Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 1. Формирование требований к информационной системе «Интернет-магазин электроники ElectroShop»

Студент: Пшенко А. Ф.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

# Введение

В условиях стремительного развития рынка интернет-торговли возникает острая необходимость в создании удобных и надежных платформ для онлайн-покупок. Магазин электроники представляет собой эффективное решение, которое позволяет пользователям легко приобретать различные товары, такие как смартфоны, ноутбуки и аксессуары, не выходя из дома.

Основная цель данного проекта заключается в разработке комплексного веб-приложения, предназначенного для автоматизации процесса покупки электроники. Приложение должно обеспечивать удобный доступ к каталогу товаров, фильтрацию по параметрам, управление корзиной и оформление заказов. Это позволит улучшить качество обслуживания клиентов и повысить эффективность работы магазина.

Цель проекта – автоматизировать процесс покупки товаров и взаимодействие пользователей с магазином. Платформа позволит пользователям легко находить нужные товары, добавлять их в корзину и совершать покупки онлайн. Администраторы магазина смогут управлять каталогом товаров.

Целевая аудитория включает обычных пользователей и бизнес-потребителей, которым необходимо быстро и удобно приобретать электронику. В современном мире онлайн-торговля становится все более популярной, что делает данный проект актуальным и востребованным.

Результатом выполнения данной работы станет полнофункциональная платформа, обеспечивающая возможность просмотра товаров, их фильтрации, добавления в корзину и оформления заказов.

# Основания для разработки

Разработка платформы для интернет-магазина электроники обусловлена актуальными требованиями рынка и стремительным развитием цифровых технологий, которые значительно влияют на процесс взаимодействия покупателей и продавцов. В условиях современного мира удобство, скорость и простота использования интернет-сервисов становятся важнейшими факторами для удовлетворения потребностей клиентов. Компании, предлагающие возможность онлайн-покупок, получают ощутимые конкурентные преимущества, привлекая внимание потребителей благодаря повышенному уровню комфорта и доступности своих услуг.

Проект реализуется в рамках курсового проекта по дисциплине «Программирование Web-сервисов» по специальности «Программное обеспечение информационных технологий» на факультете информационных технологий Белорусского Государственного Технологического Университета.

Создание такой платформы обусловлено необходимостью удовлетворить растущие ожидания пользователей, которые ценят возможность делать покупки в любое время, не привязываясь к конкретным временным и территориальным рамкам. Интернет-магазины предоставляют удобный доступ к широкому ассортименту товаров и облегчают процесс совершения покупок, делая его более адаптивным к индивидуальным потребностям покупателей. Автоматизация торговых процессов также упрощает работу компаний, снижает операционные затраты и минимизирует риски, связанные с человеческим фактором.

В условиях растущей конкуренции на рынке электронной коммерции, внедрение подобных решений позволяет компаниям повысить свою конкурентоспособность, улучшить имидж и предоставить покупателям уникальный опыт взаимодействия с брендом.

Таким образом, разработка платформы для интернет-магазина электроники способствует оптимизации внутренних процессов компаний. Это позволяет более эффективно управлять своими ресурсами, улучшать взаимодействие с клиентами и поставщиками, а также снижать затраты на организацию торговых операций. В конечном итоге, внедрение такой платформы способствует повышению конкурентоспособности компании, ее привлекательности для покупателей и, как следствие, увеличению доходов за счет роста продаж и сокращения операционных издержек.

1. **Назначение разработки**

Целью разработки является создание современного цифрового инструмента, который обеспечивает эффективное и удобное взаимодействие между покупателями и продавцами. В условиях стремительного развития технологий и увеличения спроса на онлайн-покупки, внедрение таких решений позволяет значительно улучшить процесс совершения покупок, предоставляя пользователям гибкий доступ к товарам, акциям и услугам в любое время и из любой точки мира.

Важной особенностью платформы является возможность персонализации покупок, что значительно упрощает процесс выбора и приобретения товаров для клиентов. Покупатели могут выбирать продукты, которые соответствуют их интересам и потребностям. Более того, пользователи получают доступ к детализированной информации о каждом товаре, включая описание, характеристики, отзывы и других покупателей, что способствует более осознанному выбору продукции.

Особое внимание уделяется управлению процессами продаж и отслеживанию заказов. Пользователи могут легко зарегистрироваться, выбрать товары и оформить покупку. Магазин также уведомляет покупателей о статусе заказа, сроках доставки, специальных предложениях и других важных событиях, что помогает организовать покупки и повысить уровень удовлетворенности.

Таким образом, интернет-магазин электроники играет ключевую роль в повышении эффективности торговых процессов и уровня удовлетворенности клиентов. С его помощью компании могут оптимизировать свои процессы, снизить нагрузку на сотрудников и обеспечить более высокое качество обслуживания, в то время как покупатели получают доступ к удобному и гибкому инструменту для совершения покупок. Это решение направлено на создание единой торговой экосистемы, которая объединяет интересы как компаний, так и клиентов, повышая конкурентоспособность бизнеса и улучшая потребительский опыт.

# Требования к программному изделию

# Требования к функциональным характеристикам

Функциональные характеристики приложения отличаются в зависимости от роли пользователя. В приложении доступны следующие роли:

* администратор;
* гость;
* пользователь.

# Требования к клиентской части

Клиентская часть должна поддерживаться в любом из современных браузеров, в онлайн-режиме. Клиентская часть должна поддерживать корректный режим работы в любом разрешении экрана, в том числе на мобильных устройствах. Клиентская часть должна быть реализована с помощью React.

Интерфейс веб-приложения должен быть логически завершенным и интуитивно понятным. Также интерфейс обязан обеспечивать следующий функционал:

Функционал для администратора:

* создание и редактирование карточек товаров;
* управление пользователями.

Функционал для пользователя:

* регистрация и авторизация;
* просмотр каталога;
* поиск и фильтрация по каталогу товаров;
* добавление товаров в корзину;
* удаление товаров из корзины;
* покупка товаров.

Функционал для гостя:

* регистрация и авторизация;
* просмотр каталога;
* поиск и фильтрация по каталогу товаров;

Основные страницы:

* страница регистрации и авторизации;
* главная страница с каталогом товаров и фильтрацией;
* страница поиска товаров;
* страница карточки товара;
* страница просмотра корзины;
* страница оформления заказа.

# Требования к серверной части

Серверная часть должна быть реализована с помощью Node.js и должна обеспечивать следующий функционал:

* предоставление API для клиентской части приложения;
* защиту и шифрование персональных данных;
* разграничение прав пользователей по ролям;
* валидацию введенных пользователем данных;
* доступ к базе данных;
* обработку ошибок.

# Требования к надежности

Программное средство для выполнения требований надежности должно:

* хранить пароль пользователя в зашифрованном виде;
* ограничить возможность создания более одной учетной записи на одну электронную почту;
* передавать данные по зашифрованному протоколу HTTPS;
* обеспечивать валидацию введенных пользователем данных;
* обеспечивать защиту личных данных пользователей;
* обеспечивать идентификацию, аутентификацию и авторизацию пользователей.

# Условия эксплуатации

Приложение должно быть спроецировано таким образом, чтобы предоставить возможность для комфортного использования всеми членами целевой аудитории – то есть людьми всех полов, возрастов и социальных положений. Для достижения этого, интерфейс приложения должен быть интуитивно понятным и удобным в использовании.

# Требования к составу и параметрам технических средств

Минимальные аппаратные требования к пользовательской машине:

* минимальное количество ядер процессора – 2;
* минимальный объем оперативной памяти – 4 Гб;
* минимальный доступный объем дискового пространства – 20 Гб.

Рекомендуемые аппаратные требования к пользовательской машине:

* минимальное количество ядер процессора – 4;
* минимальный объем оперативной памяти – 8 Гб;
* минимальный доступный объем дискового пространства – 40 Гб.

# Требования к информационной и программной совместимости

Язык программирования, используемый серверной частью – Javascript.

Язык программирования, используемый клиентской частью – JavaScript и TypeScript.

Исходный язык, используемый для написания структурированных запросов к базе данных – MongoDB Query Language (MQL).

Информационные структуры на входе и выходе для передачи данных между клиентом и сервером – JSON. Передача данных осуществляется по протоколу HTTP. Связь между сервером и СУБД – локальная.

Для связи между сервером и СУБД. используется облачная база данных Cloud MongoDB. Это решение позволяет хранить и обрабатывать данные на удаленных серверах, обеспечивая высокую доступность, отказоустойчивость и гибкость масштабирования системы. Cloud MongoDB поддерживает доступ к базе данных через безопасные соединения, что обеспечивает защиту данных от несанкционированного доступа.

1. **Требования к программной документации**

Предварительный состав программной документации представляет собой Swagger-документацию, которая используется для описания API веб-приложения. Swagger-документация обеспечивает удобное взаимодействие с программным интерфейсом и помогает разработчикам и пользователям системы легко понять структуру и функциональность API.

Программная документация должна содержать:

* описание пути, метода и названия конечной точки: каждый запрос должен быть четко задокументирован, включая URL-адрес, HTTP-метод (GET, POST, PUT, DELETE и т.д.) и название конечной точки, чтобы разработчики могли правильно обращаться к API;
* пример данных, подаваемых на вход: документация должна содержать примеры данных, которые должны быть отправлены на сервер, включая структуру, типы данных и обязательные поля, чтобы пользователи понимали, какие параметры передавать для корректного запроса.;
* пример данных, возвращаемых с сервера: для каждого запроса должны быть приведены примеры данных, которые возвращаются сервером, включая структуру ответа, возможные поля и их значения. Это помогает пользователям системы понять, какую информацию они получат в результате запроса;
* коды возвращаемых ответов и пояснения к ним: должны быть описаны все возможные коды ответов, которые возвращает сервер, такие как 200 (успешный запрос), 400 (неправильный запрос), 404 (ресурс не найден), 500 (внутренняя ошибка сервера), а также пояснения к каждому коду, чтобы разработчики знали, как интерпретировать результаты;
* дополнительное описание конечной точки, при необходимости: если какая-то конечная точка имеет особенности или дополнительные требования, которые не охвачены стандартными пунктами, они должны быть указаны в этом разделе. Это может включать ограничения по времени обработки запроса, требования к аутентификации или дополнительные параметры, которые могут быть необязательными, но влияют на поведение API.

Кроме того, документация может включать информацию о версиях API, ограничениях на частоту запросов (rate limits), а также советы по оптимальному использованию конечных точек для повышения производительности и безопасности работы системы.

1. **Стадии и этапы разработки**

Разработка должна быть проведена в 3 этапа:

* техническое задание;
* технический проект;
* внедрение.

На стадии «Техническое задание» происходит детальное планирование и формулирование требований к программному продукту. Составляется техническое задание (ТЗ), которое должно включать в себя всю необходимую информацию о целях, назначении, функциях и требованиях к системе. Техническое задание проходит этапы разработки, согласования с заказчиком и утверждения. Этот документ служит основой для всех последующих этапов разработки, поэтому его полнота и точность являются критически важными. Без утвержденного ТЗ невозможен переход к следующему этапу.

На стадии «Технический проект» начинается непосредственная разработка программного обеспечения. Этот этап включает несколько ключевых шагов:

* разработка программы;
* разработка программной документации;
* испытание программы.

На этапе разработки программы программа создается на основе утвержденного технического задания. Здесь выполняется непосредственно программирование, включающее создание кода, интеграцию необходимых модулей и функциональных компонентов. Программирование сопровождается процессами отладки, которые обеспечивают исправление ошибок и корректную работу всех функций. Также на этом этапе могут проводиться предварительные тестирования отдельных модулей.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101--77.

На этапе испытания программы проводится тестирование всей системы, чтобы убедиться в правильной работе всех компонентов и модулей, а также соответствие системы заданным требованиям. Проводятся функциональные, нагрузочные и интеграционные тесты, чтобы проверить работоспособность программы в различных сценариях. Испытания должны выявить и устранить все возможные ошибки перед переходом к следующему этапу.

1. **Описание информационной вербальной модели для интернет-магазина электроники**

Объекты информационной вербальной модели для платформы онлайн-курсов могут быть представлены следующим образом:

1. Пользователь:

* Описание: представляет собой физическое лицо, использующее платформу для покупки товаров. Пользователи могут просматривать каталог товаров, фильтровать их по критериям, добавлять товары в корзину и оформлять заказы.
* Атрибуты: имя, фамилия, адрес электронной почты, номер телефона, адрес доставки.
* Связи: пользователи связаны с Заказами (каждый пользователь может совершать несколько покупок, которые сохраняются в его истории заказов), а также с Корзиной (пользователи добавляют товары в корзину перед покупкой) и Отзывы (пользователи могут оставлять отзывы о товарах).

1. Товар:

* Описание: представляет собой единицу продукции, которая доступна для покупки на платформе. Товары могут быть различными по категории, цене и характеристикам (смартфоны, ноутбуки, аксессуары и т.д.).
* Атрибуты: название товара, описание, цена, количество на складе, категория.
* Связи: товар связан с Каталогом (все товары доступны в каталоге для просмотра пользователями), с Корзиной (товары добавляются пользователями перед покупкой), а также с Заказами (товары, которые были куплены пользователями, отображаются в их заказах).

1. Корзина:

* Описание: временное хранилище для товаров, которые пользователь планирует купить. Корзина позволяет пользователю собрать несколько товаров для оформления одного заказа.
* Атрибуты: идентификатор пользователя, список товаров в корзине, общая стоимость товаров.
* Связи: корзина связана с Пользователями (каждый пользователь имеет свою корзину) и Товарами (товары добавляются в корзину перед покупкой).

1. Заказ:

* Описание: заказ представляет собой завершенную транзакцию покупки, содержащую один или несколько товаров. Включает информацию о способе оплаты и статусе выполнения заказа.
* Атрибуты: номер заказа, дата создания, статус, способ оплаты, сумма заказа.
* Связи: заказ связан с Пользователем (каждый пользователь может иметь несколько заказов), а также с Товарами (каждый заказ включает товары, купленные пользователем)

**Заключение**

Разработанная платформа для интернет-магазина электроники представляет собой современное и эффективное решение для организации онлайн-торговли. Она предоставляет пользователям удобный доступ к каталогу товаров, возможность фильтрации и поиска, управления корзиной и оформления заказа. Платформа обеспечивает гибкость и простоту использования как для покупателей, так и для администраторов, которые могут управлять каталогом товаров и следить за заказами.

Для администраторов платформа предлагает инструменты создания и редактирования товаров, мониторинга продаж и управления пользователями, что повышает общую эффективность управления магазином. Пользователи получают простой и интуитивно понятный интерфейс для покупки товаров, добавления их в корзину, оформления заказа и оплаты.

В серверной части платформы использована технология Node.js с фреймворком Express, что позволило создать высокопроизводительное и масштабируемое решение для обработки пользовательских запросов. Архитектура системы включает роутеры, контроллеры и middleware для управления данными и обеспечивания стабильной работы приложения. Взаимодействие между клиентом и сервером реализовано с использованием REST API, что позволяет оперативно и надежно обрабатывать запросы пользователей.

Для обеспечения безопасности данных платформа использует протокол HTTPS, который гарантирует защиту конфиденциальной информации пользователей и безопасность транзакций. Это помогает создать доверительные отношения между пользователями и интернет-магазином, подчеркивая заботу о защите данных и надежности покупок.

Клиентская часть разработана с использованием React, что позволило создать интерактивный и удобный пользовательский интерфейс для работы с каталогом товаров, корзиной и оформлением заказов. Применение библиотеки Axios для отправки запросов на сервер оптимизировало взаимодействие с данными, а использование Redux для управления состоянием приложения сделало процесс работы с платформой более плавным и эффективным для пользователей.

В качестве системы управления базами данных была выбрана MongoDB.

Тестирование платформы подтвердило корректность работы всех функций и соответствие заявленным требованиям. Платформа готова к использованию в коммерческих целях и представляет собой надежное и эффективное решение для организации онлайн-торговли электроникой, улучшая покупательский опыт и оптимизируя процессы для администраторов магазина.