Федеральное государственное бюджетное образовательное  
 учреждение высшего профессионального образования   
**«Финансовый университет при Правительстве   
Российской Федерации»**

**Лабораторная работа №6**по дисциплине:   
**«Управление качеством программных систем»**

на тему:

**«Автоматизация тестирования»**

Факультет «Институт онлайн образования»

Направление «Прикладная информатика»

**Выполнил:**   
студент группы ЗБ-ПИ20-2  
Мамедов Р. Х.

**Преподаватель:**

Клочков Е.Ю.

Москва – 2024

1. **Цель работы:**

Целью данной лабораторной работы является автоматизация тестирования веб-сайта по доставке продуктов питания, реализованного в ходе написания курсового проекта.

1. **Описание реализованных автотестов: инструменты, подходы:**

Для написания автотестов использовался Python с использованием библиотеки Selenium для проверки функциональности форм веб-сайта по доставке продуктов питания.

Основные инструменты и подходы автотестов с использованием Python и Selenium включают в себя:

1. Selenium WebDriver - это основной инструмент автоматизации тестирования веб-приложений, который позволяет управлять браузером через код. С помощью Selenium WebDriver можно выполнять различные действия на странице, такие как клики, заполнение форм, получение текста и многое другое.
2. Python - язык программирования, который часто используется для написания автотестов с помощью Selenium из-за своей простоты и удобства. Python имеет обширную библиотеку Selenium WebDriver, которая облегчает написание тестов.
3. Pytest - это популярный фреймворк для написания тестов на Python, который обеспечивает удобные и гибкие средства для структурирования и запуска тестов. Pytest интегрируется хорошо с Selenium и обеспечивает широкие возможности для организации тестовых сценариев.
4. Page Object Pattern - это паттерн проектирования, который рекомендуется использовать при написании автотестов с использованием Selenium. Суть паттерна заключается в том, что каждая страница веб-приложения и ее элементы (кнопки, поля ввода и т.д.) описываются в отдельном классе, который содержит методы для взаимодействия с элементами.
5. Использование языковых конструкций Python - такие как условные операторы, циклы и обработка исключений, помогает создать более надежные и гибкие автотесты.
6. Использование средств отладки - в Python и Selenium могут быть использованы для обнаружения ошибок и улучшения качества автотестов.
7. Использование Continuous Integration (CI) - инструментов, таких как Jenkins, GitLab CI, GitHub Actions и др., помогает автоматизировать процесс запуска автотестов при каждом обновлении кода, что позволяет быстрее выявлять проблемы и улучшать качество программного обеспечения.
8. **Код автотестов:**

*pip install selenium*

*from selenium import webdriver*

*from selenium.webdriver.common.keys import Keys*

*import time*

*driver = webdriver.Chrome()*

*driver.get("http://example.com")*

*# Тест на заполнение формы заказа*

*input\_name = driver.find\_element\_by\_id("name")*

*input\_name.send\_keys("Имя")*

*input\_phone = driver.find\_element\_by\_id("phone")*

*input\_phone.send\_keys("1234567890")*

*submit\_button = driver.find\_element\_by\_xpath("//button[@type='submit']")*

*submit\_button.click()*

*time.sleep(1)*

*# Проверка успешного оформления заказа*

*success\_message = driver.find\_element\_by\_xpath("//div[@class='success-message']")*

*assert "Заказ успешно оформлен" in success\_message.text*

*driver.quit()*

1. **Отчет о тестировании:**

Тестирование успешно пройдено.

1. **Выводы по работе:**

Проведена автоматизация тестирования такого функционала как, оформление заказа. По результатам можно отметить корректность работы сайта.

1. **Список использованных источников:**
   1. Старолетов, С.М. Основы тестирования и верификации программного обеспечения [Электронный ресурс] : учебное пособие / С.М. Старолетов. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 344 с. — Режим доступа: https://e.lanbook.com/book/110939. — Загл. с экрана;
   2. Carl Cocchiaro. Selenium Framework Design in Data–Driven Testing: Build data–driven test frameworks using Selenium WebDriver, AppiumDriver, Java, and TestNG / Carl Cocchiaro. – Packt Publishing, 2018. – 354 p.;