**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**Лабораторная работа № 3**

**Дисциплина: Тестирование программного обеспечения**

**Тема: Написание первых автотестов на Python (или Java) + Selenium WebDriver**

**Выполнил: студент группы 221-371**

**Евдокимов Артемий Русланович**

**(Фамилия И.О )**

**Дата, подпись \_\_16.09.2025\_\_ *\_\_\_\_\_\_\_\_\_\_\_***

**(Дата) (Подпись)**

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**(Фамилия И.О., степень, звание)**

**Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_***

**(Дата) (Подпись**

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2025**

**Цель работы**

Познакомиться с базовым синтаксисом языка Python, инструментами автоматизации тестирования и написать первый UI-автотест с использованием Selenium WebDriver.

**Объект тестирования**

Страница логина веб-приложения, разработанного на Flask и развернутого на локальном сервере.

**Задание**

**Автоматическое тестирование:**

1. Установить и настроить окружение: Python, pip, библиотеки selenium, webdriver-manager, pytest
2. Написать скрипт, который инициализирует браузер с помощью WebDriver, открывает страницу и закрывает браузер
3. Написать автотест, который вводит валидные credentials на странице логина и проверяет успешный вход

**Ручное тестирование:**

Проанализировать, какие элементы на странице имеют стабильные селекторы (id, name) для использования в автотестах

**Выполнение работы**

**1. Подготовка тестового окружения**

Было разработано простое веб-приложение на Flask, состоящее из:

* Серверной части (app.py) с обработкой авторизации
* Страницы входа (login.html)
* Главной страницы (dashboard.html)

**2. Установка и настройка необходимых компонентов**

Установлены следующие библиотеки Python:

bash

pip install selenium webdriver-manager pytest flask

**3. Анализ элементов страницы и выбор селекторов**

Для элементов страницы входа были определены следующие стабильные селекторы:

* Поле логина: //input[@placeholder='Введите логин']
* Поле пароля: //input[@placeholder='Введите пароль']
* Кнопка входа: //button[text()='Вход']
* Сообщение об ошибке: //div[@id='error-message']

**4. Разработка автотестов**

Были написаны два автотеста:

1. test\_successful\_login - проверка успешной авторизации с валидными учетными данными
2. test\_login\_with\_invalid\_credentials - проверка обработки неверных учетных данных

**5. Решение проблем совместимости**

В процессе работы возникли и были решены следующие проблемы:

* Проблемы с инициализацией YandexDriver
* Ошибки при максимизации окна браузера
* Перехват вывода результатов pytest

**Код приложения**

**app.py (серверная часть)**

from flask import Flask, render\_template, request, redirect, url\_for, session, jsonify

import os

app = Flask(\_\_name\_\_)

app.secret\_key = os.urandom(24)

users = {

"testuser": {

"password": "testpass",

"name": "Тестовый Пользователь"

},

"a.r.evdokimov": {

"password": "Vfksi123",

"name": "А.Р. Евдокимов"

}

}

@app.route('/')

def index():

if 'username' in session:

return redirect(url\_for('dashboard'))

return redirect(url\_for('login'))

**test\_login\_fast.py (автотесты)**

import pytest

import os

import sys

from selenium import webdriver

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

from selenium.webdriver.chrome.options import Options

from selenium.common.exceptions import WebDriverException

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

username = request.form.get('username')

password = request.form.get('password')

if username in users and users[username]['password'] == password:

session['username'] = username

return jsonify({"success": True, "redirect": url\_for('dashboard')})

else:

return jsonify({"success": False, "message": "Неверные учетные данные"})

return render\_template('login.html')

@app.route('/dashboard')

def dashboard():

if 'username' not in session:

return redirect(url\_for('login'))

user = users[session['username']]

return render\_template('dashboard.html', username=session['username'], name=user['name'])

@app.route('/logout')

def logout():

session.pop('username', None)

return redirect(url\_for('login'))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, host='0.0.0.0', port=5000)

class TestFlaskLogin:

@pytest.fixture(autouse=True)

def setup(self):

yandex\_browser\_path = "C:\\Users\\Евдик\\AppData\\Local\\Yandex\\YandexBrowser\\Application\\browser.exe"

yandex\_driver\_path = "C:\\Users\\Евдик\\AppData\\Local\\Yandex\\YandexBrowser\\Application\\yandexdriver.exe"

if not os.path.exists(yandex\_browser\_path):

self.\_print\_result("Яндекс.Браузер не найден", "ERROR")

pytest.skip("Яндекс.Браузер не найден")

if not os.path.exists(yandex\_driver\_path):

self.\_print\_result("YandexDriver не найден", "ERROR")

pytest.skip("YandexDriver не найден")

options = Options()

options.binary\_location = yandex\_browser\_path

options.add\_argument("--disable-extensions")

options.add\_argument("--disable-gpu")

options.add\_argument("--no-sandbox")

service = Service(executable\_path=yandex\_driver\_path)

self.driver = webdriver.Chrome(service=service, options=options)

try:

self.driver.maximize\_window()

except WebDriverException:

self.driver.set\_window\_size(1920, 1080)

self.driver.get("http://localhost:5000/login")

WebDriverWait(self.driver, 10).until(

EC.presence\_of\_element\_located((By.TAG\_NAME, "body"))

)

yield

self.driver.quit()

def \_print\_result(self, message, status="INFO"):

print(f"\n{status}: {message}", file=sys.stderr)

sys.stderr.flush()

def test\_successful\_login(self):

login\_field = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.XPATH, "//input[@placeholder='Введите логин']"))

)

login\_field.clear()

login\_field.send\_keys("a.r.evdokimov")

password\_field = self.driver.find\_element(By.XPATH, "//input[@placeholder='Введите пароль']")

password\_field.clear()

password\_field.send\_keys("Vfksi123")

login\_button = self.driver.find\_element(By.XPATH, "//button[text()='Вход']")

login\_button.click()

WebDriverWait(self.driver, 15).until(

EC.presence\_of\_element\_located((By.XPATH, "//h1[contains(text(), 'Главная страница')]"))

)

user\_name = self.driver.find\_element(By.XPATH, "//strong[contains(text(), 'А.Р. Евдокимов')]")

assert user\_name.is\_displayed()

self.\_print\_result("Авторизация: УСПЕШНО", "SUCCESS")

def test\_login\_with\_invalid\_credentials(self):

self.driver.get("http://localhost:5000/login")

login\_field = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.XPATH, "//input[@placeholder='Введите логин']"))

)

login\_field.clear()

login\_field.send\_keys("invalid\_login")

password\_field = self.driver.find\_element(By.XPATH, "//input[@placeholder='Введите пароль']")

password\_field.clear()

password\_field.send\_keys("invalid\_password")

login\_button = self.driver.find\_element(By.XPATH, "//button[text()='Вход']")

login\_button.click()

error\_message = WebDriverWait(self.driver, 10).until(

EC.visibility\_of\_element\_located((By.ID, "error-message"))

)

assert "Неверные учетные данные" in error\_message.text

self.\_print\_result("Авторизация с неверными данными: ОЖИДАЕМАЯ НЕУДАЧА", "INFO")

if \_\_name\_\_ == "\_\_main\_\_":

pytest.main([\_\_file\_\_, "-v", "-s"])

**Результаты тестирования**

**Успешно выполненные тесты:**

1. **Тест успешной авторизации** - проверяет вход с валидными учетными данными
   * Ввод логина и пароля
   * Нажатие кнопки "Вход"
   * Проверка перехода на главную страницу
   * Проверка отображения имени пользователя
   * Результат: "Авторизация: УСПЕШНО"
2. **Тест авторизации с неверными данными** - проверяет обработку ошибок
   * Ввод неверного логина и пароля
   * Нажатие кнопки "Вход"
   * Проверка появления сообщения об ошибке
   * Результат: "Авторизация с неверными данными: ОЖИДАЕМАЯ НЕУДАЧА"

**Статистика тестирования:**

* Всего тестов: 2
* Успешно пройдено: 2
* Провалено: 0
* Пропущено: 0

**Выводы**

В ходе лабораторной работы были успешно выполнены все поставленные задачи:

1. Установлено и настроено тестовое окружение
2. Разработано простое веб-приложение с системой авторизации
3. Проанализированы и выбраны стабильные селекторы для элементов страницы
4. Написаны автотесты для проверки функциональности авторизации
5. Решены проблемы совместимости браузера и драйвера
6. Реализовано логирование результатов тестирования

Основные сложности возникли при настройке взаимодействия между YandexDriver и Яндекс.Браузером, а также при организации вывода результатов тестирования. Эти проблемы были успешно решены путем добавления соответствующих опций и обработки исключений.

Разработанные автотесты успешно проверяют основные сценарии работы системы авторизации и могут быть использованы как основа для более сложных тестовых комплексов.