проверка корректности процесса оформления заказа и оплаты.

- Обработка ошибок

Проверка, как приложение реагирует на неверные данные (например, неправильные учетные данные при входе).

3. Написание E2E тестов

Для написания тестов был выбран инструмент Selenium – фреймворк автоматизированного тестирования с открытым исходным кодом, который проверяет веб-приложения на различных платформах и в браузерах. Он предоставляет единый интерфейс для создания и выполнения тестовых сценариев на различных языках программирования, в том числе наиболее распространённых — Java, C# и Python.

Был написан тест (Рисунок 4). для проверки авторизации с корректными учетными данными (учетные данные предоставила платформа для тестирования).

```
14
15 ▶ def test_successful_login(setup):
16     driver = setup
17     driver.get("https://www.saucedemo.com/")
18
19     username = driver.find_element(By.ID, "user-name")
20     password = driver.find_element(By.ID, "password")
21
22     username.send_keys("standard_user")
23     password.send_keys("secret_sauce")
24     password.send_keys(Keys.RETURN)
25
26     time.sleep(2)
27     assert "Swag Labs" in driver.title
28
```

Рисунок 4 – Проверка авторизации с корректными учетными данными

Далее был написан тест с проверкой на обработку ошибок. Были введены неверные данные для авторизации (Рисунок 5).

```
30 ▶ def test_unsuccessful_login(setup):
31     driver = setup
32     driver.get("https://www.saucedemo.com/")
33
34     username = driver.find_element(By.ID, "user-name")
35     password = driver.find_element(By.ID, "password")
36
37     username.send_keys("invalid_user")
38     password.send_keys("wrong_password")
39     password.send_keys(Keys.RETURN)
40
41     time.sleep(2)
42     error_message = driver.find_element(By.XPATH, "//h3[@data-test='error']")
43     assert "Username and password do not match" in error_message.text
44
```

Рисунок 5 – Проверка авторизации с некорректными учетными данными

После был протестирован функционал добавление товара в корзину с последующей оплатой и завершением покупки (Рисунки 6–7).

```

46 def test_add_to_cart(setup):
47     driver = setup
48     driver.get("https://www.saucedemo.com/inventory.html")
49
50     add_to_cart_button = driver.find_element(By.XPATH, "//button[@data-test='add-to-cart-sauce-labs-backpack']")
51     add_to_cart_button.click()
52
53     cart_icon = driver.find_element(By.XPATH, "//a[@class='shopping_cart_link']")
54     cart_icon.click()
55
56     time.sleep(2)
57
58     assert "Sauce Labs Backpack" in driver.page_source
59

```

Рисунок 6 – Проверка добавления товара в корзину

```

61 def test_complete_purchase(setup):
62     driver = setup
63     driver.get("https://www.saucedemo.com/checkout-step-one.html")
64
65     first_name = driver.find_element(By.ID, "first-name")
66     last_name = driver.find_element(By.ID, "last-name")
67     postal_code = driver.find_element(By.ID, "postal-code")
68
69     first_name.send_keys("Test")
70     last_name.send_keys("User")
71     postal_code.send_keys("12345")
72
73     continue_button = driver.find_element(By.CSS_SELECTOR, "input.btn_primary.cart_button")
74     continue_button.click()
75
76     time.sleep(2)
77
78     finish_button = driver.find_element(By.CSS_SELECTOR, "button.btn_action.cart_button")
79     finish_button.click()
80
81     time.sleep(2)
82
83     assert "Thank you for your order!" in driver.page_source
84

```

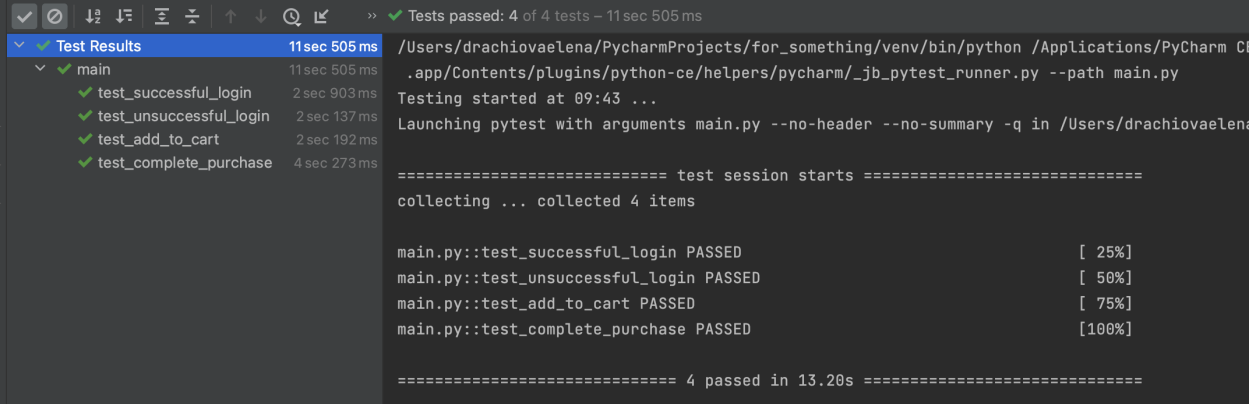
Рисунок 7 – Проверка корректного завершения заказа

4. Результаты тестирования

Перечень протестированных пользовательских сценариев:

- Успешная авторизация;
- Ошибки авторизации при неверных учетных данных;
- Процесс добавления товара в корзину;
- Завершение процесса покупки.

Данные сценарии важны для качественной проверки критических функций сайта, таких как авторизация, покупка и взаимодействие с интерфейсом. Тесты успешно прошли (Рисунок 8).



The screenshot shows the PyCharm Test Results window. On the left, a tree view shows the test hierarchy: 'Test Results' (11sec 505ms) expanded to show 'main' (11sec 505ms), which contains four sub-tests: 'test_successful_login' (2sec 903ms), 'test_unsuccessful_login' (2sec 137ms), 'test_add_to_cart' (2sec 192ms), and 'test_complete_purchase' (4sec 273ms). All tests are marked with green checkmarks. The right pane shows the command line and output of the pytest runner. The command is: `/Users/drachioaelena/PycharmProjects/for_something/venv/bin/python /Applications/PyCharm C... .app/Contents/plugins/python-ce/helpers/pycharm/_jb_pytest_runner.py --path main.py`. The output shows the test session starting, collecting 4 items, and then listing the results: `main.py::test_successful_login PASSED [25%]`, `main.py::test_unsuccessful_login PASSED [50%]`, `main.py::test_add_to_cart PASSED [75%]`, and `main.py::test_complete_purchase PASSED [100%]`. The session ends with `4 passed in 13.20s`.

```
Tests passed: 4 of 4 tests - 11 sec 505 ms

Test Results 11sec 505 ms
└─ main 11sec 505 ms
    └─ test_successful_login 2sec 903ms
    └─ test_unsuccessful_login 2sec 137ms
    └─ test_add_to_cart 2sec 192ms
    └─ test_complete_purchase 4sec 273ms

/Users/drachioaelena/PycharmProjects/for_something/venv/bin/python /Applications/PyCharm C...
.app/Contents/plugins/python-ce/helpers/pycharm/_jb_pytest_runner.py --path main.py
Testing started at 09:43 ...
Launching pytest with arguments main.py --no-header --no-summary -q in /Users/drachioaelena...

===== test session starts =====
collecting ... collected 4 items

main.py::test_successful_login PASSED [ 25%]
main.py::test_unsuccessful_login PASSED [ 50%]
main.py::test_add_to_cart PASSED [ 75%]
main.py::test_complete_purchase PASSED [100%]

===== 4 passed in 13.20s =====
```

Рисунок 8 – Результат выполнения тестов

Вывод: были спроектированы и написаны end-to-end тесты для проверки работы ключевых пользовательских сценариев.

Ссылка на репозиторий с тестами: https://github.com/Malenago/Testing_PO.