Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине: “РСиОТ”

По теме: «Тестирование функциональности веб-приложения»

**Выполнил**:

студент 4 курса

группы ПО-7

Комиссаров А.Е.

**Проверил:**

Козик И.Д.

Брест 2024

**Цель.** Провести тестирование функциональности веб-приложения, разработанного в рамках предыдущих лабораторных работ. Гарантировать правильную работу всех основных функций приложения и выявить возможные ошибки или недочеты.

**Ход работы:**

Для тестирования приложения была использована библиотека pytest для python. Созданы тесты для проверки функционала веб-приложения, а именно: проверка правильного вывода списка багов, изменение, удаление и добавление данных в базу mysql.

Код программы:

from fastapi.testclient import TestClient

from server import app

client = TestClient(app)

def test\_redirect\_root\_to\_docs():

response = client.get("/")

assert response.status\_code == 200

def test\_bugsGetAll():

response = client.get("/bugs/list/all")

assert response.status\_code == 200

assert response.json() == {"success": "[(6, 3, 2, 'black screen fix pls', 'open'), (7, 2, 3, 'font CRASH', 'open'), (8, 1, 1, 'bold text lag', 'fixed'), (9, 1, 3, 'testadd asldkfa;sld', 'open'), (11, 2, 2, '2', '2'), (14, 1, 1, 'asdf1234', 'open')]"}

def test\_bugsGetApp():

response = client.get("/bugs/list/app?appid=1")

assert response.status\_code == 200

assert response.json() == {"success": "[(8, 1, 1, 'bold text lag', 'fixed'), (9, 1, 3, 'testadd asldkfa;sld', 'open'), (14, 1, 1, 'asdf1234', 'open')]"}

def test\_bugsGetOpen():

response = client.get("/bugs/list/open")

assert response.status\_code == 200

assert response.json() == {"success": "[(6, 3, 2, 'black screen fix pls', 'open'), (7, 2, 3, 'font CRASH', 'open'), (9, 1, 3, 'testadd asldkfa;sld', 'open'), (14, 1, 1, 'asdf1234', 'open')]"}

def test\_bugsGetFixed():

response = client.get("/bugs/list/fixed")

assert response.status\_code == 200

assert response.json() == {"success": "[(8, 1, 1, 'bold text lag', 'fixed')]"}

def test\_ping():

response = client.get("/connection/ping")

assert response.status\_code == 200

assert response.json() == {"message": "pong"}

def test\_login():

response = client.post("/users/login", data={"name":"komissarov","password":"password"})

assert response.status\_code == 200

assert response.json() == {

"status": [

1,

"admin"

]

}

def test\_register():

response = client.post("/users/register", data={"name":"komissarov","password":"pass", "access":"admin"})

assert response.status\_code == 200

assert response.json() == {

"status": "already exists"

}

def test\_addBug():

response = client.post("/bugs/add", data={"appid":1, "authorid":1, "title":"1", "status":"open"})

assert response.status\_code == 200

assert response.json() == {

"status": "success"

}

def test\_editBug():

response = client.post("/bugs/edit", data={"id":20, "appid":2, "authorid":2, "title":"2", "status":"fixed"})

assert response.status\_code == 200

assert response.json() == {

"status": "success"

}

def test\_deleteBug():

response = client.post("/bugs/delete", data={"id":20})

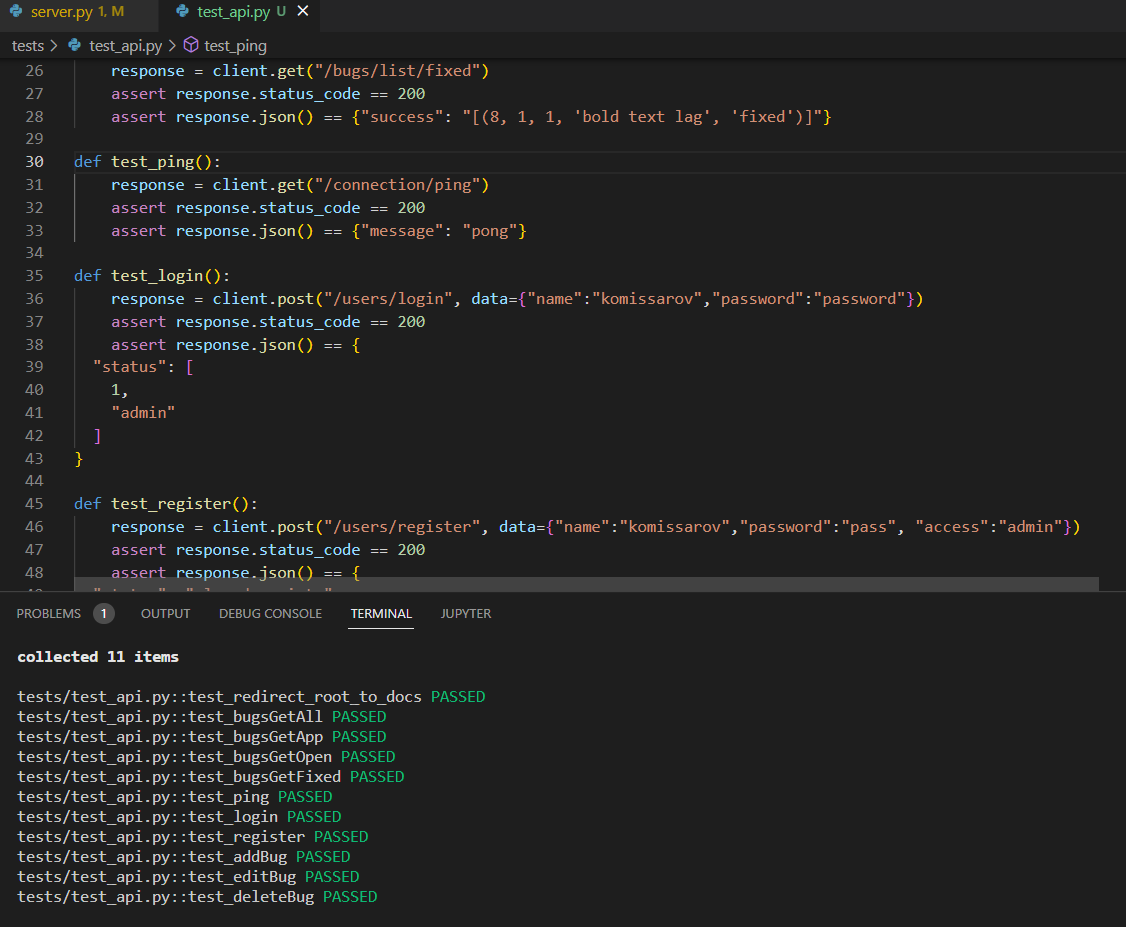
assert response.status\_code == 200

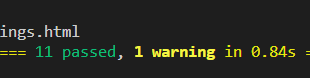
assert response.json() == {

"status": "success"

}

Результат работы программы:





**Вывод.** Провел тестирование функциональности веб-приложения и выявил возможные ошибки или недочеты.