

ЧАСТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ПЕТРОЗАВОДСКИЙ КООПЕРАТИВНЫЙ ТЕХНИКУМ  
КАРЕЛРЕСПОТРЕБСОЮЗА (ЧПОУ ПКТК)

ДИПЛОМНАЯ РАБОТА

по теме:

ЭЛЕКТРОН: РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА ЦИФРОВОЙ  
ТЕХНИКИ

Студент группы 409ИС-1

Специальность 09.02.04

Информационные системы (по отраслям) \_\_\_\_\_ Алферов В.А.

Руководитель дипломной работы,  
преподаватель \_\_\_\_\_ Жихарев М.С.

Допустить к защите:  
Председатель методической  
цикловой комиссии \_\_\_\_\_ Щепетова Е.В.

Оценка \_\_\_\_\_ Дата \_\_\_\_\_

Председатель Государственной  
экзаменационной комиссии \_\_\_\_\_ Голубник А.А.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Процесс создания веб-сайта: от планирования до запуска .....	4
1.1 Навыки для разработки .....	4
1.2 Программное обеспечение .....	5
1.3 Основные игроки рынка .....	6
2 Технические средства разработки .....	10
2.1 Языки программирования .....	10
2.2 Фреймворки .....	12
2.3 Библиотеки .....	14
3 Дизайн и разработка .....	16
3.1 Этап проектирования интерфейса .....	16
3.2 Этап верстки и разработки .....	17
ЗАКЛЮЧЕНИЕ .....	27
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	29
ПРИЛОЖЕНИЕ .....	32

## ВВЕДЕНИЕ

Сегодня сложно представить жизнь без компьютеров, смартфонов, планшетов и других устройств. Они помогают нам работать, учиться, общаться с друзьями и близкими, получать информацию и развлекаться. Особенно важную роль играет цифровая техника в образовании. Современные школьники и студенты уже не представляют свою жизнь без компьютеров и интернета. Они используют их для поиска информации, подготовки к урокам и экзаменам, общения с одноклассниками и преподавателями.

В ходе работы будет рассмотрен процесс создания сайта интернет-магазина цифровой техники с использованием современных методов и подходов к проектированию веб-приложений. Основное внимание будет уделено дизайну, верстке и написанию логики с помощью фреймворка React.

Результатом данной работы будет полноценный интернет-магазин, которая демонстрирует навыки в области создания сайтов на React.

**Актуальность проблемы:** создание сайтов как инструмент бизнеса и коммуникации.

**Цель:** разработать полноценный, удобный интернет-магазин с системой оплаты, который будет решать задачи бизнеса.

### **Задачи:**

- Проанализировать процесс создания веб-сайта;
- Ознакомиться с программным обеспечением;
- Проанализировать лидирующие интернет-магазины;
- Изучить актуальные технологии разработки;
- Разработать сайт;

## **1 Процесс создания веб-сайта: от планирования до запуска**

Разработка сайтов — это процесс создания веб-сайта, начиная с планирования и заканчивая запуском. Этот процесс включает в себя множество этапов, начиная с определения целей и аудитории сайта до разработки дизайна, программирования, тестирования и запуска.

### **1.1 Навыки для разработки**

Навыки для разработки сайтов зависят от задачи бизнеса. Если это одностраничный сайт или например, простой интернет-магазин, то для его разработки хватит и конструктора или CMS-системы.

В основном для разработки полноценного сайта, потребуются навыки:

- Дизайн;
- Иллюстрирование;
- Создание 3D-иллюстраций;
- Верстка;
- Программирование;
- Анимация интерфейсов;
- Знание фреймворков библиотек;
- И другие.

Дизайн — это первый этап в разработке сайта. Суть дизайна — спроектировать удобный и эстетичный интерфейс, который будет решать поставленные задачи бизнеса (повышать конверсию и лояльность аудитории, привлекать новую аудиторию и т.д.).

Чтобы решить эту задачу часто используют 3D-иллюстрации. Они могут наглядно показывать как выглядит продукт, как он работает и т.д.

После создания дизайна идет этап верстки. Он подразумевает перенос дизайна в код. Делается разметка элементов дизайна и их стилизация.

Чтобы оживить верстку используются языки программирования и фреймворки. Первые позволяют написать логику и функционал для статичной верстки, в то время как другие позволяют избежать написания огромного количества кода.

Анимация интерфейсов. Даже после создания полноценного сайта, он может не вызывать достаточного уровня эмоций у пользователей и не оставлять ярких воспоминаний. Это связано с тем, что в современном мире, где информация доступна в огромном количестве, пользователи становятся все более требовательными к дизайну и функциональности веб-сайтов. Чтобы привлечь и удержать внимание посетителей, дизайнеры используют различные элементы, включая анимации. Анимации в веб-дизайне вызывают эмоции у пользователей, потому что они добавляют динамику и интерактивность сайту. Они помогают создать более привлекательный и запоминающийся пользовательский опыт, делая взаимодействие с сайтом более интересным и увлекательным. Для создания анимаций используются библиотеки которые существенно упрощают и ускоряют процесс создания анимаций в коде.

## **1.2 Программное обеспечение**

Для создания веб-сайта требуется определенный набор программных инструментов. В этом разделе рассмотрены ключевые программы, которые необходимы для эффективной разработки веб-сайтов.

Для разработки сайтов необходимо иметь несколько программ:

- 1) Google Chrome (или любой другой браузер) – веб-браузер со встроенной поисковой системой Google. Главное преимущество Chrome в сравнении с другими браузерами - это его встроенные инструменты разработчика, которые включают в себя отладчик, консоль JavaScript, инструменты для анализа производительности и многое другое.

2) Visual Studio Code – бесплатный текстовый редактор, который подходит для разработки программного обеспечения. Он имеет удобный интерфейс, множество функций и расширений, которые помогают разработчикам удобно писать код, отлаживать приложения и работать с проектами. VSCode поддерживает большое количество языков программирования и интегрируется с различными инструментами разработки.

3) Node.js – это среда выполнения JavaScript, которая позволяет запускать код за пределами веб-браузера, то есть на серверной стороне. Позволяет создавать серверные приложения с помощью JavaScript, что делает его популярным инструментом для создания веб-серверов, API и других серверных приложений. Работает асинхронно, что позволяет эффективно обрабатывать большое количество запросов без блокировки потоков.

4) Figma – это онлайн-инструмент для дизайна, который позволяет создавать макеты, прототипы и дизайн-системы для веб-сайтов и мобильных приложений. Можно работать совместно с коллегами, делиться проектами и получать обратную связь в реальном времени. Сегодня для создания веб-дизайна это стандарт. В этой программе можно делать практически все - от создания прототипов, до рисования иконок и иллюстраций.

5) Cinema 4D – программа для создания 3D-графики и анимации. Для веб-дизайнера это инструмент, который позволяет создавать иллюстрации и анимации для веб-сайтов. Топовые дизайн-студии используют 3D для своих проектов. Среди них можно выделить AVA Digital, UPROCK и др.

### **1.3 Основные игроки рынка**

Основными участниками российского рынка цифровой техники являются производители устройств, дистрибьюторы, розничные сети и интернет-магазины. Среди производителей можно выделить такие компании,

как «Apple», «Samsung», «Huawei», «Xiaomi», «Sony», «Asus», «Acer», «Dell» и другие. Они занимаются созданием и выпуском различных видов цифровой техники, включая смартфоны, планшеты, ноутбуки, компьютеры и другие устройства.

Дистрибьюторы играют важную роль в распределении продукции между розничными сетями и интернет-магазинами. В России среди них можно выделить компании «Merlion», «OCS Distribution», «Marvel», «diHouse» и другие.

Розничные сети и интернет-магазины осуществляют продажу цифровой техники конечным потребителям. Среди крупных в России можно выделить «М.Видео», «Эльдорадо», «DNS», «Ситилинк» и другие. Они также играют значительную роль в продаже цифровой техники, среди которых можно выделить «Wildberries», «Ozon», «Яндекс.Маркет» и другие.

Важно отметить, что конкуренция на рынке цифровой техники в России достаточно высока, поэтому все участники стремятся предложить наиболее выгодные условия для покупателей, включая широкий выбор товаров, привлекательные цены, акции и программы лояльности.

Критерии для анализа сайтов:

- Дизайн сайта;
- Ассортимент товаров;
- Качество информации о товарах;
- Процесс оформления заказа.

1) «М.Видео» — российская компания, рисунок 1, владеющая крупнейшей сетью розничных магазинов электроники и бытовой техники. Она является лидером среди российских розничных сетей по продаже электроники и бытовой техники в России.

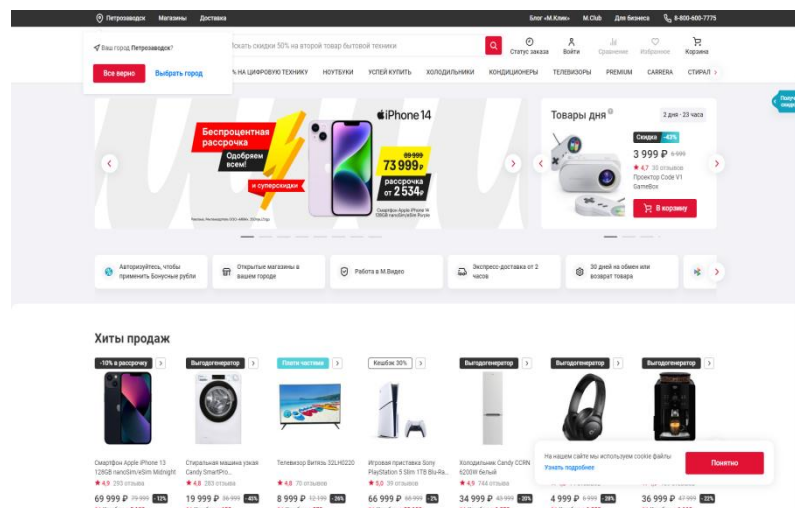


Рисунок 1 – Магазин «М.Видео»

- **Дизайн.** Сайт имеет интуитивно понятный интерфейс. Все расположено на привычных местах. Цветовая гамма позволяет делать акценты на важных элементах. На странице каталога располагается удобный фильтр [8]. Товары можно сравнить с помощью соответствующего функционала, а также посмотреть статус заказа прямо на сайте. Характеристика товара представлена в удобном виде.

- **Ассортимент.** Магазин предоставляет широчайший выбор товаров - от посуды до техники умного дома.

- **Информация о товарах.** У любого товара есть характеристики, четкое описание и удобная система отзывов [19]. Можно посмотреть наличие в магазинах, сразу узнать дату доставки, размер кэшбека и способы получения.

- **Оформление заказа.** Для совершения покупки на сайте можно войти с помощью номера телефона или не входить вообще. Главное, это выбрать способ получения (самовывоз или курьер) и оплатить. Это делает процесс очень удобным и быстрым. Также можно использовать промокод на скидку.

2) restore: — бренд и одноименное название розничной сети, рисунок 2, монобрендовых специализированных магазинов формата Apple Premium Reseller в России и Европе.



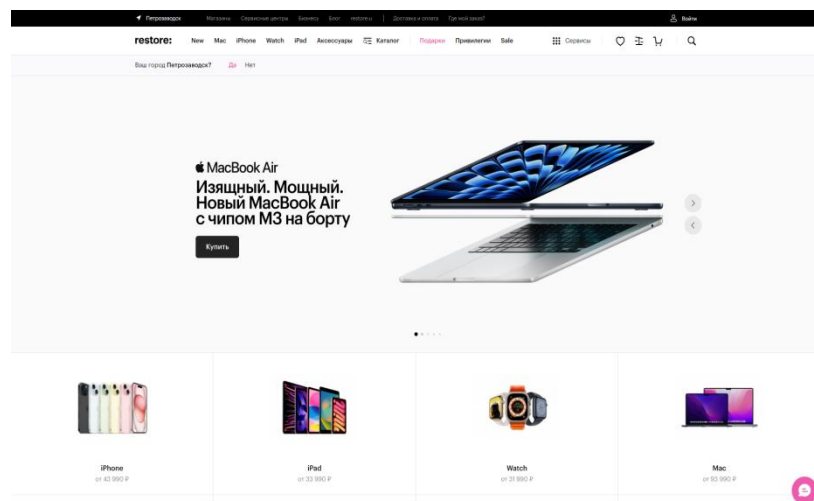


Рисунок 2 – Магазин «restore»

- Дизайн. Сайт выполнен в минималистичном стиле. Весь акцент сделан на контенте. Большие изображения лучше привлекают внимание пользователей. На странице товара есть вся необходимая информация. Характеристика товара представлена в виде удобной таблицы.
- Ассортимент. Магазин предоставляет большой выбор техники разных брендов. От умных скакалок, до новейшей техники Apple.
- Информация о товарах. Можно оплатить с помощью Яндекс Пэй или Сплит. Можно сразу узнать подходящие к товару аксессуары.
- Оформление заказа. При оформлении можно выбрать варианты гарантий и аксессуары. Есть возможность оплаты бонусной картой, а также ввести промокод для скидки. Присутствует несколько способов оплаты, один из них - при получении, что очень удобно. Доставка курьером бесплатная.

## **2 Технические средства разработки**

Сайты являются неотъемлемой частью современного цифрового мира, предоставляя доступ к информации, продуктам и услугам в Интернете. Но что лежит в основе их создания? Понимание технологий, используемых для разработки, имеет решающее значение для создания эффективного и привлекательного онлайн-присутствия.

Они создаются с использованием различных языков программирования, которые определяют структуру, функциональность и внешний вид веб-страниц.

### **2.1 Языки программирования**

Языки программирования - это формальный язык, используемый для написания инструкций, которые могут быть выполнены компьютером. Они позволяют программистам создавать программное обеспечение, которое может выполнять различные задачи, от простых математических расчетов до сложных операций. Некоторые из наиболее популярных языков программирования включают C++, Java, Python, JavaScript, Ruby и многие другие. Каждый язык имеет свои особенности и применяется в разных областях, таких как веб-разработка, мобильная разработка, системное программирование и научные вычисления.

1) HTML — раньше, когда ресурсы в интернете были простыми и представляли собой обычные страницы, его использовали для создания сайтов [1]. Но и сейчас HTML лежит в основе интернет-страниц, где отвечает за расположение различных элементов: таблиц, графики, текстовой информации и т.д.

HTML состоит из тэгов. Это команды, которые говорят браузеру, как правильно отображать содержимое.

Метод его работы прост. Когда вы вводите название сайта в адресную строку, браузер отправляет запрос к нужному ресурсу. В ответ он получает файл в формате HTML. А браузер преобразует набор команд в визуальное представление страницы.

Можно сказать, что HTML — каркас любого веб-документа и своеобразная азбука веб-разработчика. И фронтендеры, и бэкендеры обычно знают основы языка разметки.

2) CSS — если HTML — это каркас сайтов, то CSS отвечает за внешний вид [1]. Сайты, написанные на «голом» HTML, имели четкую структуру, но были некрасивыми и напоминали обычные документы со ссылками и таблицами. CSS позволил радикально улучшить их оформление.

Именно CSS дает возможность установить общий размер заголовков, цвета фона и текста, размер кнопок и т.д.

3) JavaScript — это самый популярный и практически безальтернативный язык программирования для фронтенда современных сайтов [16]. По индексу TIOBE на март 2024 JavaScript занимает шестое место. Этот рейтинг оценивает общую популярность языков программирования. А вот согласно исследованию GitHub по итогам 2023 года JavaScript занимает первое место среди веб-разработчиков. На нем пишут 19% респондентов. Вместе с его вариацией TypeScript они занимают треть рынка.

Внедрение JavaScript стало настоящим прорывом в веб-разработке. Этот язык программирования работает на основе скриптов или сценариев, которые исполняет браузер. Например, пользователь может нажать на кнопку и активировать всплывающее окно. Через эту форму можно заполнить данные и фронтенд сайта самостоятельно их обработает, без запросов на сервер.

Это позволяет, с одной стороны, уменьшить время ожидания ответа, с другой — снизить нагрузку на трафик и аппаратные мощности.

Все, что вы видите в интерфейсе сайтов: кнопки, всплывающие формы, меню, автозаполнение данных — все это реализовали на основе JavaScript. Этот язык программирования применяют и для серверной (бэкенд) разработки, но значительно реже.

Для фронтендеров в JavaScript разработали множество фреймворков — каркасов приложений и библиотек — готовых участков кода. Знание одного из них — практически обязательное условия для трудоустройства. Самые популярные технологии: Angular, React и Vue [20].

## **2.2 Фреймворки**

Фреймворки - это набор инструментов и библиотек, которые помогают разработчикам создавать пользовательские интерфейсы для веб-приложений. Они предоставляют структуру, компоненты и шаблоны, которые упрощают и ускоряют процесс разработки. Фронтенд-фреймворки обычно основаны на JavaScript и используют его для создания интерактивных и динамичных веб-страниц. Некоторые из популярных фронтенд-фреймворков включают React, Angular и Vue.js. Они предлагают множество готовых решений для типичных задач фронтенд-разработки, таких как управление состоянием приложения, маршрутизация, обработка форм и взаимодействие с сервером.

1) React — это библиотека для создания пользовательских интерфейсов, которая позволяет разработчикам создавать мощные и гибкие веб-приложения [10]. Она была создана компанией Facebook в 2013 году и с тех пор стала одной из самых популярных библиотек JavaScript.

React использует компонентный подход к разработке, что означает, что приложения строятся из отдельных компонентов, каждый из которых отвечает за свою часть интерфейса. Это делает код более модульным и упрощает его поддержку и расширение.

Одной из ключевых особенностей React является использование виртуального DOM (Document Object Model), который позволяет быстро

обновлять части страницы без необходимости перерисовывать всю страницу целиком. Это значительно ускоряет работу приложений и улучшает пользовательский опыт.

Кроме того, React поддерживает реактивность данных, что означает, что любые изменения в данных автоматически отражаются на интерфейсе пользователя. Это упрощает разработку сложных интерактивных приложений.

React широко используется в индустрии разработки программного обеспечения. Он применяется в таких компаниях, как Facebook, Instagram, Netflix, Airbnb и многих других. Кроме того, существует большое сообщество разработчиков, которые создают различные инструменты и пакеты для расширения функционала React.

2) Vue.js — это прогрессивный фреймворк для разработки одностраничных приложений (SPA), который позволяет разработчикам создавать интерактивные и мощные веб-приложения [20]. Он основан на JavaScript и может использоваться для создания различных типов веб-сайтов, включая одностраничные приложения, пользовательские интерфейсы, порталы и многое другое.

Он быстро стал одним из самых популярных фронтенд-фреймворков благодаря своей простоте, гибкости и производительности. Он использует компонентный подход к разработке, что делает его идеальным выбором для создания масштабируемых приложений.

Одной из ключевых особенностей Vue.js является его высокая степень совместимости с другими инструментами и библиотеками, такими как React и Angular. Это позволяет разработчикам легко интегрировать Vue.js в существующие проекты или использовать его вместе с другими технологиями.

Примеры компаний, использующих Vue.js в своих проектах, включают Alibaba, Baidu, GitLab и Behance. Они используют этот фреймворк для создания быстрых, отзывчивых и удобных пользовательских интерфейсов.

3) Angular — это мощный и популярный фреймворк для разработки веб-приложений, который был создан компанией Google и выпущен под лицензией MIT. Он основан на TypeScript, который является надмножеством JavaScript и добавляет поддержку статической типизации [20].

Angular предоставляет разработчикам набор инструментов для создания сложных и интерактивных веб-приложений. Фреймворк включает в себя такие компоненты, как Angular CLI (командная строка), Angular Material (библиотека для создания пользовательских интерфейсов), Angular Flex Layout (для адаптивной верстки) и другие.

## **2.3 Библиотеки**

Если фреймворк можно сравнить с ящиком инструментов, то библиотеку это определенный инструмент из этого ящика. Они предназначены для упрощения процесса разработки, повышения эффективности работы программиста и улучшения качества создаваемых программ [20].

Использование библиотек позволяет программистам избежать написания кода с нуля каждый раз при решении похожих задач. Вместо этого они могут использовать уже готовые решения из библиотеки, что значительно экономит время и усилия.

Библиотеки также способствуют стандартизации кода и повышению его качества. Когда все программисты используют одни и те же методы для решения определенных задач, это облегчает чтение и понимание чужого кода.

Более того, использование библиотек часто приводит к созданию более стабильных и надежных программ. Многие библиотеки проходят тщательное тестирование перед тем, как они становятся доступны для использования.

1) TypeScript — это надстройка над JavaScript с одним важным дополнением: он позволяет использовать типы данных [5]. Это значит, что

когда вы пишете программу на нем, вы можете указать, какой тип данных должна иметь переменная — например, число, строка и т.д.

JavaScript не требует этого, и это может привести к ошибкам, потому что программа может случайно попытаться сложить строку и число вместе или сделать что-то ещё неправильное. TypeScript помогает избежать таких ошибок, показывая их сразу же, как только вы пытаетесь их совершить, а не тогда, когда программа уже запущена и работает.

2) GSAP — это библиотека для создания анимаций на веб-сайтах. Если вы когда-нибудь видели, как элемент плавно перемещается по экрану или как что-то меняется в ответ на взаимодействие пользователя, то это, скорее всего, результат использования анимации. GSAP упрощает создание этих анимаций, позволяя разработчикам сосредоточиться на дизайне и функциональности своего сайта, а не на технических деталях анимации.

3) Swiper — это библиотека для создания слайдеров или каруселей в веб-приложениях. Если вы когда-нибудь видели, как изображения или фрагменты контента сменяют друг друга на экране, то это, скорее всего, результат использования слайдера. Swiper упрощает создание этих слайдеров, предоставляя разработчикам готовый набор инструментов для создания интерактивных и красивых слайдеров без необходимости писать сложный код с нуля.

4) Redux — это библиотека для управления состоянием в приложении. Когда вы пишете приложение, вам нужно отслеживать множество вещей: какие данные у вас есть, какие действия были выполнены пользователем, какие изменения произошли и так далее. Redux помогает организовать всё это, делая код более чистым и понятным.

### **3 Дизайн и разработка**

Веб-дизайн и разработка являются основными элементами создания успешного продукта. Они включают в себя создание эстетичного интерфейса, обеспечение функциональности и адаптивности для различных устройств.

Разработка воплощает дизайн-концепции в реальность, охватывая работы от верстки до программирования. Важно учитывать производительность, безопасность и удобство поддержки кода.

#### **3.1 Этап проектирования интерфейса**

Проектирование UX/UI является первым и одним из самых важных этапов создания интернет-магазина [19]. Это процесс разработки удобного и интуитивно понятного интерфейса, который обеспечивает положительный опыт использования сайта для пользователей. Проектирование UX (User Experience) фокусируется на том, чтобы сделать сайт максимально удобным и простым в использовании. Оно включает в себя анализ потребностей пользователей, создание прототипов и тестирование интерфейса на разных этапах разработки [8].

Целью UX является создание такого интерфейса, который позволит пользователям быстро находить нужную информацию и выполнять необходимые действия без затруднений. Проектирование UI (User Interface) занимается визуальным оформлением сайта. Это включает выбор цветовой гаммы, шрифтов, расположения элементов интерфейса и других графических элементов.

Целью UI является создание приятного для глаз дизайна, который будет соответствовать бренду компании и привлекать внимание пользователей.

Плохо спроектированный интерфейс может привести к тому, что пользователи будут испытывать трудности при работе с сайтом и откажутся



от него. Кроме того, если интерфейс не соответствует потребностям пользователей, то они могут потерять интерес к вашему продукту или услуге.

Поэтому важно уделить достаточно времени и ресурсов на проектирование UX/UI для создания успешного интернет-магазина.

### **3.2 Этап верстки и разработки**

На этом этапе происходит преобразование дизайнерских макетов в функциональный сайт. В качестве стека технологии я выбрал фреймворк Next.js для фронтенд, среду разработки Node.js для бэкенда и MongoDB в качестве базы данных [1].

Первым этапом в создании любого сайта идет веб-дизайн. Для его создания сегодня используют программу под названием Figma [19].

На данный момент это лучшее, что есть на рынке. Она позволяет дизайнерам и разработчикам работать над одним проектом в реальном времени, оставлять заметки и комментарии, имеет веб-версию, возможность интеграции с другими популярными инструментами, например, Slack и Google Drive.

Исходя из анализа конкурентов было принято решение делать сайт в минималистичном стиле так как большинство интернет-магазинов переполнены различной информацией и элементами, что отвлекает пользователей от контента [8].

В результате у меня получился вот такой дизайн, рисунок 3.

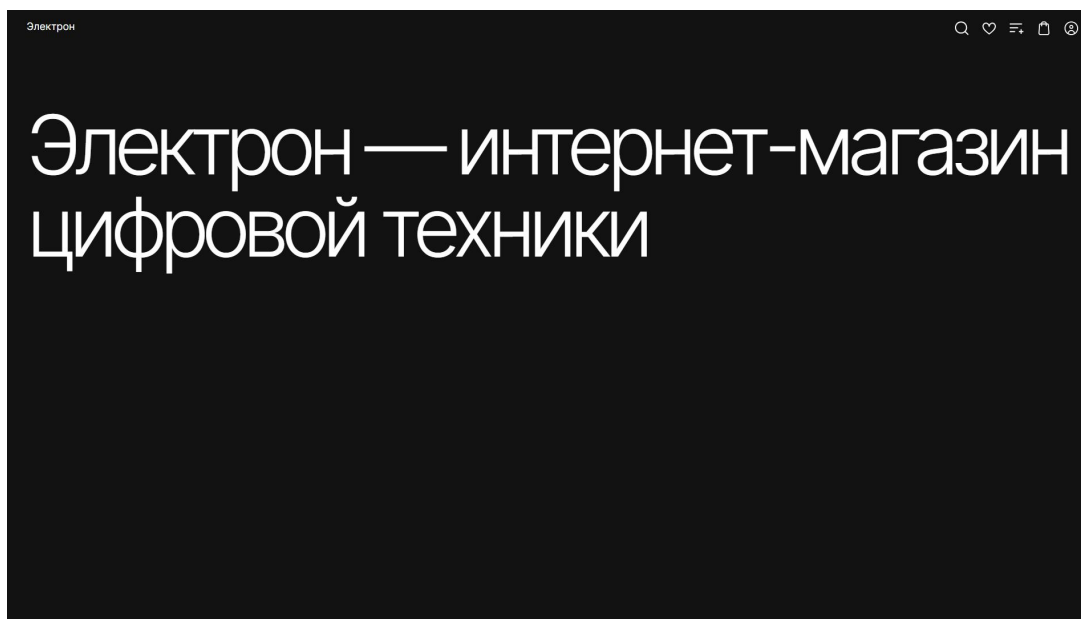


Рисунок 3 – Дизайн сайта

В качестве фреймворка был выбран Next.js [11]. Это как обычный React, но с дополнительным функционалом, который значительно упрощает разработку [6]. Например, поддержка SSG и SSR, что ускоряет загрузку страниц и улучшает продвижение сайта в интернете. Встроенная система сборки, которая автоматически собирает ваш код и оптимизирует его для лучшей производительности [1].

Исходя из макета получилась следующая структура главной страницы:

- 1) Шапка
- 2) Обложка
- 3) Категории
- 4) Новинки
- 5) Хиты

1) Шапка, рисунок 4. Это верхняя часть веб-страницы, которая содержит основную информацию о компании или бренде, навигационные элементы и другие важные ссылки. Стандартная структура шапки это логотип и меню. Обычно логотип является ссылкой на главную страницу.



Рисунок 4 – Шапка сайта

2) Обложка, рисунок 5. Представляет собой блок на странице, который пользователь видит при открытии сайта. На нем могут использовать такие элементы, как заголовок, изображение или видеоролик, призыв к действию, форма заявки. При создании первого экрана решается ряд задач:

- Формирование первого впечатления о продукте/услуге. Задача решается через создание стильного дизайна, эффектного и понятного УТП, грамотного расположения элементов. При просмотре первого экрана пользователь может испытать широкий спектр эмоций: от восторга до разочарования. Дизайнер же способен сделать так, чтобы пользователь захотел узнать о продукте больше и приобрести его.

- Знакомство с продуктом. Первый экран сайта дает пользователю краткую информацию о продукте и его преимуществах. Обычно эта информация представлена в виде УТП — уникального торгового предложения, а также изображения, которое иллюстрирует продукт или услугу. Дизайнер может в некоторых ситуациях участвовать в формировании УТП, однако в крупных компаниях это обычно делает маркетолог или бренд-менеджер. Основная работа дизайнера заключается в создании такого визуального оформления, т. е. подбора фирменных цветов, декоративных элементов и изображений, которое позволит пользователю сформировать четкое представление о продукте.

# Электрон — интернет-магазин цифровой техники

Рисунок 5 – Обложка

3) Категории, рисунок 6. Этот блок помогает пользователям быстро находить нужную им информацию, упрощает поиск товаров, а также улучшает SEO-оптимизацию сайта. Он позволяет разбить контент на логические группы, что делает его более понятным и доступным.



Рисунок 6 – Блок категории

4) Новинки и хиты, рисунок 7. Они помогают посетителям быстро найти самые актуальные товары или услуги, а также узнать о наиболее популярных предложениях среди других покупателей. Кроме того, эти блоки могут привлечь внимание к новым товарам, которые еще не получили широкой известности и стимулировать продажи.

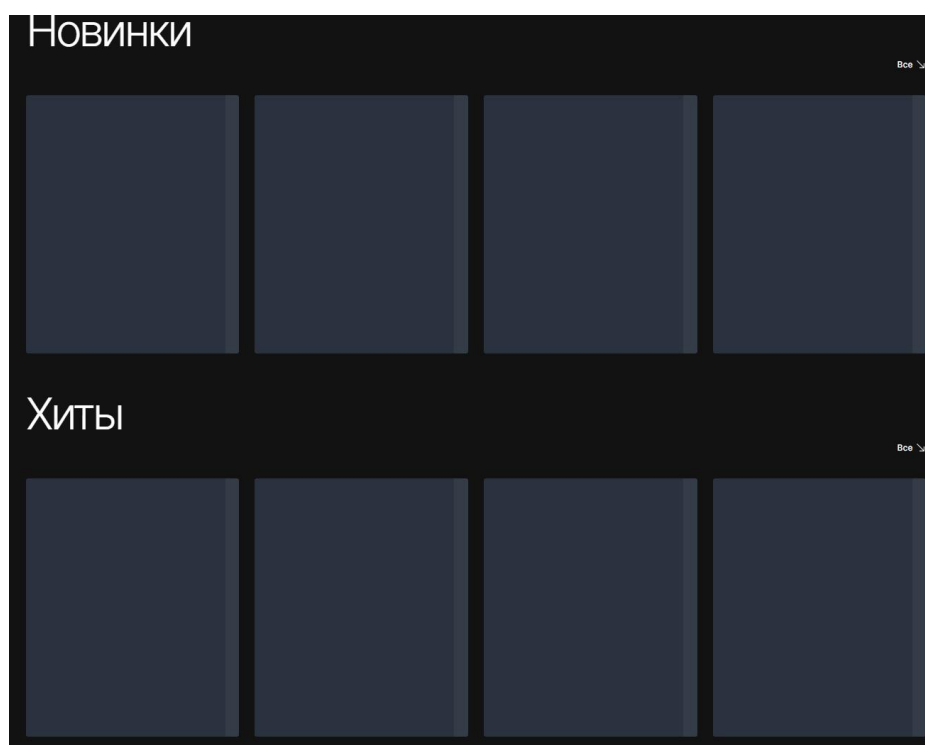


Рисунок 7 – Блоки новинки и хиты

Карточки товаров в этих блоках подгружаются из базы данных. В качестве нее я выбрал MongoDB — это СУБД, которая хранит данные в формате JSON. Она не использует таблицы, как MySQL, а представляет собой документ, который содержит данные в виде пар ключ-значение.

Она хороша для интернет-магазина тем, что позволяет легко масштабироваться и обрабатывать большие объёмы данных. Если магазин продаёт миллионы товаров, с её помощью можно быстро получить доступ к любому из них без необходимости перебирать все строки таблицы.

БД на MongoDB создается в несколько этапов. Сперва нужно создать кластер. Это группа серверов, которые работают вместе для обработки

запросов и хранения данных.

Затем идет развертывание базы данных на созданном кластере. Для этого использовался инструмент управления базами данных Mongoose.

После создается коллекция в которую необходимо добавить данные. Коллекции - это аналог таблиц в реляционных базах данных. Делают это путем добавления документов, которые являются наборами пар ключ-значение.

Далее уже в коде эти данные выводятся в виде карточек:

```
<div className={styles.main_section__inner}>
  <AllLink />
  {spinner && (
    <motion.ul
      className={skeletonStyles.skeleton}
      {...basePropsForMotion}
    >
      {Array.from(new Array(4)).map((_, i) => (
        <li key={i} className={skeletonStyles.skeleton__item}>
          <div className={skeletonStyles.skeleton__item__light} />
        </li>
      ))}
    </motion.ul>
  )}
  {!spinner && (
    <motion.ul
      className={`list-reset ${styles.main_section__list}`}
      {...basePropsForMotion}
    >
      {goods.map((item) => (
        <ProductsListItem key={item._id} item={item} title={title} />
      ))}
    </motion.ul>
  )}
</div>
```

```
    )})  
  </motion.ul>  
  })  
</div>
```

Этот код используется для отображения списка товаров на. Сначала он проверяет, загружен ли контент, используя переменную `spinner`. Если товары еще не загружены, то вместо них отображается анимация загрузки. Как только товары становятся доступны, анимация загрузки исчезает, и вместо нее отображается список.

Страница корзины, рисунок 8. Позволяет собрать все выбранные товары в одном месте, чтобы пользователь мог увидеть общую стоимость своей покупки, а также изменить количество товаров или удалить ненужные позиции перед тем, как перейти к оплате.

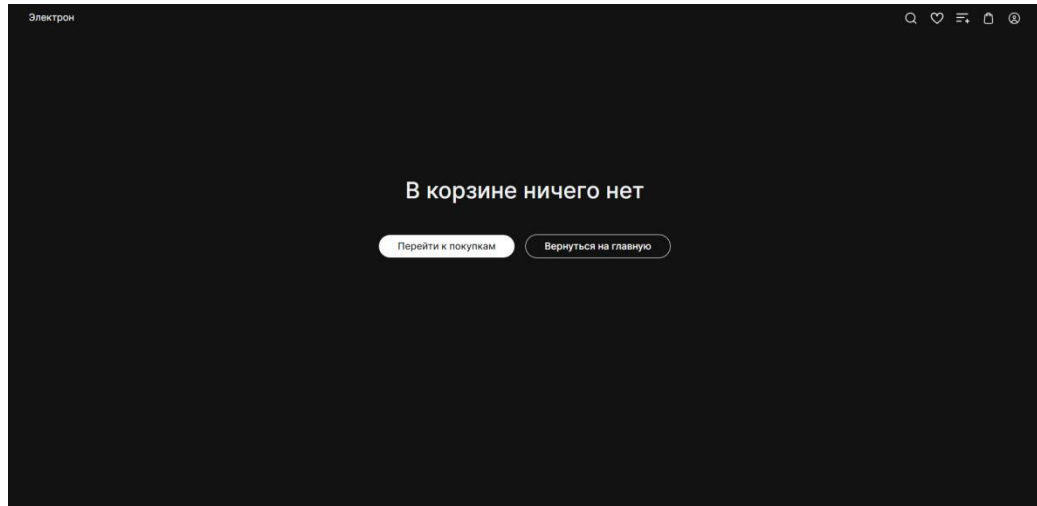


Рисунок 8 – Корзина

```
{!shouldShowEmpty ? (  
  <section className={styles.cart}>  
    <div className='container'>  
      <HeadingWithCount  
        count={countWholeCartItemsAmount(currentCartByAuth)}  
        title={translations[lang].breadcrumbs.cart}  
        spinner={cartSpinner}  
      />  
      <div className={styles.cart__inner}>  
        <div className={styles.cart__left}>  
          {(isUserAuth()  
            ? cartSpinner || loginCheckSpinner  
            : cartSpinner) && (  
            <motion.ul  
              {...basePropsForMotion}
```



```

        className={cartSkeletonStyles.skeleton}
      >
        {Array.from(new Array(3)).map((_, i) => (
          <li
            key={i}
            className={cartSkeletonStyles.skeleton__item}>
              <div
                className={cartSkeletonStyles.skeleton__item__light}
              />
            </li>
          ))}
        </motion.ul>
      )}
    {!cartSpinner && (
      <motion.ul
        {...basePropsForMotion}
        className={list-reset ${styles.cart__list}}
      >
        <CartList />
      </motion.ul>
    )}
  </div>
  <div className={styles.cart__right}>
    {isMedia930 && (
      <PromotionalCode
        setIsCorrectPromotionalCode={setIsCorrectPromotionalCode}
      />
    )}
  <div className={styles.cart__right__order}>

```

```

        <OrderInfoBlock
            isCorrectPromotionalCode={isCorrectPromotionalCode}
        />
    </div>
</div>
</div>
{!isMedia930 && (
    <PromotionalCode
        setIsCorrectPromotionalCode={setIsCorrectPromotionalCode}
    />
)}
</div>
</section>
):(
    <section>
        <div className='container'>
            <EmptyPageContent
                subtitle={translations[lang].common.cart_empty}
                description={translations[lang].common.cart_empty_advice}
                btnText={translations[lang].common.go_shopping}
                bgClassName={styles.empty_bg}
            />
        </div>
    </section>
)}

```

Этот код проверяет, должна ли быть показана страница корзины покупок. Если в корзине есть товары, то отображается список этих товаров вместе с информацией о заказе. Если же корзина пуста, то отображается сообщение об этом и предложение продолжить покупки.

## **ЗАКЛЮЧЕНИЕ**

В современную цифровую эпоху наличие веб-сайта стало неотъемлемым аспектом ведения успешного бизнеса. Внедрение интернет-магазина предоставляет предприятиям беспрецедентные возможности для расширения своей клиентской базы, увеличения доходов и повышения узнаваемости бренда.

Интернет-магазины устраняют географические барьеры, позволяя предприятиям выходить на рынки по всему миру. Они обеспечивают круглосуточный доступ к товарам и услугам, что дает потребителям возможность совершать покупки в любое удобное для них время. Кроме того, сайты служат ценным инструментом для построения отношений с клиентами, предоставления поддержки и укрепления доверия.

Помимо удобства и доступности, магазины предлагают предприятиям возможность собирать ценные данные о покупателях. Анализируя данные о поведении пользователей, предприятия могут персонализировать маркетинговые кампании, улучшать предложения продуктов и оптимизировать пользовательский опыт. Это, в свою очередь, приводит к повышению удовлетворенности клиентов и лояльности, что является основой долгосрочного успеха.

В свете растущей конкуренции на рынке наличие хорошо спроектированного и функционального интернет-магазина стало необходимостью для предприятий всех размеров. Он служит витриной для продуктов и услуг, предоставляет покупателям удобный и безопасный способ совершения покупок и помогает предприятиям выделиться среди конкурентов.

В данной курсовой работе был рассмотрен процесс разработки интернет-магазина на Next.js/Node.js/TypeScript. Была проведена работа над архитектурой проекта, реализацией клиентской и серверной части, а также интеграцией сторонних сервисов.

В результате был создан полноценный интернет-магазин, который позволяет пользователям регистрироваться, авторизовываться, просматривать товары, добавлять их в корзину и оформлять заказы. Серверная часть реализована с использованием Node.js, а клиентская часть разработана на Next.js с использованием TypeScript.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

### Книги, монографии, учебники, учебные пособия:

1. Александров А. В. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.
2. Бондаренко Е. С. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.
3. Браун Дж. Создание веб-приложений на React и Flux. — М.: ДМК Пресс, 2017. — 336 с.
4. Вайнберг М. React. Быстрый старт. — М.: ДМК Пресс, 2018. — 224 с.
5. Воробьев Д. В. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.
6. Гайсинский В. React. Essential. — М.: ДМК Пресс, 2018. — 288 с.
7. Гиффорд Д. React и React Native. — М.: ДМК Пресс, 2018. — 448 с.
8. Годунов А. В. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.
9. Дельп Ф. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.
10. Дилан Б. React. Быстрый старт. — М.: ДМК Пресс, 2018. — 224 с.
11. Дьюи М. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.
12. Иванов А. А. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.
13. Кадырова Д. В. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.
14. Купер Д. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.
15. Левин Д. React. Быстрый старт. — М.: ДМК Пресс, 2018. — 224 с.
16. Петров И. А. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.
17. Попов Е. В. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.

18. Репетто М. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.

19. Роджерс Д. Next.js. Быстрый старт. — М.: ДМК Пресс, 2021. — 224 с.

20. Фомин В. В. Next.js. Разработка современных веб-приложений. — М.: ДМК Пресс, 2022. — 208 с.

### **Электронные ресурсы:**

21. HTML & CSS Design and Build Websites by Jon Duckett. URL: <https://www.amazon.com/HTML-CSS-Design-Build-Websites/dp/1118533364> (3.06.2024)

22. You Don't Know JS (YDKJS) series by Kyle Simpson. URL: <https://github.com/getify/You-Dont-Know-JS> (5.06.2024)

23. The Modern Web Platform by Alex Russell. URL: <https://medium.com/@alexrussell/the-modern-web-platform-e8c57f1f9eb6> (4.06.2024)

24. Eloquent JavaScript by Marijn Haverbeke. URL: <https://eloquentjavascript.net/> (7.06.2024)

25. Frontend Architecture Guidelines by Adobe. URL: <https://designsystems.adobe.com/guidelines/frontend-architecture/> (3.06.2024)

26. Progressive Web Apps by Alex Russell, et al. URL: <https://developers.google.com/web/progressive-web-apps/> (4.06.2024)

27. Designing for Coherence: A Practical Guide to Visual Consistency Across the Web Platform by Adobe. URL: <https://designsystems.adobe.com/guidelines/coherence/> (9.06.2024)

28. The Principles of Beautiful Web Design by Jason Beaird. URL: <https://www.amazon.com/Principles-Beautiful-Web-Design-Jason-Beaird/dp/0321503484> (4.06.2024)

29. The Elements of Typographic Style Applied to the Web by Robert Bringhurst and Richard Rutter. URL: <https://www.amazon.com/Elements->

Typographic-Style-Applied-Web/dp/0881792066 (3.06.2024)

30. The Art of Readable Code by Mathew McCullough. URL:  
<https://www.amazon.com/Art-Readable-Code-Mathew-McCullough/dp/1449303487> (3.06.2024)

31. High Performance Browser Networking by Ilya Grigorik. URL:  
<https://www.amazon.com/High-Performance-Browser-Networking-Ilya-Grigorik/dp/1449364212> (6.06.2024)

32. Understanding ECMAScript 6 by Nicholas C. Zakas. URL:  
<https://www.amazon.com/Understanding-ECMAScript-6-Nicholas-Zakas/dp/1491950290> (6.06.2024)

33. Smashing Magazine. URL: <https://www.smashingmagazine.com/>  
(8.06.2024)

34. A List Apart. URL: <https://alistapart.com/> (8.06.2024)

35. CSS Tricks. URL: <https://css-tricks.com/> (3.06.2024)

36. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/>  
(3.06.2024)

37. JavaScript Weekly newsletter. URL: <https://javascriptweekly.com/>  
(5.06.2024)

38. Frontend Focus newsletter. URL: <https://frontendfocus.com/> (6.06.2024)

39. Reddit /r/web\_design. URL: [https://www.reddit.com/r/web\\_design/](https://www.reddit.com/r/web_design/)  
(3.06.2024)

40. Reddit /r/frontend. URL: <https://www.reddit.com/r/frontend/> (3.06.2024)

## **ПРИЛОЖЕНИЕ**