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 гр. ЗБ-ПИ20-2  
Яппаров А.Э.

**Преподаватель:** Ассистент ДАДиМО ФИТиАБД

Клочков Е.Ю.

Москва – 2024

1. **Цель работы:**

Цель работы - разработать и реализовать автотесты для проверки функциональности программной системы "Online Shopping Platform" с использованием инструментов и подходов, обеспечивающих сопровождаемость, надежность и структурированность, а также эмулирующих поведение пользователя при взаимодействии с элементами пользовательского интерфейса, и оценить результаты выполненного сценария.

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

Реализованные автотесты для проверки функциональности программной системы "Online Shopping Platform" были разработаны с использованием следующих инструментов и подходов:

* Инструмент: Selenium WebDriver - инструмент для автоматизации веб-браузера, позволяющий эмулировать поведение пользователя при взаимодействии с элементами пользовательского интерфейса.
* Подход: Modular testing - подход, который позволяет разбивать тесты на отдельные модули, каждый из которых проверяет конкретную функциональность системы. Это обеспечивает сопровождаемость и структурированность тестов.
* Язык программирования: Python - язык программирования, который был использован для написания автотестов. Python обеспечивает надежность и простоту в использовании.
* Approach: Page Object Model - подход, который позволяет абстрагироваться от конкретной реализации страницы и сосредоточиться на функциональности, которую она предоставляет. Это обеспечивает сопровождаемость и структурированность тестов.

Автотесты были реализованы в виде отдельного приложения, которое эмулирует поведение пользователя при взаимодействии с элементами пользовательского интерфейса, и обеспечивают контроль результатов выполненного сценария.

1. **Код автотестов:**

***Тестовый пример №1: вход в систему.***

* Инструмент: Selenium WebDriver.
* Подход: модульное тестирование.
* Код:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

**Код тестового примера №1.**

***Тестовый пример №2: добавить в корзину.***

* Инструмент: Selenium WebDriver.
* Подход: модульное тестирование.
* Код:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Код тестового примера №2.**

1. **Отчет о тестировании:**

***Тестовый пример №1: вход в систему.***

Результат: пройден.

Примечания: успешный вход в систему с действительными учетными данными.

***Тестовый пример №2: добавить в корзину.***

Результат: сбой.

Примечания: товар не добавлен в корзину.

Дефект: #001 - Товар не добавляется в корзину.

1. **Выводы по работе:**

Автотесты были разработаны и выполнены для платформы онлайн-покупок. Тесты охватывали функциональность входа в систему и добавления товаров в корзину. Результаты показали, что функция входа в систему работает корректно, но функция добавления в корзину имеет проблему. Был обнаружен дефект #001, о котором было сообщено в отчете о тестировании.

1. **Список использованных источников:**
   1. Документация Selenium WebDriver.
   2. Документация Python.