Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Кафедра информационных технологий

Факультет информационных технологий и анализа больших данных

**РАСЧЕТНО-АНАЛИТИЧЕСКАЯ РАБОТА**

**на тему**

**Разработка и создание веб-приложения для создания и хранения текстовых заметок**

Дисциплина «ИНФОРМАЦИОННЫЕ СИСТЕМЫ

НА ОСНОВЕ ИНТЕРНЕТ-ТЕХНОЛОГИЙ»,

Направление подготовки 09.03.03 Прикладная информатика

Выполнил:

Студент группы ЗБ-ПИ21-2

Института открытого образования

Абдусалимзода Искандар

Проверил:

к.т.н., доцент

Кафедры информационных технологий

Алюнов Александр Николаевич

Москва 2025

**Техническое описание**

Приложение построено на основе трехуровневой архитектуры:

1. **Клиентский уровень (Frontend)**

Клиентская часть реализована на React.js и отвечает за пользовательский интерфейс. Структура клиентской части организована по принципу компонентного подхода, где каждый компонент отвечает за определенную функциональность. React Context используется для управления глобальным состоянием приложения, что позволяет эффективно передавать данные между компонентами без избыточной передачи props.

Основные компоненты включают:

* Auth компоненты (Login.js, Register.js) управляют процессами аутентификации
* Notes компоненты (NotesList.js, NoteEditor.js) обеспечивают работу с заметками
* Layout компоненты (Header.js) определяют структуру интерфейса
* Policy компоненты (PrivacyPolicy.js, CookiePolicy.js) предоставляют информацию о политиках

1. **Серверный уровень (Backend)**

Серверная часть разработана на Node.js с использованием фреймворка Express.js. REST API обеспечивает взаимодействие между клиентом и сервером. Сервер обрабатывает запросы, управляет сессиями пользователей и взаимодействует с базой данных.

Основные модули сервера:

* Контроллеры обрабатывают входящие HTTP-запросы
* Middleware осуществляет промежуточную обработку запросов
* Роуты определяют пути API и связывают их с контроллерами
* Модели описывают структуру данных

1. **Уровень данных (Database)**

MongoDB используется как основная база данных. Mongoose служит ODM (Object Data Modeling) библиотекой для MongoDB и Node.js, обеспечивая строгую типизацию схем и валидацию данных. База данных расположена на облаке MongoDB Atlas, что позволяет подключаться с БД с любой точки мира.

**Система безопасности**

**Аутентификация**

Процесс аутентификации реализован через JWT (JSON Web Tokens). При успешной авторизации сервер генерирует токен, содержащий зашифрованную информацию о пользователе. Этот токен используется для последующих запросов к API.

**Хеширование паролей**

Все пароли пользователей хешируются с использованием bcrypt перед сохранением в базу данных. Bcrypt автоматически генерирует соль и добавляет ее к паролю перед хешированием, что обеспечивает дополнительный уровень защиты.

**Защита API**

API защищено несколькими уровнями безопасности:

* CORS (Cross-Origin Resource Sharing) защита предотвращает несанкционированные межсайтовые запросы
* Валидация входных данных проверяет корректность всех входящих запросов
* Middleware проверяет JWT токены для защиты приватных endpoint'ов

**Структура базы данных**

**Коллекция Users**

Хранит информацию о пользователях:

* \_id: уникальный идентификатор
* username: имя пользователя
* email: электронная почта
* password: хешированный пароль
* createdAt: дата создания аккаунта

**Коллекция Notes**

Содержит заметки пользователей:

* \_id: уникальный идентификатор
* title: заголовок заметки
* content: содержание заметки
* user: ссылка на пользователя-владельца
* createdAt: дата создания
* updatedAt: дата последнего обновления

**Взаимодействие компонентов**

При взаимодействии пользователя с интерфейсом клиентское приложение отправляет запросы к серверу через REST API. Сервер обрабатывает эти запросы, проверяет аутентификацию, выполняет необходимые операции с базой данных и возвращает результат клиенту. Все данные передаются в формате JSON, что обеспечивает эффективный обмен информацией между клиентом и сервером.  
  
**Масштабируемость**

Архитектура приложения позволяет легко масштабировать систему. MongoDB поддерживает горизонтальное масштабирование через шардинг, а Node.js сервер может быть развернут на нескольких экземплярах с использованием балансировщика нагрузки. React-приложение оптимизировано для производительности через механизм виртуального DOM и эффективное управление состоянием.

**Руководство пользователя**

**Начало работы**

При первом посещении сайта пользователь попадает на страницу входа. Для начала работы необходимо создать учетную запись, используя форму регистрации, доступную по кнопке "Регистрация" в верхней части экрана.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описаниеРис. 1 – Страница входа

**Регистрация**

На странице регистрации необходимо заполнить следующие поля:

* Имя пользователя
* Email
* Пароль

Изображение выглядит как текст, программное обеспечение, снимок экрана, дизайн

Автоматически созданное описание  
Рис. 2 – Страница регистрации

После успешной регистрации пользователь автоматически входит в систему.

**Вход в систему**

Для входа в существующую учетную запись следует использовать кнопку "Войти" в верхней части экрана и ввести email и пароль.

**Основной интерфейс**

После входа пользователь попадает на главный экран, который разделен на две части:

* Левая панель содержит список всех заметок и кнопку "Новая заметка"
* Правая панель представляет собой редактор текущей заметки

Изображение выглядит как снимок экрана, текст, программное обеспечение

Автоматически созданное описание  
Рис. 3 – Основная страница и создание заметок

**Работа с заметками**

Создание новой заметки:

1. Нажать кнопку "Новая заметка" в левой панели
2. Ввести заголовок заметки в верхнее поле
3. Ввести содержание заметки в основное поле
4. Нажать кнопку "Создать" внизу редактора

Редактирование заметки:

1. Выбрать нужную заметку из списка в левой панели
2. Внести необходимые изменения в полях заголовка или содержания
3. Нажать кнопку "Сохранить"

Удаление заметки:

1. Выбрать заметку из списка
2. Нажать кнопку "Удалить" внизу редактора
3. Подтвердить удаление

Изображение выглядит как текст, снимок экрана, программное обеспечение, дизайн

Автоматически созданное описание  
Рис. 4 – Сохранение заметки

Изображение выглядит как программное обеспечение, текст, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание  
Рис. 5 - Удаление заметки

**Поиск заметок**

Все заметки отображаются в левой панели в хронологическом порядке. Каждая заметка в списке показывает:

* Заголовок
* Начало содержания
* Дату и время создания

Изображение выглядит как текст, программное обеспечение, снимок экрана, веб-страница

Автоматически созданное описание  
Рис. 6 – Поиск заметок

**Завершение работы**

Для выхода из системы следует нажать кнопку "Выйти" в верхней части экрана. После этого все несохраненные изменения будут потеряны, и потребуется повторный вход в систему.

**Политики и конфиденциальность**

В нижней части экрана доступны ссылки на политику конфиденциальности и политику использования cookie. Рекомендуется ознакомиться с этими документами для понимания принципов обработки персональных данных.

Изображение выглядит как текст, программное обеспечение, Веб-сайт, веб-страница

Автоматически созданное описание  
Рис. 7 – Политика конфиденциальности  
  
Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описание  
Рис. 8 – Политика cookie