Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Факультет информационных технологий и анализа больших данных

Департамент анализа данных и машинного обучения

Выпускная квалификационная работа

на тему: «Разработка веб-приложения для организации совместной работы над проектами»

Направление подготовки 09.03.03 «Прикладная информатика»,

Профиль «ИТ-сервисы и технологии обработки данных в экономике и финансах»

Выполнил студент группы ПИ21-3

Балашкин Андрей Михайлович \_\_\_\_\_\_\_\_\_

Руководитель к.т.н., доцент

Хасанов Ильнур Ильдароваич \_\_\_\_\_\_\_\_\_

**ВКР соответствует предъявленным требованиям**

Руководитель Департамента

к.т.н., доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Петросов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

Москва 2025**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc1)

[1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc2)

[1.1 Существующие решения 6](#_Toc3)

[1.1.1 Trello 6](#_Toc4)

[1.1.2 Asana 7](#_Toc5)

[1.1.3 Notion 8](#_Toc6)

[1.1.4 ClickUp 8](#_Toc7)

[1.1.5 Проблемы существующих решений 9](#_Toc8)

[1.1.6 Преимущества разрабатываемого веб-приложения 9](#_Toc9)

[1.2 Определение потребностей пользователей 11](#_Toc10)

[1.2.1 Анализ целевой аудитории 11](#_Toc11)

[1.2.2 Основные проблемы пользователей 12](#_Toc12)

[1.2.3 Ожидания и требования пользователей 13](#_Toc13)

[1.2.4 Обоснование необходимости разработки 14](#_Toc14)

[1.3 Составление требований 15](#_Toc15)

[1.3.1 Функциональные требования 15](#_Toc16)

[1.3.2 Нефункциональные требования 16](#_Toc17)

[1.3.3 Технические требования 17](#_Toc18)

[1.3.4 Итоговые выводы 17](#_Toc19)

[2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 19](#_Toc20)

[3. ПРАКТИЧЕСКАЯ ЧАСТЬ 20](#_Toc21)

[ЗАКЛЮЧЕНИЕ 21](#_Toc22)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc23)

# ВВЕДЕНИЕ

Совместная работа над проектами — неотъемлемая часть современного бизнеса, образования и научных исследований. Независимо от сферы деятельности, успешная координация участников, управление задачами и обмен информацией играют ключевую роль в достижении поставленных целей. Для упрощения этих процессов используются различные цифровые инструменты, однако существующие решения не всегда удобны, универсальны и доступны.

В рамках данной работы разработано веб-приложение, которое объединяет todo-лист, канбан-доску, систему текстовых записей и механизм совместной работы в одном интерфейсе. В отличие от популярных сервисов, разрабатываемый продукт ориентирован на простоту, открытый доступ и гибкость, позволяя командам эффективно управлять проектами без необходимости сложных настроек или дорогостоящих подписок.

**Актуальность темы**

В современном мире команды работают удаленно, распределенно и в быстром темпе, что требует удобных инструментов для планирования, коммуникации и организации информации. Однако анализ существующих решений показывает, что большинство популярных платформ (Trello, Asana, ClickUp) имеют недостатки:

* сложные интерфейсы, требующие времени на обучение
* ограниченный бесплатный функционал, вынуждающий пользователей переходить на платные тарифы
* разрозненность инструментов (разные сервисы для задач, заметок, управления знаниями)

Предлагаемое веб-приложение решает эти проблемы, обеспечивая единое пространство для работы, где пользователи могут организовывать задачи, вести записи и взаимодействовать без лишних сложностей.

**Цель работы** разработка универсального веб-приложения для совместной работы над проектами, включающего:

* удобное управление задачами (todo-лист, канбан-доска)
* систему текстовых записей для хранения и организации информации
* механизм совместной работы без сложных ролей и ограничений

**Задачи работы**

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ существующих решений, выявить их сильные и слабые стороны
2. Определить ключевые потребности пользователей и сформировать требования к веб-приложению
3. Разработать архитектуру веб-приложения, обосновав выбор технологий
4. Реализовать функционал todo-листа, канбан-доски и системы заметок
5. Обеспечить безопасность данных и стабильную работу сервиса
6. Провести тестирование приложения и оценить его эффективность

**Методы исследования**

В работе используются следующие методы:

* анализ существующих решений в области управления проектами
* опрос и сбор обратной связи от потенциальных пользователей
* методы проектирования веб-приложений, включая архитектурное моделирование
* разработка и тестирование ПО на основе современных технологий

**Структура работы**

Дипломная работа состоит из трех глав:

1. Исследование предметной области – анализ существующих решений, выявление проблем и формирование требований к разрабатываемому приложению
2. Теоретическая часть – основы проектирования веб-приложений, выбор архитектуры и технологий, обеспечение безопасности данных
3. Практическая часть – процесс разработки, описание ключевых функций, тестирование и оценка эффективности полученного решения

Разрабатываемое веб-приложение представляет собой универсальный инструмент для совместной работы над проектами, устраняющий недостатки существующих решений. Оно объединяет удобство, простоту использования и гибкость, позволяя пользователям эффективно управлять задачами и информацией без дополнительных затрат. Данная работа демонстрирует полный процесс создания такого продукта – от анализа требований до реализации и тестирования.

# 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Существующие решения

Совместная работа над проектами – это ключевой аспект эффективного взаимодействия внутри команд, независимо от сферы деятельности. В настоящее время существует множество инструментов, предлагающих различные подходы к управлению задачами, организации информации и координации работы участников. Однако, несмотря на широкий выбор, многие из них имеют ограничения, которые делают их использование менее удобным или дорогим для определенных категорий пользователей.

Ниже рассмотрены наиболее популярные решения, их преимущества и недостатки, а также обоснована необходимость разработки нового веб-приложения, сочетающего в себе удобство и функциональность.

### 1.1.1 Trello

Trello – один из самых известных инструментов управления проектами, основанный на методологии Kanban. Пользователи создают карточки задач, перемещают их между колонками (например, "To Do", "In Progress", "Done"), устанавливают дедлайны и назначают исполнителей.

**Преимущества:**

* Простота интерфейса. Использовать Trello легко даже неподготовленным пользователям.
* Интерактивность. Обновления задач видны в реальном времени.
* Интеграции. Поддержка популярных сервисов (Google Drive, Slack, Dropbox и др.).

**Недостатки:**

* Ограниченный функционал в бесплатной версии. Для полноценного использования часто требуется платная подписка.
* Отсутствие расширенных возможностей управления. Например, нет встроенной системы учета времени или детальной аналитики.
* Неудобно для больших проектов. При росте числа задач доска быстро перегружается.

Trello хорошо подходит для небольших команд, но его функциональности недостаточно для комплексных проектов, где требуется больше контроля и гибкости.

### 1.1.2 Asana

Asana – мощный инструмент для управления проектами, сочетающий в себе списки задач, доски, диаграммы Ганта и систему уведомлений.

**Преимущества:**

* Гибкость. Возможность работы с задачами в разных представлениях (список, доска, календарь).
* Развитая система уведомлений. Пользователи получают напоминания и могут легко отслеживать изменения.
* Поддержка командной работы. Можно назначать ответственных, добавлять комментарии и прикреплять файлы.

**Недостатки:**

* Сложность освоения. Интерфейс перегружен функциями, что может отпугнуть новых пользователей.
* Дорогие платные планы. Бесплатная версия сильно ограничена по возможностям.
* Избыточность для небольших проектов. Многие функции могут оказаться ненужными, но усложняют работу.

Asana удобна для крупных команд, но её сложность и стоимость могут быть избыточными для малых и средних проектов.

### 1.1.3 Notion

Notion – это гибкий инструмент, сочетающий заметки, базы данных, списки задач и систему управления знаниями. Он часто используется как персональный или командный органайзер.

**Преимущества:**

* Мощная система организации данных. Поддержка вложенных страниц, тегов, баз данных.
* Гибкость. Можно создавать практически любые структуры информации.
* Удобно для ведения документации. Позволяет хранить не только задачи, но и связанные с ними заметки и инструкции.

**Недостатки:**

* Не оптимизирован для управления проектами. Первоначально Notion создавался как инструмент для ведения записей, поэтому функции управления задачами здесь вторичны.
* Медленная работа с большими объемами данных. Если проект активно растет, могут возникать задержки в загрузке страниц.
* Сложность настройки. Требует времени для создания удобной структуры рабочих процессов.

Notion хорош для хранения информации и заметок, но не подходит для полноценного управления проектами, поскольку в нём отсутствуют многие важные функции, такие как гибкое управление задачами и наглядные канбан-доски.

### 1.1.4 ClickUp

ClickUp – это комплексная платформа для управления проектами, включающая задачи, календарь, диаграммы Ганта и систему отчетности.

**Преимущества:**

* Гибкость настройки. Можно адаптировать интерфейс под разные задачи.
* Разнообразие представлений данных. Поддержка списков, досок, календарей и даже документов.
* Инструменты аналитики. Позволяет отслеживать прогресс работы.

**Недостатки:**

* Сложность интерфейса. Из-за большого количества функций ClickUp может быть перегружен для простых задач.
* Производительность. При работе с большими командами и объемами данных может снижаться скорость загрузки.

ClickUp — мощный, но сложный инструмент, который требует времени на освоение и настройку.

### 1.1.5 Проблемы существующих решений

Рассмотренные инструменты имеют свои достоинства, однако у всех них есть ряд недостатков:

1. Ограничения в бесплатных версиях. Большинство решений требуют подписки для получения полноценного функционала.
2. Сложность освоения. Некоторые системы перегружены функциями, которые не всегда нужны пользователям.
3. Отсутствие единого подхода. Одни сервисы ориентированы на канбан-доски, другие — на заметки, третьи — на диаграммы Ганта, но редко встречаются универсальные решения, сочетающие простоту и функциональность.

### 1.1.6 Преимущества разрабатываемого веб-приложения

Исходя из анализа существующих решений, разработанное веб-приложение предлагает оптимальный баланс между функциональностью, удобством и доступностью. Его ключевые преимущества:

* Интуитивно понятный интерфейс. Простота в использовании даже для неподготовленных пользователей.
* Комбинация необходимых функций. В отличие от других сервисов, здесь объединены:
* Канбан-доска для удобного управления задачами.
* Система текстовых записей для хранения информации.
* Совместная работа с равными правами участников.
* Бесплатный доступ без критических ограничений. В отличие от большинства конкурентов, проект не требует подписки для использования базового функционала.
* Гибкость и адаптивность. Возможность настройки под нужды конкретного проекта без лишних элементов.

Разрабатываемое веб-приложение решает основные проблемы существующих решений, предоставляя удобный и доступный инструмент для совместной работы над проектами.

## 1.2 Определение потребностей пользователей

Разработка веб-приложения для совместной работы над проектами должна основываться на реальных потребностях пользователей. Несмотря на существование множества инструментов, команды и отдельные пользователи сталкиваются с рядом проблем, которые не решаются или решаются неудовлетворительно. Для создания действительно удобного и эффективного решения необходимо выделить основные запросы целевой аудитории.

### 1.2.1 Анализ целевой аудитории

Основными пользователями веб-приложения являются команды, работающие над проектами в различных сферах. Среди них можно выделить несколько категорий:

**Малые и средние команды разработчиков ПО**

* Необходимость управления задачами и отслеживания прогресса.
* Удобная организация заметок и технической документации.
* Возможность быстрого доступа ко всей информации без сложных настроек.

**Маркетинговые и креативные агентства**

* Совместная работа над проектами с участием дизайнеров, копирайтеров, маркетологов.
* Хранение и организация идей, концепций и материалов.
* Гибкие инструменты планирования и координации.

**Стартапы и небольшие команды**

* Универсальный инструмент без необходимости оплачивать дорогие подписки.
* Возможность быстрой адаптации под меняющиеся задачи.
* Минимальные требования к обучению пользователей.

**Студенты и учебные группы**

* Организация учебных проектов, совместная работа над заданиями.
* Простота и удобство в использовании.
* Возможность работать без необходимости в сложных системах.

### 1.2.2 Основные проблемы пользователей

При анализе существующих инструментов и отзывов пользователей были выявлены ключевые проблемы, с которыми они сталкиваются:

**Сложность освоения**

* Многие сервисы (ClickUp, Asana) имеют сложный интерфейс, перегруженный функциями.
* Новым пользователям требуется значительное время на адаптацию.

**Разрозненность инструментов**

* Для управления задачами используется один сервис (например, Trello), для хранения заметок – другой (например, Notion), для коммуникации – третий.
* Это создает неудобство, так как данные хранятся в разных местах.

**Ограниченный бесплатный функционал**

* Большинство популярных платформ требуют подписки для доступа к продвинутым функциям.
* Это особенно проблематично для небольших команд и стартапов.

**Отсутствие гибкости**

* Многие инструменты предлагают фиксированные способы организации работы, не позволяя настроить процесс под себя.
* Например, Trello ориентирован только на канбан-доски, а Notion – на текстовые записи, но объединенных решений почти нет.

### 1.2.3 Ожидания и требования пользователей

На основе выявленных проблем можно сформировать основные ожидания пользователей от веб-приложения:

**Простота и интуитивный интерфейс**

* Минимальное время на освоение.
* Интуитивно понятные механики работы.
* Доступ к основным функциям без лишних действий.

**Универсальность и гибкость**

* Возможность работать как со списками задач, так и с текстовыми заметками.
* Настраиваемый интерфейс под разные сценарии использования.
* Поддержка командной работы без сложных ролей и прав доступа.

**Бесплатный доступ к базовым функциям**

* Возможность использовать приложение без необходимости оформления подписки.
* Отсутствие критических ограничений, препятствующих полноценной работе.

**Совместная работа в реальном времени**

* Одновременное редактирование задач и заметок несколькими пользователями.
* Автоматическая синхронизация данных.
* Комментарии и обсуждения внутри приложения.

**Кроссплатформенность и удобство доступа**

* Возможность работы через браузер без установки дополнительных программ.
* Поддержка мобильных устройств для удобного использования на ходу.

### 1.2.4 Обоснование необходимости разработки

Учитывая выявленные проблемы и потребности, очевидно, что существует необходимость в инструменте, который:

* Объединяет канбан-доску, списки задач и систему текстовых записей в одном месте.
* Не требует сложной настройки и доступен без подписки.
* Обеспечивает простоту и удобство использования, не перегружая пользователей лишними функциями.
* Позволяет работать совместно в реальном времени без дополнительных плагинов и интеграций.

Существующие решения либо сложны в освоении, либо требуют подписки, либо не объединяют в себе все необходимые функции. Разрабатываемое веб-приложение решает эти проблемы, предоставляя удобный, универсальный и доступный инструмент для совместной работы над проектами.

## 1.3 Составление требований

На основании анализа существующих решений и определения потребностей пользователей формируются требования к разрабатываемому веб-приложению. Эти требования включают функциональные, нефункциональные и технические аспекты, обеспечивающие удобство, надежность и эффективность работы сервиса.

### 1.3.1 Функциональные требования

Функциональные требования определяют основные возможности веб-приложения, которые необходимы для решения задач пользователей.

**Управление задачами и проектами**

* Возможность создавать, редактировать и удалять задачи.
* Поддержка канбан-доски для визуального управления задачами.
* Система приоритетов и сроков выполнения.
* Фильтрация и поиск задач.

**Ведение текстовых записей**

* Возможность создавать, редактировать и удалять текстовые заметки.
* Поддержка форматирования текста (жирный, курсив, списки, заголовки).
* Привязка заметок к задачам.
* Поддержка вложенных заметок для удобной организации информации.

**Совместная работа**

* Возможность добавления пользователей в проект.
* Равные права участников (все пользователи могут редактировать задачи и заметки).
* Комментарии и обсуждения внутри приложения.
* Автоматическая синхронизация данных между участниками.

**Уведомления и напоминания**

* Система уведомлений о новых задачах, изменениях и комментариях.
* Настраиваемые напоминания о дедлайнах.

**Доступность и удобство работы**

* Работа через браузер без необходимости установки.
* Кроссплатформенность (поддержка компьютеров и мобильных устройств).
* Возможность импорта и экспорта данных (например, сохранение задач в CSV или Markdown).

### 1.3.2 Нефункциональные требования

Нефункциональные требования описывают характеристики, влияющие на удобство, безопасность и производительность приложения.

**Простота и удобство использования**

* Минимальное время на освоение.
* Интуитивно понятный интерфейс без лишних элементов.
* Отсутствие сложных ролей и настроек прав доступа.

**Высокая производительность**

* Быстрое открытие страниц и загрузка данных.
* Оптимизированные алгоритмы работы с задачами и заметками.

**Надежность и отказоустойчивость**

* Минимизация потерь данных при сбоях.
* Автоматическое сохранение изменений в реальном времени.
* Возможность резервного копирования данных.

**Безопасность данных**

* Шифрование данных пользователей.
* Защита от несанкционированного доступа.
* Контроль версий для восстановления изменений.

**Масштабируемость**

* Возможность работы с большим числом задач и проектов.
* Поддержка роста числа пользователей без потери производительности.

### 1.3.3 Технические требования

Для реализации веб-приложения необходимо выбрать технологии, обеспечивающие стабильную работу и соответствие функциональным требованиям.

**Стек технологий**

* Backend: Django (Python) – надежный фреймворк с мощными возможностями работы с базой данных.
* Frontend: JavaScript (React/Vue) – для динамичного и удобного пользовательского интерфейса.
* База данных: PostgreSQL – быстрая и масштабируемая СУБД.
* API: REST/GraphQL – для удобного взаимодействия между клиентом и сервером.

**Хостинг и развертывание**

* Серверное развертывание на VPS или облачной платформе.
* HTTPS для безопасного соединения.
* Поддержка автоматического обновления без прерывания работы.

**Поддержка интеграций**

* Возможность интеграции с популярными сервисами (Google Drive, Slack, Telegram).
* Открытый API для взаимодействия с другими инструментами.

### 1.3.4 Итоговые выводы

Составленные требования обеспечивают универсальность, удобство и доступность веб-приложения. Оно будет:

* Интуитивно понятным – без сложных настроек и перегруженного интерфейса.
* Функциональным – с возможностями управления задачами, заметками и совместной работы.
* Доступным – без обязательной подписки и ограничений.
* Надежным и безопасным – с защитой данных и стабильной работой.

Разрабатываемое веб-приложение соответствует современным требованиям и потребностям пользователей, предоставляя удобный и эффективный инструмент для совместной работы над проектами.

# 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

# 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

# **ЗАКЛЮЧЕНИЕ**

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Мартин Роберт К. Чистый код. Создание анализ и рефакторинг. СПб: Питер, 2019. 464 с.
2. Django documentation [Электронный ресурс] // django. URL: <https://docs.djangoproject.com/en/4.2/> (дата обращения 10.03.2024)
3. React documentation [Электронный ресурс] // react. URL: <https://react.dev/learn> (дата обращения 25.03.2024)
4. Мартин Роберт К. Чистая архитектура. Искусство разработки программного обеспечения СПб: Питер, 2021 — 352 с.
5. Стоян Стефанов React.js. Быстрый старт. — СПб: Питер, 2017. — 304 с.