**Отчет по самостоятельной работе №9**

**по дисциплине МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил: студент

группы 319

Меркушин Евгений Романович

Дата 17.11.2024

**Цель работы**

Цель данной работы – научиться тестировать пользовательский интерфейс программного приложения, используя инструменты среды разработки.

**Основная структура задания**

# Задание 1: Проведение тестирования интерфейса

# Шаг 1: Использование инструментов среды разработки

1. **Откройте браузер** и используйте встроенные инструменты разработчика (DevTools). В Google Chrome это можно сделать через меню "View" → "Developer" → "Developer Tools" или нажав F12.
2. **Проверьте консоль браузера**. Открыв DevTools, перейдите во вкладку "Console". Здесь вы сможете увидеть любые ошибки JavaScript, которые могут возникать при загрузке страницы или взаимодействиях пользователя.
3. **Используйте режим инспектора элементов**. Перейдя во вкладку "Elements", вы можете исследовать HTML-код страниц и CSS-стили, чтобы выявить возможные проблемы с версткой.
4. **Тестируйте взаимодействия пользователя**. Попробуйте различные действия на сайте: клики, ввод данных, переходы между страницами. Обратите внимание на то, как реагирует интерфейс, нет ли задержек или ошибок.

# Шаги 2: Исправление выявленных ошибок

1. Если обнаружены ошибки в коде, откройте файлы проекта в редакторе кода (например, Visual Studio Code) и внесите исправления.
2. После внесения изменений сохраняйте их и проверяйте результат в браузере. Повторно открывайте страницу, очищая кэш (Ctrl + F5), чтобы убедиться, что изменения вступили в силу.

# Задание 2: Проверка отзывчивости интерфейса

# Шаг 1: Проверить корректность работы интерфейса на разных разрешениях экранов

1. Используйте встроенный инструмент для эмуляции различных разрешений экрана в DevTools. Для этого нажмите кнопку "Toggle device toolbar" (иконка мобильного устройства) в верхней части окна DevTools.
2. Выберите несколько популярных разрешений экранов:
   * Мобильный телефон (например, iPhone X – 375x812px)
   * Планшет (iPad Pro – 1024x1366px)
   * Десктоп (1920x1080px)
3. Просматривайте сайт на каждом разрешении, обращая внимание на следующие аспекты:
   * Корректное масштабирование контента
   * Отсутствие перекрытий элементов друг другом
   * Удобство навигации

# Шаг 2: Подготовка отчета о результатах тестирования

1. Создайте документ (например, в формате Markdown или Word), где опишите все найденные проблемы с адаптивностью сайта.
2. Включите скриншоты проблемных мест, а также предложите решения для их устранения.

# Задание 3: Анализ пользовательского опыта

# Шаг 1: Провести анализ удобства использования интерфейса (UX) на основе обратной связи пользователей

1. Соберите обратную связь от реальных или потенциальных пользователей. Это может быть сделано через опросы, интервью или фокус-группы.
2. Проанализируйте полученные данные. Обратите особое внимание на такие моменты, как:
   * Легкость поиска товаров
   * Простота оформления заказа
   * Интуитивная понятность интерфейсов

# Шаг 2: Внедрение необходимых улучшений в интерфейс

1. На основании анализа составьте список рекомендаций по улучшению UX.
2. Реализуйте эти рекомендации в проекте. Например, добавьте подсказки пользователям, упростите процесс регистрации, улучшите фильтрацию товаров.

# Задание 4: Тестирование на доступность

# Шаг 1: Соответствие стандартам доступности (WCAG)

1. Проверьте сайт на соответствие рекомендациям WCAG (Web Content Accessibility Guidelines):
   * Используйте инструменты вроде WAVE (Web Accessibility Evaluation Tool) или Lighthouse в DevTools для автоматического аудита доступности.
2. Проведите ручное тестирование:
   * Убедитесь, что контент доступен для людей с различными нарушениями зрения (использование контраста, возможность увеличения текста).
   * Проверьте доступность для пользователей с ограничениями моторики (возможность управления сайтом только клавиатурой).

# Шаг 2: Доступность для пользователей с ограниченными возможностями

1. Протестируйте сайт с помощью программ чтения с экрана (например, NVDA для Windows или VoiceOver для macOS).
2. Убедитесь, что все элементы интерфейса имеют правильные метки и описания, чтобы программы могли правильно озвучивать содержимое страницы.

# Задание 5: Автоматизация тестирования интерфейса

# Шаг 1: Разработка автоматизированных тестов

1. Выберете подходящий фреймворк для автоматизации тестирования интерфейсов, например Selenium WebDriver.
2. Напишите скрипты для проверки основных сценариев поведения пользователя:
   * Поиск товара
   * Добавление товара в корзину
   * Оформление заказа

# Шаг 2: Проведение тестирования и подготовка отчета

1. Запустите написанные тесты и убедитесь, что они проходят успешно.
2. Составьте отчет о проведенных тестах, включающий результаты каждого теста и описание обнаруженных ошибок.

**Вывод**

Из приведенного плана действий можно сделать следующий вывод: успешное создание качественного и удобного интернет-магазина требует комплексного подхода к тестированию и улучшению интерфейса. Важно не только проверить работоспособность и визуальную составляющую сайта, но и учесть удобство использования (UX), доступность для пользователей с ограниченными возможностями, а также автоматизировать процессы тестирования для повышения надежности и эффективности работы ресурса.