

Тестирование программного обеспечения

Лабораторная работа 4

Разработка автотестов

для Web UI с использованием Selenium

Дорогин Никита ИП-312

ЧАСТЬ 1: Подготовка окружения.

Устанавливаем Python 3:

```
Команда «pytest» не найдена, но может быть установлена с помощью:
sudo apt install python3-pytest
gastello123@gastello123: ~/Desktop/ТестированиеПо/testP0/Lab4$ sudo apt install python3-pytest
[sudo] пароль для gastello123:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  python3-iniconfig python3-packaging python3-pluggy
Следующие НОВЫЕ пакеты будут установлены:
  python3-iniconfig python3-packaging python3-pluggy python3-pytest
Обновлено 0 пакетов, установлено 4 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 373 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 1525 kB.
```

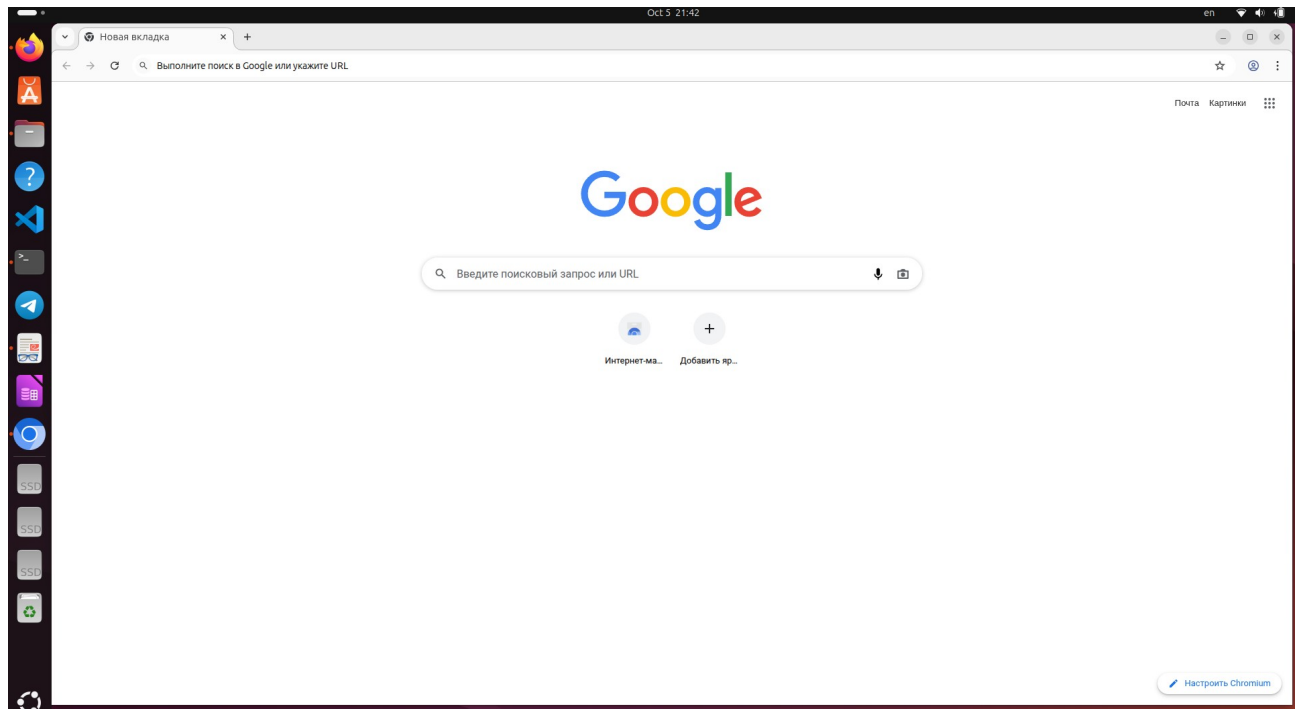
```
Команда «pytest» не найдена, но может быть установлена с помощью:
sudo apt install python3-pytest
gastello123@gastello123: ~/Desktop/ТестированиеПо/testP0/Lab4$ sudo apt install python3-pytest
[sudo] пароль для gastello123:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  python3-iniconfig python3-packaging python3-pluggy
Следующие НОВЫЕ пакеты будут установлены:
  python3-iniconfig python3-packaging python3-pluggy python3-pytest
Обновлено 0 пакетов, установлено 4 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 373 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 1525 kB.
```

```
gastello123@gastello123: $ sudo apt install python3-pip
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libxattr1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-dev libpython3.12-dev python3-dev python3-setuptools python3-wheel python3.12-dev zlib1g-dev
Предлагаемые пакеты:
  python-setuptools-doc
Следующие НОВЫЕ пакеты будут установлены:
  libxattr1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-dev libpython3.12-dev python3-dev python3-pip python3-setuptools python3-wheel python3.12-dev
Обновлено 0 пакетов, установлено 12 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 9611 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 43.5 MB.
Хотите продолжить? [Д/н] д
Пол:1 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 libxattr1-dev amd64 2.6.1-2ubuntu0.3 [140 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu noble/main amd64 libjs-jquery all 3.6.1+dfsg+~3.5.14-1 [328 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu noble/main amd64 libjs-underscore all 1.13.4-dfsg+~1.11.4-3 [118 kB]
Пол:4 http://ru.archive.ubuntu.com/ubuntu noble/main amd64 libjs-sphinxdoc all 7.2.6-6 [149 kB]
Пол:5 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 zlib1g-dev amd64 1:1.3.dfsg-3.1ubuntu2.1 [894 kB]
Пол:6 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-dev amd64 3.12.3-1ubuntu0.8 [5677 kB]
Пол:7 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3-dev amd64 3.12.3-0ubuntu2 [10.3 kB]
Пол:8 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-dev amd64 3.12.3-1ubuntu0.8 [498 kB]
Пол:9 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-dev amd64 3.12.3-0ubuntu2 [26.7 kB]
Пол:10 http://ru.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-setuptools all 68.1.2-2ubuntu1.2 [397 kB]
Пол:11 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-wheel all 0.42.0-2 [53.1 kB]
Пол:12 http://ru.archive.ubuntu.com/ubuntu noble-updates/universe amd64 python3-pip all 24.0+dfsg-1ubuntu1.3 [1320 kB]
Получено 9611 kB за 2с (5098 kB/s)
Выбор ранее не выбранного пакета libxattr1-dev:amd64.
(Чтение базы данных ... на данный момент установлено 238138 файлов и каталогов.)
Подготовка к распаковке ./00-libxattr1-dev_2.6.1-2ubuntu0.3_amd64.deb ...
Распаковывается libxattr1-dev:amd64 (2.6.1-2ubuntu0.3) ...
Выбор ранее не выбранного пакета libjs-jquery.
Подготовка к распаковке ./01-libjs-jquery_3.6.1+dfsg+~3.5.14-1_all.deb ...
Распаковывается libjs-jquery (3.6.1+dfsg+~3.5.14-1) ...
Выбор ранее не выбранного пакета libjs-underscore.
Подготовка к распаковке ./02-libjs-underscore_1.13.4-dfsg+~1.11.4-3_all.deb ...
Распаковывается libjs-underscore (1.13.4-dfsg+~1.11.4-3) ...
Выбор ранее не выбранного пакета libjs-sphinxdoc.
Подготовка к распаковке ./03-libjs-sphinxdoc_7.2.6-6_all.deb ...
Распаковывается libjs-sphinxdoc (7.2.6-6) ...
Выбор ранее не выбранного пакета zlib1g-dev.
Подготовка к распаковке ./04-zlib1g-dev_1:1.3.dfsg-3.1ubuntu2.1_amd64.deb ...
Распаковывается zlib1g-dev (1:1.3.dfsg-3.1ubuntu2.1) ...
Выбор ранее не выбранного пакета libpython3.12-dev.
Подготовка к распаковке ./05-libpython3.12-dev_3.12.3-1ubuntu0.8_amd64.deb ...
Распаковывается libpython3.12-dev (3.12.3-1ubuntu0.8) ...
Выбор ранее не выбранного пакета libpython3-dev.
Подготовка к распаковке ./06-libpython3-dev_3.12.3-0ubuntu2_amd64.deb ...
Распаковывается libpython3-dev (3.12.3-0ubuntu2) ...
Выбор ранее не выбранного пакета python3.12-dev.
Подготовка к распаковке ./07-python3.12-dev_3.12.3-1ubuntu0.8_amd64.deb ...
Распаковывается python3.12-dev (3.12.3-1ubuntu0.8) ...
Выбор ранее не выбранного пакета python3-dev.
Подготовка к распаковке ./08-python3-dev_3.12.3-0ubuntu2_amd64.deb ...
Распаковывается python3-dev (3.12.3-0ubuntu2) ...
Выбор ранее не выбранного пакета python3-setuptools.
Подготовка к распаковке ./09-python3-setuptools_68.1.2-2ubuntu1.2_all.deb ...
Распаковывается python3-setuptools (68.1.2-2ubuntu1.2) ...
Выбор ранее не выбранного пакета python3-wheel.
Подготовка к распаковке ./10-python3-wheel_0.42.0-2_all.deb ...
Распаковывается python3-wheel (0.42.0-2) ...
Выбор ранее не выбранного пакета python3-pip.
Подготовка к распаковке ./11-python3-pip_24.0+dfsg-1ubuntu1.3_all.deb ...
Распаковывается python3-pip (24.0+dfsg-1ubuntu1.3) ...
Получено 9611 kB за 2с (5098 kB/s)
```

Устанавливаем Selenium:

```
gastello123@gastello123:~$ sudo apt install python3-selenium
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  chromium-browser chromium-chromedriver python3-exceptiongroup python3-h11 python3-outcome python3-sniffio python3-sortedcontainers python3-trio python3-trio-websocket
  python3-wsproto
Предлагаемые пакеты:
  python-sortedcontainers-doc
Следующие НОВЫЕ пакеты будут установлены:
  chromium-browser chromium-chromedriver python3-exceptiongroup python3-h11 python3-outcome python3-selenium python3-sniffio python3-sortedcontainers python3-trio
  python3-trio-websocket python3-wsproto
Обновлено 0 пакетов, установлено 11 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 616 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 3177 kB.
Хотите продолжить? [Д/н] Д
Пол:1 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 chromium-browser amd64 2:1snap1-0ubuntu2 [50.0 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 chromium-chromedriver amd64 2:1snap1-0ubuntu2 [2308 B]
Пол:3 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-exceptiongroup all 1.2.0-1 [17.6 kB]
Пол:4 http://ru.archive.ubuntu.com/ubuntu noble-updates/universe amd64 python3-h11 all 0.14.0-1ubuntu0.24.04.1 [51.7 kB]
Пол:5 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-outcome all 1.2.0-1.1 [5968 B]
Пол:6 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-sniffio all 1.3.0-2 [7216 B]
Пол:7 http://ru.archive.ubuntu.com/ubuntu noble/main amd64 python3-sortedcontainers all 2.4.0-2 [27.6 kB]
Пол:8 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-trio all 0.24.0-1ubuntu1 [309 kB]
Пол:9 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-wsproto all 1.2.0-1 [23.6 kB]
Пол:10 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-trio-websocket all 0.11.1-1 [20.6 kB]
Пол:11 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 python3-selenium all 4.18.1+dfsg-1 [101 kB]
Получено 616 kB за 1с (445 kB/s)
Предварительная настройка пакетов ...
Выбор ранее не выбранного пакета chromium-browser.
(Чтение базы данных ... на данный момент установлено 239445 файлов и каталогов.)
Подготовка к распаковке ./00-chromium-browser_2%3a1snap1-0ubuntu2_amd64.deb ...
=> Installing the chromium snap
=> Checking connectivity with the snap store
=> Installing the chromium snap
2025-10-05T21:40:16+07:00 INFO task ignored
chromium: 140.0-7220-2025-09-24-160100-1-installed
```

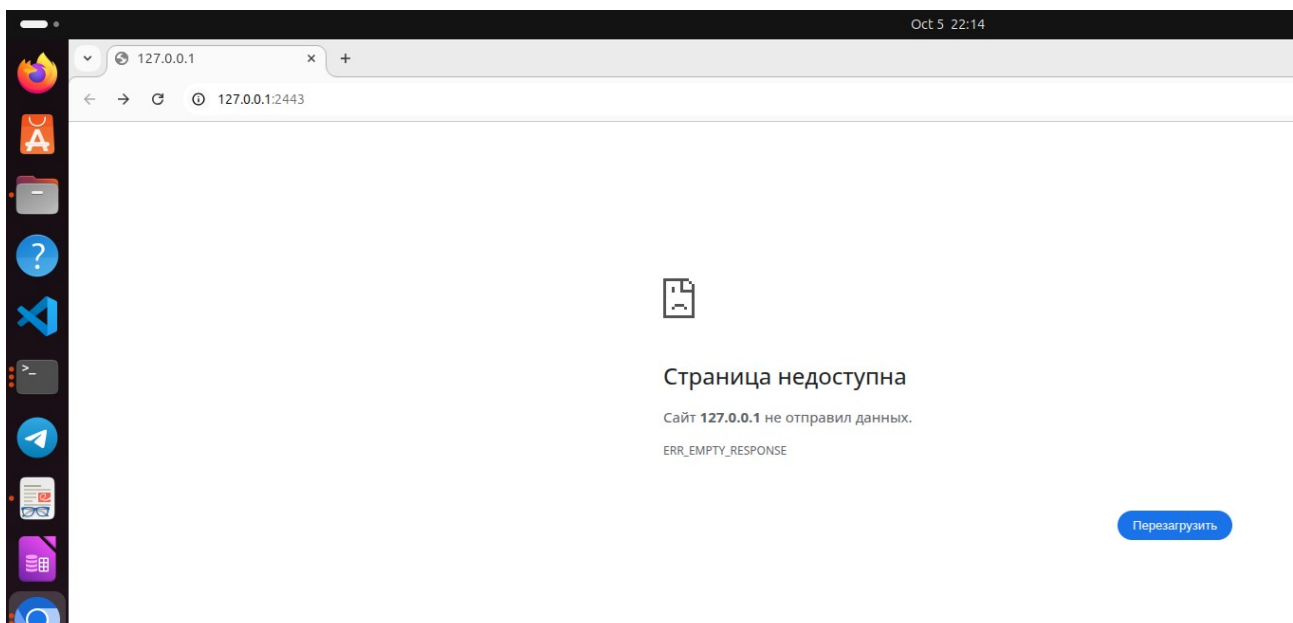
С ним же в комплекте устанавливается Chromium .и ChromiumDriver



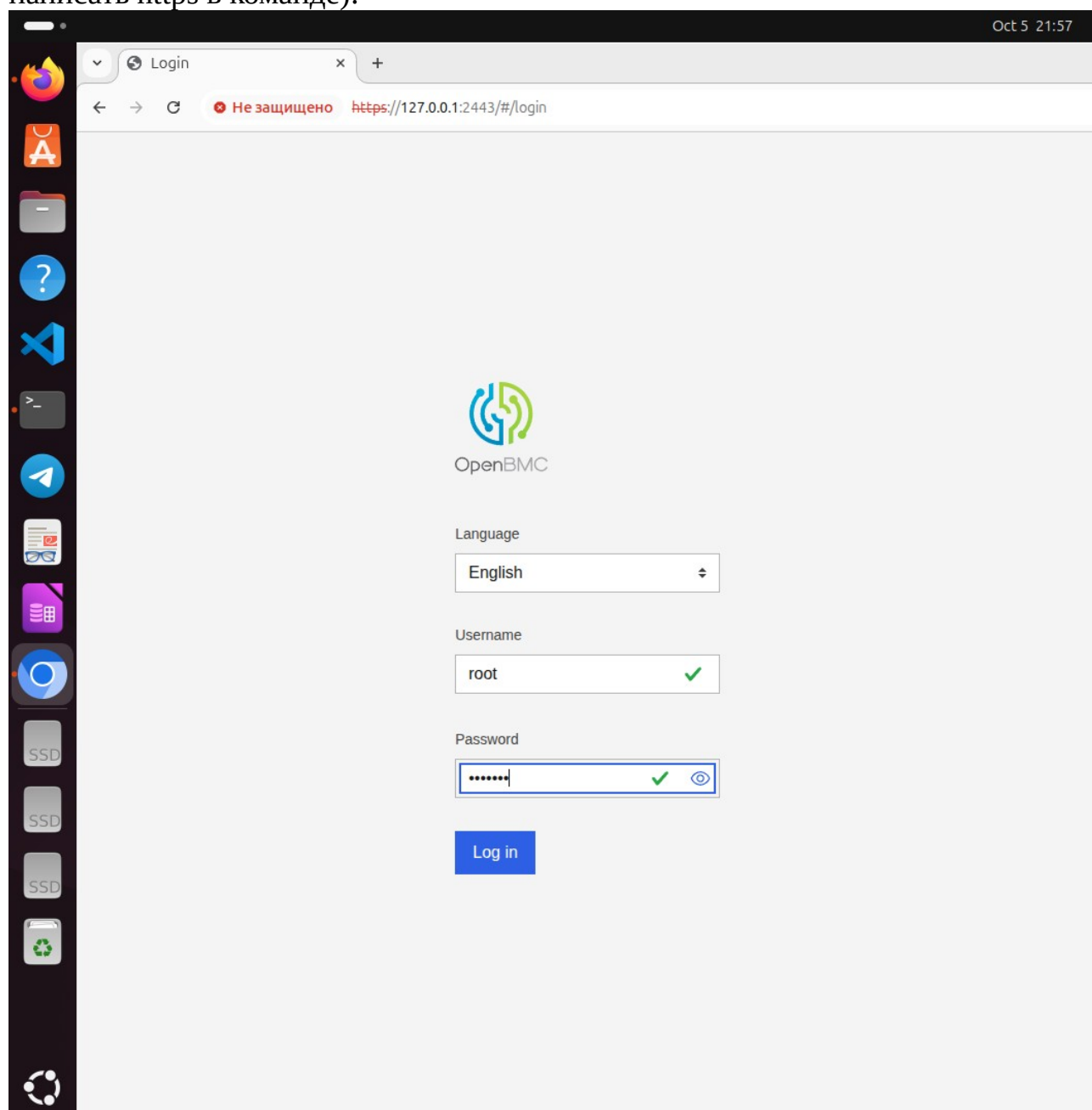
```
gastello123@gastello123:~$ snap info chromium
name:      chromium
summary:   Chromium web browser, open-source version of Chrome
publisher: Canonical✓
store-url: https://snapcraft.io/chromium
contact:   https://bugs.launchpad.net/ubuntu/+source/chromium-browser/+bugs?field.tag=snap
license:   unset
description: |
  An open-source browser project that aims to build a safer, faster, and more
  stable way for all Internet users to experience the web.
commands:
  - chromium.chromedriver
  - chromium
services:
  chromium.daemon: simple, disabled, inactive
snap-id:      XKEcBqPM06H1Z7zG0dG5fbICuf8NWK5R
tracking:     latest/stable
refresh-date: сегодня в 21:40 +07
channels:
  latest/stable: 140.0.7339.207 2025-10-02 (3259) 194MB -
  latest/candidate: 141.0.7390.54 2025-10-03 (3265) 194MB -
  latest/beta: 141.0.7390.37 2025-09-25 (3262) 194MB -
  latest/edge: 142.0.7405.0 2025-09-21 (3252) 188MB -
installed: 140.0.7339.207 (3259) 194MB -
gastello123@gastello123:~$
```

Пытаемся получить доступ к WebUI, за одно пробуем пользоваться драйвером:

```
gastello123@gastello123: ~
gastello123@gastello123:~$ chromium --new-window http://127.0.0.1:2443
Окно или вкладка откроются в текущем сеансе браузера.
gastello123@gastello123:~$
```



Если добавить в начало адресной строки https, всё работает (или изначально написать https в команде):



ЧАСТЬ 2: Написание автотестов.

Создаём файл для кода тестов.

```
gastello123@gastello123:~$ cd /home/gastello123/Desktop/ТестированиеПо/testP0/Lab4
gastello123@gastello123:~/Desktop/ТестированиеПо/testP0/Lab4$ touch ob-autotests.py
gastello123@gastello123:~/Desktop/ТестированиеПо/testP0/Lab4$
```

Настройка веб-драйвера:

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.service import Service
import time
import pytest

def setup_driver():
    s_options = Options()
    s_options.add_argument('--no-sandbox')
    s_options.add_argument('--disable-dev-shm-usage')
    s_options.add_argument('--ignore-certificate-errors')
    s_options.add_argument('--ignore-ssl-errors')
    s_options.add_argument('--headless')

    s_service = Service('/usr/bin/chromedriver')
    driver = webdriver.Chrome(service=s_service, options=s_options)
    driver.implicitly_wait(10)
    return driver
```

Функция для теста входа:

```
def login_openbmc(user, password):
    driver = setup_driver()
    try:
        driver.get('https://127.0.0.1:2443')
        time.sleep(2)

        username_field = driver.find_element(By.ID, 'username')
        password_field = driver.find_element(By.ID, 'password')
        login_button = driver.find_element(By.XPATH, '//button[@type="submit"]')

        username_field.send_keys(user)
        password_field.send_keys(password)
        login_button.click()

        WebDriverWait(driver, 5).until(
            lambda driver: "/login" not in driver.current_url
        )
        return 0

    except:
        return 1

    finally:
        driver.quit()
```

Тесты входа:

```
def test_login_success():
    assert login_openbmc('root', 'OpenBmc') == 0

@pytest.mark.xfail(reason="WRONG LOGIN AND PASSWORD")
def test_login_fail():
    assert login_openbmc('nikita', 'gastello') == 0
```

Функция для теста блокировки при нескольких неудачных попытках:


```

51 def account_block_openbmc():
52     driver = setup_driver()
53     max_attempts = 5
54
55     try:
56         for attempt in range(1, max_attempts + 1):
57             print(f"Попытка {attempt}/{max_attempts}")
58
59             driver.get('https://127.0.0.1:2443/login')
60             time.sleep(10)
61
62             try:
63                 WebDriverWait(driver, 10).until(
64                     EC.presence_of_element_located((By.ID, 'username')))
65             )
66         except TimeoutException:
67             print("Форма входа не загрузилась")
68             return False
69
70         try:
71             username_field = driver.find_element(By.ID, 'username')
72             password_field = driver.find_element(By.ID, 'password')
73             login_button = driver.find_element(By.XPATH, '//*[@type="submit"]')
74
75             username_field.clear()
76             password_field.clear()
77             username_field.send_keys('root')
78             password_field.send_keys('wrong_password_' + str(attempt))
79             login_button.click()
80
81             time.sleep(2)
82
83         except NoSuchElementException:
84             print("Не удалось найти элементы формы")
85             return False
86
87         try:
88             fail_message = driver.find_element(By.XPATH, '//*[contains(text(), "error") or contains(text(), "invalid") or contains(text(), "incorrect") or contains(text(), "wrong password")]')
89             print(f"Неудачный вход: {fail_message.text}")
90         except:
91             print("Неудачный вход (без сообщения)")
92
93         if "login" in driver.current_url.lower():
94             print("Всё ещё на странице входа")
95         else:
96             print("Ушли со страницы входа - что-то пошло не так")
97         break
98
99     print("\nПопробуем войти с правильным паролем после блокировки...")
100     driver.get('https://127.0.0.1:2443/login')
101     time.sleep(3)
102
103     try:
104         username_field = driver.find_element(By.ID, 'username')
105         password_field = driver.find_element(By.ID, 'password')
106         login_button = driver.find_element(By.XPATH, '//*[@type="submit"]')
107
108         username_field.clear()
109         password_field.clear()
110         username_field.send_keys('root')
111         password_field.send_keys('0penBmc')
112         login_button.click()
113
114         time.sleep(3)
115
116     except NoSuchElementException:
117         print("Не удалось найти форму после блокировки")
118         return 1
119
120     current_url = driver.current_url
121
122     if "login" in current_url.lower():
123         try:
124             block_message = driver.find_element(By.XPATH, '//*[contains(text(), "lock") or contains(text(), "block") or contains(text(), "temporarily")]')
125             print(f"Аккаунт заблокирован: {block_message.text}")
126             return True
127         except:
128             print("Аккаунт заблокирован (без сообщения)")
129             return True
130     else:
131         print("Аккаунт НЕ заблокирован - вошли успешно")
132         return False
133
134 except Exception as e:
135     print(f"ОШИБКА: {e}")
136     return False
137
138

```

Тест блокировки:

```

244 def test_account_block():
245     | assert account_block_openbmc() == True
246

```

Функция для теста показа инвенторики:

```
139 def inventory_show():
140     driver = setup_driver()
141     try:
142         driver.get("https://127.0.0.1:2443/login")
143         time.sleep(3)
144
145         driver.find_element(By.ID, 'username').send_keys('root')
146         driver.find_element(By.ID, 'password').send_keys('0penBmc')
147         driver.find_element(By.XPATH, '//button[@type="submit"]').click()
148         time.sleep(3)
149
150         url = "https://127.0.0.1:2443/#/hardware-status/inventory"
151
152         inventory_found = False
153         try:
154             driver.get(url)
155             time.sleep(3)
156
157             inventory_elements = driver.find_elements(By.XPATH,
158                 '//*[contains(text(), "CPU") or contains(text(), "Processor") or '
159                 'contains(text(), "Memory") or contains(text(), "RAM") or '
160                 'contains(text(), "DIMM") or contains(text(), "Hardware") or '
161                 'contains(text(), "Inventory") or contains(text(), "System")]')
162
163
164             if inventory_elements:
165                 print(f"Найдены элементы инвентаризации на {url}:")
166                 for elem in inventory_elements[:5]:
167                     print(f"    - {elem.text}")
168                 inventory_found = True
169         except Exception as e:
170             print(f"Ошибка при проверке {url}: {e}")
171
172
173         if not inventory_found:
174             print("Поиск инвентаризации в текущем интерфейсе...")
175             inventory_buttons = driver.find_elements(By.XPATH,
176                 '//*[contains(text(), "Inventory") or contains(text(), "Hardware") or '
177                 'contains(text(), "System") or contains(text(), "Components")]')
178
179
180             if inventory_buttons:
181                 print("Найдены кнопки инвентаризации:")
182                 for btn in inventory_buttons:
183                     print(f"    - {btn.text}")
184                     try:
185                         btn.click()
186                         time.sleep(3)
187                         inventory_found = True
188                         break
189                     except:
190                         continue
191
192         return inventory_found
193
194     except Exception as e:
195         print(f"Ошибка: {e}")
196         return False
197
```


Тест показа инвенторики:

```
def test_inventory_show():  
    assert inventory_show() == True
```

Функция для теста вывода логов о событиях (событие: включение питания):

```
def check_power_on_logs():  
    driver = setup_driver()  
    try:  
        driver.get("https://127.0.0.1:2443/login")  
        time.sleep(10)  
        driver.find_element(By.ID, 'username').send_keys('root')  
        driver.find_element(By.ID, 'password').send_keys('0penBmc')  
        driver.find_element(By.XPATH, '//button[@type="submit"]').click()  
        time.sleep(10)  
  
        driver.get("https://127.0.0.1:2443/?next=/login#/operations/server-power-operations")  
        time.sleep(10)  
        power_on_button = driver.find_element(By.XPATH, '//button[contains(text(), "Power on")]')  
        power_on_button.click()  
  
        time.sleep(10)  
  
        driver.get("https://127.0.0.1:2443/?next=/login#/logs/event-logs")  
        time.sleep(10)  
  
        power_logs = driver.find_elements(By.XPATH, '//*[contains(text(), "Power on")]')  
  
        if power_logs:  
            print("В логах найдена запись о включении питания")  
            for log in power_logs[:2]:  
                print(f"    - {log.text}")  
            return True  
        else:  
            print("В логах нет записи о включении питания")  
            return False  
    except Exception as e:  
        print(f"Ошибка: {e}")  
        return False
```

Тест показа логов:

```
def test_logs_show():  
    assert check_power_on_logs() == True
```

ЧАСТЬ 3: Запуск и анализ результатов.

```
gastello123@gastello123:~/Desktop/ТестированиеПо/testP0/Lab4$ pytest -v ob-autotests.py
===== test session starts =====
platform linux -- Python 3.12.3, pytest-7.4.4, pluggy-1.4.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/gastello123/Desktop/ТестированиеПо/testP0/Lab4
collected 5 items

ob-autotests.py::test_login_success PASSED [ 20%]
ob-autotests.py::test_login_fail XFAIL (WRONG LOGIN AND PASSWORD) [ 40%]
ob-autotests.py::test_account_block PASSED [ 60%]
ob-autotests.py::test_inventory_show PASSED [ 80%]
ob-autotests.py::test_logs_show FAILED [100%]

===== FAILURES =====
test_logs_show

def test_logs_show():
> assert check_power_on_logs() == True
E assert False == True
E + where False = check_power_on_logs()

ob-autotests.py:252: AssertionError
----- Captured stdout call -----
В логах нет записи о включении питания
----- Captured log call -----
Captured log call
ERROR selenium.webdriver.common.service:service.py:180 Error terminating service process.
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/selenium/webdriver/common/service.py", line 170, in _terminate_process
    self.process.terminate()
  File "/usr/lib/python3.12/subprocess.py", line 2211, in terminate
    self.send_signal(signal.SIGTERM)
  File "/usr/lib/python3.12/subprocess.py", line 2203, in send_signal
    os.kill(self.pid, sig)
PermissionError: [Errno 13] Permission denied

===== short test summary info =====
FAILED ob-autotests.py::test_logs_show - assert False == True
===== 1 failed, 3 passed, 1 xfailed in 222.86s (0:03:42) =====
gastello123@gastello123:~/Desktop/ТестированиеПо/testP0/Lab4$
```

Тесты с входом в систему проходят как запланировано безо всяких проблем.

Тест с блокировкой иногда может не проходить, потому что страница не успевает загрузиться (но таймаут уже везде по 10 секунд и его увеличение слишком растянет выполнение тестов)

Тест показа инвенторики всегда проходит, потому что ему были предъявлены минимальные требования, чтобы на странице хоть что-то было с соответствующими названиями и он что-то находит.

Последний тест в принципе и не мог пройти, потому что Web UI не показывает никаких логов.

Overview

Logs

Event logs

POST code logs

Dumps

Hardware status

Operations

Settings

Security and access

Resource management

POST code logs

0 items

From date

YYYY-MM-DD

To date

YYYY-MM-DD

Delete all

Export all

	Created	Time stamp offset	Boot count	POST code
No items available				

20

Items per page

<

1

>

Overview

Logs

Event logs

POST code logs

Dumps

Hardware status

Operations

Settings

Security and access

Resource management

Dumps

Initiate dump

Select dump type

Select an option

Initiate dump

Dumps available on BMC

0 items

From date

YYYY-MM-DD

To date

YYYY-MM-DD

Filter

	Date and time	Dump type	ID	Size
No items available				

Overview

Logs

Event logs

POST code logs

Dumps

Hardware status

Operations

Settings

Security and access

Resource management

Event logs

0 items

From date

YYYY-MM-DD

To date

YYYY-MM-DD

Filter

Delete all

Export all

	ID	Severity	Date	Description	Status
No items available					

20

Items per page

<

1

>

Ссылка на GitHub: <https://github.com/NekitD/testPO/blob/main/Lab4/ob-autotests.py>

Вывод

В ходе лабораторной работы были освоены навыки разработки автотестов для веб-интерфейса с использованием Selenium. Созданы тесты для проверки функционала авторизации в OpenBMC, покрыты основные сценарии и проанализировали результаты. Эти навыки могут быть применены для тестирования других веб-приложений.