

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ПЛАН РАБОТЫ НА ПРЕДПРИЯТИИ	4
1.1 Изучение структуры предприятия	4
1.2 Изучение должностных инструкций работников подразделения	5
1.3 Обзор задач, решаемых подразделением	7
1.4 Обзор используемых технических средств на предприятии	8
1.5 Выполнение задач на предприятии	11
2. ПЛАН ИНДИВИДУАЛЬНОГО ЗАДАНИЯ	13
2.1 Анализ существующих решений. Описание достоинств и недостатков	13
2.2 Постановка задачи и описание функций разрабатываемой системы	18
2.3 Выбор средств реализации	22
ЗАКЛЮЧЕНИЕ	30
СПИСОК СОКРАЩЕНИЙ	31
СПИСОК ЛИТЕРАТУРЫ	32

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		2

ВВЕДЕНИЕ

В современном мире эффективность образовательного процесса во многом определяется доступностью, структурированностью и удобством использования образовательных ресурсов. Несмотря на существование различных подходов к обучению и множества образовательных платформ, наблюдается значительный разрыв между структурированным представлением информации и возможностью быстрого доступа к конкретным инструкциям, что существенно снижает эффективность образовательного процесса и затрудняет освоение новых технологий.

Разработка образовательной платформы для обмена и изучения структурированных обучающих материалов представляет собой актуальную задачу, объединяющую современные технологии веб-разработки и инновационные подходы к организации образовательного контента. Основная цель проекта – создание системы, которая позволит пользователям не только получать доступ к структурированным обучающим материалам, но и создавать собственные образовательные ресурсы, визуализировать взаимосвязи между различными компонентами изучаемых технологий и отслеживать прогресс обучения.

В ходе преддипломной практики на предприятии ОАО «Савушкин продукт» в бюро перспективных разработок были изучены структура предприятия, должностные инструкции работников подразделения, задачи, решаемые подразделением, используемые технические средства, а также выполнены задания по доработке существующих на предприятии проектов и разработке нового проекта.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

1.1 Изучение структуры предприятия

Организационная структура - это схема, отражающая иерархию управления и взаимосвязи между различными подразделениями и должностями внутри организации. Она показывает, как устроено управление компанией, кто кому подчиняется, какие подразделения существуют и как они взаимодействуют. Это "скелет" организации, определяющий ее формальные коммуникации и потоки информации.

Организационная структура ОАО «Савушкин продукт»

ОАО «Савушкин продукт» представляет собой крупное производственное объединение, включающее 6 высокотехнологичных производственных площадок, расположенных в Бресте, Пинске, Столине, Берёзе, Ивацевичах и Барановичах. Организационная структура предприятия построена по территориально-производственному принципу.

Во главе предприятия стоит генеральный директор, которому подчиняются несколько заместителей по различным направлениям деятельности. Каждое функциональное направление представлено отдельным подразделением, а именно:

- Производственный департамент;
- Финансовое управление;
- Отдел сбыта и маркетинга;
- Отдел информационных технологий;
- Юридический отдел;
- Отдел кадров.

Внутри этих крупных подразделений выделены дополнительные структурные единицы, такие как секторы, группы, бюро. Управление предприятием осуществляется через головной офис, расположенный в Бресте. В структуру управления также входят:

- Совет директоров
- Генеральная дирекция
- Производственная дирекция
- Коммерческая дирекция
- Финансовая дирекция
- Служба качества
- Техническая дирекция

Товаропроводящая сеть компании включает 6 торговых филиалов в Беларуси (Минск, Гомель, Витебск, Могилёв, Гродно, Брест) и торговый дом в Москве. Каждый филиал имеет собственную структуру управления, подчиняющуюся головному офису.

Производственные площадки имеют следующие основные подразделения:

- Производственные цеха
- Лаборатории контроля качества
- Складские комплексы
- Инженерно-технические службы
- Административно-хозяйственные отделы

Все производственные площадки сертифицированы по международным стандартам СТБ ISO 9001-2015, ISO 14001, ISO 45001 и FSSC 22000, что определяет высокий

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

уровень организации производственных и управленческих процессов.

Функциональная структура ОАО «Савушкин продукт»

Функциональная структура - это способ организации деятельности предприятия, при котором каждое подразделение специализируется на выполнении определенных функций или задач. Она ориентирована на специализацию и разделение труда, что позволяет повысить профессионализм сотрудников и эффективность выполнения отдельных операций и процессов.

Основные преимущества функциональной структуры ОАО «Савушкин продукт»:

- Высокая компетентность сотрудников в своей области;
- Четкое распределение ответственности;
- Возможность контроля качества работ;
- Оперативность решения узких, специализированных задач.

Функциональная структура на ОАО «Савушкин продукт» имеет следующее распределение задач и ответственности:

- Производственный департамент отвечает за организацию и управление производственными процессами на предприятии;
- Финансовое управление занимается финансовым планированием, учетом и контролем;
- Отдел сбыта и маркетинга отвечает за продвижение продукции и взаимодействие с клиентами;
- Отдел информационных технологий разрабатывает и внедряет информационные системы, а также обеспечивает техническую поддержку;
- Юридический отдел осуществляет правовое сопровождение деятельности предприятия;
- Отдел кадров занимается управлением персоналом.

В составе технической дирекции функционирует Бюро перспективных разработок, выполняющее стратегическую функцию по поиску, разработке и внедрению инноваций. К основным задачам Бюро относится разработка и поддержка инновационных инструментов управления предприятием, создание систем описания технологических процессов, изучение новых технологий для повышения эффективности производства.

Взаимосвязь организационной и функциональной структур

Таким образом, организационная структура определяет иерархию управления, а функциональная - распределение задач и ответственности между различными подразделениями ОАО «Савушкин продукт». Предприятие построено по принципу вертикальной интеграции с четким распределением функций между подразделениями и эффективной системой управления, что обеспечивает успешное функционирование предприятия как единого производственного комплекса.

1.2 Изучение должностных инструкций работников подразделения

В процессе прохождения практики в компании ООО «Савушкин продукт» одной из важных задач являлось изучение должностных инструкций работников подразделения. Ознакомление с данными документами позволяет понять функциональные обязанности,

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

права, ответственность и квалификационные требования к сотрудникам, что способствует более эффективному вхождению в рабочий процесс. Должностные инструкции играют важную роль в организации труда и обеспечении выполнения поставленных задач.

Подразделение, в котором проходила практика, — бюро перспективных разработок, занимается автоматизированными системами управления технологическими процессами (АСУТП). Основными задачами отдела являются:

- Разработка, внедрение и сопровождение АСУТП на предприятии;
- Обслуживание программного обеспечения для автоматизации технологических процессов;
- Диагностика и устранение неисправностей в программном обеспечении;
- Обеспечение информационной безопасности систем управления;
- Техническая поддержка пользователей АСУТП.

В подразделении работают следующие категории сотрудников:

- Руководитель отдела программирования;
- Программисты АСУТП;
- Инженеры по автоматизации;
- Системные администраторы АСУТП.

В ходе изучения были рассмотрены должностные инструкции следующих работников:

- Руководителя отдела программирования;
- Программиста АСУТП;
- Инженера по автоматизации.

Руководитель отдела программирования отвечает за организацию работы отдела, распределение задач между сотрудниками, контроль выполнения поставленных задач, обеспечение соблюдения политики информационной безопасности и взаимодействие с другими подразделениями предприятия.

Программист АСУТП занимается разработкой программного обеспечения для автоматизации технологических процессов, написанием программ для контроллеров, тестированием программного обеспечения и его внедрением в производственные процессы. Основные обязанности программиста АСУТП включают:

- Разработка алгоритмов управления технологическими процессами;
- Программирование контроллеров
- Тестирование программного обеспечения;
- Ведение технической документации;
- Поддержка и обслуживание программного обеспечения АСУТП.

Инженер по автоматизации отвечает за настройку и наладку оборудования АСУТП, интеграцию программного обеспечения с аппаратной частью и проведение диагностики систем. Его обязанности включают:

- Настройку контроллеров и интерфейсов;
- Монтаж и обслуживание датчиков и исполнительных механизмов;
- Тестирование систем автоматизации;
- Взаимодействие с программистами для устранения ошибок;
- Подготовку технической документации.

Должностные инструкции являются основой для распределения обязанностей между сотрудниками, что позволяет четко определить сферу ответственности каждого работника. Они способствуют соблюдению трудовой дисциплины, помогают избежать дублирования функций и обеспечивают контроль качества выполняемой работы.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

На практике инструкции помогают оперативно решать возникающие задачи, так как каждый сотрудник знает свои обязанности и зону ответственности.

1.3 Обзор задач, решаемых подразделением

На ОАО «Савушкин продукт», можно отметить следующее распределение задач и ответственности:

- Производство продукции: выпуск молочных товаров (йогурты, сыры, творог, молоко).
- Контроль качества и безопасность: соблюдение строгих стандартов.
- Логистика и снабжение: управление цепочками поставок сырья и готовой продукции.
- Инновации: внедрение новых технологий и разработка продуктов.
- Экология и устойчивое развитие: минимизация экологического следа.
- Социальная ответственность: поддержка местных сообществ и сотрудников.

Производственные задачи:

- Оптимизация производственных процессов. Подразделение решает задачи повышения эффективности линий, снижения энергозатрат и увеличения объемов выпуска. Например, автоматизация упаковки позволила сократить время обработки заказов на 15

- Расширение ассортимента. Разработка новых продуктов, таких как линейки органических йогуртов или безлактозных сыров, отвечает запросам рынка. В 2023 году брестский филиал запустил производство сыра с пробиотиками, что стало ответом на тренд здорового питания.

- Соблюдение стандартов. Производство соответствует требованиям ISO, HACCP и белорусским ГОСТам, что критически важно для экспорта в ЕАЭС и другие страны.

Контроль качества и безопасность:

- Многоуровневый контроль. Каждая партия проходит лабораторные испытания на микробиологические и физико-химические показатели. Внедрена система отслеживания сырья от поставщика до прилавка.

- Работа с рекламациями. Подразделение оперативно реагирует на обратная связь от потребителей, анализируя причины дефектов и корректируя процессы.

- Сертификация. Поддержка международных сертификатов (например, Halal, Organic) расширяет возможности экспорта.

Логистика и снабжение:

- Управление поставками сырья. Сотрудничество с местными фермерскими хозяйствами обеспечивает стабильность поставок молока-сырья. Внедрена цифровая платформа для прогнозирования спроса.

- Дистрибуция готовой продукции. Оптимизация маршрутов доставки снизила логистические издержки на 10

- Складское хозяйство. Использование систем WMS (Warehouse Management System) повысило точность учета и скорость отгрузки.

Инновации и развитие:

- Внедрение технологий. На заводе в Бресте используются роботизированные ли-

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

нии розлива, IoT-датчики для мониторинга оборудования.

- Исследования и разработки. Локальный RD-центр тестирует новые рецептуры, например, растительно-молочные гибриды для веганского рынка.

- Цифровизация. Внедрение ERP-системы интегрировало производство, логистику и финансы, сократив время согласований.

Экологическая ответственность:

- Снижение отходов. Переход на биоразлагаемую упаковку и переработка 95

- Энергоэффективность. Установка солнечных панелей и рекуперация тепла от оборудования сократили углеродный след на 20

- Водопользование. Система замкнутого цикла очистки воды уменьшила ее потребление на 30

Социальные инициативы:

- Поддержка сотрудников. Программы обучения, медицинское страхование, корпоративный детский сад.

- Участие в жизни региона. Спонсорство школ, экологические акции (посадка деревьев), продовольственная помощь нуждающимся.

- Развитие местной экономики. Закупки сырья у брестских фермеров создают рабочие места в регионе.

Брестское подразделение ОАО «Савушкин продукт» решает комплекс задач, направленных на укрепление позиций компании как локального и международного игрока. Через инновации, качество и ответственность подразделение вносит вклад в устойчивое развитие бизнеса и общества. Дальнейшие цели включают углубление цифровизации, расширение экспорта и усиление экологических инициатив, что соответствует глобальным трендам и стратегии компании.

1.4 Обзор используемых технических средств на предприятии

На предприятии "Савушкин" в рамках автоматизации производственных процессов используется комплексная система автоматизации, состоящая из программного комплекса SCADA, программно-технического управляющего комплекса PTUSA и инструмента проектирования EasyEPLANner. Бюро перспективных разработок активно участвует в развитии и совершенствовании этих систем.

Рассмотрим аппаратное обеспечение, используемое на предприятии. Основу системы управления составляют программируемые логические контроллеры Phoenix Contact PLCnext серии AXC F 2152 и контроллеры WAGO PFC200, обеспечивающие гибкую платформу для реализации систем управления. Также используются промышленные компьютеры с операционными системами Linux для выполнения программы управления ptusa_main и различные модули ввода-вывода для подключения датчиков и исполнительных механизмов.

Для контроля технологических процессов применяются различные датчики: температуры, давления, расходомеры для измерения объемов перекачиваемых продуктов, уровнемеры для контроля наполнения емкостей. Управление потоками осуществляется с помощью клапанов и задвижек с электроприводами, а для управления электродвигателями

						Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

насосов и мешалок используются частотные преобразователи.

Сетевая инфраструктура предприятия включает промышленные коммутаторы для организации сети Ethernet, преобразователи интерфейсов для подключения устройств с различными протоколами связи и оптоволоконные линии связи для обеспечения надежной передачи данных между удаленными участками. Серверное оборудование представлено серверами для размещения SCADA-системы и баз данных, рабочими станциями операторов для мониторинга и управления технологическими процессами, а также системами резервного копирования для обеспечения сохранности данных.

Программное обеспечение предприятия включает несколько ключевых компонентов. Программный комплекс SCADA состоит из серверной части EasyServer для обработки и хранения данных, клиентской части Monitor для визуализации и управления технологическими процессами, редактора конфигурации базы данных DBXmlEditor и ряда вспомогательных библиотек: buglog.dll для ведения журнала ошибок, ConnectionLog.dll для регистрации подключений, MSUniServ.dll для универсального сервиса, ClientPLog.dll и ClientTLog.dll для ведения журналов, PGPropServ.dll и propservice.dll для работы с параметрами.

Система управления PTUSA включает основную программу управления ptusa_main, непосредственно контролирующую технологические процессы, скрипты на языке Lua для реализации логики управления технологическими объектами и модульную архитектуру программного обеспечения, обеспечивающую гибкость и масштабируемость системы.

Важным компонентом системы автоматизации является EasyEPLANner – дополнение (Add-In) для EPLAN Electric P8 версии 2.9, представляющее собой инструмент для автоматизации проектирования систем управления. EasyEPLANner позволяет генерировать Lua-скрипты для программирования контроллеров Phoenix Contact PLCnext и WAGO PFC200, а также предоставляет средства для описания технологических объектов, операций, шагов и ограничений.

Для разработки компонентов системы используются различные средства: Delphi (версии 11.3 или 12.2) для разработки компонентов SCADA-системы, C# для разработки EasyEPLANner, CMake для организации процесса сборки проекта PTUSA, Qt Creator и Microsoft Visual Studio для разработки и отладки программного обеспечения, PLCnext Technology C++ Toolchain для разработки под контроллеры Phoenix Contact. Система контроля версий Git используется для управления изменениями в программном коде, GitHub Actions – для непрерывной интеграции и автоматизированного тестирования, а SonarCloud – для анализа качества кода.

Для разработки SCADA применяются компоненты DevExpress (библиотека для создания пользовательского интерфейса), EControl SyntaxEditor (компонент для редактирования кода), Advantage Database Components (компоненты для работы с базами данных), oXml (библиотека для работы с XML), Embarcadero Sockets Components (компоненты для сетевого взаимодействия), Borland Database Engine (система управления базами данных) и QuickLogger (компонент для ведения журналов).

Тестирование системы осуществляется с помощью DUnit Tests для модульного тестирования компонентов SCADA, автоматизированного тестирования с использованием GitHub Actions для PTUSA и EasyEPLANner, а также анализа покрытия кода с помощью Codecov и SonarCloud для контроля качества тестирования.

Архитектура системы управления имеет многоуровневую структуру автоматизированной системы управления технологическими процессами, распределенную систему управления для повышения надежности и гибкости, а также интеграцию SCADA, PTUSA

						Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

и EasyEPLANner для обеспечения комплексного подхода к автоматизации.

Принцип работы программы управления PTUSA основан на выполнении технологического процесса через последовательное выполнение операций для каждого технологического объекта, иерархической структуре управления (технологические объекты → операции → шаги → действия) и событийно-ориентированной модели для обработки команд и изменений режимов работы.

Взаимодействие компонентов SCADA построено на клиент-серверной архитектуре с центральным сервером EasyServer, системе мониторинга через клиентское приложение Monitor, хранении данных в реляционной базе данных с возможностью резервного копирования и обмену данными через специализированные библиотеки и сервисы.

EasyEPLANner играет важную роль в процессе проектирования, обеспечивая автоматизацию проектирования электрических схем в EPLAN Electric P8, описание технологических объектов и их свойств, генерацию Lua-кода для программирования контроллеров и сокращение трудозатрат инженеров-автоматизаторов и программистов.

Система автоматизации предприятия обладает широкими функциональными возможностями: визуализация технологических процессов в реальном времени, управление рецептурами и партиями продукции, система учета энергоресурсов для оптимизации энергопотребления, гибкая настройка алгоритмов управления с помощью скриптов Lua, ведение журналов и отчетов для документирования производственных процессов, автоматизированное проектирование с помощью EasyEPLANner.

Преимущества собственной разработки системы автоматизации включают адаптацию под специфические требования производства молочной продукции, оперативное внесение изменений в алгоритмы управления, независимость от сторонних разработчиков SCADA-систем, оптимизацию затрат на внедрение и поддержку системы автоматизации, возможность глубокой интеграции различных компонентов системы и открытый исходный код для EasyEPLANner, обеспечивающий прозрачность и возможность модификации.

Перспективы развития системы автоматизации предприятия включают расширение функциональности системы управления, интеграцию с системами машинного обучения для предиктивной аналитики, разработку цифровых двойников технологических процессов для моделирования и оптимизации, внедрение технологий промышленного интернета вещей (IIoT) для сбора и анализа данных, а также совершенствование EasyEPLANner для поддержки новых типов контроллеров и технологических объектов.

Внедрение и постоянное совершенствование собственной системы автоматизации, состоящей из программного комплекса SCADA, программно-технического управляющего комплекса PTUSA и инструмента проектирования EasyEPLANner, позволяет предприятию "Савушкин" обеспечивать высокий уровень автоматизации производственных процессов, гибко адаптировать систему под специфические требования производства и оперативно реагировать на изменения технологических процессов. Бюро перспективных разработок активно участвует в развитии системы, внедрении новых функциональных возможностей и повышении надежности работы всего комплекса.

1.5 Выполнение задач на предприятии

В рамках преддипломной практики, проходившей в течение полутора месяцев на предприятии ОАО «Савушкин продукт», был выполнен комплекс задач, направленных на изучение структуры предприятия, ознакомление с используемыми техническими средствами и программными комплексами, а также выполнение индивидуальных заданий по доработке существующих проектов.

На начальном этапе практики была изучена организационная структура предприятия ОАО «Савушкин продукт», представляющего собой крупное производственное объединение с 6 высокотехнологичными производственными площадками в различных городах Беларуси. Были рассмотрены основные подразделения предприятия. Особое внимание было уделено изучению структуры бюро перспективных разработок, занимающегося автоматизированными системами управления технологическими процессами (АСУТП).

В процессе изучения должностных инструкций работников подразделения были детально рассмотрены обязанности руководителя отдела программирования, программиста АСУТП и инженера по автоматизации. Это позволило понять распределение задач и ответственности между сотрудниками, что является основой эффективной организации труда в подразделении. Были изучены основные задачи, решаемые бюро перспективных разработок.

Значительное внимание было уделено изучению технических средств, используемых на предприятии. Были рассмотрены программируемые логические контроллеры Phoenix Contact PLCnext серии AXC F 2152 и контроллеры WAGO PFC200, промышленные компьютеры с операционными системами Linux, различные датчики (температуры, давления, расходомеры, уровнемеры), исполнительные механизмы (клапаны, задвижки с электроприводами), а также сетевая инфраструктура предприятия, включающая промышленные коммутаторы, преобразователи интерфейсов и оптоволоконные линии связи.

После ознакомления с теоретическими аспектами работы предприятия была выполнена практическая часть, включающая сборку и запуск ключевых проектов, используемых на предприятии. Был успешно собран и запущен проект $ptusa_{main}$, $EasyEPLANner-EPL$

После успешной сборки и запуска проектов было проведено их детальное изучение. Были рассмотрены архитектура и принципы работы программы управления PTUSA, основанные на выполнении технологического процесса через последовательное выполнение операций для каждого технологического объекта. Был изучен проект EasyEPLANner, позволяющий генерировать Lua-скрипты для программирования контроллеров и предоставляющий средства для описания технологических объектов. Также был изучен проект SCADA, построенный на клиент-серверной архитектуре с центральным сервером EasyServer и системой мониторинга через клиентское приложение Monitor.

Особое внимание было уделено изучению готового решения веб-версии SCADA, которое позволяет осуществлять мониторинг и управление технологическими процессами через веб-интерфейс, что обеспечивает доступ к системе с различных устройств без необходимости установки специального программного обеспечения.

В рамках индивидуальных заданий были выполнены доработки существующих проектов. Для проекта SCADA было реализовано автозаполнение полей при импорте тегов OPC UA Browser, что значительно упростило процесс настройки системы и сократило время, необходимое для импорта тегов. Для проекта $ptusa_{main}/$,

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

Завершающим этапом практики стала разработка клиентской части веб-версии SCADA на React. Данная разработка направлена на создание удобного и отзывчивого пользовательского интерфейса для системы мониторинга и управления технологическими процессами, доступного через веб-браузер. Использование React обеспечило компонентный подход к разработке, высокую производительность при обновлении интерфейса и возможность создания сложных пользовательских интерфейсов.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

2. ПЛАН ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

2.1 Анализ существующих решений. Описание достоинств и недостатков

2.1.1 Обзор существующих подходов к обучению

В данном разделе проводится анализ существующих подходов к обучению и образовательных платформ с точки зрения эффективности образовательного процесса, доступности знаний и удобства их структурирования. Рассмотрены их функциональные возможности, преимущества и недостатки для обучающихся.

На сегодняшний день существует несколько основных подходов к получению образовательной информации, каждый из которых имеет свои особенности с точки зрения образовательного процесса:

Обучение в университете представляет собой традиционный формат получения знаний, который характеризуется следующими особенностями:

- Структурированная программа обучения с последовательным изложением материала
- Непосредственное взаимодействие с преподавателями и возможность получения обратной связи
- Практические занятия и лабораторные работы для закрепления теоретических знаний
- Социальное взаимодействие с другими студентами и возможность групповой работы
- Получение официального документа об образовании

Однако с точки зрения образовательного процесса университетское образование имеет ряд недостатков:

- Фиксированное расписание и темп обучения
- Ограниченная гибкость программы обучения
- Учебные программы часто отстают от быстро меняющихся технологий
- Ограниченная возможность для самостоятельного углубленного изучения отдельных тем
- Сложность быстрого поиска конкретной информации по решению определенной задачи

Онлайн-курсы (Udemy, Coursera, Pluralsight) предлагают более гибкий подход к обучению, который имеет следующие преимущества для образовательного процесса:

- Доступ к материалам в любое время и из любого места
- Возможность выбора конкретных курсов в зависимости от интересов и потребностей
- Видеоформат обучения с дополнительными материалами
- Система оценок и отзывов для определения качества курсов
- Возможность обучаться в собственном темпе, повторно просматривая сложные моменты или ускоряя просмотр знакомого материала

Однако с точки зрения образовательного процесса онлайн-курсы имеют следующие недостатки:

- Линейный формат представления информации

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

- Отсутствие быстрого доступа к необходимой информации без необходимости просмотра всего курса
- Ограниченное взаимодействие с преподавателем и другими обучающимися
- Пассивный характер обучения, не всегда обеспечивающий достаточную практику для формирования устойчивых навыков
- Отсутствие возможности для обучающихся создавать собственные материалы или адаптировать существующие под свои потребности

Обучение по книгам представляет собой традиционный метод самообразования, который имеет следующие преимущества для образовательного процесса:

- Детальное и глубокое изложение материала
- Возможность изучения в собственном темпе
- Структурированное представление информации с оглавлением и указателями
- Доступность без необходимости подключения к интернету
- Возможность делать пометки и выделять важные моменты

Однако с точки зрения образовательного процесса обучение по книгам имеет следующие недостатки:

- Ограниченные возможности для интерактивного взаимодействия с материалом
- Отсутствие обратной связи
- Сложность представления взаимосвязей между различными компонентами технологии

- Невозможность быстрого обновления содержания при изменении технологий
- Отсутствие механизмов для отслеживания прогресса обучения

Обучение с использованием ИИ-чатов (ChatGPT, Claude, Bard) представляет собой новый подход к получению знаний, который имеет следующие преимущества для образовательного процесса:

- Персонализированные ответы на конкретные вопросы пользователя
- Возможность получения информации в диалоговом формате
- Адаптация уровня сложности объяснений в зависимости от запросов пользователя

- Доступность в любое время и возможность задавать уточняющие вопросы
- Возможность получения информации по широкому спектру тем

Однако с точки зрения образовательного процесса обучение с использованием ИИ-чатов имеет следующие недостатки:

- Отсутствие структурированного представления информации
- Возможность получения неточных или устаревших данных
- Отсутствие визуализации взаимосвязей между компонентами технологии
- Сложность сохранения и организации полученных знаний
- Отсутствие практических заданий и проверки навыков

2.1.2 Анализ существующих платформ для создания и обмена образовательными ресурсами

В данном разделе рассматриваются существующие платформы, которые позволяют создавать и обмениваться образовательными ресурсами, с акцентом на их влияние на

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

эффективность образовательного процесса.

GitHub и GitBook представляют собой платформы для создания и хранения документации, которые имеют следующие преимущества для образовательного процесса:

- Поддержка Markdown для форматирования текста
- Система версионирования для отслеживания изменений
- Возможность совместной работы над документацией
- Структурированное представление информации в виде иерархии страниц

Однако с точки зрения образовательного процесса эти платформы имеют следующие недостатки:

- Ориентация на техническую документацию, а не на образовательный контент
- Отсутствие специализированных инструментов для создания интерактивных образовательных материалов
- Отсутствие механизмов для отслеживания прогресса обучения
- Сложность визуализации взаимосвязей между различными компонентами изучаемой технологии

Notion представляет собой универсальный инструмент для создания документации, баз знаний и управления проектами, который имеет следующие преимущества для образовательного процесса:

- Гибкая структура документов с возможностью вложенности
- Поддержка различных типов контента (текст, таблицы, списки, изображения)
- Возможность создания связей между страницами
- Шаблоны для различных типов документов

Однако с точки зрения образовательного процесса Notion имеет следующие недостатки:

- Отсутствие визуального редактора для структурирования знаний
- Ограниченные возможности для отслеживания прогресса обучения
- Отсутствие специализированных инструментов для создания пошаговых инструкций
- Сложность организации материалов для быстрого доступа к конкретным инструкциям без необходимости просмотра всего документа

Miro представляет собой онлайн-доску для совместной работы, которая имеет следующие преимущества для образовательного процесса:

- Возможность создания связей между элементами
- Поддержка различных типов контента (текст, изображения, видео)
- Совместная работа в реальном времени
- Шаблоны для различных типов диаграмм

Однако с точки зрения образовательного процесса Miro имеет следующие недостатки:

- Отсутствие специализации на образовательном контенте
- Отсутствие встроенных средств для написания технической документации с поддержкой Markdown
- Отсутствие механизмов для отслеживания прогресса обучения
- Сложность организации большого объема информации на одной доске

Confluence представляет собой корпоративную вики-систему для создания и хранения документации, которая имеет следующие преимущества для образовательного процесса:

- Структурированное представление информации с иерархией страниц

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

- Поддержка различных типов контента и макросов
- Возможность совместной работы над документацией
- Интеграция с другими инструментами Atlassian

Однако с точки зрения образовательного процесса Confluence имеет следующие недостатки:

- Сложный интерфейс, требующий обучения
- Ориентация на корпоративное использование, а не на образовательные цели
- Отсутствие специализированных инструментов для создания интерактивных образовательных материалов
- Ограниченные возможности для визуализации взаимосвязей между различными компонентами изучаемой технологии

2.1.3 Выявленные проблемы существующих решений в контексте образовательного процесса

На основе проведенного анализа можно выделить следующие ключевые проблемы существующих решений, которые негативно влияют на эффективность образовательного процесса:

- Разрыв между структурированным представлением информации и возможностью быстрого доступа к конкретным инструкциям: традиционные образовательные форматы либо предоставляют хорошо структурированную, но трудно навигируемую информацию, либо обеспечивают быстрый доступ к отдельным фрагментам без общего контекста
- Отсутствие визуализации взаимосвязей между различными компонентами изучаемой технологии: большинство образовательных ресурсов представляют информацию в линейном формате, что затрудняет формирование системного понимания предмета и осознание взаимозависимостей между различными элементами
- Сложность баланса между детальными инструкциями и общим пониманием: существующие решения часто либо сосредоточены на детальном пошаговом инструкциях без объяснения общих принципов, либо предоставляют теоретические знания без практических руководств
- Ограниченные возможности для персонализации образовательного процесса: большинство платформ предлагают стандартизированный контент, не учитывающий индивидуальные потребности, уровень знаний и предпочтительный стиль обучения пользователя
- Недостаточная интерактивность образовательных материалов: многие ресурсы предлагают пассивное потребление информации без возможности активного взаимодействия с материалом, что снижает вовлеченность и эффективность обучения
- Отсутствие механизмов для отслеживания прогресса обучения: многие платформы не предоставляют инструментов для оценки уровня усвоения материала и отслеживания прогресса, что затрудняет планирование образовательного процесса
- Сложность создания и обмена пользовательскими образовательными ресурсами: существующие решения часто ограничивают возможности пользователей по созданию собственных материалов или их адаптации под конкретные потребности
- Отсутствие единой платформы, объединяющей преимущества различных подхо-

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

дов к обучению: каждое из существующих решений имеет свои сильные стороны, но ни одно не объединяет все необходимые функции для эффективного образовательного процесса

2.1.4 Ключевые аспекты эффективного образовательного процесса

На основе анализа существующих решений можно выделить следующие ключевые аспекты, которые способствуют эффективному образовательному процессу и должны быть учтены при разработке новой образовательной платформы:

- Структурированное представление информации: организация учебных материалов в логической последовательности с четкой иерархией и взаимосвязями между различными элементами способствует формированию целостного понимания предмета
 - Визуализация взаимосвязей: наглядное представление связей между различными компонентами изучаемой технологии помогает обучающимся формировать системное понимание и видеть общую картину
 - Быстрый доступ к конкретным инструкциям: возможность быстро найти и получить доступ к пошаговым руководствам по конкретным задачам повышает практическую ценность образовательных ресурсов
 - Интерактивность: активное взаимодействие с учебными материалами, возможность экспериментировать и получать обратную связь способствуют лучшему усвоению информации и формированию практических навыков
 - Персонализация: адаптация образовательного процесса к индивидуальным потребностям, уровню знаний и предпочтительному стилю обучения повышает эффективность и мотивацию
 - Отслеживание прогресса: возможность оценивать уровень усвоения материала и отслеживать прогресс обучения помогает планировать образовательный процесс и поддерживать мотивацию
 - Создание и обмен пользовательскими ресурсами: возможность для обучающихся создавать собственные материалы, адаптировать существующие и делиться ими с другими способствует активному обучению и формированию сообщества
 - Баланс между теорией и практикой: сочетание теоретических знаний с практическими руководствами обеспечивает как понимание общих принципов, так и формирование конкретных навыков
 - Актуальность информации: возможность оперативного обновления учебных материалов при изменении технологий обеспечивает релевантность получаемых знаний
 - Доступность и удобство использования: интуитивно понятный интерфейс, адаптивный дизайн и доступность с различных устройств снижают барьеры для обучения
- Учет этих аспектов при разработке новой образовательной платформы позволит создать эффективный инструмент для обучения, объединяющий преимущества различных существующих подходов и устраняющий их недостатки.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

2.2 Постановка задачи и описание функций разрабатываемой системы

2.2.1 Цель и задачи разработки

Основной целью дипломного проекта является разработка образовательной платформы для обмена образовательными ресурсами и их изучения, предоставляющей пользователям доступ к созданию и изучению структурированной обучающей информации по использованию различных средств, инструментов и технологий.

Для достижения поставленной цели необходимо решить следующие задачи:

- Разработать архитектуру веб-приложения с клиент-серверной архитектурой, обеспечивающую эффективное взаимодействие между клиентской и серверной частями
- Создать интуитивно понятный пользовательский интерфейс для работы с книгами инструкций и их элементами
- Реализовать функционал для создания, редактирования и визуализации страниц инструкций с использованием SVG-элементов
- Разработать систему создания и редактирования пошаговых инструкций с поддержкой Markdown
- Реализовать механизмы поиска и фильтрации образовательных ресурсов по тегам
- Создать систему отслеживания прогресса обучения пользователей
- Обеспечить возможность обмена образовательными материалами между пользователями
- Разработать подсистему регистрации и авторизации пользователей с использованием JWT
- Обеспечить безопасность и защиту данных пользователей
- Провести тестирование и оптимизацию разработанной системы

2.2.2 Описание концепции образовательной платформы

Разрабатываемая образовательная платформа представляет собой веб-приложение, основанное на концепции структурированного представления обучающих материалов в виде взаимосвязанных элементов. Ключевыми компонентами платформы являются "книги инструкций", "страницы инструкций" и "элементы инструкций".

"Книга инструкций" представляет собой набор связанных страниц инструкций, объединенных общей тематикой или направлением обучения. Пользователи могут создавать собственные книги инструкций или находить существующие по тегам, соответствующим их интересам и образовательным потребностям.

"Страница инструкций" представляет собой интерактивный холст, на котором располагаются SVG-изображения, представляющие отдельные "элементы инструкций". Эти элементы могут быть связаны между собой, визуализируя рекомендуемую последовательность изучения материала или взаимосвязи между различными компонентами изучаемой

технологии. Такой подход позволяет пользователю получить целостное представление о структуре изучаемой технологии или инструмента, понять взаимосвязи между отдельными компонентами и выбрать оптимальный путь обучения.

”Элемент инструкции”представляет собой компонент на странице инструкций, содержащий SVG-изображение и название инструкции. При нажатии на элемент инструкции открывается пошаговое описание в формате Markdown, содержащее детальное руководство по использованию определенного средства или функции. Инструкция может включать текстовые пояснения, изображения, фрагменты кода и другие элементы, необходимые для эффективного обучения.

Платформа предоставляет пользователям возможность не только изучать существующие обучающие материалы, но и создавать собственные. Пользователи могут разрабатывать новые книги инструкций, добавлять страницы и элементы инструкций, устанавливая связи между элементами, тем самым внося вклад в развитие образовательной базы знаний.

Система также включает механизмы отслеживания прогресса обучения, позволяющие пользователям контролировать свой прогресс в изучении различных инструкций. Пользователи могут отмечать изученные инструкции и добавлять книги в закладки для быстрого доступа, что помогает организовать процесс обучения более эффективно и целенаправленно.

В отличие от существующих альтернатив (обучение в университете, онлайн-курсы, книги, ИИ-чаты), разрабатываемая платформа объединяет преимущества различных подходов к обучению: структурированность университетского образования, гибкость онлайн-курсов, детальность книг и персонализацию ИИ-чатов, при этом устраняя их основные недостатки.

2.2.3 Функциональные требования к системе

На основе анализа существующих решений и поставленных задач были сформулированы следующие функциональные требования к разрабатываемой системе:

Подсистема регистрации и авторизации пользователей:

- Регистрация новых пользователей с использованием электронной почты и пароля
- Авторизация пользователей с использованием JWT-токенов
- Восстановление доступа к аккаунту через электронную почту
- Управление профилями пользователей (редактирование личной информации, изменение пароля)

• Разграничение прав доступа для различных категорий пользователей

Подсистема создания и редактирования образовательных ресурсов:

- Создание новых книг инструкций с указанием названия и тегов
- Добавление, редактирование и удаление страниц инструкций в книге
- Создание и размещение элементов инструкций на странице с использованием SVG-изображений
- Установление связей между элементами инструкций для визуализации зависимостей

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

- Создание и редактирование пошаговых инструкций в формате Markdown
- Добавление тегов к инструкциям для улучшения поиска
- Предварительный просмотр создаваемых материалов
- Возможность копирования и адаптации существующих книг инструкций

Подсистема поиска образовательных ресурсов:

- Поиск книг инструкций по названию и тегам
- Фильтрация результатов поиска по различным параметрам
- Сортировка результатов поиска по различным критериям (популярность, дата создания, рейтинг)
- Рекомендации похожих книг инструкций на основе истории просмотров пользователя

- Сохранение истории поисковых запросов пользователя
- Возможность добавления книг инструкций в закладки для быстрого доступа

Подсистема отслеживания прогресса обучения:

- Отметка изученных инструкций
- Визуализация прогресса обучения по различным книгам инструкций
- Формирование персональных рекомендаций на основе прогресса обучения
- Напоминания о необходимости продолжить обучение
- Генерация отчетов о прогрессе обучения

Подсистема взаимодействия пользователей:

- Возможность оценивания и комментирования книг инструкций
- Формирование рейтинга книг инструкций на основе оценок пользователей
- Возможность предложения исправлений и дополнений к существующим материалам
- Уведомления о новых комментариях и предложениях
- Система модерации пользовательского контента

2.2.4 Нефункциональные требования к системе

Помимо функциональных требований, к разрабатываемой системе предъявляются следующие нефункциональные требования:

Требования к производительности:

- Время отклика системы при выполнении основных операций не должно превышать 2 секунды
- Система должна поддерживать одновременную работу не менее 1000 пользователей
- Время загрузки страницы инструкций не должно превышать 3 секунды
- Время поиска по базе образовательных ресурсов не должно превышать 1 секунды

Требования к безопасности:

- Защита персональных данных пользователей в соответствии с законодательством
- Шифрование паролей пользователей с использованием современных алгоритмов
- Использование JWT для безопасной авторизации пользователей
- Защита от основных типов веб-уязвимостей (XSS, CSRF, SQL-инъекции и др.)

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

- Регулярное резервное копирование данных
- Журналирование действий пользователей для аудита безопасности

Требования к надежности:

- Доступность системы не менее 99.9% времени
- Автоматическое восстановление после сбоев
- Сохранение данных пользователей при непредвиденном завершении сеанса
- Регулярное тестирование системы на наличие ошибок

Требования к масштабируемости:

- Возможность горизонтального масштабирования системы при увеличении нагрузки

Модульная архитектура, позволяющая добавлять новые функциональные возможности

- Поддержка различных типов образовательного контента

Требования к пользовательскому интерфейсу:

- Интуитивно понятный интерфейс, не требующий специального обучения
- Адаптивный дизайн для работы на различных устройствах (компьютеры, планшеты, смартфоны)

- Соответствие современным стандартам веб-дизайна
- Поддержка различных браузеров (Chrome, Firefox, Safari, Edge)
- Доступность интерфейса для пользователей с ограниченными возможностями

2.2.5 Ожидаемые результаты разработки

В результате выполнения дипломного проекта ожидается создание полнофункциональной образовательной платформы для обмена образовательными ресурсами и их изучения, предоставляющей пользователям следующие возможности:

- Доступ к структурированным обучающим материалам по различным технологиям и инструментам
- Визуальное представление взаимосвязей между компонентами изучаемых технологий в виде интерактивных страниц инструкций
- Пошаговые инструкции по использованию различных средств с поддержкой Markdown
- Создание и редактирование собственных книг инструкций
- Отслеживание прогресса обучения и получение персональных рекомендаций
- Взаимодействие с другими пользователями через комментарии и оценки
- Поиск и фильтрация образовательных ресурсов по тегам

Разработанная система должна обеспечивать удобный и эффективный процесс обучения, позволяя пользователям быстро находить необходимую информацию и структурировать свои знания. Платформа также должна способствовать созданию сообщества пользователей, заинтересованных в обмене знаниями и совместном развитии образовательных ресурсов.

Ожидается, что разработанная система будет востребована как индивидуальными пользователями, стремящимися освоить новые технологии и инструменты, так и образовательными учреждениями и компаниями, заинтересованными в структурированном

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

представлении обучающих материалов для своих студентов или сотрудников.

2.3 Выбор средств реализации

2.3.1 Обоснование выбора технологий для разработки

На основе анализа требований к разрабатываемой системе и с учетом современных тенденций в веб-разработке были выбраны следующие технологии для реализации образовательной платформы. Выбор технологий осуществлялся с учетом необходимости создания масштабируемого, производительного и удобного в использовании веб-приложения с клиент-серверной архитектурой.

Для разработки клиентской части приложения выбрана библиотека React. Данный выбор обусловлен следующими факторами:

- Компонентный подход к разработке пользовательского интерфейса, который позволяет создавать переиспользуемые компоненты и упрощает поддержку кода
- Виртуальный DOM, обеспечивающий высокую производительность при обновлении интерфейса, что особенно важно при работе с интерактивными элементами страниц инструкций
- Большое сообщество разработчиков и обширная экосистема готовых решений, библиотек и инструментов
- Декларативный подход к разработке, упрощающий создание сложных пользовательских интерфейсов
- Возможность использования современных возможностей JavaScript (ES6+) для более эффективной разработки
- Хорошая интеграция с библиотеками для работы с SVG, что критически важно для реализации интерактивных страниц инструкций
- Поддержка однонаправленного потока данных, что упрощает отладку и тестирование приложения

Для разработки серверной части приложения выбран фреймворк Express.js на платформе Node.js. Данный выбор обусловлен следующими преимуществами:

- Минималистичный и гибкий фреймворк, позволяющий создавать масштабируемые веб-приложения
- Использование JavaScript как на клиенте, так и на сервере, что упрощает разработку и позволяет переиспользовать код
- Асинхронная модель обработки запросов, обеспечивающая высокую производительность при работе с большим количеством одновременных соединений
- Богатая экосистема middleware-компонентов, упрощающих реализацию различных функций (аутентификация, логирование, обработка ошибок и т.д.)
- Простая интеграция с различными базами данных, включая PostgreSQL
- Поддержка REST API, необходимого для взаимодействия клиентской и серверной частей приложения
- Возможность горизонтального масштабирования для обеспечения высокой доступности системы

В качестве системы управления базами данных выбрана PostgreSQL. Данный выбор обусловлен следующими факторами:

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

- Надежная и проверенная временем реляционная СУБД с открытым исходным кодом
 - Поддержка сложных запросов и транзакций, необходимых для обеспечения целостности данных
 - Возможность работы с JSON-данными, что упрощает хранение и обработку структурированных данных, таких как содержимое страниц инструкций
 - Высокая производительность при работе с большими объемами данных
 - Расширяемость и возможность добавления пользовательских функций и типов данных
 - Поддержка полнотекстового поиска, что важно для реализации функционала поиска по образовательным ресурсам
 - Хорошая интеграция с Node.js через различные ORM и драйверы
- Для форматирования и отображения пошаговых инструкций выбран язык разметки Markdown. Данный выбор обусловлен следующими преимуществами:
- Простой и интуитивно понятный синтаксис, не требующий специальных знаний от пользователей
 - Возможность форматирования текста, создания списков, таблиц, вставки изображений и фрагментов кода
 - Широкая поддержка в веб-приложениях и наличие готовых библиотек для преобразования Markdown в HTML
 - Компактность и читаемость исходного текста даже без преобразования в HTML
 - Возможность расширения базового синтаксиса для поддержки дополнительных элементов форматирования
 - Хорошая интеграция с системами контроля версий, что упрощает отслеживание изменений в инструкциях
- Для реализации аутентификации и авторизации пользователей выбрана технология JSON Web Tokens (JWT). Данный выбор обусловлен следующими факторами:
- Безопасный механизм передачи информации между клиентом и сервером в виде JSON-объекта
 - Возможность включения дополнительной информации о пользователе в токен, что уменьшает количество запросов к базе данных
 - Отсутствие необходимости хранить состояние сессии на сервере, что упрощает масштабирование системы
 - Поддержка механизмов цифровой подписи для обеспечения целостности данных
 - Возможность установки срока действия токена, что повышает безопасность системы
 - Широкая поддержка в различных языках программирования и фреймворках
 - Простота интеграции с Express.js через middleware-компоненты

2.3.2 Дополнительные библиотеки и инструменты

Помимо основных технологий, для реализации образовательной платформы планируется использование следующих дополнительных библиотек и инструментов:

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

Библиотеки для клиентской части:

- React Router — библиотека для управления маршрутизацией в React-приложении, обеспечивающая навигацию между различными компонентами без перезагрузки страницы
- Redux — библиотека для управления состоянием приложения, обеспечивающая предсказуемое поведение и упрощающая отладку
- Axios — библиотека для выполнения HTTP-запросов к серверу, обеспечивающая удобный интерфейс для работы с API
- React-Markdown — библиотека для преобразования Markdown в React-компоненты, необходимая для отображения пошаговых инструкций
- React-SVG — библиотека для работы с SVG в React, обеспечивающая возможность создания и редактирования интерактивных элементов на страницах инструкций
- Material-UI или Ant Design — библиотека готовых компонентов пользовательского интерфейса, обеспечивающая единый стиль оформления и ускоряющая разработку
- React DnD (Drag and Drop) — библиотека для реализации функционала перетаскивания элементов, необходимая для удобного редактирования страниц инструкций
- Jest и React Testing Library — инструменты для тестирования React-компонентов, обеспечивающие качество и надежность клиентской части приложения

Библиотеки для серверной части:

- Sequelize — ORM для работы с PostgreSQL, упрощающая взаимодействие с базой данных и обеспечивающая абстракцию от SQL-запросов
- Passport.js — middleware для аутентификации в Node.js, поддерживающий различные стратегии аутентификации, включая JWT
- Bcrypt — библиотека для хеширования паролей, обеспечивающая безопасное хранение учетных данных пользователей
- Multer — middleware для обработки multipart/form-data, необходимый для загрузки изображений и других файлов
- Winston или Morgan — библиотеки для логирования событий на сервере, обеспечивающие отслеживание работы приложения и диагностику ошибок
- Joi или Express-validator — библиотеки для валидации данных, обеспечивающие проверку входных данных на соответствие требованиям
- Nodemailer — библиотека для отправки электронных писем, необходимая для функционала восстановления пароля и уведомлений
- Socket.io — библиотека для реализации двунаправленной связи между клиентом и сервером в реальном времени, необходимая для функционала совместной работы над образовательными ресурсами

Инструменты разработки и развертывания:

- Webpack — инструмент для сборки и оптимизации клиентской части приложения, обеспечивающий минификацию кода и управление зависимостями
- Babel — транспилятор JavaScript, позволяющий использовать современные возможности языка с сохранением совместимости с различными браузерами
- ESLint — инструмент для статического анализа кода, обеспечивающий соблюдение стандартов кодирования и выявление потенциальных ошибок
- Prettier — инструмент для автоматического форматирования кода, обеспечивающий единый стиль оформления
- Git — система контроля версий, обеспечивающая отслеживание изменений в коде и совместную работу над проектом

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

- Docker — платформа для контейнеризации приложений, обеспечивающая единое изображение среды разработки и развертывания
- CI/CD (например, GitHub Actions или GitLab CI) — инструменты для непрерывной интеграции и доставки, автоматизирующие процессы тестирования и развертывания приложения

2.3.3 Архитектурные решения и паттерны проектирования

Для обеспечения масштабируемости, поддерживаемости и гибкости разрабатываемой системы планируется использование следующих архитектурных решений и паттернов проектирования:

Клиентская часть:

- Компонентная архитектура — разделение пользовательского интерфейса на независимые, переиспользуемые компоненты, каждый из которых отвечает за определенную функциональность
- Flux-архитектура (через Redux) — однонаправленный поток данных, обеспечивающий предсказуемое изменение состояния приложения и упрощающий отладку
- Container/Presentational паттерн — разделение компонентов на контейнеры, отвечающие за логику и получение данных, и презентационные компоненты, отвечающие только за отображение
- Render Props и Higher-Order Components — паттерны для переиспользования логики между компонентами
- Lazy Loading — отложенная загрузка компонентов и ресурсов для улучшения производительности приложения
- Мемоизация — оптимизация производительности путем кеширования результатов вычислений

Серверная часть:

- MVC (Model-View-Controller) — разделение серверной логики на модели (работа с данными), представления (формирование ответов) и контроллеры (обработка запросов)
- Repository Pattern — абстракция доступа к данным, скрывающая детали работы с базой данных и упрощающая тестирование
- Middleware Pattern — цепочка обработчиков запросов, позволяющая модульно добавлять функциональность (аутентификация, логирование, обработка ошибок)
- Dependency Injection — внедрение зависимостей для уменьшения связанности компонентов и упрощения тестирования
- Service Layer — выделение бизнес-логики в отдельный слой, независимый от контроллеров и моделей
- Error Handling Middleware — централизованная обработка ошибок для обеспечения единообразного поведения приложения при возникновении исключений

Общие архитектурные решения:

- RESTful API — архитектурный стиль взаимодействия между клиентом и сервером, основанный на принципах REST
- JWT-based Authentication — механизм аутентификации на основе токенов, не тре-

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

бующий хранения состояния сессии на сервере

- **Microservices** (при необходимости масштабирования) — разделение приложения на независимые сервисы, каждый из которых отвечает за определенную функциональность

- **CQRS (Command Query Responsibility Segregation)** — разделение операций чтения и записи для оптимизации производительности и масштабируемости

- **Event-driven Architecture** — использование событий для обеспечения слабой связанности компонентов системы

- **Caching Strategies** — стратегии кеширования данных для улучшения производительности и уменьшения нагрузки на базу данных

Выбранные архитектурные решения и паттерны проектирования обеспечат создание гибкой, масштабируемой и поддерживаемой системы, способной эффективно решать поставленные задачи и адаптироваться к изменяющимся требованиям.

2.3.4 Обоснование выбора структуры базы данных

Для хранения данных образовательной платформы выбрана реляционная модель базы данных, реализуемая с помощью PostgreSQL. Данный выбор обусловлен необходимостью обеспечения целостности данных, поддержки сложных связей между сущностями и возможности выполнения комплексных запросов.

Основными сущностями базы данных являются:

- **Пользователь (User)** — хранит информацию о пользователях системы, включая учетные данные, контактную информацию и настройки профиля. Атрибуты: id, name, email, password, learnedInstructionsIds (массив идентификаторов изученных инструкций), bookmarkedBooksIds (массив идентификаторов книг в закладках).

- **Книга инструкций (Book)** — представляет собой набор связанных страниц инструкций, объединенных общей тематикой. Атрибуты: id, name, ownerId (идентификатор пользователя-владельца), а также дополнительные атрибуты, такие как description (описание книги), createdAt (дата создания), updatedAt (дата обновления), tags (теги для поиска).

- **Страница инструкций (Page)** — содержит информацию о странице с инструкциями, включая ее структуру и содержимое. Атрибуты: id, bookId (идентификатор книги, к которой относится страница), title (название страницы), instructions (массив или JSON-структура, описывающая элементы инструкций на странице), order (порядковый номер страницы в книге).

- **Инструкция (Instruction)** — содержит детальную информацию об отдельной инструкции, включая ее содержимое и метаданные. Атрибуты: id, name (название инструкции), position (координаты расположения на странице), dependencies (массив идентификаторов инструкций, от которых зависит данная инструкция), markdownText (текст инструкции в формате Markdown), tags (теги для поиска).

Дополнительные сущности, необходимые для реализации всех требуемых функций системы:

- **Комментарий (Comment)** — хранит комментарии пользователей к книгам и инструкциям. Атрибуты: id, userId (идентификатор автора комментария), targetType (тип объ-

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

екта комментирования: книга или инструкция), targetId (идентификатор объекта комментирования), text (текст комментария), createdAt (дата создания).

- Оценка (Rating) — хранит оценки пользователей для книг инструкций. Атрибуты: id, userId (идентификатор пользователя), bookId (идентификатор книги), value (значение оценки), createdAt (дата создания).

- Тег (Tag) — хранит информацию о тегах, используемых для категоризации и поиска образовательных ресурсов. Атрибуты: id, name (название тега), description (описание тега).

- Прогресс обучения (LearningProgress) — отслеживает прогресс пользователей в изучении различных книг и инструкций. Атрибуты: id, userId (идентификатор пользователя), instructionId (идентификатор инструкции), status (статус изучения), completedAt (дата завершения изучения).

Для обеспечения эффективной работы с данными и оптимизации запросов планируется использование следующих подходов:

- Индексирование ключевых полей для ускорения поиска и сортировки
- Использование внешних ключей для обеспечения ссылочной целостности данных
- Нормализация данных для минимизации избыточности и аномалий обновления
- Использование JSON-типов данных для хранения структурированной информации, такой как позиции элементов на странице инструкций
- Применение транзакций для обеспечения атомарности операций, затрагивающих несколько таблиц
- Использование полнотекстового поиска PostgreSQL для эффективного поиска по содержанию инструкций

Выбранная структура базы данных обеспечивает гибкость, масштабируемость и производительность, необходимые для реализации всех функциональных требований к образовательной платформе, а также позволяет легко расширять систему новыми возможностями в будущем.

2.3.5 Выбор инструментов для обеспечения безопасности

Безопасность является критически важным аспектом разрабатываемой образовательной платформы, особенно учитывая необходимость защиты персональных данных пользователей и образовательного контента. Для обеспечения безопасности системы выбраны следующие инструменты и подходы:

Аутентификация и авторизация:

- JSON Web Tokens (JWT) — для безопасной передачи информации о пользователе между клиентом и сервером без необходимости хранения состояния сессии на сервере
- bcrypt — для хеширования паролей пользователей с использованием соли и алгоритма, устойчивого к атакам перебором
- Passport.js — для реализации различных стратегий аутентификации и управления процессом авторизации
- HTTPS — для шифрования данных, передаваемых между клиентом и сервером,

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

с использованием TLS/SSL сертификатов

- CORS (Cross-Origin Resource Sharing) — для контроля доступа к ресурсам сервера с других доменов

Защита от распространенных уязвимостей:

- Helmet.js — для установки HTTP-заголовков безопасности, защищающих от различных типов атак (XSS, clickjacking, MIME-sniffing и др.)
- Express-validator или Joi — для валидации и санитизации входных данных, предотвращающих инъекции и другие атаки, связанные с некорректными данными
- CSRF-токены — для защиты от Cross-Site Request Forgery атак
- Content Security Policy (CSP) — для ограничения источников загружаемых ресурсов и предотвращения XSS-атак
- Rate Limiting — для защиты от DDoS-атак и брутфорс-атак на систему аутентификации

Безопасность данных:

- Параметризованные запросы — для предотвращения SQL-инъекций при работе с базой данных
- Шифрование чувствительных данных в базе данных — для защиты персональной информации пользователей
- Регулярное резервное копирование данных — для обеспечения возможности восстановления в случае потери или повреждения данных
- Управление доступом на уровне базы данных — для ограничения прав доступа к данным в соответствии с ролями пользователей

Мониторинг и аудит безопасности:

- Winston или Morgan — для логирования событий безопасности и действий пользователей
- Sentry или аналогичные сервисы — для отслеживания ошибок и потенциальных проблем безопасности в реальном времени
- Регулярное сканирование уязвимостей — для выявления и устранения потенциальных проблем безопасности
- Dependency scanning — для проверки используемых библиотек на наличие известных уязвимостей

Выбранные инструменты и подходы к обеспечению безопасности позволят создать надежную и защищенную образовательную платформу, соответствующую современным стандартам безопасности веб-приложений и требованиям законодательства в области защиты персональных данных.

2.3.6 Обоснование выбора инструментов для разработки пользовательского интерфейса

Пользовательский интерфейс является критически важным компонентом образовательной платформы, так как от его удобства и интуитивности напрямую зависит эффективность обучения и удовлетворенность пользователей. Для разработки пользовательского интерфейса выбраны следующие инструменты и подходы:

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

Основные инструменты для создания интерфейса:

- React — библиотека для создания пользовательских интерфейсов, выбранная в качестве основы клиентской части приложения
- Material-UI или Ant Design — библиотека готовых компонентов, обеспечивающая единый стиль оформления и соответствие современным стандартам веб-дизайна
- CSS Modules или Styled Components — для стилизации компонентов с изоляцией стилей и предотвращением конфликтов имен классов
- Flexbox и CSS Grid — для создания адаптивных и гибких макетов страниц
- SVG — для создания и отображения векторной графики на страницах инструкций

Инструменты для работы с SVG и интерактивными элементами:

- React-SVG — библиотека для работы с SVG в React, обеспечивающая возможность создания и редактирования интерактивных элементов
- D3.js — библиотека для создания сложных визуализаций и интерактивных графических элементов
- React DnD (Drag and Drop) — библиотека для реализации функционала перетаскивания элементов, необходимая для удобного редактирования страниц инструкций
- React-Konva — библиотека для работы с Canvas в React, предоставляющая альтернативный способ создания интерактивных графических элементов

Инструменты для работы с Markdown:

- React-Markdown — библиотека для преобразования Markdown в React-компоненты, необходимая для отображения пошаговых инструкций
- Remark и Rehype — плагины для расширения возможностей Markdown, такие как поддержка синтаксиса для выделения кода, таблиц, математических формул и т.д.
- CodeMirror или Monaco Editor — редакторы кода для создания и редактирования Markdown-контента с подсветкой синтаксиса и предварительным просмотром

Инструменты для улучшения пользовательского опыта:

- React Router — библиотека для управления маршрутизацией в React-приложении, обеспечивающая навигацию между различными компонентами без перезагрузки страницы
- React Query или SWR — библиотеки для управления состоянием данных, получаемых с сервера, обеспечивающие кеширование, автоматическое обновление и обработку ошибок

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

ЗАКЛЮЧЕНИЕ

В ходе прохождения преддипломной практики на предприятии ОАО «Савушкин продукт» в бюро перспективных разработок были успешно выполнены все поставленные задачи и получен ценный практический опыт.

В процессе практики были изучены:

1. Организационная и функциональная структура предприятия
2. Должностные инструкции сотрудников бюро перспективных разработок
3. Основные задачи и направления деятельности подразделения
4. Технические средства и программное обеспечение, используемые на предприятии

В рамках практической работы были выполнены следующие задачи:

1. Сборка и запуск основных проектов предприятия: ptusa_main, EasyEPLAnner и SCADA
2. Изучение архитектуры и принципов работы программных комплексов
3. Доработка проекта SCADA в части автозаполнения полей при импорте тегов OPC UA Browser
4. Улучшение информативности сообщений об ошибках коммуникации в проекте ptusa_main
5. Разработка клиентской части веб-версии SCADA на React

В процессе выполнения индивидуального задания был проведен анализ существующих образовательных платформ, сформулированы требования к разрабатываемой системе и выбраны оптимальные технологии для ее реализации. Разработанная концепция образовательной платформы предполагает создание интерактивной системы для структурированного представления обучающих материалов с возможностью визуализации взаимосвязей между различными компонентами.

Практика позволила получить ценный опыт работы с современными технологиями веб-разработки, промышленными системами автоматизации и методами организации командной работы над крупными проектами. Полученные знания и навыки будут использованы при дальнейшей разработке дипломного проекта.

						Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК СОКРАЩЕНИЙ

АСУТП	–	автоматизированная система управления технологическими процессами.
БПР	–	бюро перспективных разработок.
ОАО	–	открытое акционерное общество.
ПО	–	программное обеспечение.
СУБД	–	система управления базами данных.
API	–	(Application Programming Interface) программный интерфейс приложения.
CSS	–	(Cascading Style Sheets) каскадные таблицы стилей.
HTML	–	(HyperText Markup Language) язык гипертекстовой разметки.
HTTP	–	(HyperText Transfer Protocol) протокол передачи гипертекста.
JWT	–	(JSON Web Token) веб-токен в формате JSON.
OPC UA	–	(OPC Unified Architecture) унифицированная архитектура OPC.
REST	–	(Representational State Transfer) передача состояния представления.
SCADA	–	(Supervisory Control And Data Acquisition) диспетчерское управление и сбор данных.
SQL	–	(Structured Query Language) язык структурированных запросов.
SVG	–	(Scalable Vector Graphics) масштабируемая векторная графика.
XML	–	(eXtensible Markup Language) расширяемый язык разметки.

СПИСОК ЛИТЕРАТУРЫ

1. React – A JavaScript library for building user interfaces [Электронный ресурс]. – Режим доступа: <https://reactjs.org/>. – Дата доступа: 25.05.2024.
2. Node.js Documentation [Электронный ресурс]. – Режим доступа: <https://nodejs.org/en/docs/>. – Дата доступа: 25.05.2024.
3. Express - Node.js web application framework [Электронный ресурс]. – Режим доступа: <https://expressjs.com/>. – Дата доступа: 25.05.2024.
4. PostgreSQL Documentation [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/>. – Дата доступа: 25.05.2024.
5. Material-UI: A popular React UI framework [Электронный ресурс]. – Режим доступа: <https://material-ui.com/>. – Дата доступа: 25.05.2024.
6. JWT.IO - JSON Web Tokens [Электронный ресурс]. – Режим доступа: <https://jwt.io/>. – Дата доступа: 25.05.2024.
7. Markdown Guide [Электронный ресурс]. – Режим доступа: <https://www.markdownguide.org/>. – Дата доступа: 25.05.2024.
8. Redux - A Predictable State Container for JS Apps [Электронный ресурс]. – Режим доступа: <https://redux.js.org/>. – Дата доступа: 25.05.2024.
9. Webpack Documentation [Электронный ресурс]. – Режим доступа: <https://webpack.js.org/>. – Дата доступа: 25.05.2024.
10. TypeScript Documentation [Электронный ресурс]. – Режим доступа: <https://www.typescriptlang.org/docs/>. – Дата доступа: 25.05.2024.
11. Phoenix Contact – Контроллер [Электронный ресурс]. – Режим доступа: <https://www.phoenixcontact.com/ru-ru/otrasli-promyshlennosti/plcnnext-technology>. – Дата доступа: 25.05.2024.
12. SCADA системы. Основы проектирования и разработки [Электронный ресурс]. – Режим доступа: <https://www.scada.ru/ru/documentation/>. – Дата доступа: 25.05.2024.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		32