**Ручное и автоматизированное тестирование веб-сайта**

**"Product Store" без технического задания**

Программа:

Автоматизация тестирования. Специалист

Бондарь Георгий Михайлович

Москва

2024

**Содержание:**

**Введение**

**Глава 1. Теоретические основы тестирования веб-приложений**

1.1 Что такое веб-приложения

1.2 Понятие тестирования и роль тестировщика

1.3 Требования к тестированию: определение технического задания и методы проведения ручного тестирования

1.4 Этапы ручного тестирования веб-приложений

1.5 Этапы автоматизированного тестирования веб-приложений

1.6 Обзор инструментов для тестирования веб-приложений

**Глава 2. Подготовка к тестированию веб-приложения**

2.1 Разработка тест-кейсов

2.2 Создание баг-репортов

2.3 Использование системы контроля версий Git

**Глава 3. Проведение тестирования и разработка предложений по улучшению Интернет-магазина Product Store**

3.1 Тестирование функциональности

3.2 UX тестирование

3.3 UI тестирование

3.4 Кроссбраузерное тестирование

3.5 Автоматизированное тестирование

3.6 Анализ результатов тестирования и разработка предложений по улучшению

**Заключение**

**Список используемой литературы**

**Приложения**

# Введение:

Тема проекта: **“Тестирование и улучшение веб-приложения интернет-магазина Product Store”**

Цель проекта: **Оценить качество веб-приложения интернет-магазина Product Store с помощью комплексного тестирования и предложить улучшения для повышения его эффективности и удобства использования.**

Задачи проекта:

1. Изучить теоретические основы тестирования веб-приложений.
2. Разработать тест-кейсы для тестирования веб-приложения.
3. Создать баг-репорты на основе результатов тестирования.
4. Провести тестирование функциональности, UX, UI, кроссбраузерное и автоматизированное тестирование веб-приложения.
5. Проанализировать результаты тестирования и разработать предложения по улучшению веб-приложения.

Проблема, которую решает проект: **Недостаточное качество веб-приложения интернет-магазина Product Store, которое может привести к ухудшению пользовательского опыта, снижению продаж и ущербу для репутации бренда.** Проект направлен на выявление и устранение проблем, а также на предложение улучшений для повышения качества веб-приложения.

**Инструменты:** Google.docs, Dev Tools, Git, Pycharm

**Состав команды**: Бондарь Георгий Михайлович (Тестировщик)

**Глава 1. Теоретические основы тестирования веб-приложений**

**Что такое веб-приложения?**

**Понимание веб-приложений** Веб-приложение представляет собой тип программного обеспечения, который работает внутри веб-браузера. Это инструмент, который компании используют для обмена информацией и предоставления услуг на расстоянии. Веб-приложения обеспечивают удобство и безопасность взаимодействия с клиентами. Обычные функции веб-сайта, такие как корзины для покупок, поиск и фильтрация товаров, обмен мгновенными сообщениями и новостные ленты социальных сетей, на самом деле являются веб-приложениями. Они предоставляют доступ к сложным функциям без необходимости установки или настройки программного обеспечения.

**Преимущества использования веб-приложений** Веб-приложения предлагают множество преимуществ, благодаря которым они стали неотъемлемой частью предложений большинства крупных компаний. Вот некоторые из ключевых преимуществ веб-приложений. Доступность Веб-приложения доступны из любого веб-браузера и на различных устройствах, как личных, так и рабочих. Команды, находящиеся в разных местах, могут получить доступ к общим документам, системам управления контентом и другим бизнес-услугам через веб-приложения на основе подписки. Эффективность разработки Процесс разработки веб-приложений относительно прост и экономически выгоден для компаний. Малые команды могут достичь коротких циклов разработки, что делает веб-приложения эффективным и доступным способом создания компьютерных программ. Более того, поскольку одна и та же версия работает во всех современных браузерах и устройствах, не требуется создавать разные версии для разных платформ. Удобство для пользователя Веб-приложения не требуют загрузки пользователем, что делает их легкодоступными и избавляет от необходимости обслуживания конечных пользователей и ограничения емкости жесткого диска. Веб-приложения автоматически получают обновления программного обеспечения и безопасности, что означает, что они всегда актуальны и меньше подвержены риску нарушения безопасности. Масштабируемость Компании, которые используют веб-приложения, могут добавлять пользователей по мере необходимости, без необходимости в дополнительной инфраструктуре и дорогостоящем оборудовании. Большинство данных веб-приложений хранится в облаке, что означает, что вашей компании не придется инвестировать в дополнительные объемы хранения для работы веб-приложений.

**Некоторые распространенные веб-приложения** Есть множество различных типов веб-приложений. Вот некоторые из наиболее популярных. Веб-приложения для совместной работы Веб-приложения для совместной работы позволяют членам команды получать доступ к документам, общим календарям, службам мгновенного обмена сообщениями и другим корпоративным инструментам. Веб-приложения для электронной коммерции Веб-приложения для электронной коммерции, такие как Amazon.com, позволяют пользователям просматривать, искать и оплачивать товары в Интернете. Веб-приложения электронной почты Приложения веб-почты широко используются корпоративными и персональными пользователями для доступа к своей электронной почте. Они часто включают другие средства коммуникации, такие как обмен мгновенными сообщениями и видеовстречи. Веб-приложения для онлайн-банков Бизнес- и индивидуальные пользователи широко используют веб-приложения онлайн-банков для доступа к своим счетам и другим финансовым продуктам, таким как кредиты и ипотека. Техническая документация Вы можете использовать веб-приложения для создания и обмена технической документацией, такой как руководства пользователя, руководства по эксплуатации и спецификации устройств Как работают веб-приложения? Веб-приложения имеют архитектуру клиент-сервер. Их код разделен на два компонента: скрипты на стороне клиента и скрипты на стороне сервера.  
Архитектура на стороне клиента Скрипт на стороне клиента отвечает за функциональность пользовательского интерфейса, например, кнопки и выпадающие поля. Когда конечный пользователь нажимает на ссылку веб-приложения, веб-браузер загружает скрипт на стороне клиента и отображает графические элементы и текст для взаимодействия с пользователем. Например, пользователь может читать контент, смотреть видео или заполнять данные в форме контакта. Такие действия, как нажатие кнопки отправки, передаются на сервер в виде клиентского запроса. Архитектура на стороне сервера Скрипт на стороне сервера обрабатывает данные. Сервер веб-приложений обрабатывает запросы клиентов и отправляет ответ. Обычно запросы касаются получения дополнительных данных или изменения или сохранения новых данных. Например, если пользователь нажимает на кнопку Подробнее, сервер веб-приложений отправляет контент обратно пользователю. Если пользователь нажмет кнопку Отправить, сервер приложений сохранит данные пользователя в базе данных. В некоторых случаях сервер завершает запрос данных и отправляет полную HTML-страницу обратно клиенту. Это называется рендерингом на стороне сервера. В чем разница между веб-приложением и веб-сайтом? Когда Интернет только появился, веб-сайты обладали значительно меньшей функциональностью, чем веб-приложения. Они были способны доставлять информацию пользователям только посредством статического контента. Нужно было устанавливать и запускать программное обеспечение со сложной функциональностью. Веб-приложения были созданы для того, чтобы преодолеть разрыв между программным обеспечением и статичными сайтами. Они обладали функциональностью и интерактивными пользовательскими элементами, подобно программному обеспечению, но предоставлялись с помощью URL-адреса веб-браузера.

Однако технологии веб-разработки значительно продвинулись с тех пор. Большинство современных веб-сайтов теперь являются сложными веб-приложениями.

**Веб-приложения против встроенных приложений** Встроенное приложение - это компьютерная программа, разработанная специально для определенной пользовательской среды. Мобильные приложения являются одним из наиболее распространенных типов встроенных приложений, которые разрабатываются с использованием специализированного программного кода. В отличие от веб-приложений, встроенные или мобильные приложения загружаются пользователем на его мобильное устройство, обычно через магазины приложений. Доступ к встроенным приложениям возможен только на том устройстве, на которое они были загружены, и пользователи несут ответственность за загрузку любых обновлений.

Разработчикам приходится создавать различные версии своих мобильных приложений, если они хотят сделать их доступными для разных операционных систем, таких как iOS или Android, обеспечивая максимально широкий охват. Разработчики также должны убедиться, что их мобильные приложения соответствуют стандартам магазинов приложений, иначе их заявки могут быть отклонены.

В целом, веб-приложения и встроенные приложения имеют свои преимущества и недостатки, и выбор между ними зависит от конкретных потребностей и целей бизнеса или пользователя. Однако с учетом все большей популярности облачных технологий и мобильного интернета, веб-приложения становятся все более важными инструментами для бизнеса и повседневной жизни.

Практическая часть:

Тест Кейсы:

<https://docs.google.com/spreadsheets/d/1A5vsqnp0d7VgZdIEQvn1M-RxkFlzYGeuxInIxxXfxxQ/edit?usp=sharing>

Кроссбраузерное тестирование:

-<https://docs.google.com/spreadsheets/d/1OTe9Njzfr1J-qDFZyWFzvzvXTGndQ-86UyjOZOBvpd8/edit?usp=sharing>

Ссылка на гитхаб - <https://github.com/Gbon2/Automation_Testing.git>

сайт для тестирования -<https://www.demoblaze.com/index.html>