



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

---

**Институт кибербезопасности и цифровых технологий**  
**Кафедра КБ-14 «Цифровые технологии обработки данных»**

## **ОТЧЁТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ**

**Тема практики:** «Разработка приложения для поиска и  
возврата утерянных вещей»

приказ Университета о направлении на практику от «18» апреля 2024 г. №3389-С

Отчет представлен к  
рассмотрению:

«\_\_» \_\_\_\_\_ 2024 г.

Студент группы: БСБО-05-20

\_\_\_\_\_  
(Подпись)

Верхотуров  
Валерий  
Сергеевич

Отчет утвержден.  
Допущен к защите:

Руководитель практики от кафедры

«\_\_» \_\_\_\_\_ 2024 г.

\_\_\_\_\_  
(Подпись)

Иванова Ирина  
Алексеевна

Москва 2024 г.



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

---

Институт кибербезопасности и цифровых технологий  
Кафедра КБ-14 «Цифровые технологии обработки данных»

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРЕДДИПЛОМНУЮ ПРАКТИКУ**

**Студенту 4 курса учебной группы БСБО-05-20  
Верхотурову Валерию Сергеевичу**

**Место и время практики:** РТУ МИРЭА, кафедра КБ-14 «Цифровые технологии обработки данных», с «19» апреля 2024 г. по «16» мая 2024 г.

**Должность на практике:** студент

**1. ЦЕЛЕВАЯ УСТАНОВКА:** развитие способностей в области проектной деятельности путем применения современных методов и инструментальных средств прикладной информатики, в том числе путем анализа и моделирования прикладных процессов с учетом выбранной темы исследования.

**2. СОДЕРЖАНИЕ ПРАКТИКИ:**

2.1. Изучить: исследовать информационные и прикладные процессы.

2.2. Практически выполнить: применить современные инструментальные средства для моделирования информационных и прикладных процессов.

2.3. Ознакомиться: с уровнем развития информационных и прикладных процессов с учетом темы исследования.

**3. ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:** оформить презентационный материал.

**4. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:** в процессе практики рекомендуется использовать периодические издания и отраслевую литературу годом издания не старше 5 лет.

Заведующий кафедрой:

«\_\_» \_\_\_\_\_ 2024 г.

\_\_\_\_\_  
(подпись) (Иванова И.А.)

**СОГЛАСОВАНО:**

Руководитель практики от кафедры

«\_\_» \_\_\_\_\_ 2024 г.

\_\_\_\_\_  
Иванова Ирина  
Алексеевна  
(подпись)

Задание получил

«\_\_» \_\_\_\_\_ 2024 г.

\_\_\_\_\_  
(Верхотуров, В.С.)  
(подпись)

**Проведенные  
инструктажи:**

Охрана труда:

«19» \_\_04\_\_ 2024 г.

Инструктирующий

\_\_\_\_\_  
(подпись)  
Иванова Ирина Алексеевна,  
кандидат технических наук  
доцент кафедры «Цифровые  
технологии обработки  
данных» ИКЦТ РТУ  
МИРЭА

Инструктируемый

\_\_\_\_\_  
(подпись)  
Верхотуров В.С.

Техника безопасности:

«19» \_\_04\_\_ 2024 г.

Инструктирующий

\_\_\_\_\_  
(подпись)  
Иванова Ирина Алексеевна,  
кандидат технических наук  
доцент кафедры «Цифровые  
технологии обработки  
данных» ИКЦТ РТУ  
МИРЭА

Инструктируемый

\_\_\_\_\_  
(подпись)  
Верхотуров В.С.

Пожарная безопасность:

«19» \_\_04\_\_ 2024 г.

Инструктирующий

\_\_\_\_\_  
(подпись)  
Иванова Ирина Алексеевна,  
кандидат технических наук  
доцент кафедры «Цифровые  
технологии обработки  
данных» ИКЦТ РТУ  
МИРЭА

Инструктируемый

\_\_\_\_\_  
(подпись)  
Верхотуров В.С.

С правилами внутреннего распорядка ознакомлен:

«19» \_\_04\_\_ 2024 г.

\_\_\_\_\_  
Верхотуров В.С.  
(подпись)



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ  
ПРЕДДИПЛОМНОЙ ПРАКТИКИ**

студента Верхотурова В.С. 4 курса группы БСБО-05-20 очной формы обучения,  
обучающегося по направлению подготовки 09.03.02 «Информационные системы и  
технологии» программа «Технологии искусственного интеллекта в безопасности»

Неделя	Сроки выполнения	Этап	Отметка о выполнении
1	19.04.2024	Подготовительный этап, включающий в себя организационное собрание (Вводная лекция о порядке организации и прохождения производственной практики, инструктаж по технике безопасности)	
2	20.04.2024	Выполнение задания по практике в соответствии с выданным заданием студента. (Мероприятия по сбору, обработке и структурированию материала, выполнение поставленной задачи)	
3	06.05.2024	Подготовка отчета и презентационного материала по практике (Оформление материалов отчета и презентации в полном соответствии с требованиями на оформление учебных работ студентов)	
4	16.05.2024	Защита отчета по производственной практике у руководителя практики. (Представление отчета по практике к защите)	

Руководитель  
практики от кафедры  
Обучающийся

\_\_\_\_\_/Иванова И.А./  
\_\_\_\_\_/Верхотуров В.С./

**Согласовано:**

Заведующий кафедрой

\_\_\_\_\_/Иванова И.А., к.т.н., доцент/

# ОТЧЁТ

## по преддипломной практике

студента 4 курса учебной группы БСБО-05-20 института КБ

Верхотуров Валерий Сергеевич

1. Практику проходил с 19.04.2024 г. по 16.05.2024 г. в ФГБОУ ВО «МИРЭА – Российский технологический университет», на кафедре КБ-14 «Цифровые технологии обработки данных», студент

*(место прохождения практики и должность)*

2. Задание на практику выполнил

в полном объеме/частично

*(нужное подчеркнуть)*

Не выполнены следующие задания:

*(указать также причины невыполнения)*

Подробное содержание выполненной на практике работы и достигнутые результаты:  
проведено исследование прикладной области в части изучения

Предложения по совершенствованию организации и прохождения практики:

предложений нет

Студент \_\_\_\_\_ (Верхотуров В.С.)

*(подпись)*

«\_\_» \_\_\_\_\_ 2024 г.

Заключение руководителя практики

Приобрел следующие профессиональные навыки: студент продемонстрировал профессиональные умения и навыки, знание и понимание прикладной области, задач, требующих решения в прикладной области, современные подходы и средства решения прикладных задач разных классов, умение находить и работать с различными источниками информации по профессиональной деятельности, структурировать отчет с учетом тематики исследования

Проявил себя как: студент соблюдал/не соблюдал сроки календарного графика практики.

*(нужное подчеркнуть)*

«\_\_» \_\_\_\_\_ 2024 г.

**Отчет проверил:**

**Руководитель практики от Университета**

\_\_\_\_\_ (Иванова И.А.)

*(подпись)*

## СОДЕРЖАНИЕ

1	ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ . . . . .	7
1.1	Выбор технологий для реализации системы . . . . .	7
1.2	Реализация модулей автоматизации процессов . . . . .	8
1.2.1	Модуль регистрации и авторизации пользователей . .	8
1.2.2	Модуль бесконечных лент объявлений потерянных, найденных вещей . . . . .	10
1.2.3	Модуль добавления и поиска утерянных вещей . . . .	11

# 1 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 1.1 Выбор технологий для реализации системы

Были выбрали надежный и современный набор технологий для своего веб-сервиса «Бюро находок». Вот краткий обзор некоторых из наиболее важных технологий в вашем стеке и почему они полезны для проекта:

- 1) Next.js — это мощная платформа React, которая обеспечивает рендеринг на стороне сервера и генерацию статических сайтов, что может помочь улучшить производительность и SEO веб-приложения. Он также поддерживает маршруты API, что упрощает внутреннюю разработку в рамках одного проекта [1].
- 2) React — популярная библиотека JavaScript для создания пользовательских интерфейсов, особенно одностраничных приложений, где требуется быстрый ответ на взаимодействие с пользователем. Это помогает в создании повторно используемых компонентов пользовательского интерфейса [2].
- 3) tRPC позволяет создавать типобезопасные API без необходимости писать схемы или генерировать типы. Он использует TypeScript, чтобы обеспечить проверку типов входных и выходных данных API, что снижает вероятность ошибок во время выполнения и повышает производительность разработчиков [3].
- 4) Prisma — это ORM (реляционное сопоставление объектов), которое упрощает доступ к базе данных и обеспечивает безопасность типов. Адаптер Prisma для NextAuth.js упрощает создание надежных систем аутентификации, привязанных непосредственно к схеме вашей базы данных [4].
- 5) NextAuth.js — комплексное решение для аутентификации в приложениях Next.js. Он поддерживает различные стратегии входа в систему с минимальной настройкой, повышая безопасность приложения [5].
- 6) React Query — это библиотека для эффективного извлечения, кэширования и обновления данных в приложениях React. Это помогает управлять состоянием сервера и синхронизировать пользовательский

интерфейс с данными без использования бойлерплейт кода [6].

- 7) Zustand — это минималистичное решение для управления состоянием, которое работает «из коробки» с React. Это просто, быстро и не требует шаблонного кода, как это делают другие библиотеки управления состоянием [8].
- 8) Tailwind CSS — это ориентированная на утилиты CSS-инфраструктура, наполненная классами, которые можно создавать для создания любого дизайна прямо в вашей разметке. Flowbite расширяет Tailwind дополнительными компонентами, ускоряя создание красивых и отзывчивых интерфейсов [?].
- 9) AWS SDK позволяет легко использовать веб-сервисы Amazon, такие как S3 для хранения, SES для отправки электронной почты и т. д., прямо из приложения. Это имеет решающее значение для масштабирования и управления инфраструктурой [10].
- 10) Headless UI — библиотека, которая предоставляет полностью неоформленные, полностью доступные компоненты пользовательского интерфейса, предназначенные для прекрасной интеграции с Tailwind CSS. Это помогает в создании пользовательских и доступных выпадающих списков, модальных окон и т.д [11].

## **1.2 Реализация модулей автоматизации процессов**

### **1.2.1 Модуль регистрации и авторизации пользователей**

Модуль представляет реализацию OAuth с единовременной передачей данных пользователя.

OAuth (Open Authorization) [12] — это открытый стандарт авторизации, который позволяет пользователям предоставлять безопасный делегированный доступ к своим учетным записям на различных сервисах, не раскрывая свои пароли. Реализация клиента OAuth включает несколько ключевых этапов и может варьироваться в зависимости от используемой версии OAuth (например, OAuth 1.0 или OAuth 2.0). Страница логина представлена на рис. 1.

Ниже описан процесс реализации клиента для OAuth 2.0.



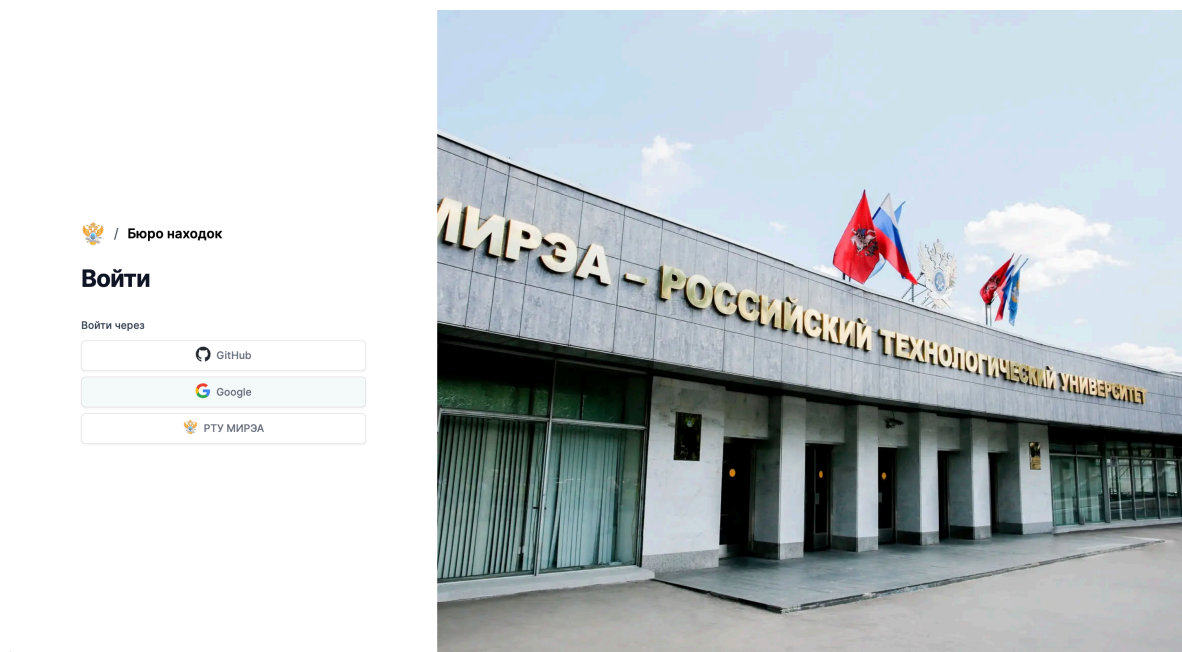


Рисунок 1 – Страница логина

### 1) Регистрация приложения

Перед тем как начать, вам нужно зарегистрировать ваше приложение на платформе, которую вы хотите использовать (например, Google, Facebook, GitHub). В процессе регистрации вам нужно будет указать:

- Название приложения
- URL, на который будет перенаправлен пользователь после авторизации (callback URL)

После регистрации вы получите client ID и client secret, которые будут использоваться для аутентификации вашего приложения.

### 2) Получение кода авторизации

Клиент должен перенаправить пользователя на URL авторизации, предоставленный сервисом, с необходимыми параметрами запроса:

- *response\_type* (обычно *code*)
- *client\_id*
- *redirect\_uri*
- *scope* (необязательно, зависит от того, к каким данным приложение стремится получить доступ)
- *state* (рекомендуется для защиты от CSRF-атак)

Пользователь входит в систему (если еще не вошел) и подтверждает

доступ к своим данным. После этого сервис перенаправляет пользователя обратно на *redirect\_uri* с кодом авторизации в параметрах URL.

### 3) Обмен кода авторизации на токен доступа

После получения кода авторизации, приложение делает запрос на сервер сервиса для обмена кода на токен доступа. Этот запрос должен содержать:

- *grant\_type* (обычно *authorization\_code*)
- *code* (полученный код авторизации)
- *redirect\_uri*
- *client\_id*
- *client\_secret*

Если запрос успешен, сервер ответит JSON-объектом, содержащим *access\_token* (и возможно *refresh\_token* и другие данные).

### 4) Использование токена доступа

Токен доступа используется для доступа к защищенным ресурсам пользователя. Он добавляется в заголовок HTTP-запроса как *Authorization: Bearer <token>*.

### 5) Обновление токена

Если получен *refresh\_token*, его можно использовать для получения нового *access\_token* после его истечения без необходимости повторной аутентификации пользователя.

## 1.2.2 Модуль бесконечных лент объявлений потерянных, найденных вещей

Для создания бесконечной ленты объявлений в React можно воспользоваться следующим подходом, основанным на базовых принципах React и встроенных возможностях JavaScript:

### 1) Структура состояния

Создается состояние в компоненте, которое будет хранить массив объявлений и переменную для отслеживания, загружены ли все данные.

## 2) Загрузка данных

Реализуется функция, которая будет загружать порции данных (например, по 10 объявлений за раз). Эта функция должна обновлять ваше состояние, добавляя новые объявления к уже загруженным.

## 3) Отслеживание прокрутки

Добавляется обработчик события прокрутки к элементу, в котором отображаются объявления. Когда пользователь достигает конца списка, вызывайте функцию загрузки данных.

## 4) Условия для загрузки

Необходимо убедиться, что новые данные загружаются только когда предыдущая загрузка завершена, чтобы избежать повторных запросов. Также нужно проверить, не достигнут ли конец списка данных, чтобы прекратить загрузку новых данных.

## 5) Отображение данных

Используется метод `map` для преобразования массива объявлений в `JSX`-элементы, которые будут отображаться в пользовательском интерфейсе.

## 6) Оптимизация производительности

Используются техники оптимизации, такие как `React.memo` для компонентов объявлений, чтобы избежать ненужных ререндеров при добавлении новых объявлений.

## 7) Обработка ошибок

Также нужно добавить обработку ошибок для ситуаций, когда загрузка данных может завершиться неудачей, например, при проблемах с сетью.

Реализация бесконечной ленты объявлений представлена на рис. 2.

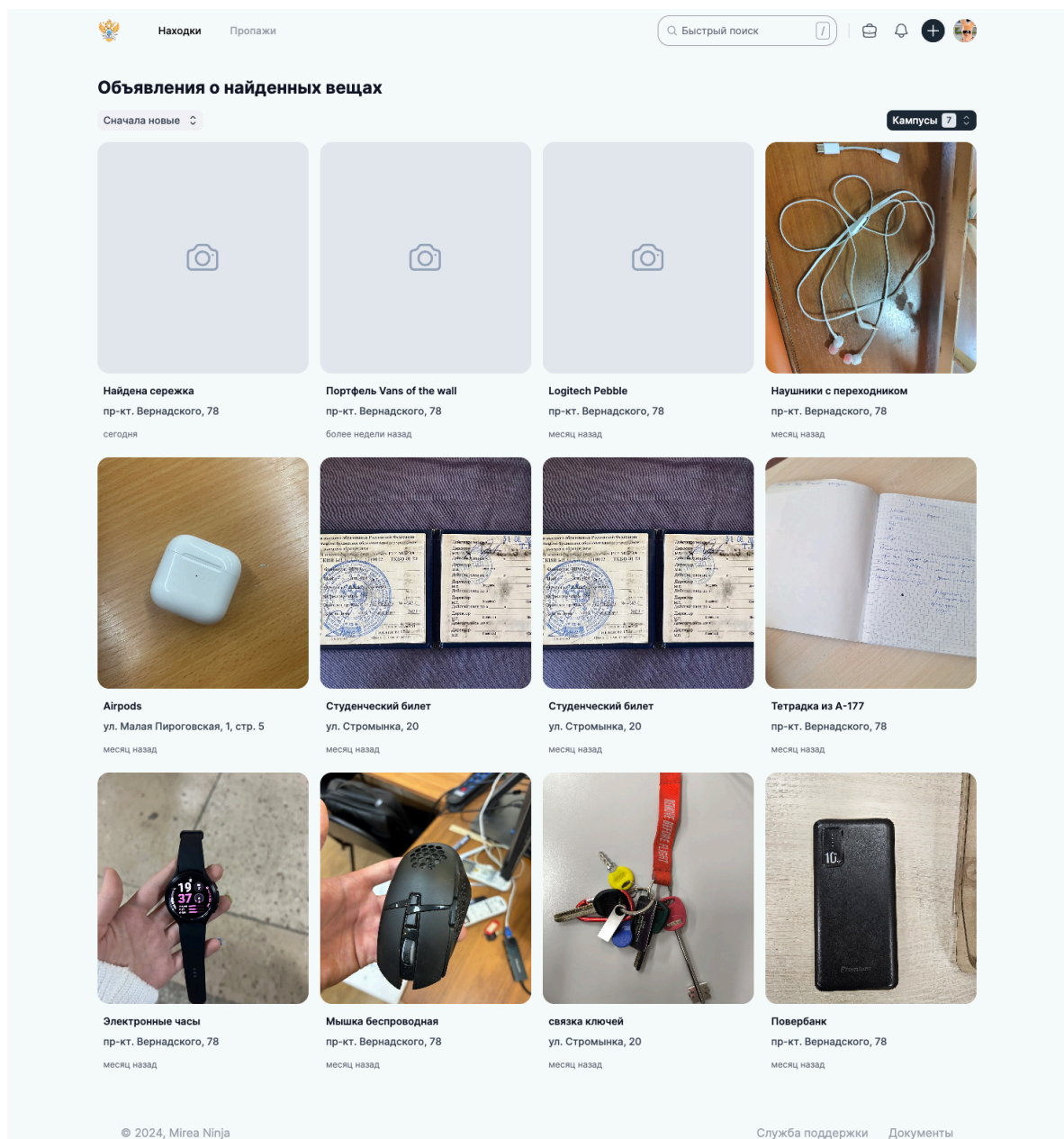


Рисунок 2 – Страница бесконечной ленты объявлений

### 1.2.3 Модуль добавления и поиска утерянных вещей

Для добавления утерянных вещей реализована форма с вводом названия, описания, возможностью загружать картинки в объектное хранилище. Форма представлена на рисунке 3.

Полнотекстовый поиск по названиям и описаниям происходит по названиям и описанию объявлений. Реализация клиентской части поиска представлена на рисунке 4.

Находки

Пропажи

Быстрый поиск

Сообщить о пропаже

Пропажа

Опишите пропажу, чтобы можно было определить, что она Ваша

Фотографии

Загрузите фотографии или перетащите

PNG, JPG, JPEG, GIF, WEBP до 10MB

Название

Описание

Место нахождения или контактная информация

Кампус

пр-кт. Вернадского, 78

☐ Я отнес(ла) находку в А-131
 

А-131 - это аудитория, в которую вы можете отнести находку. Сотрудники студенческого союза заберут её и позаботятся о том, чтобы с предметом ничего не случилось! Мы рекомендуем отнести найденный предмет туда
 [Аудитория на карте >](#)

Отменить

Сохранить

© 2024, Mirea Ninja

Служба поддержки

Документы

Рисунок 3 – Форма ввода утерянных вещей

Находки

Пропажи

Быстрый поиск

Все объявления

Сначала новые

Кампусы

Найдена сережка

В коридоре корпуса А найдена золотая сережка. Отнесли в А-430. За фотографией пишите в соц сети

Бумажник

Чёрный бумажник. Внутри пропуск на имя Новикова Артемия Сергеевича, найден на пуфике в чилл-аут зоне на 1 эт. корпуса А Вернадского 78. Лежит на охране.

AirPods 1 поколения

Валялись в чилл зоне на диване

Золотое кольцо с красными камнями

Найдено в лектории А-4 после 3 пары

Пропажи

Найдена сережка

Чёрная капка с ушками

Портфель Vans of the wall

Блок питания от ноутбука MSI

Рисунок 4 – Форма ввода утерянных вещей

## **Вывод по разделу**

Были реализованы все необходимые модули для полноценного функционирования приложения, включая современные технологии авторизации и создания веб-приложений. В частности, мы внедрили систему аутентификации и авторизации пользователей, которая обеспечивает высокий уровень безопасности и удобства. Эта система поддерживает метод входа по протоколу OAuth 2.0.

Для обеспечения надежности и отказоустойчивости приложения были внедрены механизмы логирования и мониторинга. Это позволяет оперативно выявлять и устранять возможные проблемы, а также анализировать поведение пользователей для дальнейшего улучшения приложения.

В итоге, все эти модули вместе обеспечивают надежное, безопасное и удобное функционирование приложения, соответствующее современным стандартам и требованиям.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Next.js – The React Framework for Production. URL: <https://nextjs.org/> (дата обращения: 16.05.2024).
2. React – JavaScript библиотека для построения пользовательских интерфейсов. URL: <https://reactjs.org/> (дата обращения: 16.05.2024).
3. tRPC – End-to-end типобезопасные программные интерфейсы.. URL: <https://trpc.io/> (дата обращения: 16.05.2024).
4. Prisma – открытая Node.js и TypeScript ORM с читаемыми моделями и автоматическими миграциями. URL: <https://www.prisma.io/> (дата обращения: 16.05.2024).
5. NextAuth – JavaScript библиотека для аутентификации. URL: <https://next-auth.js.org/> (дата обращения: 16.05.2024).
6. React Query – асинхронный менеджер состояний. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
7. React Query – асинхронный менеджмент состояний для сетевых запросов. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
8. Zustand – менеджер состояний клиентского приложения. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
9. Tailwind CSS – CSS фреймворк для быстрого построения современных вебсайтов. URL: <https://tailwindcss.com/> (дата обращения: 16.05.2024).
10. AWS SDK – JavaScript библиотека для взаимодействия с интерфейсами Amazon Web Services. URL: <https://aws.amazon.com/sdk-for-javascript/> (дата обращения: 16.05.2024).
11. Headless UI – полностью нестилизованная библиотека пользовательских интерфейсов для интеграции с Tailwind CSS. URL: <https://headlessui.com/> (дата обращения: 16.05.2024).

12. OAuth 2.0 - промышленный стандарт авторизации. URL: <https://oauth.net/2/> (дата обращения: 16.05.2024).