СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc208413477)

[1. ОБЩАЯ ЧАСТЬ 6](#_Toc208413478)

[1.1. Цель разработки 6](#_Toc208413479)

[1.2. Средства разработки 6](#_Toc208413480)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 8](#_Toc208413481)

[2.1. Постановка задачи 8](#_Toc208413482)

[2.1.1. Входные данные предметной области 8](#_Toc208413483)

[2.1.2. Выходные данные предметной области 8](#_Toc208413484)

[2.1.3. Требования к проекту 9](#_Toc208413485)

[2.2. Внешняя спецификация 10](#_Toc208413486)

[2.2.1. Описание задачи 10](#_Toc208413487)

[2.2.2. Входные и выходные данные 10](#_Toc208413488)

[2.2.3. Тесты 11](#_Toc208413489)

[2.2.4. Контроль целостности данных 11](#_Toc208413490)

[2.3. Проектирование 11](#_Toc208413491)

[2.3.1. Схема архитектуры приложения 11](#_Toc208413492)

[2.3.2. Логическая модель данных 12](#_Toc208413493)

[2.3.3. Физическая модель данных 13](#_Toc208413494)

[2.3.4. Диаграмма классов 13](#_Toc208413495)

[2.3.5. Схема тестирования 14](#_Toc208413496)

[2.3.6. Структурная схема 14](#_Toc208413497)

[2.3.7. Функциональная схема 14](#_Toc208413498)

[2.3.8. Схема пользовательского интерфейса 15](#_Toc208413499)

[2.4. Результат работы программы 15](#_Toc208413500)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 18](#_Toc208413501)

[3.1. Инструментальные средства 18](#_Toc208413502)

[3.2. Отладка программы 18](#_Toc208413503)

[3.3. Характеристики программы 18](#_Toc208413504)

[ЗАКЛЮЧЕНИЕ 20](#_Toc208413505)

[СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ 22](#_Toc208413506)

ПРИЛОЖЕНИЕ А. Техническое задание

ПРИЛОЖЕНИЕ Б. Сценарий и результаты тестовых испытаний

ПРИЛОЖЕНИЕ В. Руководство пользователя

ПРИЛОЖЕНИЕ Г. Текст программы

ПРИЛОЖЕНИЕ Д. Диаграмма классов

# ВВЕДЕНИЕ

Современный рынок искусственного интеллекта и нейронных сетей активно развивается, предлагая пользователям всё более разнообразные инструменты и сервисы. Среди множества ИИ-решений особой популярностью пользуются системы управления подписками, которые сочетают в себе удобство управления, персонализацию и высокий уровень безопасности. Веб-приложения, благодаря своей доступности и постоянно растущей производительности, становятся идеальной платформой для таких систем, позволяя пользователям эффективно управлять подписками на нейросети в любое время и в любом месте.

Система «Neuro Store» представляет собой веб-приложение для управления подписками на различные нейросети, которое разработано специально для пользователей, заинтересованных в доступе к ИИ-сервисам. Проект ориентирован на широкую аудиторию, включая как опытных пользователей, предпочитающих сложные функции управления подписками, так и новичков, которые только начинают осваивать подобные инструменты.

Актуальность разработки обусловлена растущим спросом на качественные веб-приложения для управления подписками на ИИ-сервисы с интуитивно понятным интерфейсом и широкими возможностями настройки. Современные пользователи ожидают не только удобного управления подписками, но и интеграции с платежными системами, а также адаптации под индивидуальные потребности.

Разработка «Neuro Store» направлена на создание качественного инструмента для управления подписками на нейросети, который сможет занять достойное место на рынке веб-приложений. Внедрение современных технологий, продуманный дизайн интерфейса и акцент на пользовательский опыт позволят проекту привлечь внимание аудитории и обеспечат его успешное использование.

В данной работе рассматриваются ключевые аспекты разработки системы, включая проектирование архитектуры, функциональные возможности (такие как управление пользователями, продуктами и подписками), интеграцию с базой данных PostgreSQL и тестирование.

# ОБЩАЯ ЧАСТЬ

## Цель разработки

Целью разработки является предоставление функционала веб-приложения, который включает в себя различные задачи, такие как: создание и управление пользователями, управление каталогом нейросетей, создание и управление подписками, обработка платежей, администрирование системы, а также интеграция с базой данных для хранения и синхронизации данных. Система «Neuro Store» стремится помочь пользователям эффективно управлять подписками на нейросети, повышать продуктивность использования ИИ-сервисов и адаптировать инструмент под индивидуальные потребности.

## Средства разработки

Для разработки ПО использовались следующие технические средства, представлены в Таблице 1.

Таблица 1 – Технические средства для разработки.

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Разрешение экрана: | 3024 x 1964 |
| 2 | Линейка процессора: | Apple M4 |
| 3 | Количество ядер процессора: | 10 |
| 4 | Оперативная память: | 16 ГБ |
| 5 | Тип видеокарты: | Встроенная |

В Таблице 2 представлены минимальные и рекомендованные технические средства для комфортного использования программного продукта.

Таблица 2 – Технические средства для использования приложения

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Минимальные системные требования |  |  |
| 1 | Разрешение экрана: | 1920 x 1080 |
| 2 | Количество ядер процессора: | 4 ядра |
| 3 | Оперативная память: | 8 ГБ |
| 4 | Встроенная память: | 10 ГБ |
| 5 | Операционная система: | Windows 10, macOS 12, Ubuntu 20.04 |
| 6 | Браузер: | Chrome 90+, Firefox 88+, Safari 14+ |

Таблица 3 – Рекомендованные требования

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Рекомендованные системные требования |  |  |
| 1 | Разрешение экрана: | 2560 x 1440 |
| 2 | Количество ядер процессора: | 8 ядер |
| 3 | Оперативная память: | 16 ГБ |
| 4 | Встроенная память: | 50 ГБ |
| 5 | Операционная система: | Windows 11, macOS 13, Ubuntu 22.04 |
| 6 | Браузер: | Chrome 100+, Firefox 95+, Safari 15+ |

Для разработки использовались следующие программные средства, представленные в Таблице 4.

Таблица 4 – Программные средства

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средств | Наименование средства | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | Среда разработки мобильного приложения на SwiftUI | Xcode 16.4 | Написание кода |
| 2 | Средство симуляции мобильной среды iOS | Xcode Simulator | Симулятор устройства |

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Разработать веб-приложение «Neuro Store» для управления подписками на нейросети с использованием современных технологий веб-разработки.

## Входные данные предметной области

Входные данные приложения включают:

Ввод email, username и пароля для регистрации нового пользователя

Ввод email и пароля для авторизации в системе

Ввод названия, описания и цены для создания продукта (нейросети)

Ввод названия, описания, цены и длительности для создания тарифного плана

Ввод product\_id и plan\_id для создания подписки

Ввод суммы и способа оплаты для обработки платежа

Ввод роли для назначения пользователю через административную панель

## Выходные данные предметной области

Выходные данные приложения включают:

Список активных подписок пользователя после авторизации

Каталог доступных нейросетей с описанием и ценами

Детальная информация о продукте с доступными тарифными планами

Статистика использования и аналитика для администраторов

История платежей и транзакций пользователя

Уведомления о статусе подписки (активна, истекла, отменена)

Отчеты по доходам и пользователям для административной панели

## Требования к проекту

Разработка системы управления подписками с помощью современного веб-интерфейса, включающего REST API для backend и React-компоненты для frontend, а также интеграцию с базой данных PostgreSQL и системой кэширования Redis. Создание пользовательских интерфейсов и страниц приложения (например, каталог продуктов, управление подписками и административная панель).

Система «Neuro Store» предоставляет следующие функции:

* Создание, редактирование и удаление пользователей с полным набором параметров: email, username, пароль, роли, статус активности
* Управление каталогом нейросетей с описанием, ценами, категориями и статусом активности
* Создание и управление тарифными планами с различной длительностью и стоимостью
* Покупка подписок на нейросети с автоматическим расчетом стоимости
* Управление активными подписками с возможностью продления и отмены
* Обработка платежей с интеграцией внешних платежных систем
* Система ролей и разграничения доступа (admin, moderator, user, viewer)
* Административная панель для управления пользователями и аналитики
* Аудит действий пользователей с логированием всех операций
* API для интеграции с внешними системами
* Автоматическое резервное копирование данных
* Мониторинг производительности и логирование ошибок

## Внешняя спецификация

## Описание задачи

Разработать веб-приложение «Neuro Store» для управления подписками на нейросети. Программа должна эксплуатироваться в веб-браузере с поддержкой современных стандартов HTML5, CSS3 и JavaScript. Конечными пользователями приложения будут люди, стремящиеся эффективно управлять подписками на ИИ-сервисы, организовать доступ к нейросетям, а также те, кто заинтересован в удобных и функциональных инструментах для управления подписками и взаимодействия с администраторами системы.

## Входные и выходные данные

Входные данные веб-приложения «Neuro Store» представлены в Таблице 5.

Таблица – 5 Таблица входных данных

| № | Название | Тип | Ограничения | Краткое описание |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | UserEmail | string | [a-z, A-Z, 0-9, @, .] {5, 100} | Email пользователя для регистрации и входа |
| 2 | Username | string | [a-z, A-Z, 0-9, \_] {3, 50} | Имя пользователя |
| 3 | Password | string | [a-z, A-Z, 0-9, !@#$%^&\*] {8, 128} | Пароль пользователя |
| 4 | ProductName | string | [a-z, A-Z, 0-9, -] {1, 200} | Название нейросети |
| 5 | ProductDescription | text | [a-z, A-Z, 0-9, .,!?] {1, 1000} | Описание продукта |
| 6 | ProductPrice | decimal | [0-9, .] {1, 10} | Цена продукта |
| 7 | PlanName | string | [a-z, A-Z, 0-9, -] {1, 100} | Название тарифного плана |
| 8 | PlanDuration | integer | [0-9] {1, 4} | Длительность плана в днях |
| 9 | SubscriptionStatus | enum | [active, expired, cancelled] | Статус подписки |
| 10 | PaymentAmount | decimal | [0-9, .] {1, 10} | Сумма платежа |
| 11 | UserRole | enum | [admin, moderator, user, viewer] | Роль пользователя |

Выходные данные мобильного приложения «Planer\_1» представлены в Таблице 6.

Таблица – 6 Таблица выходных данных.

| Имя | Тип | Описание |
| --- | --- | --- |
| 1 | 2 | 3 |
| UserList | array | Список пользователей с ролями и статусами |
| ProductCatalog | array | Каталог нейросетей с ценами и описаниями |
| SubscriptionList | array | Список подписок пользователя |
| PaymentHistory | array | История платежей и транзакций |
| AnalyticsData | object | Статистика и аналитика для администраторов |
| UserProfile | object | Профиль пользователя с подписками |
| SystemStats | object | Статистика системы (пользователи, доходы) |

## Тесты

Сценарий и результаты тестовых испытаний приведены в Приложении Б.

## Контроль целостности данных

Система обеспечивает контроль целостности данных через:

Ограничения внешних ключей в базе данных PostgreSQL

Валидацию данных на уровне API с использованием Pydantic схем

Проверку прав доступа через систему ролей RBAC

Аудит всех изменений данных через таблицу audit\_log

## Проектирование

## Схема архитектуры приложения

Веб-приложение «Neuro Store» построено по принципу многоуровневой архитектуры и развернуто в контейнерах Docker. Данные хранятся в реляционной базе данных PostgreSQL, а кэширование осуществляется через Redis. Система обеспечивает высокую доступность и масштабируемость.

На Рисунке 1 представлена архитектурная схема веб-приложения, где «Клиент» обозначает пользователя, взаимодействующего с приложением через браузер, а «Сервер» представляет серверную часть, включающую API, базу данных и систему кэширования.

Изображение выглядит как текст, Шрифт, диаграмма, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – Архитектурная схема

## Логическая модель данных

Логическая модель данных представлена в виде ER-диаграммы, включающей 11 основных сущностей:

users - пользователи системы

roles - роли пользователей

user\_roles - связь пользователей и ролей (M:M)

products - каталог нейросетей

plans - тарифные планы

product\_plans - связь продуктов и планов (M:M)

subscriptions - подписки пользователей

orders – заказы

payments – платежи

usage\_events - события использования

audit\_log - журнал аудита

Изображение выглядит как текст, диаграмма, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 - ER-диаграмма

## Физическая модель данных

Физическая модель данных реализована в PostgreSQL 15 с использованием следующих типов данных:

BIGINT для первичных ключей

VARCHAR с ограничениями длины для текстовых полей

DECIMAL для денежных сумм

TIMESTAMP для дат и времени

JSONB для хранения структурированных данных

BOOLEAN для флагов состояния

## Диаграмма классов

Диаграмма классов приведена в Приложении Г «Диаграмма классов».

## Схема тестирования

Схема тестирования приведена в Приложении Б «Сценарий и результаты тестовых испытаний».

## Структурная схема

На Рисунке 3 представлена структурная схема веб-приложения «Neuro Store», показывающая основные компоненты системы и их взаимодействие.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 - Структураная схема

## Функциональная схема

На Рисунке 3 представлена функциональная схема для веб-приложения «Neuro Store», иллюстрирующая основные бизнес-процессы системы.

Пользователь → Регистрация/Вход → Выбор продукта →   
Выбор плана → Создание подписки → Платеж →   
Активация подписки → Управление подпиской

Рисунок 4 – Функциональная схема

## Схема пользовательского интерфейса

На Рисунке 5 представлена схема пользовательского интерфейса веб-приложения «Neuro Store», которая отражает основные страницы и элементы управления, обеспечивающие интуитивное взаимодействие пользователя с функционалом приложения.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 - схема пользовательского интерфейса

## Результат работы программы

В результате выполнения поставленной задачи было разработано веб-приложение «Neuro Store» для управления подписками на нейросети с использованием современных технологий веб-разработки.

На рисунках ниже представлен результат работы программы. Более подробно результат работы описан в приложении В «Руководство пользователя».

На рисунке 6 представлено, как выглядит главная страница приложения с каталогом нейросетей.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 - Главная страница

На рисунке 7 представлено, как выглядит страница управления подписками пользователя.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 - Страница подписок

На рисунке 8 представлено, как выглядит административная панель.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 - Административная панель

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Инструментальные средства

В качестве инструментального средства разработки программы была выбрана среда Visual Studio Code с расширениями для Python и JavaScript, так как данная платформа является оптимальным и удобным решением для разработки веб-приложений с использованием современных технологий

Для backend разработки использовался Python 3.11 с фреймворком FastAPI, для frontend - Next.js с TypeScript, для базы данных - PostgreSQL 15, для кэширования - Redis 7, для контейнеризации - Docker и Docker Compose.

## Отладка программы

Отладка кода проводилась с помощью встроенных инструментов отладки в Visual Studio Code и браузерных инструментов разработчика. Проверка ошибок в приложении осуществлялась через логи FastAPI и консоль браузера. При тестировании программы дефекты не были обнаружены.

Для отладки API использовались инструменты:

* Swagger UI для тестирования эндпоинтов
* Postman для комплексного тестирования
* pytest для автоматизированного тестирования
* Логирование через structlog для отслеживания ошибок

## Характеристики программы

Разработанное веб-приложение «Neuro Store» предназначено для работы в современных веб-браузерах с поддержкой HTML5, CSS3 и JavaScript ES6+.

Характеристики программы представлены в Таблице 7.

| № | Параметр | Значение |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | Язык программирования backend: | Python 3.11 |
| 2 | Фреймворк backend: | FastAPI |
| 3 | Язык программирования frontend: | TypeScript |
| 4 | Фреймворк frontend: | Next.js |
| 5 | База данных: | PostgreSQL 15 |
| 6 | Система кэширования: | Redis 7 |
| 7 | Контейнеризация: | Docker |
| 8 | Размер приложения: | ~50 MB |
| 9 | Время запуска: | < 30 секунд |
| 10 | Поддерживаемые браузеры: | Chrome 90+, Firefox 88+, Safari 14+ |

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы было успешно разработано веб-приложение «Neuro Store» для управления подписками на нейросети, представляющее собой удобный инструмент для управления доступом к ИИ-сервисам. Проект реализован в полном соответствии с техническим заданием и включает все запланированные функциональные возможности: создание и управление пользователями, управление каталогом нейросетей, создание и управление подписками, обработку платежей, администрирование системы и интеграцию с базой данных.

Реализация данного проекта позволила существенно расширить и закрепить практические навыки в области разработки веб-приложений. В процессе работы были освоены ключевые аспекты работы с современными технологиями, включая:

* проектирование и настройку REST API с использованием FastAPI;
* программирование frontend на Next.js с TypeScript;
* интеграцию с реляционными базами данных PostgreSQL;
* реализацию системы аутентификации и авторизации;
* оптимизацию производительности через кэширование Redis.

Особое внимание было уделено созданию безопасной и масштабируемой архитектуры. В ходе работы были изучены принципы проектирования веб-приложений и интеграции с внешними сервисами, что обеспечило баланс между функциональностью и производительностью.

Важным достижением проекта стала успешная реализация системы ролей и разграничения доступа, а также интеграция с платежными системами. Это создало удобный и гибкий инструмент, способный удовлетворить потребности широкой аудитории.

Дополнительно в рамках проекта были изучены современные подходы к контейнеризации приложений и автоматизации развертывания. Перспективы дальнейшего развития включают:

* добавление поддержки мобильных приложений;
* расширение аналитических функций с новыми типами отчетов;
* внедрение уведомлений в реальном времени;
* интеграцию с дополнительными платежными системами.

Практическая значимость выполненной работы заключается в создании полноценного инструмента для управления подписками на ИИ-сервисы, а также в приобретении опыта, который может быть применен в профессиональной деятельности. Разработанное приложение демонстрирует возможность создания качественных веб-решений с использованием современных инструментов разработки.

Таким образом, все поставленные в курсовой работе задачи были успешно выполнены. Созданный продукт соответствует современным требованиям к веб-приложениям и обладает потенциалом для дальнейшего развития и использования. Полученные знания и навыки существенно расширили профессиональные компетенции в области разработки веб-приложений и могут стать основой для реализации более сложных проектов в будущем.

# СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. ГОСТ Р – 7.0.5-2008 Система стандартов по информации, библиотечному и издательскому делу.
2. ГОСТ 19.101-77 Единая система программной документации виды программ и программных документов.
3. FastAPI Documentation, URL: https://fastapi.tiangolo.com/.
4. Next.js Documentation, URL: https://nextjs.org/docs.
5. PostgreSQL Documentation, URL: https://www.postgresql.org/docs/.
6. Docker Documentation, URL: https://docs.docker.com/.
7. SQLAlchemy Documentation, URL: https://docs.sqlalchemy.org/.
8. Фаулер, М. Архитектура корпоративных программных приложений. – М.: Вильямс, 2019. – 544 с.
9. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – М.: Вильямс, 2020. – 366 с.
10. Макконнелл, С. Совершенный код. – М.: Русская редакция, 2019. – 896 с.
11. Redis Documentation, URL: https://redis.io/documentation.
12. GitHub Actions Documentation, URL: https://docs.github.com/en/actions.
13. TypeScript Documentation, URL: https://www.typescriptlang.org/docs/.
14. Pytest Documentation, URL: https://docs.pytest.org/.
15. Alembic Documentation, URL: https://alembic.sqlalchemy.org/.
16. React Documentation, URL: https://react.dev/.
17. Tailwind CSS Documentation, URL: https://tailwindcss.com/docs.
18. JWT.io Documentation, URL: https://jwt.io/.
19. OpenAPI Specification, URL: https://swift.org/documentation/.
20. Docker Compose Documentation, URL: https://docs.docker.com/compose/.