

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Современные платформы программирования (СПП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

Приложение «Электронный журнал»

БГУИР КП I-40 01 01 05 ПЗ

Выполнил

студент гр. 051003

Гаркушенко С.С.

Проверил:

Мискевич П. Л.

Минск 2023

Учреждение образования

«Белорусский государственный университет информатики и
радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ
Заведующий кафедрой ПОИТ

(подпись)
Лапицкая Н.В. 2023г.

ЗАДАНИЕ

по курсовому проектированию

Студенту Гаркушенко Станиславу Сергеевичу

1. Тема работы Веб-приложение «Заметки»

2. Срок сдачи законченной работы 12.05.2023г.

3. Исходные данные к работе Язык программирования JavaScript.
Использование фреймворка Express в качестве бэкэнд-сервера и библиотеки
React для клиентской части. Возможность регистрации и авторизации с
использованием JWT-токенов, создания разделов и подразделов, создания тем
и оформление сообщений с помощью языка разметки Markdown.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

Введение

1 Анализ литературных источников

2 Постановка задачи

3 Моделирование предметной области

4 Проектирование программного обеспечения

5 Разработка программного средства

6 Тестирование и проверка работоспособности программного средства

Заключение

Список использованных источников

Приложения

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

Схема алгоритма в формате A1

6. Консультант по курсовой работе Мискевич П. Л.

7. Дата выдачи задания 03.02.2023г.

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объема работы):

Раздел 1, 2 к 15.03.2023 – 30 % готовности работы;

Раздел 3, 4 к 15.04.2023 – 60 % готовности работы;

Раздел 5, 6 к 28.04.2023 – 90 % готовности работы;

Оформление пояснительной записки и графического материала к

05.05.2023 – 100 % готовности работы;

Защита курсового проекта с 08.05.2023 по 12.05.2023

РУКОВОДИТЕЛЬ Мискевич П. Л.
(подпись)

Задание принял к исполнению Гаркушенко С.С. 03.02.2023
(дата и подпись студента)

СОДЕРЖАНИЕ

Введение.....	5
1 Анализ литературных источников	6
1.1 Анализ существующих аналогов	6
1.1.1 Google Keep	6
1.1.2 Evernote	6
2 Постановка задачи	8
3 Моделирование предметной области	9
4 Проектирование программного средства	10
4.1 Анализ методов и способов разработки	10
4.2 Выбор технологии, языка и среды программирования	10
5 Разработка программного средства	12
5.1 Описание алгоритма updateNote.....	12
5.2 Описание алгоритма verifyJWT.....	13
5.3 Описание алгоритма login.....	14
5.4 Описание алгоритма deleteUser	15
6 Тестирование и проверка работоспособности программного средства	16
Заключение	18
Список использованной литературы	19
Приложение А	20

ВВЕДЕНИЕ

Современный ритм жизни требует организованности и эффективности в выполнении задач. Однако, с увеличением объема информации, которую люди получают каждый день, становится все сложнее запомнить и упорядочить все нужные данные. Именно поэтому приложения для заметок становятся неотъемлемой частью повседневной жизни. Приложения заметок обычно предлагают разнообразные функциональные возможности, которые помогают пользователям структурировать свою информацию.

Важно отметить, что существует широкий спектр приложений заметок на рынке с разными особенностями и возможностями. Некоторые из них фокусируются на минимализме и простоте использования, предлагая базовые функции заметок без лишней сложности. Другие приложения предоставляют расширенные функциональные возможности, такие как интеграция с календарем, возможность добавления медиафайлов, синхронизация с облачными сервисами или даже возможность записи голосовых заметок.

Большинство приложений заметок доступны как на мобильных устройствах (iOS и Android), так и на компьютерах (Windows и macOS), что обеспечивает гибкость и удобство использования независимо от предпочтений пользователя. Некоторые приложения также предлагают версии для веб-браузеров, позволяя получать доступ к заметкам прямо из интернета.

В процессе выполнения проекта будут анализироваться основные требования пользователей, изучаться существующие приложения заметок, исследоваться инновационные технологии и методы, которые могут быть использованы для создания более эффективного и удобного приложения. Будет обращено внимание на вопросы безопасности и защиты данных, чтобы обеспечить надежность и конфиденциальность информации пользователей.

Целью данного курсового проекта является исследование и разработка приложения заметок, которое поможет пользователям управлять и структурировать их информацию, сохранять идеи, задачи, и другую важную информацию в удобном и доступном формате. Основное внимание будет уделено созданию удобного и интуитивно понятного пользовательского интерфейса, а также разработке функциональности, позволяющей эффективно организовывать и хранить заметки.

В результате успешной реализации проекта, ожидается создание интуитивно понятного и функционального приложения заметок, которое поможет пользователям повысить их продуктивность и организованность, а также обеспечить удобное хранение и доступ к их важной информации.

1 АНАЛИЗ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ

1.1 Анализ существующих аналогов

Существует множество веб-версий приложений для ведения заметок, которые можно использовать в качестве аналогов. Некоторые из них специализируются на конкретных тематиках, таких как ведение заметок или список задач. Иные объединяют в себе все эти функции.

1.1.1 Google Keep

Одно из наиболее известных веб-приложений для ведения заметок – Google Keep. Является достаточно старым удобным для большинства пользователей, потому что практически не изменяло свой дизайн. Соответствует большинству требований обычного пользователя.

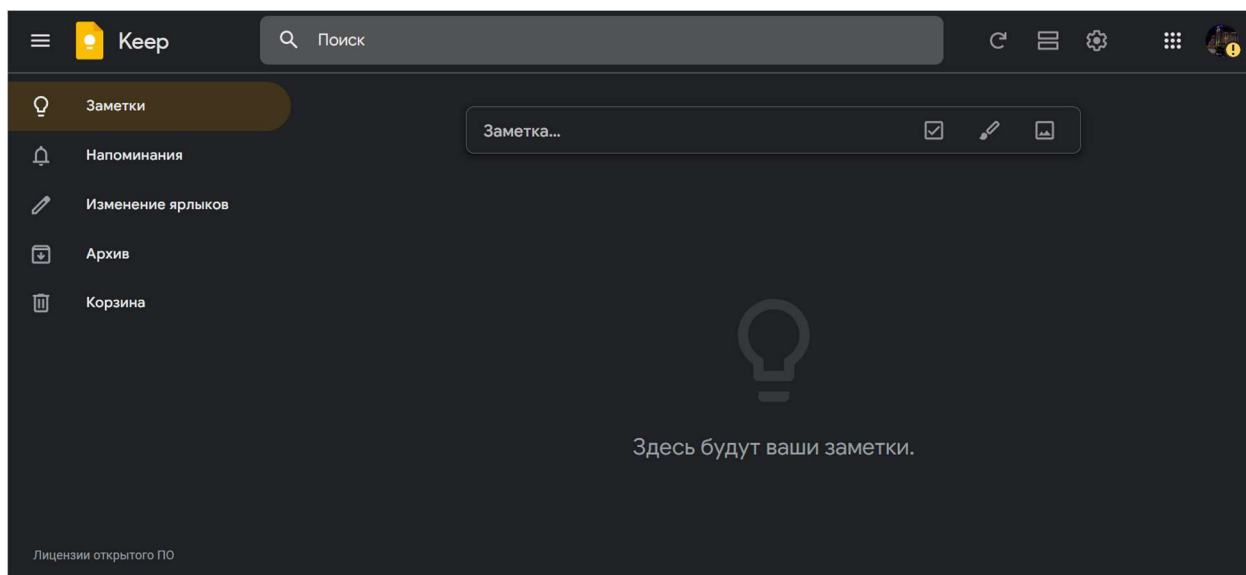


Рисунок 1.1 – Интерфейс приложения Google Keep

Плюсы:

- активная аудитория;
- удобный графический интерфейс;
- наличие корзины;
- возможность создать фотозаметки, заметки с рисунками.

Минусы:

- неудобство использования заметок несколькими пользователями

1.1.2 Evernote

Еще один аналог, который можно использовать в качестве примера для веб-приложения заметок – это Evernote. Приложение имеет современный нагроможденный дизайн, поэтому иногда тяжело быстро найти что-либо необходимое. Evernote позволяет добавлять к заметкам веб-клипы, изображения, документы, аудиофайлы и электронные письма.

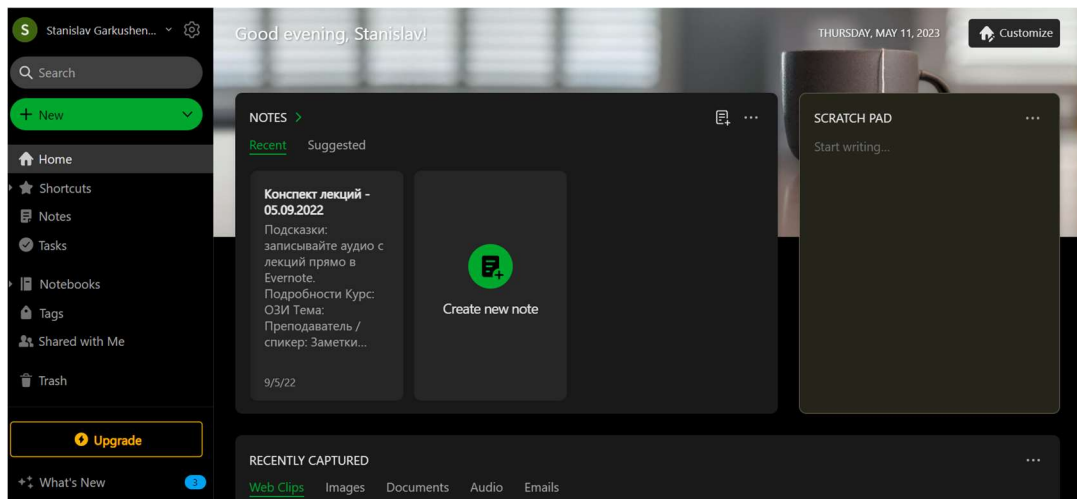


Рисунок 1.3

– Интерфейс приложения Evernote

Плюсы:

- высокая скорость работы сайта благодаря SSR;
- огромное количество пользователей;
- большое количество существующих ответов на популярные вопросы;
- продвинутый редактор заметок.

Минусы:

- наличие большого количества информации;
- нагромождённый интерфейс;
- сайт выглядит устаревшим в силу использования простых стилей.

2 ПОСТАНОВКА ЗАДАЧИ

Приложение заметок, разрабатываемое в рамках данного курсового проекта, должно иметь следующий функционал:

- поддержка аутентификации и авторизации для получения данных о текущем пользователе и проверки его прав на выполнение различных действий;
- создание пользователей, которые будут иметь различные роли;
- создание заметок пользователями;
- просмотр и редактирование заметок администраторами;
- удаление заметок их авторами или администраторами;
- возможность удаления заметок пользователей администраторами;
- просмотр данных пользователей администраторами и их редактирование;
- внизу страницы должна быть Markdown кнопка, позволяющая вернуться на домашнюю страницу.

3 МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Для разработки приложения заметок было произведено моделирование предметной области. В качестве основных сущностей были выделены следующие:

1 Пользователь – сущность, которая представляет зарегистрированного человека. У пользователя есть логин, пароль и ряд других атрибутов, один из которых – роль, которая определяет его возможности в приложении.

2 Заметка – сущность, которая представляет собой отдельное сообщение пользователя в рамках темы. Заметка содержит текст и информацию об авторе, дату создания, дату редактирования.

При создании заметки автор указывает название заметки, которое будет отображаться среди всех заметок. Затем указывает сам текст заметки и указывает выполнена ли заметка.

Моделирование приложения заметок можно продолжить и вот несколько идей, которые можно было бы добавить в данный проект:

- возможность загружать файлы в заметки;
- возможность цитировать другие заметки;
- оценка скорости выполнения задач, сохраняемых в заметках, и просмотр репутации пользователя, основанной на оценках его исполнительности;
- возможность вставлять видео из различных источников (таких как YouTube, Twitch и др.).

4 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

4.1 Анализ методов и способов разработки

Оптимальное функционирование большинства алгоритмов зависит от организации данных, поэтому проектирование программ следует начинать с разработки структуры, необходимой для хранения и обработки входных, выходных и промежуточных данных. Структура данных - это программная единица, которая обеспечивает удобное хранение и просмотр множества однотипных или логически связанных данных в вычислительной технике. Часто структура данных является реализацией абстрактного типа данных.

Понимание значения данных, используемых в программе, включая символы и числа, а также способы их обработки, является необходимым для эффективной разработки программной обработки данных. Все данные, необходимые для решения практических задач, могут быть разделены на несколько типов, включая основные (простые) и сложные типы, которые могут быть созданы пользователями для решения конкретных задач.

Для разработки программного продукта, необходимо спроектировать и разработать пользовательский интерфейс, который обеспечит удобное визуальное отображение результатов работы и эргономичное взаимодействие с компьютером пользователя. Определение пользовательского интерфейса является неотъемлемой частью проектирования приложения "Моделирование сложных связанных неисправностей для тестирования запоминающих устройств". Пользовательский интерфейс включает в себя не только графический элемент, но и способы взаимодействия пользователя с программным обеспечением.

Программный интерфейс - это набор унифицированных связей, предназначенных для обмена информацией между компонентами вычислительной системы. Он задает необходимый набор процедур, параметров и способов доступа. Интерфейс пользователя включает элементы и компоненты программы, которые позволяют пользователю взаимодействовать с программным обеспечением.

Интерфейс пользователя — это элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением.

4.2 Выбор технологии, языка и среды программирования

Технологией программирования называют совокупность методов и средств, используемых в процессе разработки программного обеспечения. Как любая другая технология, технология программирования представляет собой набор технологических инструкций, включающих:

- указание последовательности выполнения технологических операций;
- перечисление условий, при которых выполняется та или иная операция;

– описания самих операций, где для каждой операции определены исходные данные, результаты, а также инструкции, нормативы, стандарты, критерии и методы оценки и т. п.

Кроме набора операций и их последовательности, технология также определяет способ описания проектируемой системы, точнее модели, используемой на конкретном этапе разработки.

Основной целью данного курсового проекта является знакомство с современными платформами программирования, фреймворками и библиотеками, популярными приёмами программирования и реализация основного функционала для работоспособного приложения заметок.

Всё приложение должно быть разделено на три основных слоя:

– База данных: в рамках данного проекта было принято решение использовать СУБД MongoDB.

– Бэкэнд: для реализации серверной части был выбран фреймворк Express, а для типизированного взаимодействия с базой данных была выбрана библиотека Mongoose, позволяющая оформлять запросы в качестве вызовов методов на языке JavaScript.

– Фронтэнд: для клиентской части приложения будет использована одна из самых популярных библиотек React. Для визуального оформления сайта была выбрана библиотека кастомизируемых компонентов React Material UI.

Для разработки будет использоваться среда программирования VS Code, которая довольно сильно облегчает разработку программного средства за счет обширного количества возможностей, встроенных библиотек и модулей.

JavaScript является одним из современных самых популярных языков программирования.

5 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

5.1 Описание алгоритма updateNote

Алгоритм обновления заметки.



Рисунок 5.1 – Схема алгоритма updateNote

5.2 Описание алгоритма verifyJWT

Алгоритм проверки подписи JWT.

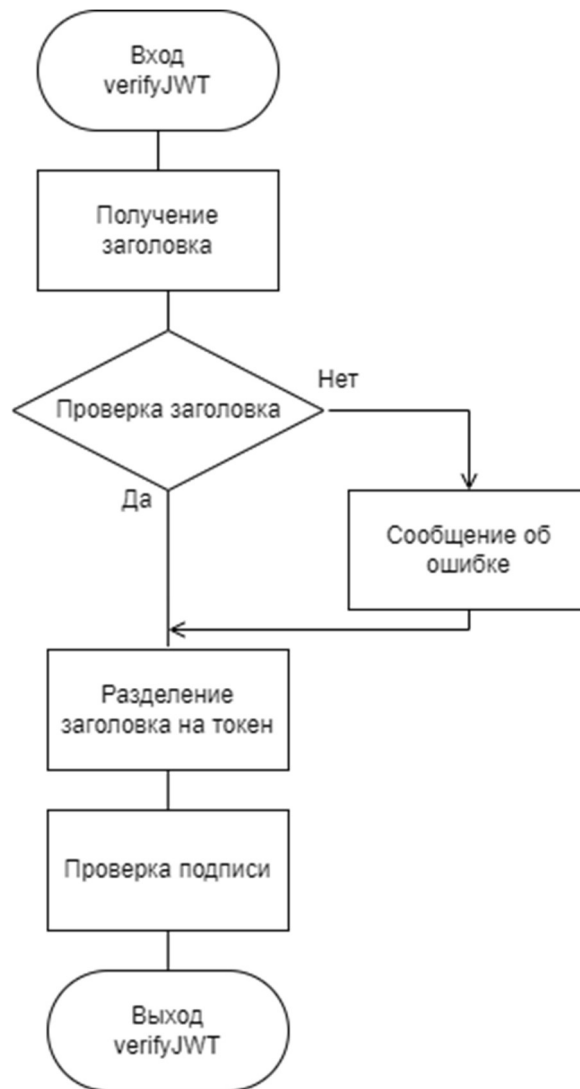


Рисунок 5.2 – Схема алгоритма SetGrade

5.3 Описание алгоритма login

Алгоритм авторизации.

Схема алгоритма см. приложение к курсовому проекту схема А1

5.4 Описание алгоритма deleteUser

Алгоритм удаления пользователя.

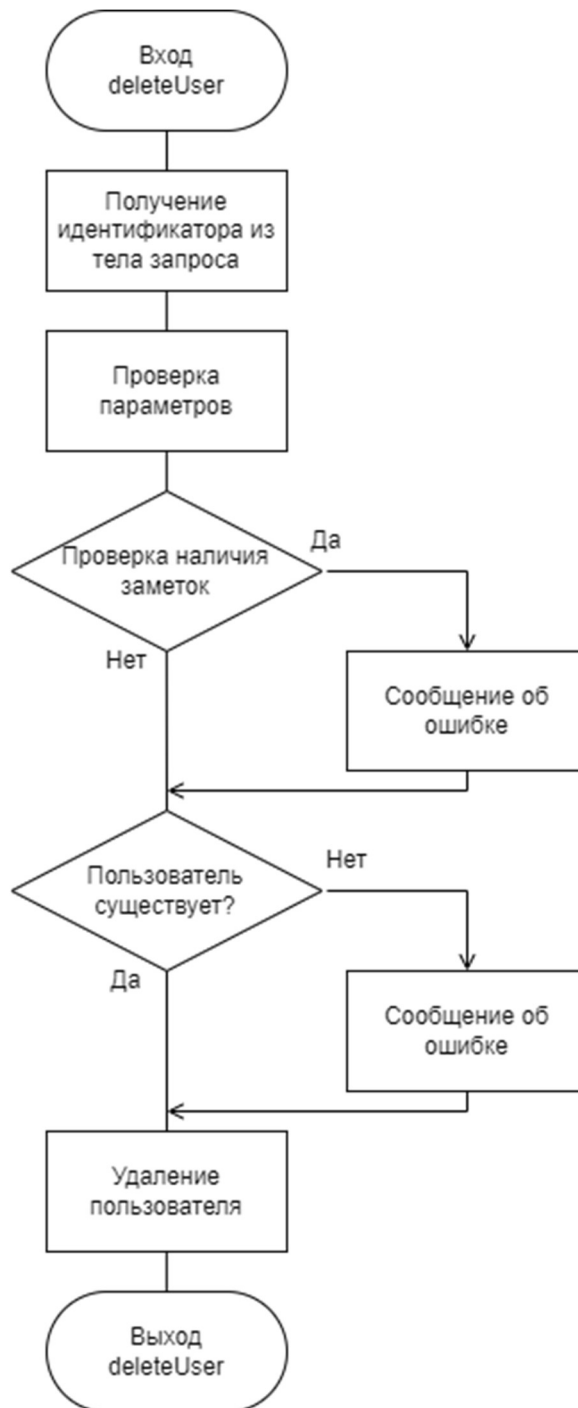


Рисунок 5.3 – Схема алгоритма deleteUser

6 ТЕСТИРОВАНИЕ И ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО СРЕДСТВА

Таблица 1 – Тесты

Идентификатор теста	Шаги воспроизведения	Ожидаемый результат
Открытие сайта	1) Открыть сайт.	Отображение страницы авторизации
Авторизация	1) Открыть сайт. 2) Авторизоваться, используя верные логин и пароль.	Успешная авторизация, отображение списка доступных секций в виде списка
Просмотр заметок	1) Авторизироваться в системе. 2) Перейти на страницу заметок.	Корректное отображение заметок для пользователя с данными привилегиями
Добавление пользователя	1) Авторизироваться в системе под администратором/менеджером. 2) Перейти на страницу добавления пользователей.	Корректное добавление пользователя
Добавление заметок	1) Авторизироваться в системе. 2) Перейти на добавления заметки.	Корректное добавление заметки
Редактирование заметки	1) Авторизироваться в системе. 2) Перейти на страницу заметок. 3) Выбрать изменение заметки.	Корректное редактирование заметки
Изменение информации о пользователях администратором	1) Авторизироваться в системе под администратором. 2) Перейти на страницу управления пользователями. 3) Выбрать пользователя для редактирования.	Корректное редактирование данных выбранного пользователя
Просмотр пользователем заметок администратора	1) Авторизироваться в системе под пользователем. 2) Перейти на страницу заметок.	Отображение заметок, доступных к просмотру только для пользователя

Продолжение Таблицы 1

Просмотр менеджером заметок пользователя	1) Авторизироваться в системе под менеджером. 2) Перейти на страницу заметок.	Корректное отображение заметок пользователя и менеджера
Нажатие кнопки “Домой”	1) Авторизироваться в системе под пользователем. 2) Перейти на страницу заметок. 3) Нажать кнопку “Домой”	Корректное возвращение на домашнюю страницу
Удаление заметок администратором	1) Авторизироваться в системе. 2) Перейти на страницу заметок. 3) Выбрать заметку для удаления.	Корректное удаление выбранной заметки
Выход из аккаунта	1) Авторизироваться в системе. 2) Перейти на главную страницу. 3) Нажать кнопку выхода из аккаунта	Пользователь выходит из своего аккаунта и переходит на главную страницу

Все выбранные функции выполнили свою работу корректно. Такого набора функций достаточно для работоспособности электронного журнала.

ЗАКЛЮЧЕНИЕ

По итогу работы над курсовым проектом было разработано программное средство «Веб-приложение для ведения заметок», обладающее графическим интерфейсом для взаимодействия с пользователем, в том числе поддерживающим навигацию между различными URL-маршрутами.

Приложение прошло все этапы тестирования и продемонстрировало корректную быструю работу, обработав исключительные ситуации и выполнив операции по созданию разделов, тем и сообщений. Также был проверен вспомогательный функционал, помогающий проверять роль авторизованного пользователя на выполнение различных задач, требующих определённых разрешений.

Среди преимуществ программы можно отметить интуитивно понятный интерфейс, простоту реализации и использование современных средств разработки, увеличивающих стабильность приложения даже в очень неоднозначных ситуациях.

Бесценным результатом курсового проекта является полученный опыт работы с современным языком программирования JavaScript и различными готовыми решениями, такие как:

- библиотека для разработки клиентских веб-приложений React;
 - Mongoose для типизированного доступа к базе данных Mongo;
 - мощный фреймворк для создания серверных решений на Express;
 - кастомизируемая библиотека React компонентов Material UI,
- позволяющая легко использовать готовые решения большинства дизайн-задач.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] Paul, Daniels RxJS in Action / Daniels Paul, Atencio Luis. — 1. — Shelter Island: Manning Publications Co., 2017. — 354 с.
- [2] Документация по языку JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 27.03.2023.
- [3] Документация библиотеки Express [Электронный ресурс]. – Режим доступа: <https://expressjs.com/ru/>. – Дата доступа: 25.03.2023.
- [4] Документация MongoDB [Электронный ресурс]. – Режим доступа: <https://www.mongodb.com/docs/>. – Дата доступа: 07.04.2023.
- [5] Документация библиотеки React [Электронный ресурс]. – Режим доступа: <https://react.dev/>. – Дата доступа: 14.04.2023.
- [6] Документация библиотеки компонентов Material UI [Электронный ресурс]. – Режим доступа: <https://mui.com/material-ui/>. – Дата доступа: 05.05.2023.
- [7] Документация NodeJS [Электронный ресурс]. – Режим доступа: <https://nodejs.org/en/docs>. – Дата доступа: 05.05.2023.

ПРИЛОЖЕНИЕ А
(обязательное)
Исходный код программы

```
import { Routes, Route } from 'react-router-dom'
import Layout from '../components/Layout'
import Public from '../components/Public'
import Login from '../features/auth/Login';
import DashLayout from '../components/DashLayout'
import Welcome from '../features/auth/Welcome'
import NotesList from '../features/notes/NotesList'
import UsersList from '../features/users/UsersList'
import EditUser from '../features/users/EditUser'
import NewUserForm from '../features/users/NewUserForm'
import EditNote from '../features/notes/EditNote'
import NewNote from '../features/notes/NewNote'
import Prefetch from '../features/auth/Prefetch'
import PersistLogin from '../features/auth/PersistLogin'
import RequireAuth from '../features/auth/RequireAuth'
import { ROLES } from '../config/roles'
import useTitle from '../hooks/useTitle';

function App() {
  useTitle('Technical Notes')

  return (
    <Routes>
      <Route path="/" element={<Layout />}>
        {/* public routes */}
        <Route index element={<Public />} />
        <Route path="login" element={<Login />} />

        {/* Protected Routes */}
        <Route element={<PersistLogin />}>
          <Route element={<RequireAuth
allowedRoles={([...Object.values(ROLES)]} />}>
            <Route element={<Prefetch />}>
              <Route path="dash" element={<DashLayout />}>

                <Route index element={<Welcome />} />

                <Route element={<RequireAuth
allowedRoles={[ROLES.Manager, ROLES.Admin]} />}>
                  <Route path="users">
                    <Route index element={<UsersList />} />
                    <Route path=":id" element={<EditUser />} />
                    <Route path="new" element={<NewUserForm />}>
                  />

                  </Route>
                </Route>

                <Route path="notes">
```

```

        <Route index element={<NotesList />} />
        <Route path=":id" element={<EditNote />} />
        <Route path="new" element={<NewNote />} />
    </Route>

    </Route>{/* End Dash */}
  </Route>
</Route>
</Route>{/* End Protected Routes */}

</Route>
</Routes >
);
}

export default App;

```