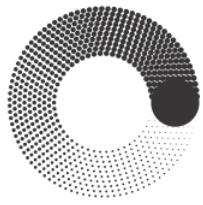


**федеральное государственное автономное образовательное
учреждение высшего образования**



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(а): студент(ка) группы 221-3711

Морозов К.А.
(Фамилия И.О.)

Дата, подпись _____
(Дата) _____
(Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) _____
(Оценка)

Дата, подпись _____
(Дата) _____
(Подпись)

Замечания: _____

Москва 2025

Лабораторная работа 3 – Написание первых автотестов на Python (или Java) + Selenium WebDriver

Цель работы

Познакомиться с базовым синтаксисом языка, инструментами и написать первый UI-автотест.

Задание

Объект тестирования: Страница логина любого публичного веб-сайта или специально подготовленный стенд.

Задание (Автоматическое тестирование):

1. Установите и настройте окружение: Python, pip, библиотеки `selenium`, `webdriver-manager`, `pytest`.
2. Напишите скрипт, который инициализирует браузер (Chrome/Firefox) с помощью WebDriver, открывает страницу и закрывает браузер.
3. Напишите автотест, который вводит валидные credentials на странице логина и проверяет успешный вход (например, появлению элемента "Logout").

Задание (Ручное тестирование): Проанализируйте, какие элементы на странице имеют стабильные селекторы (id, name) для использования в автотестах..

Практическая часть

Подготовка

Для начала я установил python, создал папку лабораторной и импортировал нужные библиотеки:

```
python -m venv .venv  
pip install selenium  
pip install webdriver-manager  
pip install pytest
```

Далее я создал файл .py, в котором будет написан скрипт для автотеста. В качестве объекта автотеста я использую сайт

<https://practicetestautomation.com/practice-test-login/> (первый попавшийся сайт при запросе тестирования логина на selenium). На нем уже есть вся нужная информация: данные тестового аккаунта, ответы, примеры тестов.

Скрипт

В .py файле прописал создание окна браузера для автотестов и 3 автотеста: правильные данные, неправильный логин и неправильный пароль. Элементы для автозаполнения ищутся по id на странице сайта, которые были вручную найдены с помощью просмотра кода элемента:

Test login

s is a simple Login page. Students can use this page to practice writing simple positive and negative Login tests. Login functionality is something that most of the test automation engineers need to automate.

Use next credentials to execute Login:

Username: student
Password: Password123

me

xrd

Submit

```
*<body id="modern-store-modified" class="wp-singular page-template page-template-e_simple_login page-template-e_simple_login-php page-page-id-251 wp-custom-logo wp-theme-modern-store-modified full-post">
  <a class="skip-content" href="#main-container">Press "Enter" to skip to content!</a>
  <div id="overflow-container" class="overflow-container">
    <header class="max-width" class="max-width">
      <header class="site-header" id="site-header" role="banner"></header>
      <section id="main-container" class="main-container" role="main">
        <section id="login">
          <h2>Test login</h2>
          <ul></ul>
          <div id="form">
            <div>
              <label for="username">Username</label>
              <input type="text" name="username" id="username">
            </div>
            <div>
              <label for="password">Password</label>
              <input type="password" name="password" id="password">
            </div>
            <button id="submit" class="btn">Submit</button>
          </div>
        </section>
        <div id="error">Your username is invalid!</div>
      <hr class="wp-block-separator">
      <h5>Test case 1: Positive Login test</h5>
      <p></p>
      <hr class="wp-block-separator">
      <h5>Test case 2: Negative username test</h5>
      <p></p>
      <hr class="wp-block-sePARATOR">
      <h5>Test case 3: Negative password test</h5>
      <p></p>
    </section>
  </div>

```

Результат

Было написано приложение на python для автоматического тестирования сайта. Были вручную найдены элементы тестирования, выбран объект тестирования. Все автотесты были успешно выполнены:

```
PS C:\Users\mrj\Desktop\TPO\tpo\Тестирование\З\Testing_3_TPO_Morozov\test3_Morozov> pytest -q

DevTools listening on ws://127.0.0.1:55009/devtools/browser/e3b7b065-f248-4af0-8db4-ce0084922fb6
[2960:3960:1125/170843.295:ERROR:chrome\browser\task_manager\providers\fallback_task_provider.cc:126] Every renderer should have at least one task provided by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/739782.

.
DevTools listening on ws://127.0.0.1:56764/devtools/browser/51566168-12d9-4618-8c27-14d29d50bc9e
.
DevTools listening on ws://127.0.0.1:59093/devtools/browser/47e3534e-8f2d-40a8-8981-e0c2005aecba
.

[100%]

3 passed in 12.05s
PS C:\Users\mrj\Desktop\TPO\tpo\Тестирование\З\Testing_3_TPO_Morozov\test3_Morozov>
```

[Ссылка на github с кодом лабораторной:](#)

https://github.com/Sollimba/Testing_3_TPO_Morozov

Ссылка на яндекс диск с видео-защитой:

https://disk.yandex.ru/d/Jg3tB_oH4RDkOA