Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

ОТЧЕТ

«Лабораторная №7»

по дисциплине «Внедрение и тестирование программного обеспечения»

Выполнил:

студент группы № 932205

Викторов Всеволод Андреевич

Проверил:

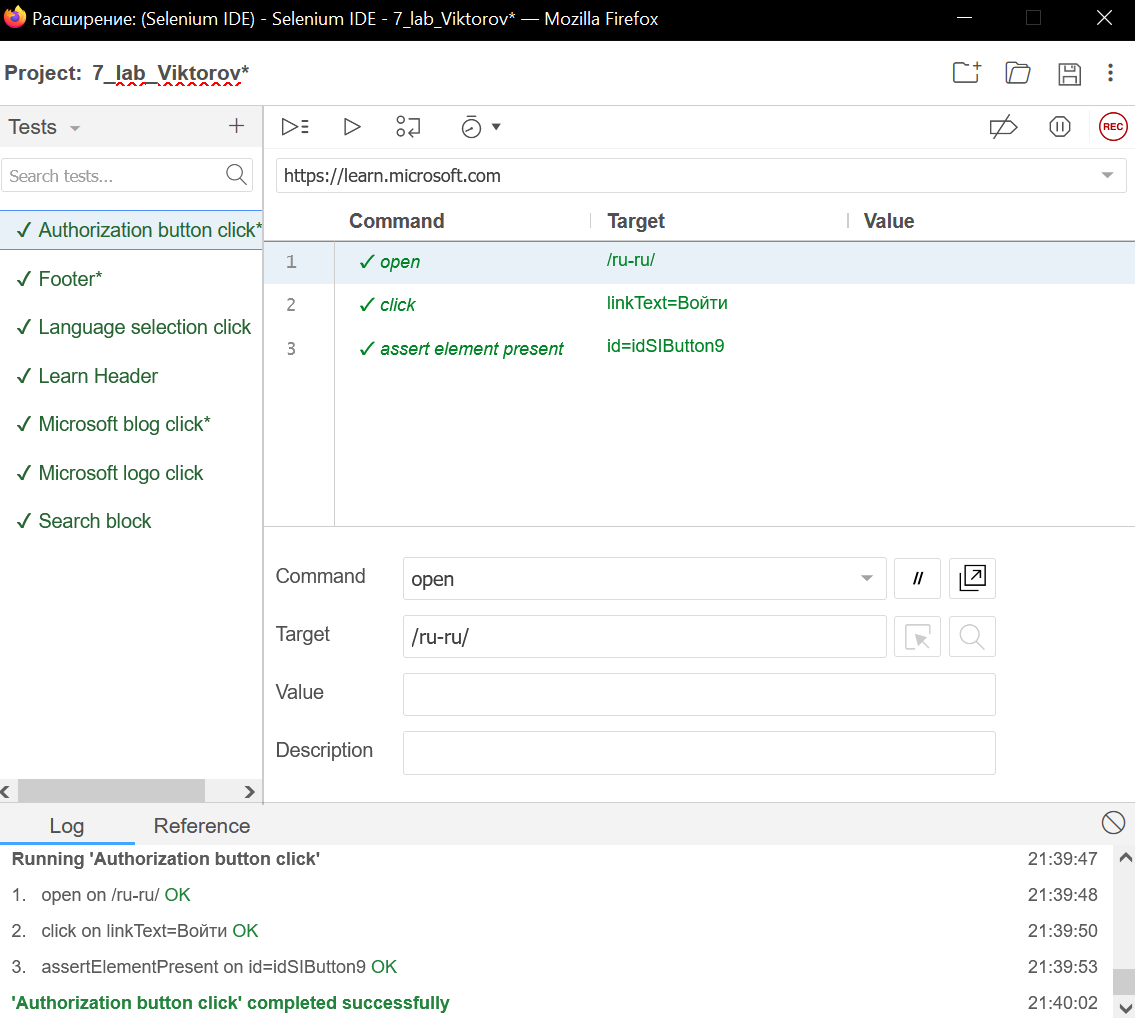
Преподаватель

Мокина Елена Евгеньевна

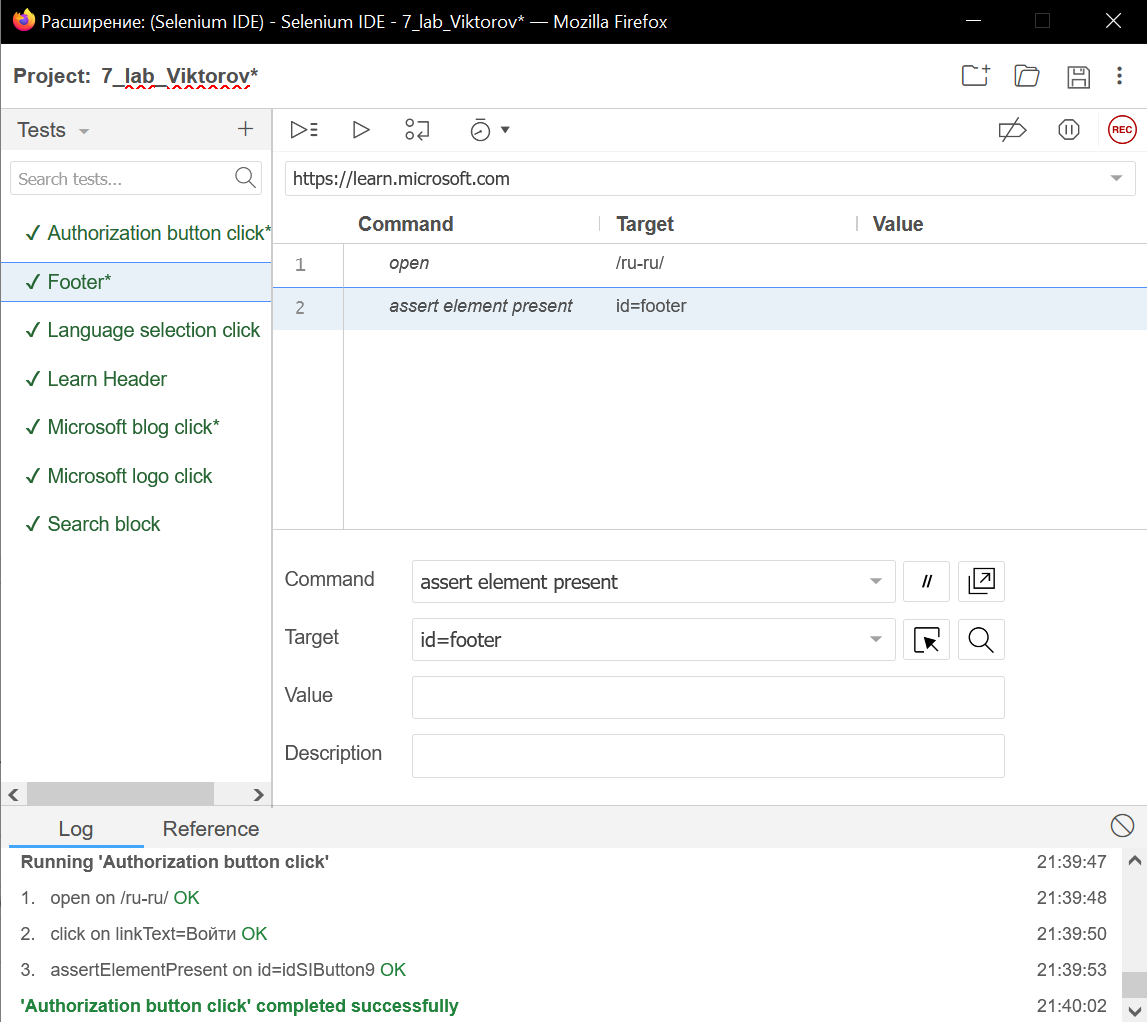
Томск – 2025.

**ЧАСТЬ 1**

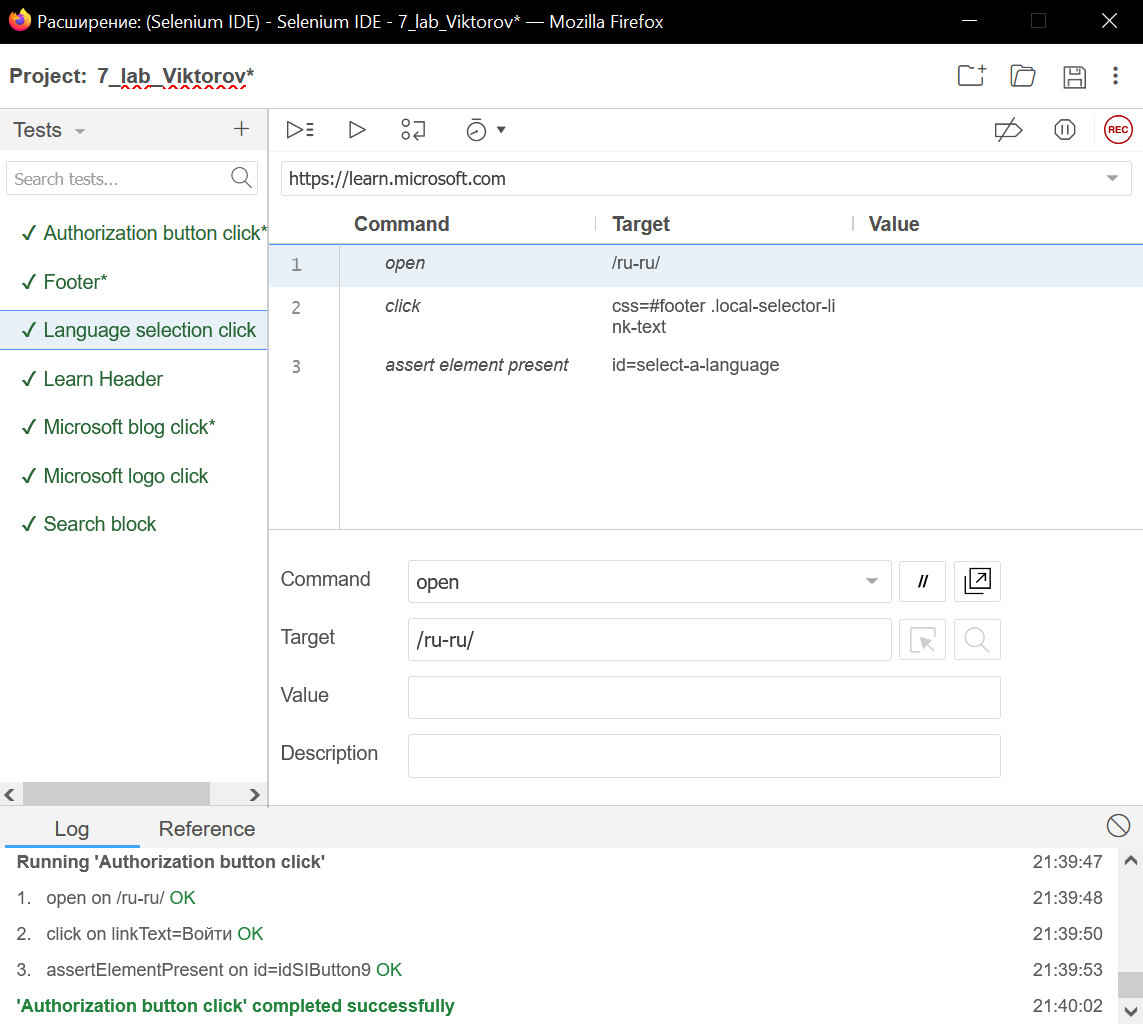
Тест 1: Проверка перехода по ссылки на сайт авторизации



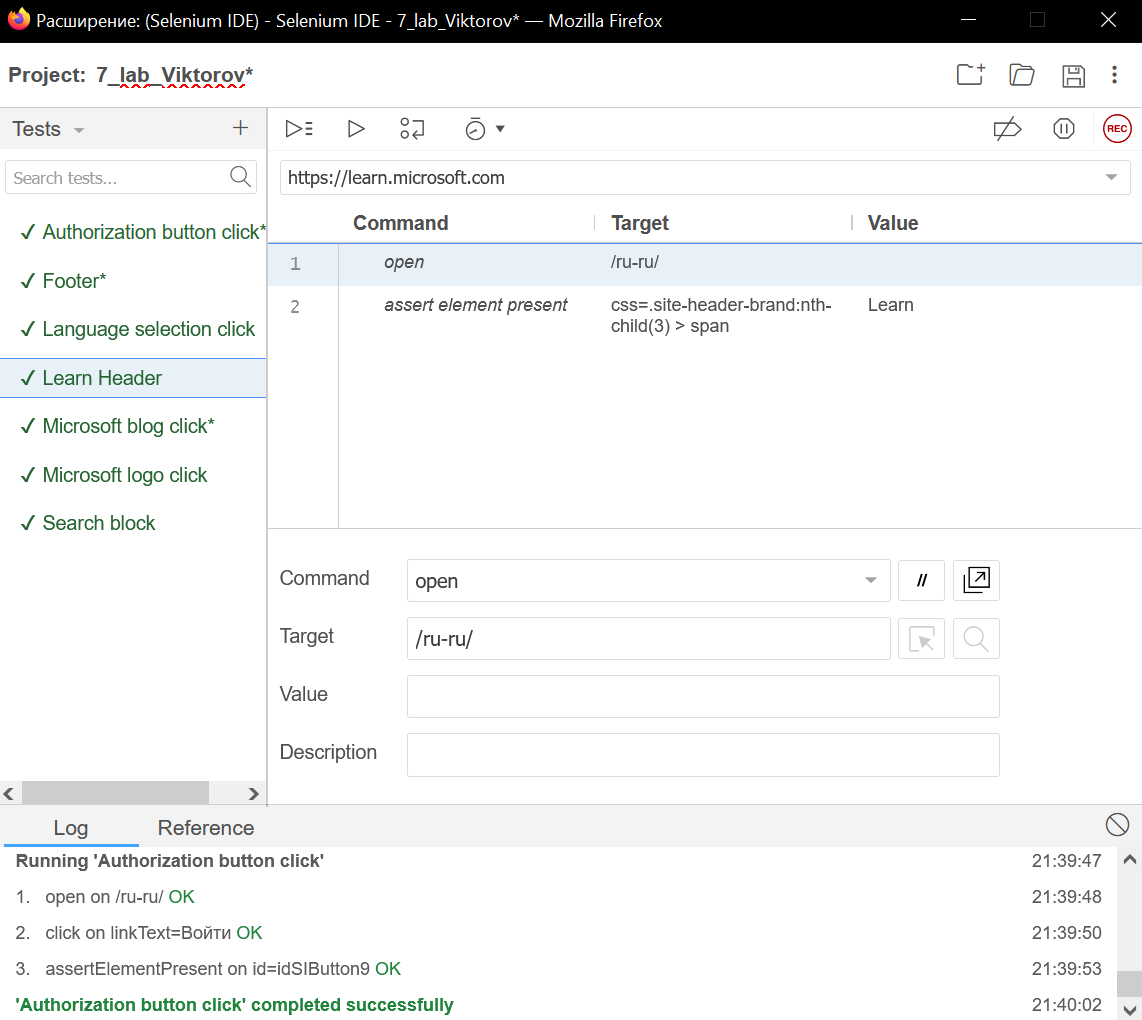
Тест 2: Проверка на наличие подвала сайта на главной странице



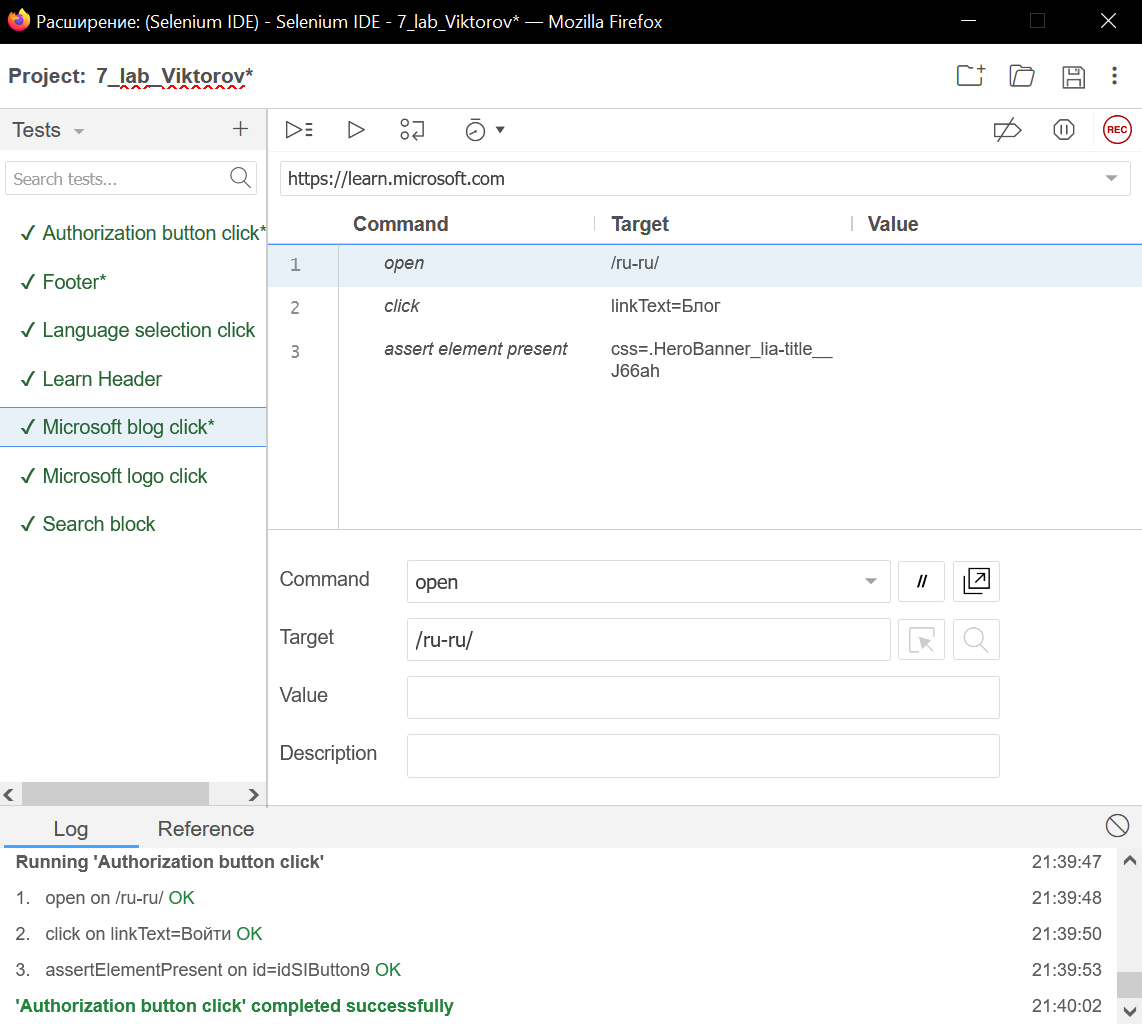
Тест 3: Проверка перехода на страницу с выбором языка



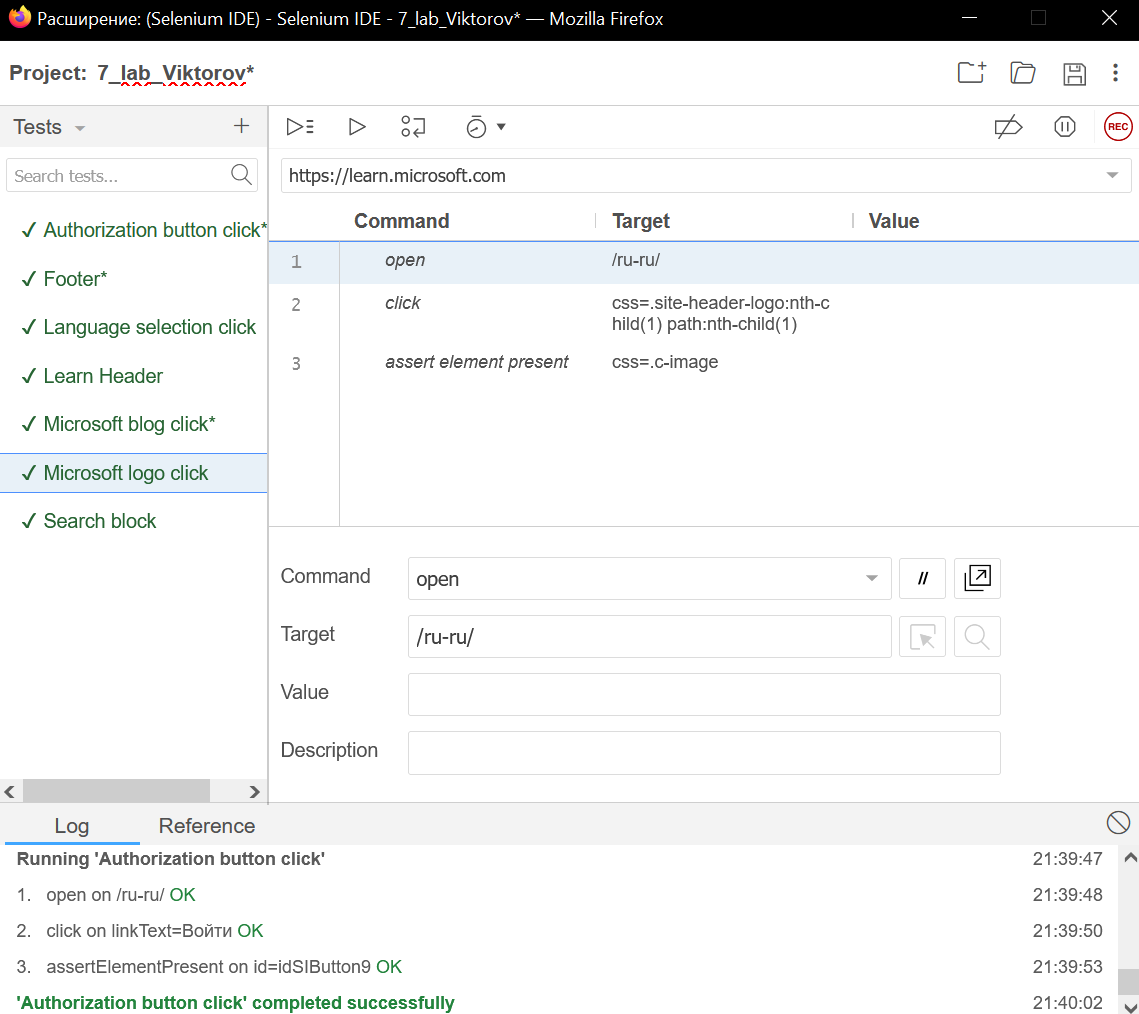
Тест 4: Проверка на наличие заголовка learn на сайте



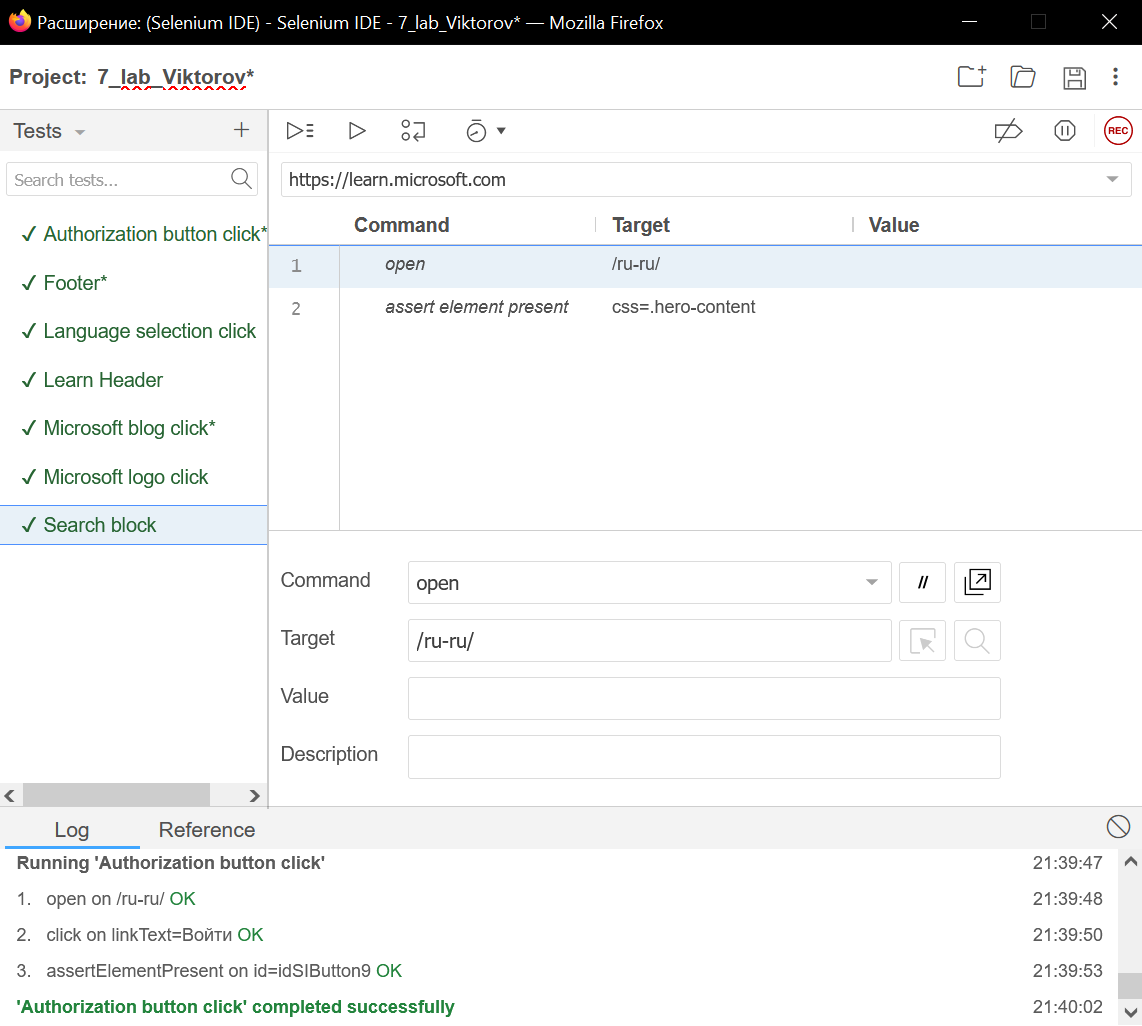
Тест 5: Проверка перехода по ссылке “Блог”



Тест 6: Проверка перехода на главный сайт microsoft по клику на логотип



Тест 7: Проверка на наличие блока поиска в Теле сайта



**ЧАСТЬ 2**

Все тесты были написаны на python с использованием pytest и selenium webdriver, также все тесты имеют одинаковые импорты и методы “setup\_method” и “teardown\_method”. Первый - запускает веб-драйвер, второй - завершает его работу. Оба метода запускаются автоматически, при начале и конце, соответственно, взаимодействия с веб-драйвером. Комментарии приведены в коде.

В каждом из файлов используются следующие строки

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.desired\_capabilities import DesiredCapabilities

class TestAuthorizationbuttonclick():

    def setup\_method(self, method):

        self.driver = webdriver.Chrome()

        self.vars = {}

    def teardown\_method(self, method):

        self.driver.quit()

Тест 1: Проверка перехода по ссылки на сайт авторизации

    def test\_authorizationbuttonclick(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        time.sleep(2)  # Оставляем sleep вне шага

        # Step 2: Клик по кнопке "Войти"

        step\_tracker.step(lambda: self.driver.find\_element(By.LINK\_TEXT, "Войти").click())

        time.sleep(10)  # Оставляем sleep вне шага

        def check\_logo():

            elements = self.driver.find\_elements(By.CSS\_SELECTOR, ".logo")

            assert len(elements) > 0, "Логотип Microsoft не найден"

            return elements

        # Step 3: Проверка логотипа

        step\_tracker.step(check\_logo)

Тест 2: Проверка на наличие подвала сайта на главной странице

    def test\_footer(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        def check\_footer():

            elements = self.driver.find\_elements(By.ID, "footer")

            assert len(elements) > 0, "Футер не найден"

            return elements

        # Step 2: Проверка футера

        step\_tracker.step(check\_footer)

Тест 3: Проверка перехода на страницу с выбором языка

    def test\_languageselectionclick(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        # Step 2: Клик по меню выбора языка

        step\_tracker.step(lambda: self.driver.find\_element(By.CSS\_SELECTOR, "#footer .local-selector-link-text").click())

        def check\_language\_selector():

            elements = self.driver.find\_elements(By.ID, "select-a-language")

            assert len(elements) > 0, "Заголовок 'Выбор языка' не найден"

            return elements

        # Step 3: Проверка заголовка "Выбор языка"

        step\_tracker.step(check\_language\_selector)

Тест 4: Проверка на наличие заголовка learn на сайте

    def test\_learnHeader(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        time.sleep(2)  # Оставляем sleep вне шага

        def check\_header():

            elements = self.driver.find\_elements(By.CSS\_SELECTOR, ".site-header-brand:nth-child(3) > span")

            assert len(elements) > 0, "Заголовок 'Learn' не найден"

            return elements

        # Step 2: Проверка заголовка Learn

        step\_tracker.step(check\_header)

Тест 5: Проверка перехода по ссылке “Блог”

    def test\_microsoftblogclick(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        # Step 2: Клик по кнопке "Блог"

        step\_tracker.step(lambda: self.driver.find\_element(By.LINK\_TEXT, "Блог").click())

        def check\_blog\_header():

            WebDriverWait(self.driver, 10).until(

                EC.presence\_of\_element\_located((By.CSS\_SELECTOR, ".HeroBanner\_lia-title\_\_J66ah"))

            )

            elements = self.driver.find\_elements(By.CSS\_SELECTOR, ".HeroBanner\_lia-title\_\_J66ah")

            assert len(elements) > 0, "Заголовок блога не найден на странице"

            return elements

        # Step 3: Проверка наличия заголовка на странице блога

        step\_tracker.step(check\_blog\_header)

Тест 6: Проверка перехода на главный сайт microsoft по клику на логотип

    def test\_microsoftlogoclick(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        # Step 2: Клик по логотипу Microsoft

        step\_tracker.step(lambda: self.driver.find\_element(By.CSS\_SELECTOR, ".site-header-logo:nth-child(1) path:nth-child(1)").click())

        def check\_logo():

            WebDriverWait(self.driver, 10).until(

                EC.presence\_of\_element\_located((By.CSS\_SELECTOR, ".c-image"))

            )

            elements = self.driver.find\_elements(By.CSS\_SELECTOR, ".c-image")

            assert len(elements) > 0, "Элемент с логотипом (.c-image) не найден на странице"

            return elements

        # Step 3: Проверка наличия элемента с логотипом

        step\_tracker.step(check\_logo)

Тест 7: Проверка на наличие блока поиска в Теле сайта

    def test\_searchblock(self, step\_tracker):

        # Step 1: Открытие сайта

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        time.sleep(2)  # Оставляем sleep вне шага

        def check\_search\_block():

            elements = self.driver.find\_elements(By.CSS\_SELECTOR, ".hero-content")

            assert len(elements) > 0, "Блок поиска (.hero-content) не найден"

            return elements

        # Step 2: Проверка блока поиска

        step\_tracker.step(check\_search\_block)

Код комплексного теста:  
import pytest

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.wait import WebDriverWait

class TestComplexsearchUpdated():

    def setup\_method(self, method):

        self.driver = webdriver.Chrome()

        self.vars = {}

    def teardown\_method(self, method):

        self.driver.quit()

    def test\_complexsearch\_updated(self, step\_tracker):

        # Step 1: Открытие главной страницы

        step\_tracker.step(lambda: self.driver.get("https://learn.microsoft.com/ru-ru/"))

        # Step 2: Клик по кнопке поиска для активации поля

        step\_tracker.step(lambda: WebDriverWait(self.driver, 10).until(

            EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, ".margin-left-xxs > .button .docon"))).click())

        # Step 3: Ввод запроса "Python" в поле поиска

        step\_tracker.step(lambda: [WebDriverWait(self.driver, 10).until(

            EC.element\_to\_be\_clickable((By.ID, "site-header-search-autocomplete-input"))).send\_keys("python"),

            WebDriverWait(self.driver, 10).until(

            EC.element\_to\_be\_clickable((By.ID, "site-header-search-autocomplete-input"))).submit()])

        # Step 4: Переход по первой ссылке в результатах

        step\_tracker.step(lambda: WebDriverWait(self.driver, 10).until(

            EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Руководство по Python в Visual Studio 1. Создание проекта"))).click())

        def check\_element():

            header = WebDriverWait(self.driver, 10).until(

                EC.presence\_of\_element\_located((By.ID, "tutorial-work-with-python-in-visual-studio"))

            )

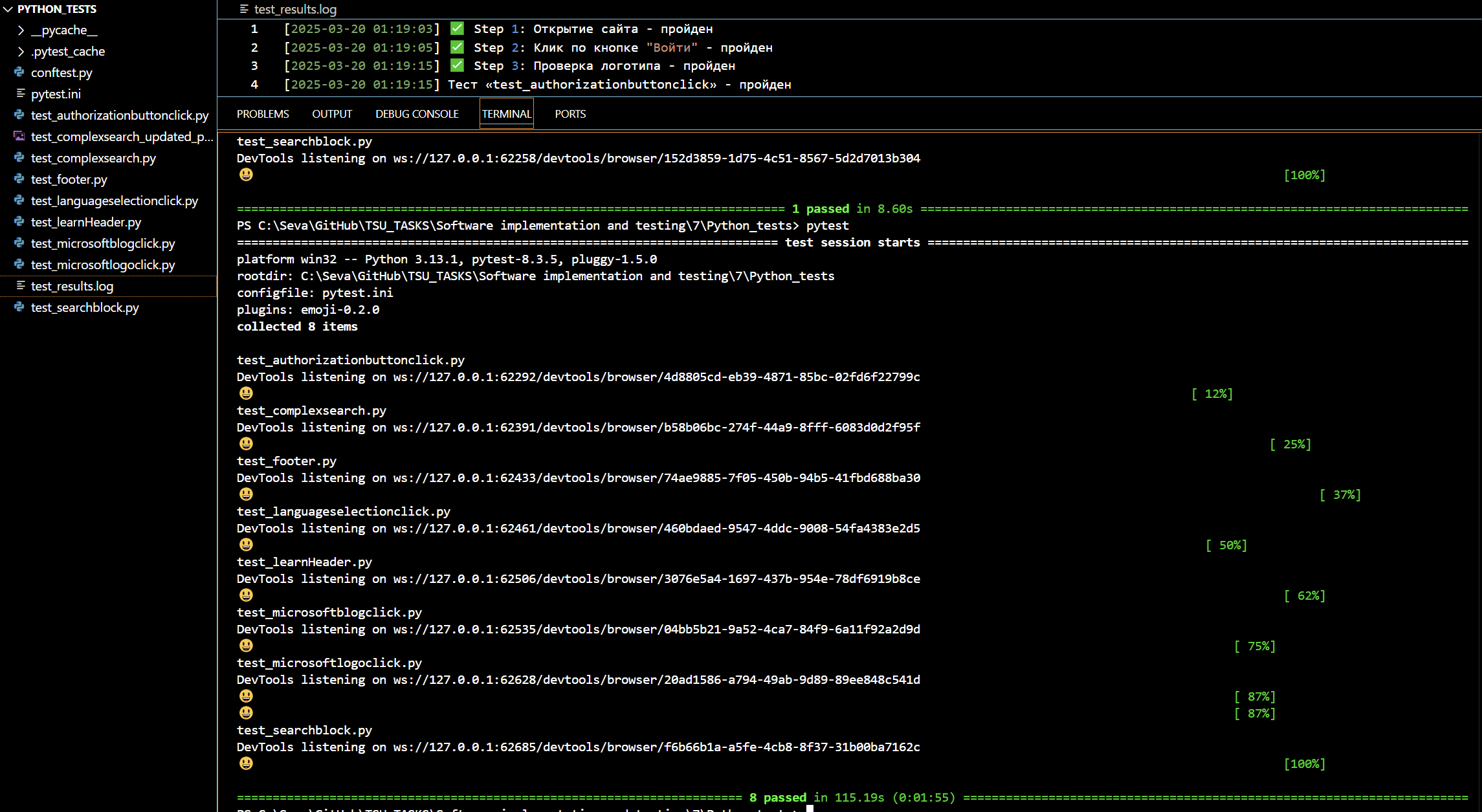
            assert header.is\_displayed(), "Уникальный элемент страницы (id=tutorial-work-with-python-in-visual-studio) не найден"

            return header

        # Step 5: Проверка уникального элемента на странице

        step\_tracker.step(check\_element)

Результат тестов:



Поскольку во время своей работы pytest блокирует поток вывода в терминал, пришлось создавать отдельный скрипт, который будет универсально использоваться во всех предыдущих скриптах и выводить в файл результаты выполнения каждого тест-степа. Как итог создается файл «test\_results.log» с вот таким содержимым после выполнения всех автотестов:  
