

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание предметной области.....	5
1.2 Сравнительный анализ аналогов и прототипов	6
1.3 Постановка задачи проектирования.....	9
1.4 Выбор и обоснование средств и методов решения задач	10
1.5 Разработка технического задания	11
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	12
2.1 Функциональная структура приложения.....	12
2.2 Проектирование диаграммы вариантов использования	13
2.3 Информационное обеспечение проекта.....	14
2.4 Разработка структуры конфигурации	15
2.5 Разработка концепции пользовательского интерфейса	21
2.6 Безопасность и защита данных	22
3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ	24
3.1 Результаты реализации функциональных частей	24
3.2 Разработка программной документации	28
3.3 Тестирование.....	28
4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....	32
4.1 Обоснование необходимости вывода продукта на рынок	32
4.2 Структура (этапы) работ по созданию программного обеспечения	32
4.3 Составление сметы затрат на разработку программного обеспечения.....	32
4.4 Расчет экономического эффекта разработчика программного обеспечения.....	32
4.5 Выводы по экономической части.....	33
ЗАКЛЮЧЕНИЕ.....	34
СПИСОК ИСТОЧНИКОВ.....	35
ПРИЛОЖЕНИЕ А (обязательное) Техническое задание	36
ПРИЛОЖЕНИЕ Б (обязательное) Диаграмма вариантов использования	41
ПРИЛОЖЕНИЕ В (справочное) Схема базы данных.....	42
ПРИЛОЖЕНИЕ Г (обязательное) Описание программы.....	43
ПРИЛОЖЕНИЕ Д (обязательное) Программа и методика испытаний	44
ПРИЛОЖЕНИЕ Е (обязательное) Руководство пользователя	45

					<i>Отчет по преддипломной практике</i>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Гилимович А.Н.			Веб-сайт для филиала Новополоцк хлебозавод ОАО «Витебскхлебпром», г. Новополоцк	для	Лист
Провер.		Голубева О.В.					Листов
Реценз.							3
Н. Контр.							38
Утверд						Учреждение образования «Полоцкий государственный университет имени Евфросинии Полоцкой» гр.19–ИТ–2	

ВВЕДЕНИЕ

В современном информационном обществе веб-сайты становятся неотъемлемой частью бизнеса и организаций, предоставляя возможность эффективного взаимодействия с клиентами и расширения рынка сбыта. В рамках дипломного проекта представляется разработка веб-сайта для филиала Новополоцк хлебозавода ОАО «Витебскхлебпром», расположенного в городе Новополоцк.

Филиал Новополоцк хлебозавода ОАО «Витебскхлебпром» является крупным предприятием пищевой промышленности, занимающимся производством и реализацией хлебобулочных изделий. Он предлагает широкий ассортимент продукции, включая хлеб, булочки, пироги и другие деликатесы. Стремясь удовлетворить потребности клиентов и повысить качество обслуживания, филиал стремится внедрить современные технологии и информационные системы, включая веб-сайт, который позволит обеспечить удобный доступ к информации о продукции и услугах компании.

Целью данного дипломного проекта является создание уникального и привлекательного веб-сайта для филиала Новополоцк хлебозавода ОАО «Витебскхлебпром», который будет способствовать улучшению взаимодействия с клиентами, расширению рынка сбыта и повышению узнаваемости бренда. В рамках проекта планируется разработка дизайна, функциональности, структуры и содержания веб-сайта с учетом требований компании и потребностей клиентов.

Используя современные технологии веб-разработки и дизайна, данный дипломный проект будет представлять полноценный веб-сайт, который обеспечит эффективное представление продукции, информации о филиале, его контактах и работе, а также новостях и акциях компании.

Разработка веб-сайта для филиала Новополоцк хлебозавода ОАО «Витебскхлебпром» имеет важное значение для развития компании, повышения ее конкурентоспособности и удовлетворения потребностей клиентов. Поэтому данное исследование является актуальным и полезным для практической реализации и внедрения веб-сайта в деятельность филиала.

1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

1.1 Описание предметной области

Филиал Новополоцк хлебозавод ОАО «Витебскхлебпром» занимается производством и реализацией хлебобулочных изделий в городе Новополоцк. Как крупное предприятие пищевой промышленности, филиал стремится удовлетворить потребности своих клиентов и повысить уровень обслуживания. В этом контексте разработка веб-сайта является важным шагом для улучшения взаимодействия с клиентами и расширения рынка сбыта.

Веб-сайт будет представлять собой платформу, предназначенную для предоставления информации о продукции и услугах компании, а также облегчения процесса ознакомления и приобретения хлебобулочных изделий. Он будет разработан с учетом требований и особенностей компании, а также потребностей клиентов.

Проект позволит хлебозаводу ОАО представить свою продукцию в привлекательном и информативном формате, включая описания, фотографии, состав и другую полезную информацию. Клиенты смогут ознакомиться с разнообразием хлебобулочных изделий, а также получить информацию о компании, ее истории, ценностях и качестве продукции.

Основные функциональные возможности веб-сайта будут включать возможность просмотра и поиска продукции, а также доступ к акциям и специальным предложениям. Кроме того, клиенты смогут получать информацию о работе филиала, его контактах, расписании работы и новостях компании.

Для обеспечения безопасности и конфиденциальности данных клиентов на веб-сайте филиала будет использоваться соответствующая система шифрования и защиты. Это позволит клиентам быть уверенными в сохранности своих личных данных при обмене информацией.

Стоит учитывать, что в современном мире мобильные устройства играют все более значительную роль в нашей повседневной жизни. Это влечет за собой изменения в поведении пользователей и их предпочтениях при поиске информации и использовании веб-ресурсов.

В этом контексте важно учесть растущую потребность в доступе к информации через мобильные устройства. Множество пользователей предпочитают использовать малые экраны для просмотра веб-сайтов, ведь они обладают высокой мобильностью и доступностью в любое время и в любом месте. Поэтому включение мобильной версии сайта становится необходимым шагом для успешного онлайн-присутствия.

Основная цель мобильной версии сайта заключается в создании оптимального и удобного пользовательского опыта на мобильных устройствах. Это подразумевает адаптацию веб-сайта под размеры меньших экранов, улучшение навигации и взаимодействия, упрощение процесса загрузки страниц,

					Отчет по преддипломной практике	/лсм 5
Изм.	Лист	№ докум.	Подпись	Дата		

а также оптимизацию контента для более эффективного отображения на мобильных устройствах.

Добавление мобильной версии сайта для филиала Новополоцк хлебозавод ОАО «Витебскхлебпром» имеет ряд важных преимуществ. Во-первых, это улучшит доступность информации о продукции и услугах филиала для пользователей, которые предпочитают мобильные устройства. Во-вторых, мобильная версия сайта повысит удобство использования и улучшит пользовательский опыт, что положительно скажется на общем впечатлении о компании. В-третьих, это повысит конкурентоспособность филиала, так как многие его конкуренты уже оснастили свои сайты мобильными версиями.

Проектирование и разработка мобильной версии сайта требует специальных подходов и технологий, чтобы учесть особенности мобильных устройств и обеспечить оптимальное взаимодействие с пользователем. В этом процессе следует учитывать различные параметры, такие как размеры экрана, управление сенсорным экраном, ограниченные ресурсы устройства и пропускную способность сети.

Таким образом, добавление мобильной версии сайта для Новополоцкого хлебозавода является важным шагом для улучшения пользовательского опыта, расширения аудитории и повышения конкурентоспособности компании в современном мобильном мире.

В целом, разработка веб-сайта для филиала Новополоцк хлебозавод ОАО «Витебскхлебпром» имеет большое значение для улучшения клиентского опыта, эффективного продвижения продукции и укрепления позиции компании на рынке. Предлагаемый веб-сайт будет служить эффективным инструментом для установления и развития взаимодействия с клиентами, а также для повышения видимости и привлекательности филиала на рынке хлебопродуктов.

1.2 Сравнительный анализ аналогов и прототипов

Сравнительный анализ аналогов и прототипов представляет собой важный этап исследовательского процесса, направленного на оценку существующих аналогичных продуктов или решений, а также прототипов, разработанных в рамках проекта. Цель данной части заключается в выявлении преимуществ, недостатков, особенностей и потенциала каждого аналога и прототипа, а также определении их соответствия поставленным целям и требованиям.

Сравнительный анализ является неотъемлемой частью процесса разработки и инновационных исследований, поскольку он позволяет более осознанно принимать и выбирать наиболее оптимальные решения для дальнейшего развития продукта или проекта. Аналоги и прототипы рассматриваются с различных точек зрения, таких как функциональность, производительность, стоимость, надежность, удобство использования и другие

					Отчет по преддипломной практике	/лсм 6
Изм.	Лист	№ докум.	Подпись	Дата		

факторы, которые могут быть важными для конечного пользователя.

Далее будут представлены результаты сравнительного анализа нескольких выбранных аналогов и прототипов, с акцентом на их сильные и слабые стороны, а также потенциал для дальнейшего усовершенствования. Важным аспектом будет также оценка соответствия аналогов и прототипов поставленным требованиям и целям проекта, что поможет определить их пригодность для дальнейшей реализации и коммерциализации.

Первым сайтом для сравнительного анализа будет веб-сайт ОАО «Витебскхлебпром». Данный ресурс предназначен для ознакомления с производимой продукцией предприятия, а также предоставляет возможность просматривать новости об предприятии, узнать контакты и вакансии.

Рассмотрим положительные стороны данного веб-сайта:

- многоязычность, сайт представлен на четырех языках (русский, белорусский, английский и китайский);
- простая и понятная навигация.



Рисунок 1.1 – Внешний вид сайта ОАО «Витебскхлебпром»

Далее рассмотрим недостатки:

- устаревший дизайн сайта, яркие плохо контрастирующие цвета плохо влияют на читаемость сайта;
- веб-сайт не адаптирован для мобильных устройств;
- недостаточно подробно представлена информация о продукте;
- некачественные, низкого разрешения изображения представленные на сайте.

Следующим объектом анализа будет сайт Полоцкого хлебозавода.



Рисунок 1.2 – Главная страница сайта Полоцкого хлебозавода

Сразу можно отметить положительные стороны ресурса:

- наполненность главной страницы сайта, здесь сразу видна вся информация, которая тебе может пригодиться;
- простота навигации;
- представлена и работает версия для слабовидящих.

Недостатками являются:

- устаревший дизайн;
- сайта, также как и первый не адаптирован для мобильных устройств;
- каталог продукции представлен в неудобном формате для просмотра;
- встречаются ошибки воспроизведения сайта, на некоторых страницах страница обрезана и нет возможности просмотреть эту информацию;
- много продукции без изображения.

Исходя из результатов сравнительного анализа сделан вывод о наиболее перспективных аналогах или прототипах, которые обладают наибольшим

					Отчет по преддипломной практике	Лист 8
Изм.	Лист	№ докум.	Подпись	Дата		

потенциалом и могут быть дальше развиты или применены в практической реализации проекта. Будут взяты положительные стороны для добавления к реализации дипломного проекта и недопущения добавления рассмотренных недостатков.

Результаты анализа позволили сформулировать функциональные требования к проектируемому веб-сайту, представленные в техническом задании приложения А.

1.3 Постановка задачи проектирования

В современном бизнес-мире веб-сайты играют ключевую роль в установлении и поддержании присутствия компании в интернете. Веб-сайт является важным инструментом для представления информации о компании, ее продукции и услугах, а также для взаимодействия с клиентами и потенциальными партнерами.

Основная цель разработки веб-сайта для филиала Новополоцк хлебозавода ОАО «Витебскхлебпром» заключается в создании привлекательной и информативной онлайн-платформы, которая позволит представить продукцию и услуги филиала, обеспечить удобство использования и обмена информацией, а также укрепить связи с клиентами и привлечь новых потенциальных партнеров.

В рамках данной задачи будут решены следующие подзадачи:

- Анализ требований и потребностей: первоначально необходимо провести анализ требований и потребностей филиала Новополоцк хлебозавода ОАО «Витебскхлебпром» и его целевой аудитории. Это позволит определить основные функциональные и дизайнерские требования к веб-сайту, а также учесть особенности и предпочтения клиентов.

- Разработка информационной структуры: следующим шагом будет разработка информационной структуры веб-сайта. Это включает определение основных разделов, подразделов и их иерархии, а также логической организации информации для удобства навигации пользователей.

- Создание привлекательного дизайна: для успешного привлечения и удержания внимания пользователей важно разработать привлекательный и современный дизайн веб-сайта. Это включает выбор цветовой схемы, шрифтов, графических элементов и создание пользовательского интерфейса, который будет легким в использовании и интуитивно понятным.

- Реализация функциональности: веб-сайт должен обладать необходимой функциональностью для представления информации о продукции и услугах филиала, а также для обратной связи с клиентами. Это может включать онлайн-каталог продукции, систему обратной связи и другие функциональные элементы, удовлетворяющие потребностям филиала и его клиентов.

- Адаптация под мобильные устройства: в современном мобильном

					Отчет по преддипломной практике	/лист 9
Изм.	Лист	№ докум.	Подпись	Дата		

мире важно учесть возможность просмотра и использования веб-сайта на различных мобильных устройствах. Поэтому важным аспектом задачи будет адаптация веб-сайта под мобильные платформы, чтобы обеспечить оптимальный пользовательский опыт и доступность для широкой аудитории.

– Тестирование и оптимизация: после разработки веб-сайта необходимо провести тестирование его функциональности, производительности и совместимости с различными браузерами и устройствами. По результатам тестирования будут внесены необходимые исправления и оптимизации для обеспечения стабильной и эффективной работы веб-сайта.

В итоге успешная реализация поставленных задач позволит филиалу Новополоцк хлебозавода ОАО «Витебскхлебпром» создать современный и привлекательный веб-сайт, который будет эффективным инструментом для представления продукции и услуг, установления связи с клиентами и привлечения новых возможностей для развития бизнеса.

1.4 Выбор и обоснование средств и методов решения задач

Для разработки веб-сайта был сделан выбор в пользу следующих технологий: Strapi, базы данных SQLite и Next.js. Этот выбор был сделан на основе их функциональности, гибкости и популярности в веб-разработке.

1 Strapi. Strapi является гибкой и мощной системой управления контентом (CMS) с открытым исходным кодом. Она позволяет создавать и управлять API для различных типов контента, таких как продукты, услуги, новости и другие элементы, которые могут быть представлены на веб-сайте филиала. Strapi обладает интуитивным пользовательским интерфейсом, который позволяет администраторам управлять контентом без необходимости в программировании. Благодаря гибкости Strapi можно настроить и адаптировать под требования филиала Новополоцк хлебозавода ОАО «Витебскхлебпром».

2 База данных SQLite. SQLite является легковесной и встроенной веб-базой данных, которая обеспечивает эффективное хранение и управление данными для веб-сайта филиала. SQLite хорошо подходит для небольших и средних проектов, таких как филиалы предприятий, и обеспечивает высокую производительность и надежность. База данных SQLite интегрируется с Strapi и позволяет эффективно хранить и получать информацию о продуктах, услугах, контактах и других данных, необходимых для веб-сайта филиала.

3 Next.js. Next.js является популярным фреймворком React для разработки серверного рендеринга и статически генерируемых веб-приложений. Он предоставляет мощные инструменты для создания быстрых и масштабируемых веб-сайтов. Next.js обладает превосходной производительностью благодаря оптимизации рендеринга и предзагрузке страниц. Он также обеспечивает удобную разработку, предоставляя готовые компоненты и маршрутизацию.

Next.js позволит создать современный и отзывчивый веб-сайт для Новополоцкого хлебозавода.

Выбор этих технологий обусловлен их совместимостью, расширяемостью и широкой поддержкой веб-разработчиками сообщества. Они позволяют разработать функциональный, удобный и привлекательный веб-сайт для филиала, обеспечивая эффективное управление контентом, хранение данных и отзывчивый пользовательский интерфейс.

1.5 Разработка технического задания

На основании анализа сущности проблемы и требований пользователей, были прописаны следующие требования к функциональности разрабатываемого веб-сайта:

1 Регистрация и аутентификация пользователей.

- Возможность регистрации новых пользователей с указанием обязательных полей, таких как имя, фамилия, адрес электронной почты и пароль.
- Подтверждение регистрации через электронную почту или другой подходящий механизм.
- Аутентификация пользователей для доступа к персонализированному контенту.

2 Галерея продукции.

- Предоставление веб-страницы с галереей фотографий продукции, сопровождаемой описанием и характеристиками.

3 Контактная информация и обратная связь.

- Предоставление контактной информации филиала и формы обратной связи для вопросов, предложений или жалоб пользователей.

4 Административная панель.

- Разработка административной панели для управления контентом, заказами и пользователями.
- Возможность добавления, редактирования и удаления контента на веб-сайте, включая информацию о продукции, акциях и новостях.

Техническое задание для проектируемого веб-сайта филиала Новополоцк хлебозавода ОАО «Витебскхлебпром» представлено в Приложении А. Данное техническое задание содержит более подробное описание требований к функциональности, дизайну, структуре базы данных, безопасности и производительности веб-сайта.

					Отчет по преддипломной практике	/лсм 11
Изм.	Лист	№ докум.	Подпись	Дата		

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Функциональная структура приложения

Функциональная структура разрабатываемого веб-сайта должна быть тщательно спроектирована с учетом потребностей и требований пользователей, а также основных целей и задач, стоящих перед данной платформой. Функциональная структура веб-сайта представлена на рисунке 2.1, которая обеспечит эффективную работу и удобство использования для всех заинтересованных сторон.

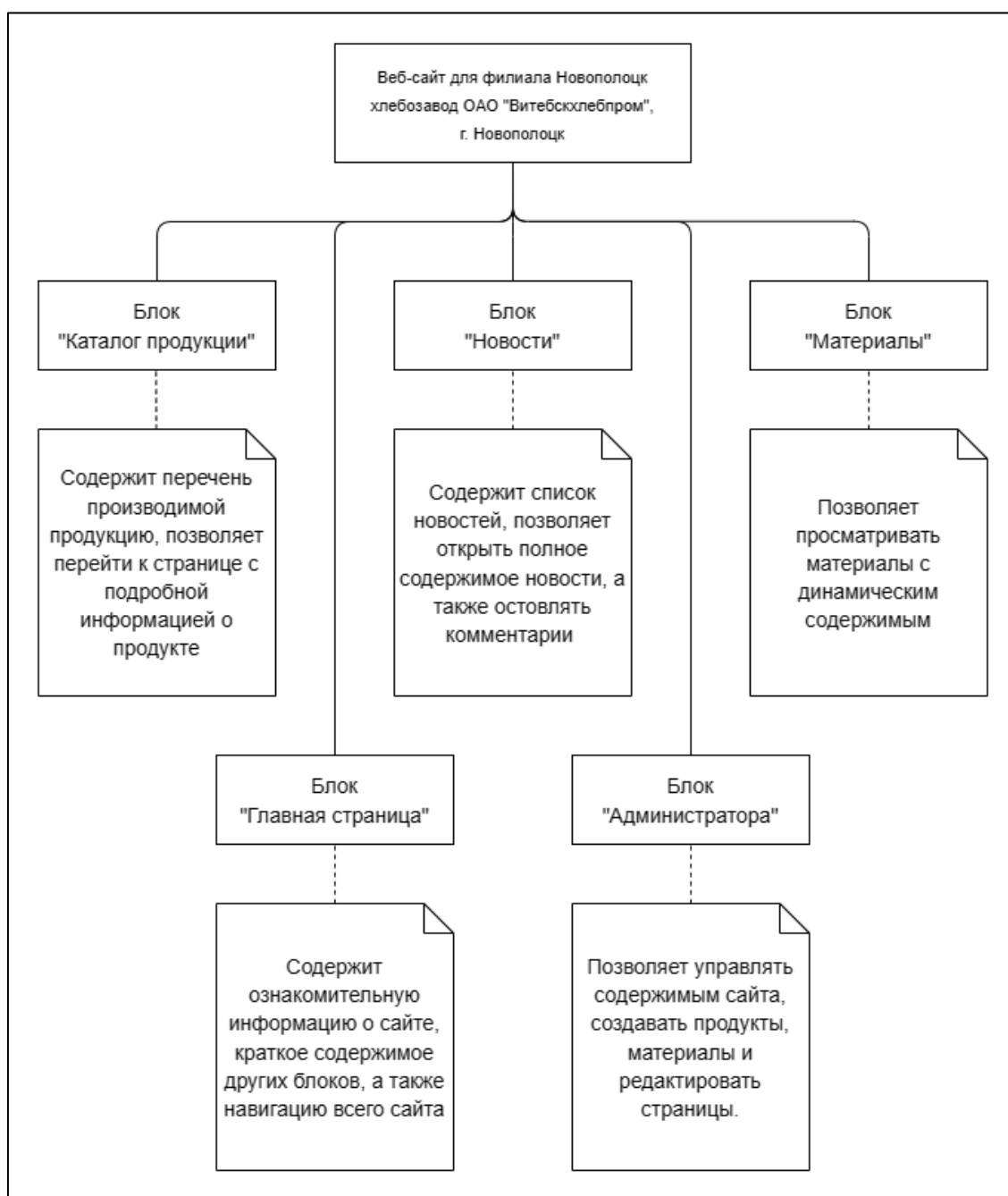


Рисунок 2.1 – Функциональная структура веб-сайта

Блок Главная страница содержит:

- основную информацию о филиале Новополоцк хлебозавод ОАО «Витебскхлебпром»;
- виджеты с несколькими продуктами и новостями для краткого ознакомления и перехода ко всем продуктам и новостям;
- контактная информация и режим работы.

Блок Каталог продукции содержит:

- перечень доступной продукции с описанием и фотографиями;
- фильтры для удобного поиска и сортировки товаров.

Блок Новости содержит список новостей о предприятии с фотографиями и датой публикации, а также возможность оставлять комментарии.

Блок Материалы содержит материалы редактируемые в свободной форме.

Блок Администратора содержит:

- управление контентом сайта (новости, продукция, информация о предприятии и др.);
- обработка комментариев и обращений;
- управления пользователями и их правами доступа.

Функциональная структура веб-сайта для филиала Новополоцк хлебозавода ОАО «Витебскхлебпром» предоставляет разнообразные возможности для пользователей, позволяет удобно ознакомиться с предлагаемой продукцией и получить необходимую информацию о филиале. Административная панель обеспечивает эффективное управление контентом, а также позволяет осуществлять контроль и настройку веб-сайта.

2.2 Проектирование диаграммы вариантов использования

Для веб-сайта необходимо разработать диаграмму вариантов использования, которая представит основные функциональные возможности системы и взаимодействие с актерами. Данная диаграмма поможет понять основные сценарии использования и обеспечит понимание взаимодействия между актерами и системой.

Варианты использования (use case) представляют собой спецификации сервисов или функций, которые система предоставляет актерам. Они определяют действия, которые система выполняет при взаимодействии с актером, не задавая конкретную реализацию этих действий.

Для веб-сайта филиала Новополоцк хлебозавод ОАО «Витебскхлебпром» выделяются два основных актера (приложение Б):

1 Администратор

- Конфигурирует работу веб-сайта.
- Обрабатывает жалобы и предложения пользователей.

- Управляет содержимым сайта.

2 Пользователь

- Просматривает материалы и другой контент ресурса.
- Оставляет комментарии и формирует рейтинг товара.
- Оставляет жалобы и предложения.

Диаграмма вариантов использования будет включать следующие элементы:

1 Актеры:

- администратор;
- пользователь.

2 Варианты использования:

- конфигурация веб-сайта администратором;
- обработка комментариев, жалоб и предложений администратором;
- создание статей, материалов и продукции администратором;
- просмотр статей и материалов пользователем;
- общение с другими пользователями;
- оставление отзыва и выставление оценки к продукции пользователем;
- подача жалоб и предложений пользователем.

Диаграмма вариантов использования поможет визуализировать основные сценарии взаимодействия между актерами и системой. Она позволит более четко определить функциональные требования к системе и обеспечит понимание основных возможностей, которые должны быть реализованы.

2.3 Информационное обеспечение проекта

Информационное обеспечение проекта играет важную роль в достижении целей. Включает в себя программные и аппаратные компоненты, базы данных, методологии и стандарты, которые необходимы для разработки, тестирования и развертывания системы.

В данном проекте используется SQLite в качестве СУБД, что обеспечивает легкость использования и портативность. SQLite обладает хорошей производительностью и эффективностью при обработке данных. Она предлагает компактное хранение данных и оптимизированные операции чтения и записи. Также поддерживает транзакции, что обеспечивает целостность данных и возможность отката изменений.

Для создания клиентской части веб-сайта был использован фреймворк Next.js. Next.js является популярным инструментом разработки, основанным на языке JavaScript и базирующимся на фреймворке React. Он обеспечивает удобные возможности для создания современных и производительных веб-приложений.

Next.js предоставляет набор инструментов и функциональных возможностей, которые позволяют разработчикам эффективно работать над клиентской частью веб-сайта. Он поддерживает серверный рендеринг (Server-Side Rendering), статическую генерацию (Static Site Generation) и инкрементальную перестройку (Incremental Static Regeneration), что обеспечивает быструю загрузку страниц и оптимальную производительность. \

Для создания административной панели был выбран инструмент Strapi. Strapi является гибкой системой управления контентом (CMS), специально разработанной для разработчиков. Он предоставляет удобный интерфейс и мощные функциональные возможности для управления контентом, настройки прав доступа и обработки данных.

Позволяет создавать и настраивать схемы данных, определять типы полей и связей между ними. Он также предоставляет гибкие возможности для создания пользовательских конечных точек API и выполнения запросов к базе данных.

Использование Strapi для создания административной панели позволяет разработчикам удобно управлять содержимым веб-сайта, добавлять новые записи, редактировать существующие данные и управлять правами доступа пользователей.

Для моделирования информационной системы на этапе проектирования используется стандарт UML. UML (Unified Modeling Language) – это язык для описания систем, который позволяет визуализировать компоненты системы, их свойства, атрибуты и отношения. UML обеспечивает простоту создания диаграмм и возможность детальной настройки системы.

В проекте применяется CASE-средство ERwin Data Modeler r7.2 для моделирования базы данных. Эта среда предоставляет графический интерфейс для создания схем базы данных, что облегчает процесс разработки и увеличивает качество проекта. ERwin Data Modeler r7.2 также обеспечивает возможность автоматической генерации скриптов базы данных, что упрощает процесс развертывания системы.

2.4 Разработка структуры конфигурации

При проектировании и разработке базы данных требуется выбрать оптимальное количество таблиц для хранения данных, полей, входящих в ту или иную таблицу, а также планирование отношений между таблицами. Для решения данного рода задач и используется нормализация.

Нормализация баз данных заключается в приведении структуры хранения данных к нормальным формам (NF). Всего таких форм существует 8, но часто достаточным является соблюдение первых трех. Рассмотрим их более подробно на примере учебной базы данных. Примеры будут строиться по принципу «что было бы, если было иначе, чем сейчас».

					<i>Отчет по преддипломной практике</i>	/лст
Изм.	Лист	№ докум.	Подпись	Дата		
						15

Первые три нормальные формы можно описать следующим образом:

- основным правилом 1NF является необходимость неделимости значения в каждом поле (столбце) строки – атомарность значений. Строки таблиц не должны зависеть друг от друга, т.е. первая запись не должна влиять на вторую и наоборот, вторая на третью и т.д. Размещение записей в таблице не имеет никакого значения. Аналогичная ситуация со столбцами записей. Их порядок не должен влиять на понимание информации. Каждая строка должна быть уникальна, поэтому для нее определяется первичный ключ, состоящий из одного либо нескольких полей (составной ключ). Первичный ключ не может повторяться в пределах таблицы и служит идентификатором записи;

- отношение находится во 2NF, если оно находится в 1NF и каждый не ключевой атрибут неприводимо зависит от первичного ключа. Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов;

- 3NF схожа по логике с 2NF, но с некоторым отличием. Если 2 форма ликвидирует зависимости не ключевых полей от части ключа, то третья нормальная форма исключает зависимость не ключевых полей от других не ключевых полей.

Реляционная база данных – это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов – атрибутов (например, наименование, код товара, сведения о клиенте). Записи, то есть строки таблицы, имеют одинаковую структуру – они состоят из полей, хранящих атрибуты объекта. Каждое поле, то есть столбец, описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта.

Схема базы данных веб-сайта филиала Новополоцк хлебозавод ОАО «Витебскхлебпром», г. Новополоцк представлена в приложении В.

В ходе проектирования базы данных были выявлены следующие основные сущности:

- сущность Администраторы представляет собой список администраторов сайта. Характеризуется логином, паролем, почтой, ФИО, датой регистрации и уровнем доступа;

- сущность Продукция хранит в себе список продукции предприятия. Характеризуется названием продукта и изображением к нему;

- подсущность Характеристика представляет собой информацию о продукте. Характеризуется составом продукта, сроком хранения, а также дополнительной информацией к продукту;

- подсущность Энергетическая ценность представляет собой информацию об энергетической ценности продукта;

- подсущность Масса хранит в себе различные форматы упаковки

					<i>Отчет по преддипломной практике</i>	/лист 16
Изм.	Лист	№ докум.	Подпись	Дата		

продукта. Характеризуется видом упаковки и массой;

- подсущность Пищевая ценность представляет собой химический состав продукта. Характеризуется белками, жирами и углеводами;

- подсущность Отзывы представляет собой список отзывов к продукту, оставляемых пользователями. Характеризуется именем пользователя, почтой, оценкой и содержимым отзыва;

- сущность Меню представляет собой список навигационных элементов сайта. Характеризуется родительской категорией, названием категорией и ссылкой куда ведет пункт меню;

- сущность Материалы представляет собой список материалов, открываемых через меню. Характеризуется пунктом меню, заголовком и содержимым материала;

- сущность Новости представляет собой список новостей об предприятии. Характеризуется заголовком, изображением к новости и датой публикации;

- подсущность Контент новости представляет собой содержимое новости. Характеризуется идентификационным номером новости и содержимым статьи;

- сущность Слайдер представляет собой список слайдов для главной страницы сайта. Характеризуется изображением слайда, ссылкой куда ведет при нажатии на слайд, описанием слайда и порядком отображения;

- сущность Баннеры представляет собой список баннеров для страниц сайта. Характеризуется изображением баннера, ссылкой, описанием и позицией отображения баннера;

- сущность Обращения хранит в себе список обращений пользователей. Характеризуется почтой, номером телефона, ФИО, типом обращения и содержимым изображения.

Далее будут описаны все таблицы, созданные в базе данных веб-сайта.

Таблица admins содержит информацию об администраторах. Её структура представлена в таблице 3.1.

Таблица 3.1 – Характеристика атрибутов таблицы admins

Имя атрибута	Тип	Описание
id	int	Идентификатор администратора
login	varchar	Логин администратора
password	varchar	Пароль
email	varchar	Электронная почта
name	varchar	Имя
surname	varchar	Фамилия
registr_date	date	Дата регистрации

lavel	int	Уровень доступа
-------	-----	-----------------

Таблица products содержит краткую информацию о продукте для отображения в каталоге. Её структура представлена в таблице 3.2.

Таблица 3.2 – Характеристика атрибутов таблицы products

Имя атрибута	Тип	Описание
id	int	Идентификатор продукта
name	varchar	Название продукта
image	varchar	Ссылка на изображение продукта

Таблица products_charcteristic содержит состав и другую информацию о товаре. Её структура представлена в таблице 3.3.

Таблица 3.3 – Характеристика атрибутов таблицы products_charcteristic

Имя атрибута	Тип	Описание
id	int	Идентификатор характеристики продукта
id_product	int	Идентификатор продукта
composition	varchar	Название продукта
shelf_life	varchar	Срок хранения продукта
inf	text	Дополнительная информация о продукте

Таблица products_energetic содержит энергетическую ценность товара в калориях и джоулях. Её структура представлена в таблице 3.4.

Таблица 3.4 – Характеристика атрибутов таблицы products_ energetic

Имя атрибута	Тип	Описание
id	int	Идентификатор энергетической ценности
id_product	int	Идентификатор продукта
ccal	int	Количество калорий
energy	int	Энергетическая ценность в джоулях

Таблица products_mass содержит информацию о видах упаковки и их массе. Её структура представлена в таблице 3.5.

Таблица 3.5 – Характеристика атрибутов таблицы products_ mass

Имя атрибута	Тип	Описание
id	int	Идентификатор массы продукта
id_product	int	Идентификатор продукта

package	int	Вид упаковки продукта
mass	int	Масса товара

Таблица products_value содержит информацию о пищевой ценности продукта. Её структура представлена в таблице 3.6.

Таблица 3.6 – Характеристика атрибутов таблицы products_value

Имя атрибута	Тип	Описание
id	int	Идентификатор пищевой ценности
id_product	int	Идентификатор продукта
proteins	int	Белки
fats	int	Жиры
carbohydrates	int	Углеводы

Таблица products_comments содержит отзывы, оставляемые пользователями. Её структура представлена в таблице 3.7.

Таблица 3.7 – Характеристика атрибутов таблицы products_comments

Имя атрибута	Тип	Описание
id	int	Идентификатор комментария
id_product	int	Идентификатор продукта
name	int	Имя пользователя
email	int	Электронная почта
score	int	Оценка
comment	varchar	Комментарий

Таблица menu содержит список пунктов меню для навигации по сайту. Её структура представлена в таблице 3.8.

Таблица 3.8 – Характеристика атрибутов таблицы menu

Имя атрибута	Тип	Описание
id	int	Идентификатор пункта меню
parent	varchar	Родительский пункт меню
title	varchar	Название
alias	varchar	Алиас пункта меню

Таблица materials содержит материалы привязанные к пунктам меню. Её структура представлена в таблице 3.9.

Таблица 3.9 – Характеристика атрибутов таблицы materials

Имя атрибута	Тип	Описание
id	int	Идентификатор материала
id_menu	int	Идентификатор пункта меню
title	varchar	Название материала
content	text	Содержимое материала

Таблица news содержит список новостей о предприятии. Её структура представлена в таблице 3.10.

Таблица 3.10 – Характеристика атрибутов таблицы news

Имя атрибута	Тип	Описание
id	int	Идентификатор новости
title	varchar	Заголовок новости
image	varchar	Изображение новости
date	varchar	Дата публикации

Таблица news_content содержит контент новости. Её структура представлена в таблице 3.11.

Таблица 3.11 – Характеристика атрибутов таблицы news_content

Имя атрибута	Тип	Описание
id	int	Идентификатор контента новости
id_news	varchar	Идентификатор новости
content	varchar	Содержимое новости

Таблица slider содержит слайдер для отображения на главной странице. Её структура представлена в таблице 3.12.

Таблица 3.12 – Характеристика атрибутов таблицы slider

Имя атрибута	Тип	Описание
id	int	Идентификатор слайда
image	varchar	Ссылка на изображение слайда
links	varchar	Ссылка
title	varchar	Заголовок слайда
order	int	Номер порядка отображения

Таблица banners содержит баннеры, которые могут быть отображены на любой странице сайта. Её структура представлена в таблице 3.13.

Таблица 3.13 – Характеристика атрибутов таблицы banners

Имя атрибута	Тип	Описание
id	int	Идентификатор баннера
image	varchar	Ссылка на изображение слайда
links	varchar	Ссылка
title	varchar	Заголовок баннера
position	int	Позиция отображения

Таблица appeals содержит обращения пользователей к администратору. Её структура представлена в таблице 3.14.

Таблица 3.14 – Характеристика атрибутов таблицы appeals

Имя атрибута	Тип	Описание
id	int	Идентификатор обращения
email	varchar	Электронная почта пользователя
telephone	varchar	Номер телефона
name	varchar	Имя
surname	varchar	Фамилия
second_name	varchar	Отчество
type	int	Тип обращения
message	text	Сообщение
feedback	bool	Требуется ли обратная связь

Таким образом при разработке базы данных веб-сайта было выявлено 14 сущностей и спроектированы 14 таблиц, которые связаны между собой связями «Один ко многим» либо «один к одному».

2.5 Разработка концепции пользовательского интерфейса

При создании веб-сайта для филиала Новополоцкого хлебозавода аспектом является разработка правильной концепции пользовательского интерфейса (UI). Удобный и интуитивно понятный пользовательский интерфейс играет ключевую роль в успешном взаимодействии пользователей с веб-сайтом.

После анализа аналогов можно выделить следующие основные разделы и функциональные элементы: главная страница, каталог продукции, новости, награды и материалы.

Главная страница должна содержать навигационные ссылки на видном месте, не нагружая при этом интерфейс. Помимо этого, на главной странице должен быть слайдер и виджеты с новостями и продукцией.

На странице каталога продукции должна выводиться производимая продукция в кратком виде (изображение и название), при нажатии на продукт должна открываться полная информация о продукте.

На странице новостей должен выводиться список новостей предприятия, при нажатии на изображение новости должна открываться полная статья.

На странице материала должен выводиться любой другой материал редактируемы в произвольной форме, за пунктом меню должен быть прикреплен один материал

При разработке концепции пользовательского интерфейса важно учитывать принципы удобства использования, понятности и лаконичности. Интерфейс должен быть интуитивно понятным для пользователей разного уровня опыта и обладать привлекательным дизайном, соответствующим корпоративному стилю филиала Новополоцкого хлебозавода.

Кроме того, следует обратить внимание на адаптивность пользовательского интерфейса. Веб-сайт должен корректно отображаться на различных устройствах, таких как компьютеры, планшеты и мобильные телефоны, чтобы обеспечить удобство пользования независимо от выбранного устройства.

Важно создать удобный, понятный и лаконичный интерфейс, который удовлетворит потребности пользователей и способствует эффективному взаимодействию с веб-сайтом.

2.6 Безопасность и защита данных

Обеспечение безопасности и защиты данных является одним из наиболее важных аспектов при разработке веб-сайта. Защита конфиденциальности пользовательской информации и предотвращение несанкционированного доступа к данным являются первоочередными задачами.

Одним из распространенных методов обеспечения безопасности передачи данных является протокол HTTPS. Протокол защищенной версией стандартного протокола HTTP, который используется для передачи данных между веб-сервером и клиентским браузером. HTTPS обеспечивает шифрование данных и аутентификацию, что делает передачу информации по сети более безопасной.

Основной механизм безопасности HTTPS основан на использовании шифрования с помощью SSL (Secure Sockets Layer) или его более новой версии TLS (Transport Layer Security). Этот механизм шифрования обеспечивает конфиденциальность данных путем их защиты от перехвата и прослушивания третьими лицами.

Когда клиентское устройство (например, браузер) устанавливает безопасное соединение с сервером через HTTPS, происходит следующий процесс:

1 Клиент отправляет запрос на установку защищенного соединения с сервером.

2 Сервер отправляет свой публичный SSL-сертификат клиенту. Сертификат содержит публичный ключ сервера, который будет использоваться для шифрования данных.

3 Клиент проверяет доверенность сертификата, используя цепочку сертификации и список доверенных центров сертификации (Certificate Authorities).

4 Если сертификат признается доверенным, клиент генерирует сеансовый ключ шифрования, который будет использоваться для шифрования и расшифровки данных.

5 Клиент зашифровывает сеансовый ключ с использованием публичного ключа сервера и отправляет его на сервер.

6 Сервер получает зашифрованный сеансовый ключ и расшифровывает его с помощью своего приватного ключа.

7 Теперь клиент и сервер могут использовать сеансовый ключ для шифрования и расшифровки данных, передаваемых между ними.

Использование HTTPS необходимо на веб-сайтах, где вводится и передается конфиденциальная информация, такая как данные авторизации, информация о платежах и почтовые сервисы. Шифрование данных с использованием HTTPS позволяет предотвратить их перехват и злоупотребление третьими лицами.

Для реализации передачи данных через протокол HTTPS на веб-сервере, обрабатывающем запросы от клиентов, необходимо установить специальный SSL-сертификат. Существуют сертификаты, защищающие только один домен, а также сертификаты Wildcard SSL, которые обеспечивают защиту информации на всех поддоменах. Наличие SSL-сертификата также может повлиять на ранжирование в поисковой выдаче, хотя в настоящее время его влияние не имеет основополагающего значения, но может увеличиться в будущем.

Основные принципы обеспечения безопасности данных строятся на обеспечении целостности информации и предотвращении несанкционированного доступа. Пользователям предоставляется доступ только к тем данным, которые им разрешено редактировать, исходя из их статуса или роли. Это гарантирует, что пользователи не смогут изменять или портить данные, к которым у них нет прав доступа.

Для предотвращения возможных ошибок в базе данных предусмотрен механизм отката транзакций. В случае возникновения ошибок или несоответствия вводимых данных в базе данных, система автоматически откатывает изменения и возвращает данные к предыдущему состоянию. Это позволяет избежать сохранения некорректных данных и обеспечить целостность информации.

3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

3.1 Результаты реализации функциональных частей

При разработке веб-сайта для филиала Новополоцкого хлебозавода ОАО «Витебскхлебпром» в качестве инструментов были использованы Next.js на клиентской стороне и Strapi для административной панели. Оба инструмента предоставили мощные возможности для создания и управления функциональными частями веб-сайта.

Next.js - это популярный фреймворк для разработки веб-приложений на базе React. Он предоставляет мощные инструменты и функциональности для создания современных и эффективных веб-сайтов. Вот некоторые ключевые особенности и возможности Next.js:

- Серверный рендеринг (Server-Side Rendering - SSR): Next.js позволяет выполнять серверный рендеринг веб-страниц. Это означает, что контент страницы может быть сгенерирован на сервере перед отправкой клиенту. SSR позволяет создавать страницы, которые могут быть полностью индексированы поисковыми системами, а также улучшает время загрузки страниц и SEO-оптимизацию.

- Статическая генерация (Static Site Generation - SSG): Next.js поддерживает статическую генерацию, при которой страницы предварительно генерируются во время сборки проекта. Это позволяет достичь быстрой загрузки страниц и повышает производительность веб-сайта. Статически сгенерированные страницы могут быть кэшированы и обслуживаться без необходимости выполнения серверного кода для каждого запроса.

- Клиентский рендеринг (Client-Side Rendering - CSR): Next.js также поддерживает клиентский рендеринг, позволяя веб-страницам обновляться и взаимодействовать с пользователем без полной перезагрузки страницы. Это создает более динамичные и интерактивные пользовательские интерфейсы.

- Удобство маршрутизации: Next.js предоставляет простой и интуитивно понятный способ определения маршрутов веб-приложения. Он автоматически связывает файлы в папке «app» с соответствующими URL-адресами, что делает настройку маршрутизации простой и естественной.

- Горячая перезагрузка модулей (Hot Module Replacement - HMR): Next.js поддерживает HMR, что позволяет разработчикам вносить изменения в код и мгновенно видеть результаты без полной перезагрузки приложения. Это значительно ускоряет процесс разработки и улучшает опыт работы разработчика.

- Расширяемость и плагины: Next.js предоставляет множество плагинов и возможностей для расширения функциональности. Вы можете легко интегрировать сторонние библиотеки, добавлять пользовательские плагины и настроить среду разработки по своим потребностям.

Next.js является мощным фреймворком для создания современных веб-приложений. Он сочетает в себе преимущества серверного и клиентского рендеринга, обладает удобной маршрутизацией, поддерживает горячую перезагрузку модулей и предоставляет множество инструментов для разработки и расширения функциональности.

Next.js также легко интегрируется с SCSS (Sass), позволяя разработчикам использовать препроцессор CSS для создания стилей в своих проектах. SCSS предоставляет мощные инструменты и функции, которые упрощают процесс написания и управления стилями.

С помощью Next.js и SCSS можно создавать модульные стили, разделять стили на компоненты и повторно использовать их в разных частях приложения. SCSS позволяет использовать переменные, миксины, наследование и другие функции, что делает процесс стилизации более гибким и эффективным.

Next.js автоматически обрабатывает импорты SCSS-файлов и компилирует их в CSS стили во время сборки проекта. Также можно настроить Next.js для автоматической генерации отдельных CSS файлов или встраивания стилей непосредственно в HTML страницу.

Использование SCSS с Next.js упрощает организацию и поддержку стилей в проекте, позволяет создавать гибкие и модульные стили, а также повышает производительность и эффективность разработки веб-приложений.

Strapi в свою очередь это гибкая и мощная головная система управления контентом (CMS), построенная на Node.js. Она предоставляет возможность создавать и администрировать API-интерфейсы для ваших приложений или веб-сайтов с минимальными усилиями. Strapi является открытым исходным кодом, что означает, что вы можете настроить и расширить его в соответствии с вашими потребностями.

Основные преимущества использования Strapi для административной панели включают:

- Гибкость и масштабируемость: Strapi позволяет вам легко создавать и определять свои модели данных, а также настраивать их поля, типы и отношения. Можно создавать пользовательские конечные точки API, добавлять плагины и настраивать взаимодействие с другими инструментами и сервисами.

- Административная панель: Strapi предоставляет удобную административную панель, где можно управлять содержимым и настройками приложения. Можно создавать, редактировать и удалять записи, загружать медиафайлы, настраивать права доступа пользователей и многое другое, все через интуитивно понятный пользовательский интерфейс.

- Гибкость развертывания: Strapi может быть развернут на вашем сервере, в облачной среде или даже в контейнерной среде. Можно выбрать подходящий вариант развертывания, основываясь на требованиях проекта.

- Безопасность и авторизация: Strapi обеспечивает возможность настройки авторизации и аутентификации пользователей, а также управления доступом к данным и функциям API. Можно настраивать роли и разрешения для

различных пользовательских групп, обеспечивая безопасность контента и приложения.

– **Расширяемость:** Strapi предоставляет множество плагинов и модулей, которые позволяют расширять его функциональность и интегрировать сторонние сервисы. Можно использовать готовые плагины или создать собственные, чтобы добавить нужные вам функции.

Использование Next.js на клиентской стороне и Strapi для административной панели позволяет создать функциональный и удобный веб-сайт для филиала Новополоцкого хлебозавода ОАО «Витебскхлебпром». Эти инструменты обеспечивают высокую производительность, гибкость в управлении контентом и повышенный уровень безопасности.

Пример вывода каталога продукции представлен в листинге 3.1.

Листинг 3.1 – Страница вывода кталога продукции

```
'use client'
import styles from '../..//styles/pages/products.module.scss'
import MenuComponent from '@components/global/menu';
import Link from "next/link";
import { usePathname } from 'next/navigation'
import { host, getMenuProductItem } from
'@/utils/data.client.request';
import Product from '@components/products/product';
import TitleProductsPreloader from '@components/preloaders/title';
import ProductsPreloader from '@components/preloaders/products';
export default function Categories() {
  const pathname = usePathname();
  const { menuItem, getMenuItemIsLoading, getMenuItemIsError } =
  getMenuProductItem(pathname);
  if (getMenuItemIsLoading) {
    return (
      <>
      <TitleProductsPreloader />
      <ProductsPreloader />
      </>
    )
  }
  const content = menuItem[0];
  const renderProducts = (products) => {
    // console.log(content);
    return (
      products.map(product => {
        console.log(product)
        return (
          <Product
            key={product.id}
            host={host}
            title={product.title}
            image={product.image}
          />)))
  }
```

```

}
return (
<>
<div className={styles.title}>
<div className={styles.h1}>
<h1>{content.title ? content.title : ''}</h1>
<p>{content.related.products.length ?
content.related.products.length : 'Нет'} товара</p>
</div>
<div className={styles.baner}>
</div>
</div>
<div className={styles.products}>
{renderProducts(content.related.products)}
</div>
</>)
}

```

Пример формирования меню представлен в листинге 3.2.

Листинг 3.2 – Компонент вывода меню

```

'use client'
import styles from
'../../styles/components/dynamicPage/menu.module.scss'
import { getMenu } from '@/utils/data.client.request';
import Link from 'next/link';
import { usePathname } from 'next/navigation';
import MenuPreloader from '../preloaders/menu';
export default function MenuComponent() {
const pathname = usePathname();
const { menu, getMenuIsLoading, getMenuIsError } = getMenu();
const renderMenu = (items) => {
return (
<ul>
{items.map((item) => {
return (
<li
key={item.id}
className={item.path == pathname || pathname.includes(item.path) ?
styles.active : ''}
>
<Link href={item.path}>{item.title}</Link>
{item.items.length ? renderMenu(item.items) : ''}
</li>
)
)}}
</ul>
)
}
if (getMenuIsLoading) {
return <MenuPreloader />
}
}

```

					Отчет по преддипломной практике	Исч 27
Изм.	Лист	№ докум.	Подпись	Дата		

```

console.log(menu)
return (
<nav className={styles.nav}>
{renderMenu(menu) }
</nav>
)
}

```

В ходе реализации программного продукта был разработан веб-сайт для филиала Новополоцкого хлебозавода ОАО «Витебскхлебпром» г. Новополоцк, отвечающим всем требованиям технического задания. Представление результатов реализации приведено в документе «Описание программы» в приложении Г.

3.2 Разработка программной документации

В результате создания программного продукта была разработана следующая документация:

- техническая документация (приложение А);
- описание программы (приложение Г);
- программа и методика испытаний (приложение Д);
- руководство пользователя (приложение Е) – для помощи пользователю в ознакомлении и освоении программного продукта.

3.3 Тестирование

Тестирование программного обеспечения является важной частью процесса разработки и создания программных продуктов. От того, насколько досконально проведены тесты, зависит время окончательной сдачи проекта и необходимость последующего устранения ошибок. Тестирование программного продукта на разных этапах разработки является залогом качественного выполнения заказа. Отсутствие должного тестирования может вызвать негативные отзывы и отрицательное отношение к программе.

Виды тестирования программного обеспечения можно разделить на три основные группы:

- функциональные;
- нефункциональные;
- связанные с изменениями.

Функциональное тестирование направлено на проверку функциональности системы, то есть на то, насколько система соответствует

					Отчет по преддипломной практике	/лст 28
Изм.	Лист	№ докум.	Подпись	Дата		

требованиям и выполняет заданные функции.

В процессе функционального тестирования проверяются внешние аспекты системы, ее возможности, интерфейсы и взаимодействие с пользователем. Целью этого вида тестирования является убедиться, что программа выполняет то, для чего она предназначена, и что она работает корректно в различных ситуациях.

Тестирование может быть проведено на различных уровнях тестирования:

- Компонентное или модульное тестирование: В этом случае тестируются отдельные компоненты или модули системы. Тестировщики проверяют, что каждый модуль функционирует правильно в изоляции, а также взаимодействие между модулями.

- Интеграционное тестирование: В данном виде тестирования проверяется взаимодействие между различными модулями или компонентами системы. Целью является обнаружение возможных ошибок при интеграции и корректность передачи данных между модулями.

- Системное тестирование: Этот вид тестирования проводится на уровне готовой системы. Он направлен на проверку функциональности системы в целом. Тестировщики проверяют выполнение основных сценариев использования системы и ее поведение в различных условиях.

- Приемочное тестирование: В конечном итоге система подвергается приемочному тестированию, которое проводится заказчиком или конечными пользователями. Целью этого тестирования является убедиться, что система соответствует требованиям и ожиданиям пользователей.

При проведении функционального тестирования используются различные методы и техники, включая тестирование функций, сравнение ожидаемых и фактических результатов, использование тестовых сценариев и данных, а также автоматизацию тестирования с помощью специальных инструментов.

Важной частью функционального тестирования является разработка тестовых случаев, которые описывают шаги, ожидаемые результаты и условия тестирования. Тестировщики выполняют эти тестовые случаи и регистрируют результаты для последующего анализа и исправления ошибок.

Важно отметить, что функциональное тестирование не оценивает производительность, надежность или безопасность системы. Для проверки этих аспектов существуют другие виды тестирования, такие как тестирование производительности, нагрузочное тестирование и тестирование безопасности.

Нефункциональное тестирование направлено на проверку нефункциональных аспектов системы, таких как производительность, надежность, удобство использования, безопасность и другие качественные характеристики.

Основная цель нефункционального тестирования состоит в проверке поведения системы в различных условиях, а также в обнаружении и устранении проблем, связанных с нефункциональными аспектами ее работы.

Вот некоторые из основных видов нефункционального тестирования:

– Тестирование производительности: Этот вид тестирования направлен на оценку производительности системы в различных условиях, таких как нагрузка, объем данных и интенсивность использования. Он позволяет определить, насколько система способна обрабатывать запросы и выполнять операции в заданных ограничениях времени.

– Тестирование надежности: Это тестирование проводится для оценки стабильности и надежности системы. Оно включает в себя проверку способности системы справляться с ошибками, восстанавливаться после сбоев и предотвращать потерю данных.

– Тестирование безопасности: Данный вид тестирования направлен на обнаружение уязвимостей и проверку уровня защиты системы от несанкционированного доступа, атак и утечек данных. В ходе тестирования проверяются механизмы аутентификации, авторизации, шифрования и другие аспекты безопасности.

– Тестирование удобства использования: Этот вид тестирования оценивает, насколько система удобна в использовании для конечных пользователей. В ходе тестирования анализируются интерфейсы, навигация, интуитивность и другие аспекты, влияющие на пользовательский опыт.

– Тестирование совместимости: Этот вид тестирования проводится для проверки совместимости системы с различными операционными системами, браузерами, устройствами и другими зависимостями. Цель состоит в обнаружении возможных проблем, которые могут возникнуть при использовании системы в различных окружениях.

– Тестирование масштабируемости: В данном виде тестирования проверяется способность системы масштабироваться в зависимости от увеличения объема данных, пользователей или нагрузки. Это позволяет убедиться в том, что система может эффективно функционировать в условиях роста и расширения.

Нефункциональное тестирование позволяет выявить и устранить проблемы, связанные с производительностью, безопасностью, удобством использования и другими аспектами, которые могут оказывать влияние на общую эффективность и успешность системы.

Тестирование, связанное с изменениями, направлено на проверку работоспособности системы после внесения изменений, таких как исправление ошибок (дефектов) или внедрение новых функций. Цель этого тестирования – убедиться, что система продолжает функционировать должным образом и не возникают нежелательные побочные эффекты.

Вот некоторые виды тестирования, которые проводятся в связи с изменениями:

– Дымовое тестирование: Это первичное тестирование, которое выполняется после внесения изменений или исправления дефектов. Оно направлено на проверку основных функций системы, чтобы убедиться, что основные сценарии работы работают корректно. Если дымовое тестирование

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						<i>30</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

проходит успешно, это указывает на то, что система готова для дальнейшего тестирования.

– Регрессионное тестирование: Это тестирование проводится для проверки работоспособности системы после внесения изменений. Оно направлено на обнаружение возможных побочных эффектов или нарушений функциональности, которые могут возникнуть из-за изменений в системе. Регрессионное тестирование включает повторное выполнение предыдущих тестовых случаев и проверку, что система продолжает работать как ожидается.

– Тестирование сборки: Это тестирование проводится после сборки новой версии программного обеспечения. Оно направлено на проверку работоспособности базовых функций системы, чтобы убедиться, что сборка прошла успешно и основные компоненты функционируют без проблем.

– Санитарное тестирование или проверка согласованности/исправности: Это тестирование выполняется после внесения незначительных изменений или исправления незначительных дефектов. Оно направлено на проверку, что система остается согласованной и работает без существенных ошибок после внесенных изменений.

Тестирование, связанное с изменениями, необходимо для подтверждения работоспособности системы после изменений и обеспечения стабильности и надежности программного обеспечения. Правильное проведение такого тестирования помогает обнаружить и устранить возможные проблемы, связанные с изменениями, и гарантирует, что система продолжает работать эффективно и соответствует требованиям пользователей.

Результаты тестирования показали, что система является готовым к эксплуатации программным продуктом с устойчивым использованием и интуитивно понятным интерфейсом. Подробные результаты тестирования приведены в таблице Д.1 (приложение Д).

4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

4.1 Обоснование необходимости вывода продукта на рынок

Целью экономического раздела данного дипломного проекта является расчёт затрат на разработку программного обеспечения и определение экономической эффективности от его внедрения.

Веб-сайт позволит филиалу Новополоцк хлебозавод ОАО «Витебскхлебпром» эффективно привлекать новых клиентов и удерживать существующую клиентскую базу. Виртуальное присутствие в интернете является важным инструментом маркетинга и продвижения товаров и услуг. Современные потребители активно используют интернет для поиска информации о предприятиях и продуктах, поэтому наличие профессионального и информативного веб-сайта будет способствовать увеличению узнаваемости и привлечению новых клиентов к филиалу Новополоцк хлебозавода.

Веб-приложение содержит информацию о продукции, предоставляемой филиалом. Сайт имеет удобный и интуитивно понятный интерфейс, позволяющий пользователям легко ориентироваться на страницах и находить необходимую информацию. Кроме того, сайт адаптирован для работы на различных устройствах, таких как компьютеры, планшеты и смартфоны, что обеспечит его доступность для широкого круга потенциальных потребителей.

Потребителем разрабатываемого программного обеспечения являются клиенты и партнеры филиала Новополоцк хлебозавод ОАО «Витебскхлебпром» города Новополоцк. Веб-сайт предоставляет им удобный и информативный инструмент для получения необходимой информации о продукции, услугах и контактной информации. Клиенты могут легко ознакомиться с ассортиментом хлебобулочных изделий и ознакомиться с акциями, а партнеры смогут оперативно получать информацию о сотрудничестве и условиях работы с филиалом. Все это содействует удовлетворению потребностей клиентов и повышению эффективности взаимодействия с партнерами.

4.2 Структура (этапы) работ по созданию программного обеспечения

4.3 Составление сметы затрат на разработку программного обеспечения

4.4 Расчет экономического эффекта разработчика программного обеспечения

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>32</i>

4.4.1 Экономический эффект у разработчика программного обеспечения

4.4.2 Экономический эффект от использования программного обеспечения у пользователя (заказчика)

4.4.3 Расчет экономической эффективности разработки ПО для заказчика

4.5 Выводы по экономической части

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						<i>33</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

ЗАКЛЮЧЕНИЕ

СПИСОК ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А
(обязательное)
Техническое задание

Введение

Наименование продукта: веб-сайт для филиала Новополоцк хлебозавод ОАО «Витебскхлебпром», г. Новополоцк.

Данный проект будет использоваться для размещения производимой продукция предприятия, администрирования сайта, добавления и редактирования материалов и новостей.

А.1 Основания для разработки

Основанием для разработки является приказ №__ от 05.2021. «О закреплении тем дипломных проектов студентов факультета информационных технологий специальности 1-40-01-01 «Программное обеспечение информационных технологий» по кафедре технологий программирования УО «Полоцкий государственный университет имени Евфросинии Полоцкой»

Тема разработки – веб-сайт для филиала Новополоцк хлебозавод ОАО «Витебскхлебпром», г. Новополоцк, выданная студенту группы 19ИТ-2 Гилимовичу Артёму Николаевичу, руководителем назначен старший преподаватель кафедры технологий программирования Кравченко Юрий Николаевич.

А.2 Назначение разработки

Основной целью разработки интернет-магазина является облегчение администрирования и автоматизация работы предприятия. Веб-приложение представляет следующие возможности для пользователя сервисом:

- просмотр продукции;
- просмотр новостей;
- добавления отзыва к продукту;
- отправка жалоб и предложений о предприятии;
- возможность связаться с представителями организации;
- просмотр сведений об организации.

Также для администратора доступна следующая функциональность:

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		<i>36</i>

- добавление, редактирование, удаление продукции;
- добавление, редактирование, удаление новостей;
- добавление, редактирование, удаление категорий и добавление категории к продукту;
- редактирование материалов об предприятии и контакты;
- добавление, редактирование, удаление слайдов для слайдера;
- добавление, редактирование, удаление баннеров к любой странице сайта;
- обработка отзывов, сообщений;
- отправка писем на обращения пользователей.

А.3 Требование к программе или программному изделию

А.3.1 Требования к функциональным характеристикам

К функциональным характеристикам для клиента выдвигаются следующие требования:

- клиент может просмотреть продукцию с возможностью сортировки различными способами, просматривать всю продукцию либо отдельной категории, оставить отзыв к товару и поставить оценку;
- клиент может просматривать материалы сайта, ознакомиться с информацией об предприятии и узнать контакты для связи с представителем организации;
- клиент может связаться с представителями предприятия, отправив письмо прямо на сайте сервиса, либо перейти к другим средствам связи;
- клиент может просматривать новости предприятия.

Для администратора:

- администратор может добавить, удалить, отредактировать информацию об продукте производимом предприятием;
- администратор может добавлять, удалять, редактировать новости об предприятии;
- редактировать материалы об предприятии и контакты;
- возможность создавать весь контент на различных языках;
- просматривать отзывы пользователей и отвечать письмами либо с помощью других средств связи;
- администратор может добавлять, редактировать, удалять слайды для слайдера;
- возможность добавить, удалить отредактировать баннер на любой странице приложения.

А.3.2 Требования к надёжности

Программное средство должно обеспечивать контроль корректности входных данных. В случае обнаружения ошибок во входных данных пользователю должны выводиться соответствующие сообщения без сброса введенных данных.

А.3.3 Условия эксплуатации

Данное программное средство не требует предварительной подготовки.

А.3.4 Требования к составу и параметрам технических средств

Минимальными и достаточными требованиями по конфигурации оборудования клиента для функционирования системы являются:

- процессоры с частотой не менее 233 МГц;
- наличие браузера;
- объем оперативной памяти не мене 128 Мб;
- свободное дисковое пространство не менее 50 Мб.

А.3.5 Требования к информационной и программной совместимости

Для программного обеспечения клиента требуется:

- Операционная система: любая операционная система;
- Любой браузер.

Для программного обеспечения сервера требуется:

- Node.js.

А.3.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке отсутствуют.

А.3.7 Требования к транспортированию и хранению

Требования к транспортировке и хранению отсутствуют.

А.4 Требования к программной документации

Программная документация должна состоять из следующих документов:

- техническое задание в соответствии с ГОСТ 19.201-78 ЕСПД;
- описание программы в соответствии с ГОСТ 19.402-2000 ЕСПД;
- руководство оператора в соответствии с ГОСТ 19.505-79 ЕСПД;
- программа и методика испытаний в соответствии с ГОСТ 19.301-2000 ЕСПД.

Содержание и структура программной документации соответствует требованиям ЕСПД.

А.5 Технико-экономические показатели

Эффективность данного программного продукта в основном обуславливается скоростью работы и широкими возможностями администратора без использования дополнительных узконаправленных инструментов.

А.6 Стадии и этапы разработки

Этапы разработки:

- постановка задачи;
- анализ исходных данных;
- разработка программного средства;
- тестирование программного средства;
- разработка технической документации к данному программному средству.

А.7 Порядок контроля и приёмки

Для контроля выполнения требования технического задания необходимо

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						<i>39</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

провести испытания. Порядок и состав испытания определяются программой и методикой испытаний.

Приемка готового программного средства осуществляется комиссией по защите дипломного проекта в соответствии с программой и методикой испытаний.

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						40
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

ПРИЛОЖЕНИЕ Б
(обязательное)
Диаграмма вариантов использования

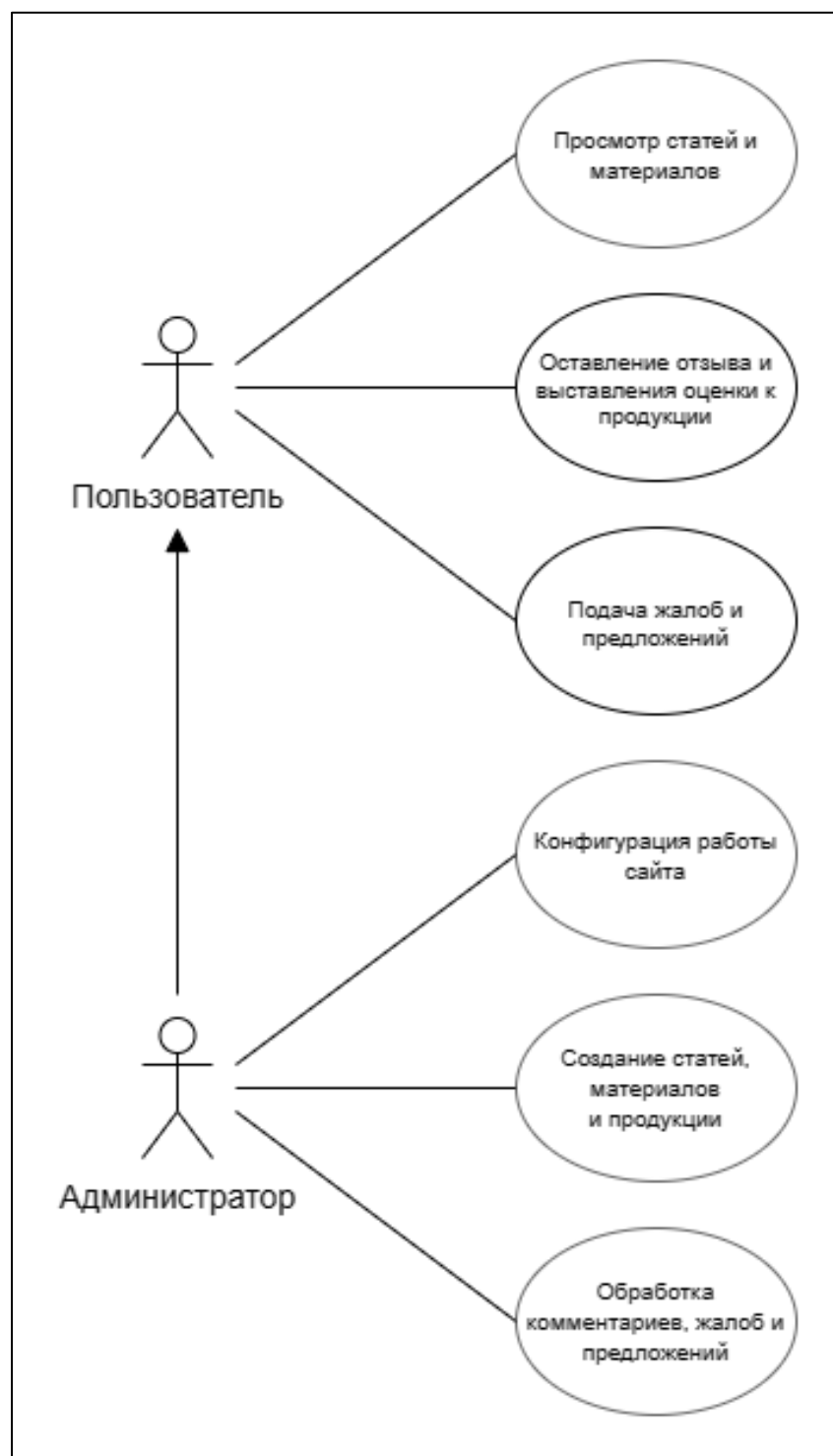


Рисунок Б.1 – Диаграмма вариантов использования

ПРИЛОЖЕНИЕ В

(справочное)

Схема базы данных

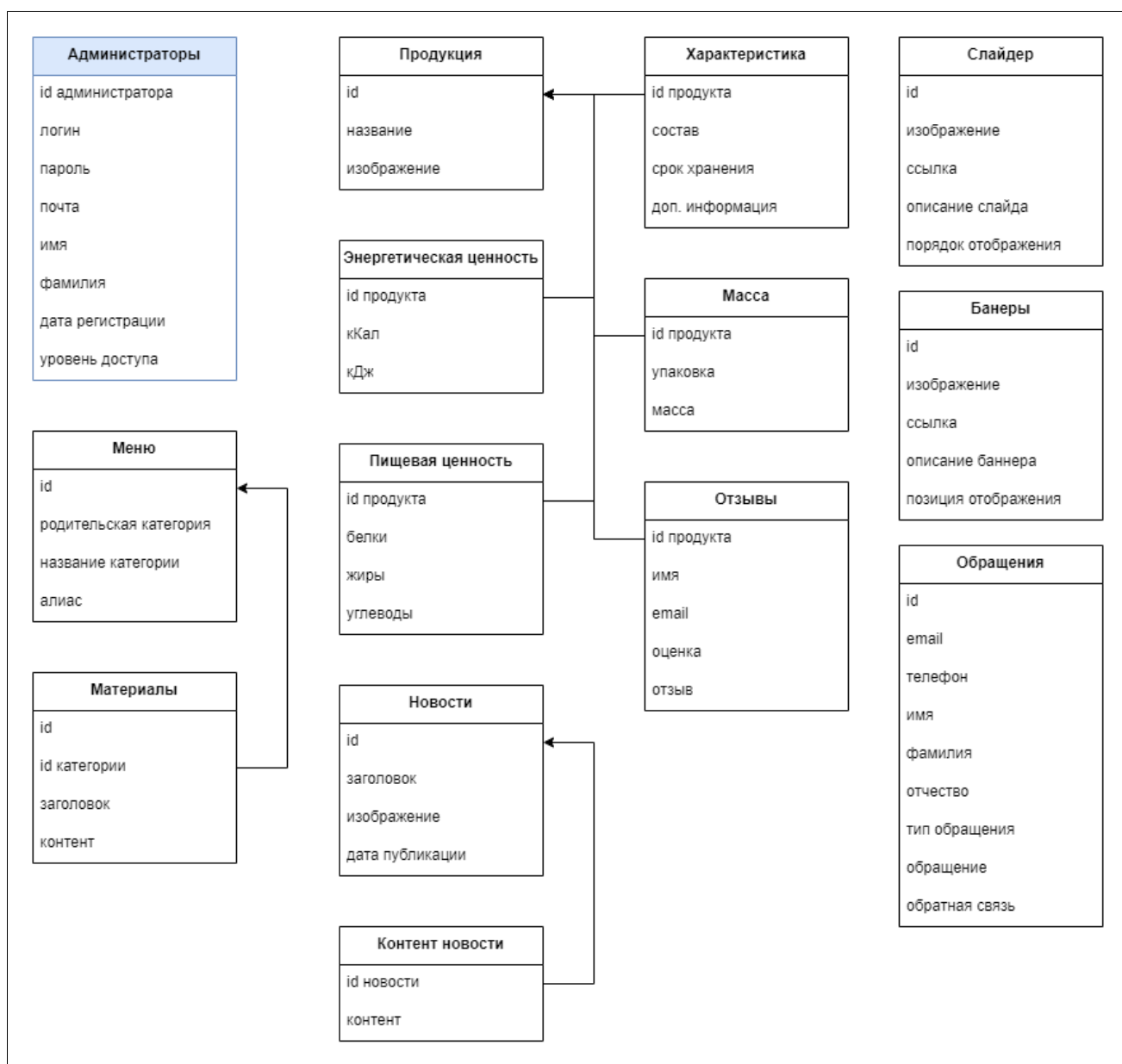


Рисунок В.1 – Схема базы данных

ПРИЛОЖЕНИЕ Г
(обязательное)
Описание программы

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						43
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

ПРИЛОЖЕНИЕ Д
(обязательное)
Программа и методика испытаний

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						44
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

ПРИЛОЖЕНИЕ Е
(обязательное)
Руководство пользователя

					<i>Отчет по преддипломной практике</i>	<i>Лист</i>
						45
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		