

Дипломный проект

“Ручное и автоматизированное тестирование веб-приложения “Triangle”: исследовательский подход.”

Ночевой Сергей



Ночевной Сергей

Инженер по тестированию, 2022 - 2023

35 лет, проживаю в городе Анапа.

Имею среднее техническое образование по специальности: “Строительство и эксплуатация зданий и сооружений”
И высшее образование по специальности: “Менеджмент организации”.

С 2012 года работаю в агентстве недвижимости.
Решил сменить область деятельности на более интересную и перспективную, ориентированную на сферу IT.

Увлекаюсь горными лыжами, веду активный образ жизни.



Цель проекта

Цель данной работы заключается в исследовании различных методов тестирования веб-приложения "Triangle", включая как ручные, так и автоматизированные подходы. Выявить не только ожидаемые, но и неожиданные аспекты функциональности приложения, проверить его на корректность и надежность, а также закрепить полученные знания на этапе обучения.

Задачи проекта

1. Ознакомиться с веб-приложением "Triangle".
2. Составить условный чек-лист для ручного тестирования, базируясь на исследовательском подходе.
3. На основе ожидаемых выводов программы, полученных в результате ручных проверок, написать автотесты на языке Python.



План работы

1

Ознакомиться с веб-приложением “Triangle”

2

Определить элементы и функционал для тестирования

3

Определить методы и техники тестирования

4

Составить условный чек-лист для ручного тестирования

5

Выполнить ручное тестирование

6

Выбрать инструменты для автоматизированного тестирования

7

Создание проекта в Visual Studio Code

8

Написать автотесты

9

Выполнить тестирование

10

Сгенерировать отчет в Pytest HTML Reporter

11

Оформить баг-репорты по найденным ошибкам, если имеются

12

Создать отчет о проделанном исследовании



Выбор инструментов



Visual Studio Code - среда разработки.



Python 3.9 - язык для написания автотестов, имеет набор готового ПО для разработки и запуска приложений.



Pytest - фреймворк для написания тестов.



Selenium - фреймворк, который позволяет управлять браузерами.



Pytest HTML Reporter - плагин для создания отчетов о тестировании.



Git и GitHub - для ведения репозитория по проектам.



Знакомство с веб-приложением “Triangle”

Веб-приложение "Triangle" представляет собой простой калькулятор, который принимает три стороны треугольника от пользователя и определяет тип треугольника на основе введенных данных. Основные функции приложения включают в себя определение равностороннего, равнобедренного и разностороннего треугольника.

[Index](#) [v001](#) [v002](#) [About](#)

Triangle v001

Enter the lengths of the three sides of a triangle. The program will inform you if the triangle is equilateral, isosceles or scalene.

Side 1:

Side 2:

Side 3:

Identify Triangle Type

Ручное тестирование по условному чек-листу



	A	B	C	D
1	Проверка	Ожидание	Результат	Комментарий
2	Проверка кнопки "Indetefy Triangle Type"	Кнопка реагирует на "клик"	passed	При нажатии кнопки выводится сообщение, что говорит о её работоспособности
3	Ввод валидных значений (3, 4, 5)	Приложение определило треугольник как разносторонний	passed	Выводится сообщение "Scalene", что значит разносторонний
4	Ввод валидных значений (5, 5, 5)	Приложение определило треугольник как равносторонний	passed	Выводится сообщение "Equilateral", что значит равносторонний
5	Ввод валидных значений (7, 7, 4)	Приложение определило треугольник как равнобедренный	passed	Выводится сообщение "Isosceles", что значит равнобедренный
6	Ввод невалидных значений (0, 0, 0)	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Not a Triangle", что говорит о правильной обработке ввода приложением некорректных данных
7	Ввод невалидных значений с отрицательными числами (-2, 3, 4)	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Not a Triangle", что говорит о правильной обработке ввода приложением некорректных данных
8	Ввод невалидных значений с невозможным соотношением сторон треугольника (10, 20, 10)	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Not a Triangle", что говорит о правильной обработке ввода приложением некорректных данных
9	Ввод невалидных значений в виде букв/символов в графу "Side 1"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 1 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных и оно не подвержено XSS уязвимости.
10	Ввод невалидных значений в виде букв/символов в графу "Side 2"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 2 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных
11	Ввод невалидных значений в виде букв/символов в графу "Side 3"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 3 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных
12	Ввод больших чисел в графы (4294967295, 4294967295, 4294967000)	Приложение либо не выдаёт ответ, либо выдаёт предупреждение, либо считывает треугольник, если такой получается.	passed	Выводится сообщение "Isosceles", что значит равнобедренный. Это говорит о том, что приложение не только справилось с потенциально большим числом, но и смогло определить треугольник, верно считывая данные
13	Ввод пустого поля/полей в след комбинации (пусто, пусто, пусто)	Приложение либо не выдаёт ответ, либо предупреждает о пустом поле/полях.	passed	Выводится сообщение "Error: Side 1 is missing", что говорит о правильной обработке ввода некорректных данных
14	Ввод пустого поля/полей в след комбинации (пусто, валидное значение, валидное значение)	Приложение либо не выдаёт ответ, либо предупреждает о пустом поле/полях.	passed	Выводится сообщение "Error: Side 1 is missing", что говорит о правильной обработке ввода некорректных данных
15	Ввод пустого поля/полей в след комбинации (валидное значение, валидное значение, пусто)	Приложение либо не выдаёт ответ, либо предупреждает о пустом поле/полях.	passed	Выводится сообщение "Error: Side 3 is missing", что говорит о правильной обработке ввода некорректных данных
16	Ввод пустого поля/полей в след комбинации (валидное значение, пусто, валидное значение)	Приложение либо не выдаёт ответ, либо предупреждает о пустом поле/полях.	passed	Выводится сообщение "Error: Side 2 is missing", что говорит о правильной обработке ввода некорректных данных
17	Ввод скрипта <script>alert('XSS')</script> в графу "Side 1"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 1 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных и оно не подвержено XSS уязвимости.
18	Ввод скрипта <script>alert('XSS')</script> в графу "Side 2"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 2 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных и оно не подвержено XSS уязвимости.
19	Ввод скрипта <script>alert('XSS')</script> в графу "Side 3"	Приложение либо не выдаёт ответ, либо сообщает о некорректных значениях.	passed	Выводится сообщение "Error: Side 3 is not a Number", что говорит о правильной обработке ввода приложением некорректных данных и оно не подвержено XSS уязвимости.

Автоматизация на Python

Паттерн Page Object



BaseApp.py - файл содержит класс с методами, который представляет базовую страницу для автотестов веб-приложения “Triangle”.

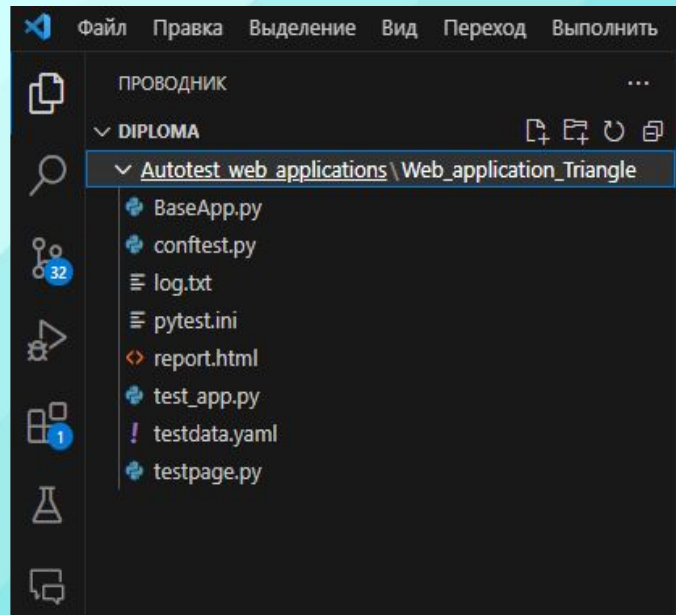
conftest.py - файл содержит фикстуру, создающую экземпляр браузера, который будет использоваться в тестах.

testpage.py - файл содержит классы для удобной работы с локаторами элементов страницы, а также методы для выполнения операций на веб-странице.

testdata.yaml - содержит параметры для последующего тестирования (браузер, тесты с данными в виде списка).

pytest.ini - содержит настройки, связанные с логированием и выводом отчета о тестировании.

test_app.py - файл с тестами.





Автоматизация на Python

Тестирование проекта

```
test_app.py X
Autotest_web_applications > Web_application_Triangle > test_app.py > ...
1  from testpage import OperationsHelper
2  import logging
3  import yaml
4  import pytest
5
6  with open("testdata.yaml", encoding="utf-8") as f:
7      testdata = yaml.safe_load(f)
8
9
10 def test_1(browser):
11     """
12     Проверяем треугольник на равнобедренность,
13     вводя валидные данные в приложение.
14     """
15     logging.info("Test Starting")
16     testpage = OperationsHelper(browser)
17     testpage.go_to_site()
18     testpage.enter_side1(testdata.get("test1")[0])
19     testpage.enter_side2(testdata.get("test1")[1])
20     testpage.enter_side3(testdata.get("test1")[2])
21     testpage.click_button()
22     assert "isosceles" in testpage.get_text().lower(), "test FAILED"
23
24
25 def test_2(browser):
26     """
27     Проверяем треугольник на равносторонность,
28     вводя валидные данные в приложение.
29     """
30     logging.info("Test Starting")
31     testpage = OperationsHelper(browser)
32     testpage.go_to_site()
33     testpage.enter_side1(testdata.get("test2")[0])
34     testpage.enter_side2(testdata.get("test2")[1])
35     testpage.enter_side3(testdata.get("test2")[2])
36     testpage.click_button()
37     assert "equilateral" in testpage.get_text().lower(), "test FAILED"
```

```
40 def test_3(browser):
41     """
42     Проверяем треугольник на разносторонность,
43     вводя валидные данные в приложение.
44     """
45     logging.info("Test Starting")
46     testpage = OperationsHelper(browser)
47     testpage.go_to_site()
48     testpage.enter_side1(testdata.get("test3")[0])
49     testpage.enter_side2(testdata.get("test3")[1])
50     testpage.enter_side3(testdata.get("test3")[2])
51     testpage.click_button()
52     assert "scalene" in testpage.get_text().lower(), "test FAILED"
53
54
55 @pytest.mark.parametrize("test_name, side_values", [
56     ("test4", [0, 0, 0]),
57     ("test5", [-2, 3, 4]),
58     ("test6", [10, 20, 10])
59 ])
60 def test_4_5_6(browser, test_name, side_values):
61     """
62     Вводим невалидные данные в приложение,
63     объединив три однотипных теста с помощью фикстуры @pytest.mark.parametrize.
64     """
65     logging.info("Test Starting")
66     testpage = OperationsHelper(browser)
67     testpage.go_to_site()
68     side1, side2, side3 = side_values
69     testpage.enter_side1(side1)
70     testpage.enter_side2(side2)
71     testpage.enter_side3(side3)
72     testpage.click_button()
73     assert "error: not a triangle" in testpage.get_text(
74         ).lower(), f"{test_name} FAILED"
```



```
77 @pytest.mark.parametrize("test_name, side_values", [
78     ("test7", ["aba", 20, 10]),
79     ("test8", [10, "aba", 10]),
80     ("test9", [10, 20, "aba"])
81 ])
82 def test_7_8_9(browser, test_name, side_values):
83     """
84     Вводим невалидные данные в приложение,
85     объединив три однотипных теста с помощью фикстуры @pytest.mark.parametrize.
86     """
87     logging.info("Test Starting")
88     testpage = OperationsHelper(browser)
89     testpage.go_to_site()
90     side1, side2, side3 = side_values
91     testpage.enter_side1(side1)
92     testpage.enter_side2(side2)
93     testpage.enter_side3(side3)
94     testpage.click_button()
95     assert "error: side 1 is not a number" or "error: side 2 is not a number" \
96     or "error: side 3 is not a number" in testpage.get_text(
97     ).lower(), f"{test_name} FAILED"
98
99
100 def test_10(browser):
101     """
102     Ввод большого валидного числа.
103     """
104     logging.info("Test Starting")
105     testpage = OperationsHelper(browser)
106     testpage.go_to_site()
107     testpage.enter_side1(testdata.get("test10")[0])
108     testpage.enter_side2(testdata.get("test10")[1])
109     testpage.enter_side3(testdata.get("test10")[2])
110     testpage.click_button()
111     assert "isosceles" in testpage.get_text().lower(), "test FAILED"
```

```
113 def test_11(browser):
114     """
115     Ввод пустых полей.
116     """
117     logging.info("Test Starting")
118     testpage = OperationsHelper(browser)
119     testpage.go_to_site()
120     testpage.enter_side1(testdata.get("test11"))
121     testpage.enter_side2(testdata.get("test11"))
122     testpage.enter_side3(testdata.get("test11"))
123     testpage.click_button()
124     assert "error: side 1 is missing" in testpage.get_text().lower(), "test FAILED"
125
126
127 def test_12(browser):
128     """
129     Проверка на XSS уязвимость
130     """
131     logging.info("Test Starting")
132     testpage = OperationsHelper(browser)
133     testpage.go_to_site()
134     testpage.enter_side1(testdata.get("test12")[0])
135     testpage.enter_side2(testdata.get("test12")[1])
136     testpage.enter_side3(testdata.get("test12")[2])
137     testpage.click_button()
138     assert "error: side 1 is not a number" in testpage.get_text().lower(), "test FAILED"
```



Автоматизация на Python. Генерация отчета

PYTEST REPORT

October 28, 2023

⌚ Time taken 15.85 secs

12

TEST CASES



12

PASSED

0

FAILED

0

SKIPPED

0

XPASSED

0

XFAILED

0

ERROR

0

RERUN



Автоматизация на Python. Генерация отчета

Pytest HTML Reporter

Suite	Test Case	Status	Time (s)
test_app.py	test_1	PASS	5.32
test_app.py	test_12	PASS	0.59
test_app.py	test_4_5_6[test6-side_values2]	PASS	0.48
test_app.py	test_7_8_9[test7-side_values0]	PASS	0.4
test_app.py	test_2	PASS	0.36
test_app.py	test_3	PASS	0.36
test_app.py	test_7_8_9[test8-side_values1]	PASS	0.36
test_app.py	test_10	PASS	0.36
test_app.py	test_4_5_6[test4-side_values0]	PASS	0.35
test_app.py	test_4_5_6[test5-side_values1]	PASS	0.35
test_app.py	test_7_8_9[test9-side_values2]	PASS	0.35
test_app.py	test_11	PASS	0.35



Итог

Цели и задачи данного проекта выполнены.

- Проведена подготовительная работа для тестирования веб-приложения “Triangle”.
- Составлен чек-лист с исследовательским подходом для ручного тестирования веб-приложения
- Выбраны инструменты для написания кода и проведения тестирования.
- Написаны авто-тесты на основе ожидаемых выводов программы, полученных в результате ручного тестирования.

Все тесты проходят успешно!

Идеи на будущее

Продолжить развиваться в направлении автоматизации тестирования.

Поиск работы в сфере тестирования, применение полученных навыков на практике.