**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

кафедра физики и прикладной математики

**ОТЧЕТ**

**ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ**

Студента Руденко Александры Максимовны

Колледжа инновационных технологий и предпринимательства

Специальности 09.02.07 Информационные системы и программирование

Группы ИПсп-121

Время прохождения практики

с «20» апреля 2025 г.

по «17» мая 2025 г.

*Руководитель от ВлГУ*:

преподаватель КИТП  
(кафедра ФиПМ)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*М.Ю. Кабанова

Владимир, 2025

|  |  |
| --- | --- |
|  | Утверждаю  Зав. Кафедрой ФиПМ\_\_\_\_\_\_\_\_\_\_\_\_  «19» апреля 2025 г. |

З А Д А Н И Е

на преддипломную практику

студентки ***Руденко Александры Максимовны***

4 курса, группы ИПсп-121 специальности 09.02.07 Информационные системы и программирование

Место прохождения практики **кафедра ФиПМ ВлГУ**

Тема ***Разработка веб приложения управления проектами для малых команд и стартапов с функционалом анализа производительности, рейтингом участников***

Последовательность прохождение практики получить задание, пройти инструктаж по технике безопасности и охране труда, выполнить все задания, оформить отчет согласно требованиям.

За время прохождения практики необходимо:

1. Изучить вопросы, предусмотренные программой по всем разделам.

2. Изучить предметную область

3. Проанализировать существующие решения и потребности пользователей

4. Разработать требования к приложению

5. Выбрать и обосновать выбор инструментальных средств разработки приложения

6. Спроектировать приложение: архитектурное проектирование, проект базы данных, макет пользовательского интерфейса, UML диаграммы

7. Разработать приложение: архитектура, база данных, модули, пользовательский интерфейс

8. Протестировать функциональность приложения

9. Подготовить отчет по результатам практики

10. Задание по стандартизации: оформить отчет по результатам практики согласно требованиям к оформлению документации

11. Задание по охране труда, технике безопасности и охране окружающей среды

изучить правила техники безопасности при работе за компьютером, вопросы охраны труда на месте прохождения практики

Отчет по практике составить к 17.05.2025

Задание выдал:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ преподаватель КИТП (кафедра ФиПМ) М.Ю. Кабанова

Задание получил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.М. Руденко

Согласовано: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ преподаватель КИТП А.В. Лоханов

19.04.2025

СОДЕРЖАНИЕ

[Введение 6](#_Toc197704204)

[1 Анализ предметной области 7](#_Toc197704205)

[1.1 Описание предметной области 7](#_Toc197704206)

[1.2 Анализ существующих решений 10](#_Toc197704207)

[1.3 Потребности пользователей и заинтересованных лиц 19](#_Toc197704208)

[2 Постановка задачи 20](#_Toc197704209)

[2.1 Требования к разрабатываемой системе 20](#_Toc197704210)

[2.1.1 Функциональные требование 20](#_Toc197704211)

[2.1.2 Нефункциональные требование 25](#_Toc197704212)

[2.2 Обоснование выбора инструментальных средств 26](#_Toc197704213)

[2.3 Определение этапов реализации проекта 28](#_Toc197704214)

[3 Проектирование приложения 30](#_Toc197704215)

[3.1 Архитектурное проектирование 30](#_Toc197704216)

[3.2 Проектирование базы данных 30](#_Toc197704217)

[3.3 Проектирование интерфейса пользователя 30](#_Toc197704218)

[3.4 Проектирование модулей и компонентов 30](#_Toc197704219)

[3.5 Диаграммы UML 30](#_Toc197704220)

[3.5. Диаграммы UML 31](#_Toc197704221)

[3.5.1 Диаграмма классов 31](#_Toc197704222)

[3.5.2 Диаграмма компонентов 31](#_Toc197704223)

[3.5.3 Диаграмма последовательности 31](#_Toc197704224)

[3.5.4 Диаграмма состояний 31](#_Toc197704225)

[3.5.5 Диаграмма развертывания 32](#_Toc197704226)

[4 Разработка приложения 33](#_Toc197704227)

[4.1 Реализация архитектуры 33](#_Toc197704228)

[4.2 Реализация базы данных 33](#_Toc197704229)

[4.3 Реализация модулей 33](#_Toc197704230)

[4.4 Реализация пользовательского интерфейса 33](#_Toc197704231)

[5 Тестирование 34](#_Toc197704232)

[5.1 Виды тестирования 34](#_Toc197704233)

[5.1.1 Unit-тестирование 34](#_Toc197704234)

[5.1.2 Тестирование пользовательского интерфейса 34](#_Toc197704235)

[5.2 Инструменты и среды тестирования 34](#_Toc197704236)

[5.2.1 NUnit/xUnit/MSTest 34](#_Toc197704237)

[5.2.2 Selenium 34](#_Toc197704238)

[5.3 Результаты тестирования 35](#_Toc197704239)

[5.3.1 Таблицы с результатами тестов 35](#_Toc197704240)

[5.3.2 Анализ выявленных ошибок и их исправлений 35](#_Toc197704241)

[Список использованных источников 36](#_Toc197704242)

[Приложение А 37](#_Toc197704243)

введение

Целью выполнения преддипломной практики является создание веб-приложения для управления проектами, рассчитанного на малые команды и стартапы. Приложение должно содержать функционал для простого контроля и планирования проектов, анализа производительности участников. Для достижения поставленной цели необходимо решить следующие задачи:

* Провести анализ существующих решений для управления проектами и выявить их достоинства и недостатки, особенно в контексте небольшими малыми командами.
* Определить основные требования к веб-приложению, исходя из потребностей целевой аудитории.
* Разработать архитектуру и дизайн интерфейса веб-приложения.
* Реализовать функциональные модули приложения.
* Провести тестирование и оптимизацию приложения для обеспечения его стабильной и эффективной работы.

Для малых команд, которые часто работают в условиях ограниченных ресурсов, использование эффективных инструментов управления проектами может существенно повысить их конкурентоспособность и успешность. Однако, большинство существующих решений на рынке ориентированы на крупные организации и не всегда соответствуют потребностям небольших коллективов. Поэтому создание простого веб-приложения, адаптированного под нужды небольших команд и, является важным и актуальным направлением разработки.

1. Анализ предметной области

1.1 Описание предметной области

Темой дипломного проекта является «Разработка веб-приложения управления проектами для малых команд и стартапов с функционалом анализа производительности, рейтингом участников». Основная цель приложения — предоставить пользователям удобный и простой инструмент для управления задачами и проектами в команде, что позволит повысить эффективность и прозрачность рабочего процесса.

В процессе разработки веб-приложения будут реализованы следующие функциональные возможности:

1. Регистрация и авторизация
   1. Пользователь сможет зарегистрироваться в системе, указав свой адрес электронной почты.
   2. После регистрации пользователю будет доступна функция авторизации с использованием электронной почты и пароля.
   3. В случае если пользователь забыл пароль от учетной записи, есть возможность его восстановить.
2. Работа с проектами.
   1. Пользователь получит возможность создавать новые проекты для управления задачами.
   2. Реализованная функциональность позволит изменять названия, описания, срок и статус созданных проектов, а также удалять их при необходимости.
3. Работа с задачами.
   1. Веб-приложение позволит пользователям добавлять задачи на канбан-доску проекта, а также редактировать их, включая такие параметры, как название, описание, ответственный, срок завершения, статус, приоритет и категория.
   2. Задачи можно удалять при необходимости.
   3. Удаленные задачи будут помещаться в архив, который можно просмотреть и при необходимости восстановить выбранные задачи.
4. Работа с категориями
   1. Пользователь получит возможность добавлять и редактировать категории в проекте.
   2. Каждому пользователю администратор может назначить несколько категорий.
   3. При создании и редактировании задач будет возможность назначить задаче категорию, в зависимости от которой будет подобран список доступных ответственных для задачи.
5. Совместная работа.
   1. Администраторы досок смогут приглашать других пользователей для совместной работы над проектами.
   2. Администратор также может управлять ролями пользователей проекта.
   3. Администратор может передать любому пользователю право на администрирование проекта.
6. Логирование действий
   1. Каждое действие, совершенное в рамках проекта, сохраняется с указанием пользователя выполнившего действия и даты его выполнения.
   2. При необходимости администратор проекта может просмотреть всю историю действий проекта и сохранить в формате .xlsx.
7. Статистика и анализ
   1. Администратору будет доступна страница со статистикой созданных и выполненных задач с возможностью фильтрации по категориям и выбора периода времени для просмотра.

В рамках веб-приложения будут предусмотрены ключевых роли пользователей:

* Администратор доски — это пользователь, который создает проект и управляет им. Он имеет полный контроль над проектом, включая возможность добавления, редактирования и удаления задач. Администратор также отвечает за добавление других участников, настраивая их уровни доступа.
* Менеджер — пользователь, которому администратор доски предоставляет право редактирования. Этой роли доступны функции добавления, редактирования и удаления категорий.
* Сотрудник – пользователь, который может просматривать информацию о проекте, задачах, имеет возможность создавать, редактировать и удалять собственные задачи.

Входные параметры:

* Электронная почта и пароль для регистрации и авторизации пользователя.
* Информация о создаваемых проектах и задачах (названия, описания и т. д.).

Выходные параметры:

* Успешное создание учетной записи, проекта, задачи.
* Сообщения об ошибках при регистрации или авторизации.
* Измененные данные о проектах и задачах после редактирования.

Таким образом, разработка данного веб-приложения обеспечит пользователей мощным инструментом для эффективного управления рабочими процессами и коллективной работы.

1.2 Анализ существующих решений

Jira — один из самых популярных инструментов для управления проектами, особенно в сфере разработки ПО. Предлагает комплексное решение для планирования, отслеживания и управления задачами. Интерфейс Jira представлен на рисунках 1.1 и 1.2.

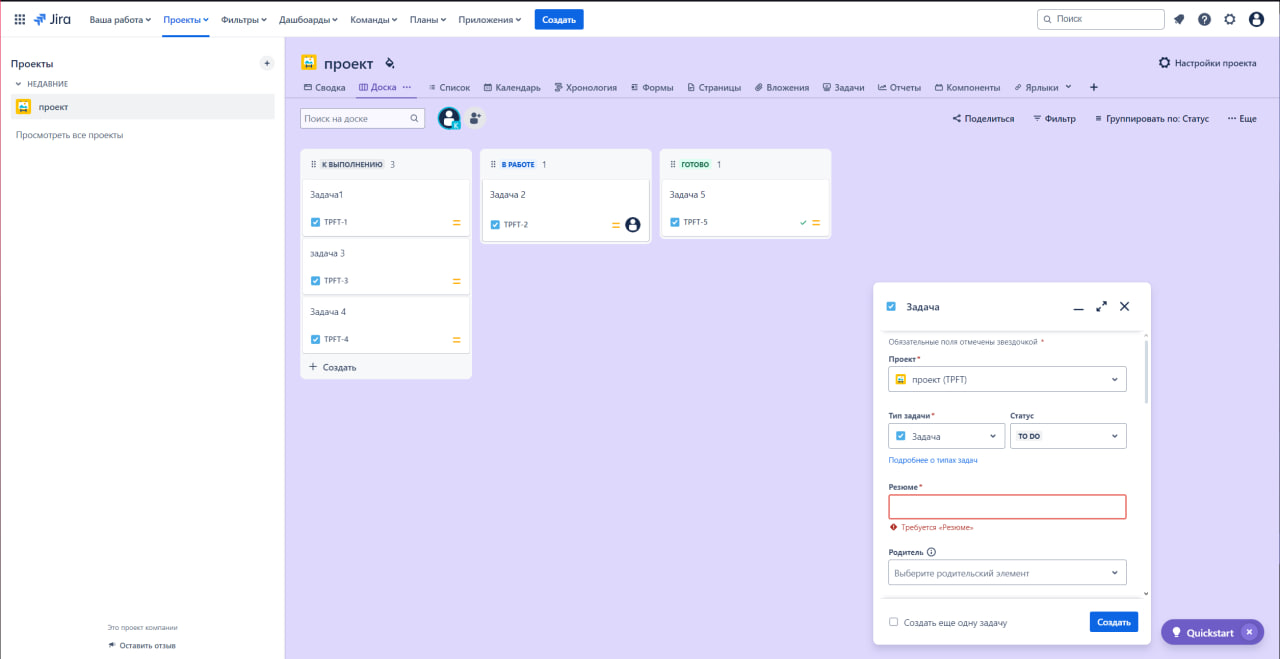


Рисунок 1.1 – доска Jira

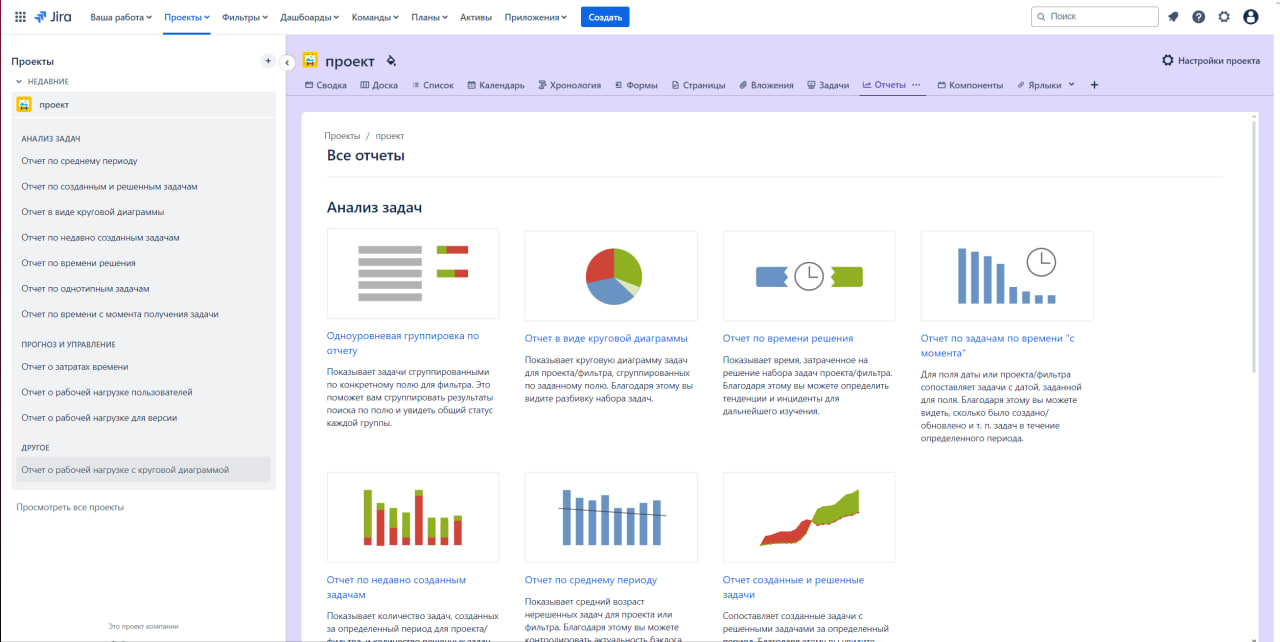


Рисунок 1.2 – отчеты проектов Jira

Плюсы:

* Гибкость: поддерживает Agile методологии (Scrum, Kanban) и позволяет настраивать рабочие процессы.
* Интеграции: Широкий выбор интеграций с другими инструментами (Confluence, Bitbucket, Slack и т.д.).
* Мощные функции отчетности: Доступны отчеты о производительности команды, диаграммы Ганта и другие аналитические инструменты.
* Система прав доступа: позволяет гибко настраивать права пользователей на уровне проекта и задачи.

Минусы:

* Сложность обучения: Интерфейс может быть перегруженным, что требует времени на освоение.
* Цена: на фоне других платформ может показаться дорогой при увеличении числа пользователей.
* Функциональные ограничения в бесплатной версии: Бесплатная версия имеет множество ограничений по функционалу.
* Недоступность в России: Ограничение доступа делает продукт недоступным для пользователей в России.

Функционал:

* Создание и управление задачами.
* Настройка досок Scrum или Kanban.
* Возможность создания пользовательских полей и типов задач.
* Многоуровневое планирование и распределение задач.

Asana это платформа для управления проектами, ориентированная на командную работу и простоту использования. Интерфейс Asana представлен на рисунках 1.3 и 1.4.

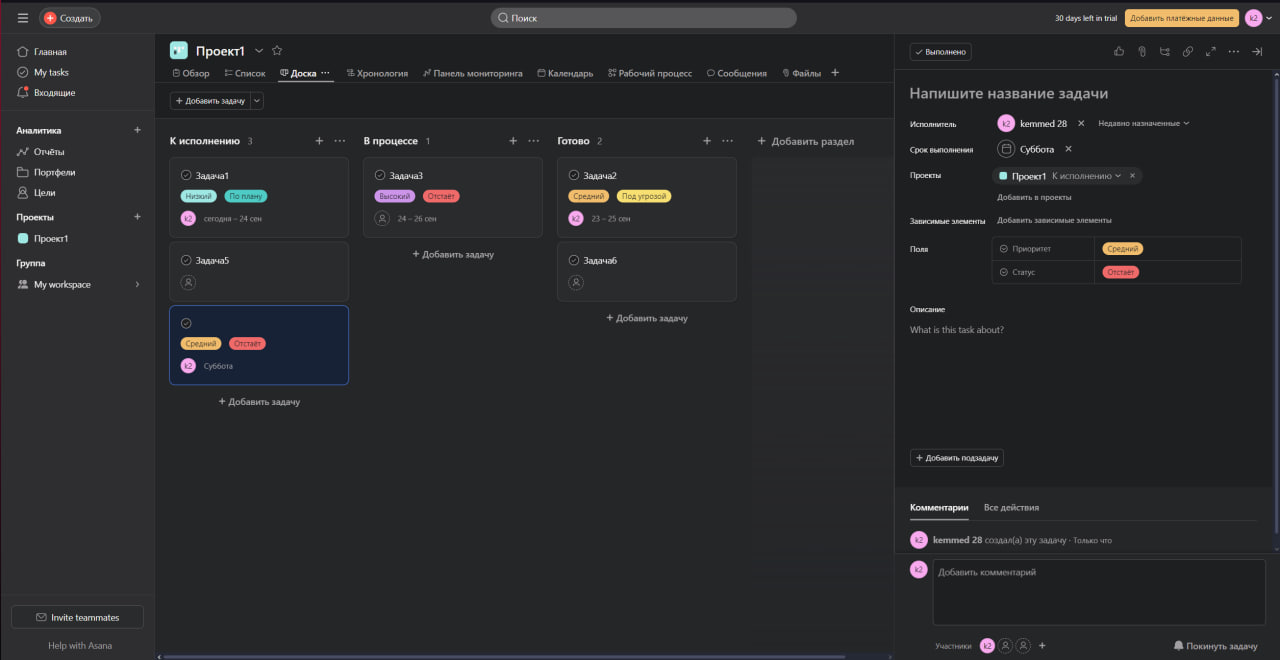


Рисунок 1.3 – доска Asana

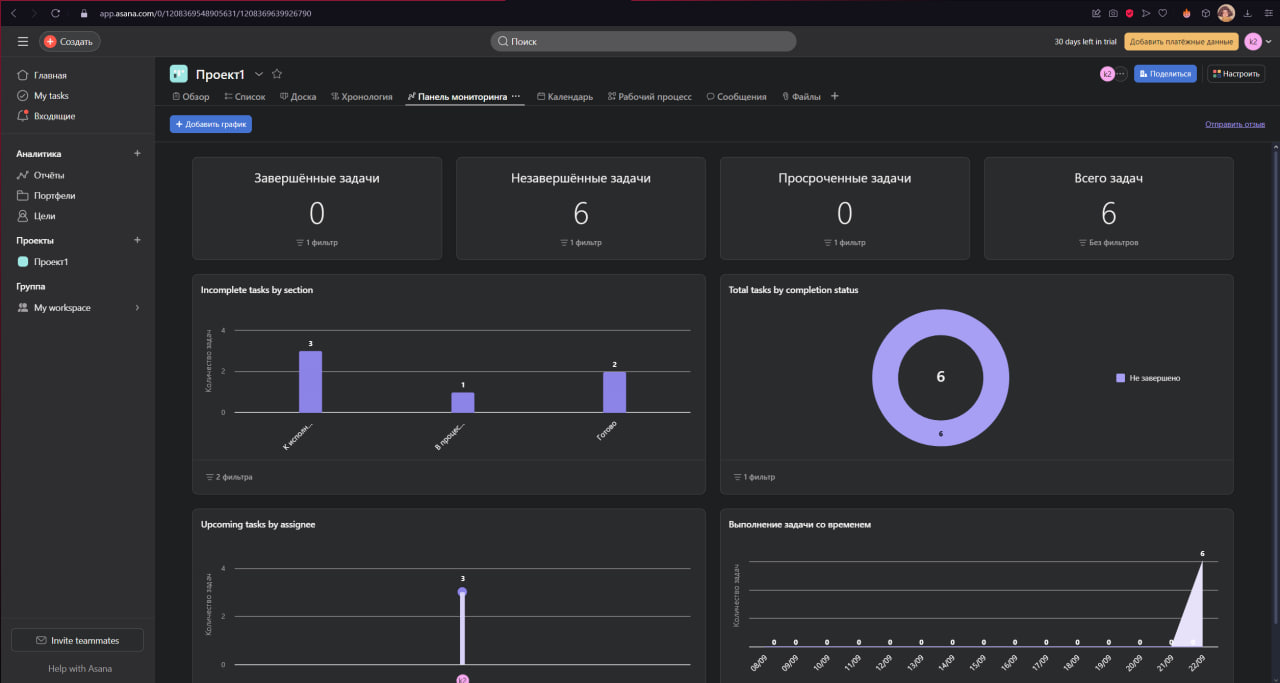


Рисунок 1.4 – статистика проекта Asana

Плюсы:

* Простой интерфейс: Удобный и интуитивно понятный интерфейс, быстрое создание задач и проектов.
* Поддержка нескольких видов представления: Возможность переключения между списком, доской и календарем.
* Интеграция с другими сервисами: хорошо работает с Google Drive, Slack и другими приложениями.
* Шаблоны проектов: позволяют быстро запускать повторяющиеся проекты.

Минусы:

* Ограниченные возможности для сложных проектов: может не подойти для больших разработок с множеством зависимостей и задач.
* Меньше возможностей для кастомизации: Меньшее количество настроек по сравнению с Jira.
* Отсутствие поддержки русского языка: это может стать проблемой для команд, работающих в России.

Функционал:

* Управление задачами и проектами.
* Календарь и временные линии.
* Комментарии и обсуждения внутри задач.
* Отчеты о состоянии проектов.

ClickUp — универсальный инструмент для управления проектами, предлагающий множество функций для различных команд. Интерфейс ClickUp представлен на рисунках 1.5 и 1.6.

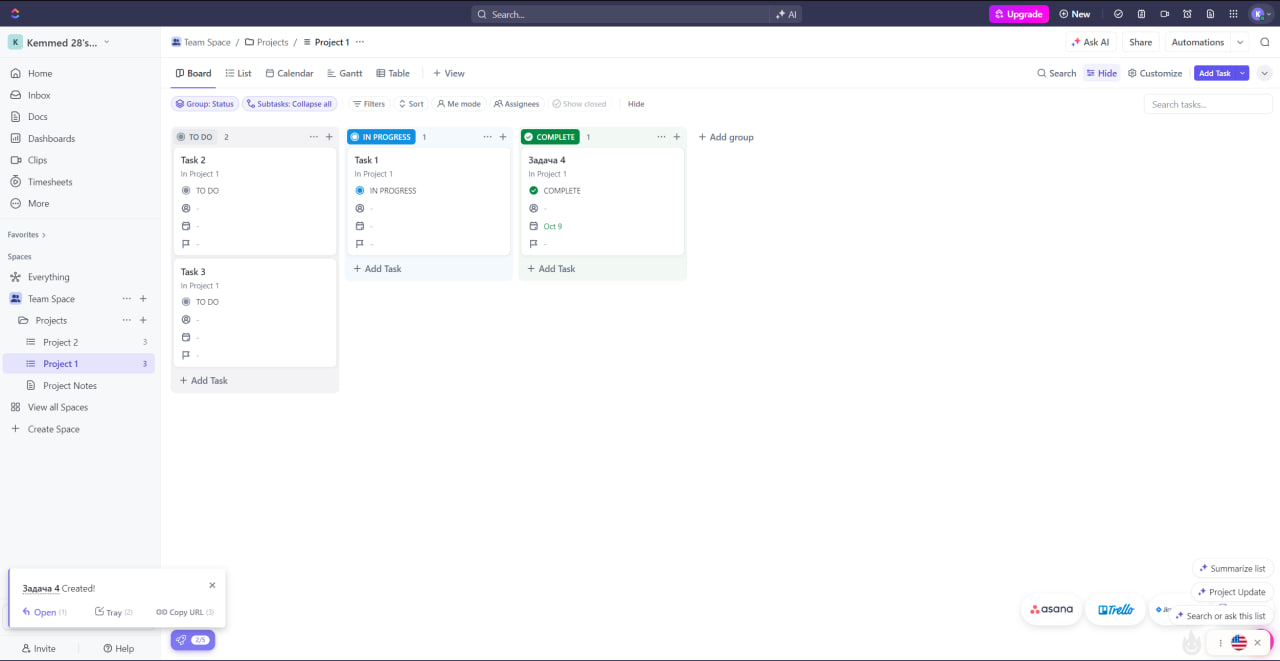


Рисунок 1.5 – доска ClickUp

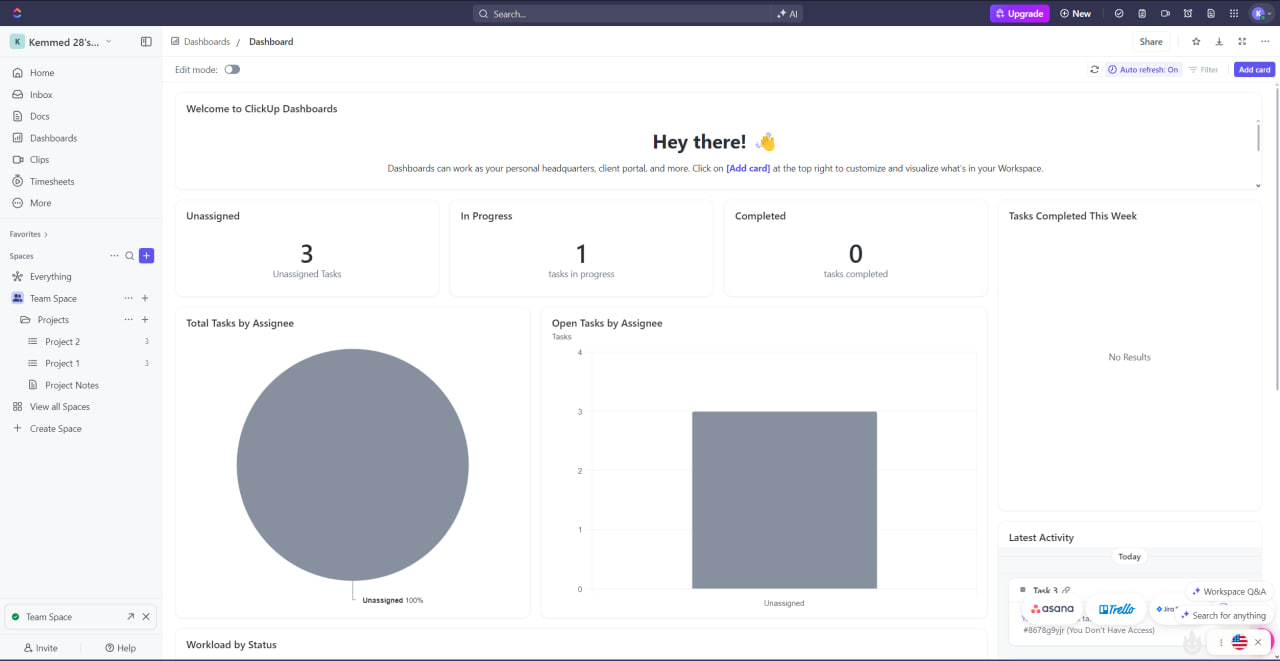


Рисунок 1.6 – статистика проекта ClickUp

Плюсы:

* Многофункциональность: объединяет в себе задачи, документы, цели и расписания в одном инструменте.
* Высокая степень кастомизации: Пользователи могут настраивать рабочие пространства под свои нужды.
* Визуальные инструменты: Диаграммы Ганта, доски Kanban и модули для отслеживания времени.
* Мобильное приложение: Удобное приложение для работы на ходу.

Минусы:

* Сложный интерфейс: Высокая функциональность может затруднять использование, особенно для новых пользователей.
* Ограниченная поддержка: Некоторые функции доступны только в платных версиях.
* Неполная локализация: Интерфейс на русском языке не всегда полный.
* Недоступность в России: Ограничение доступа делает продукт недоступным для пользователей в России.

Функционал:

* Управление задачами и подзадачами.
* Диаграммы Ганта, доски Kanban и различные виды представлений задач.
* Инструменты для совместной работы (комментарии, упоминания).
* Аналитика и отчеты.

Notion — мощный инструмент для организации информации, который сочетает в себе возможности ведения заметок, управления задачами и базы данных. Интерфейс Notion представлен на рисунках 1.7 и 1.8.

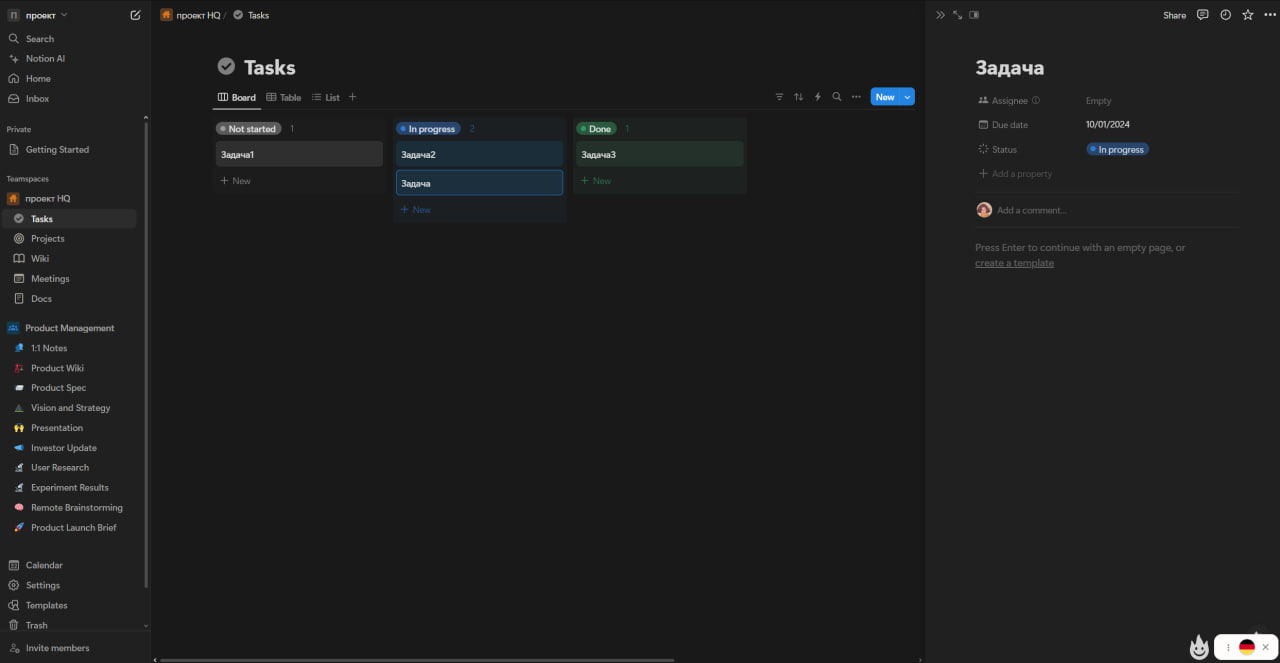


Рисунок 1.7 – доска Notion

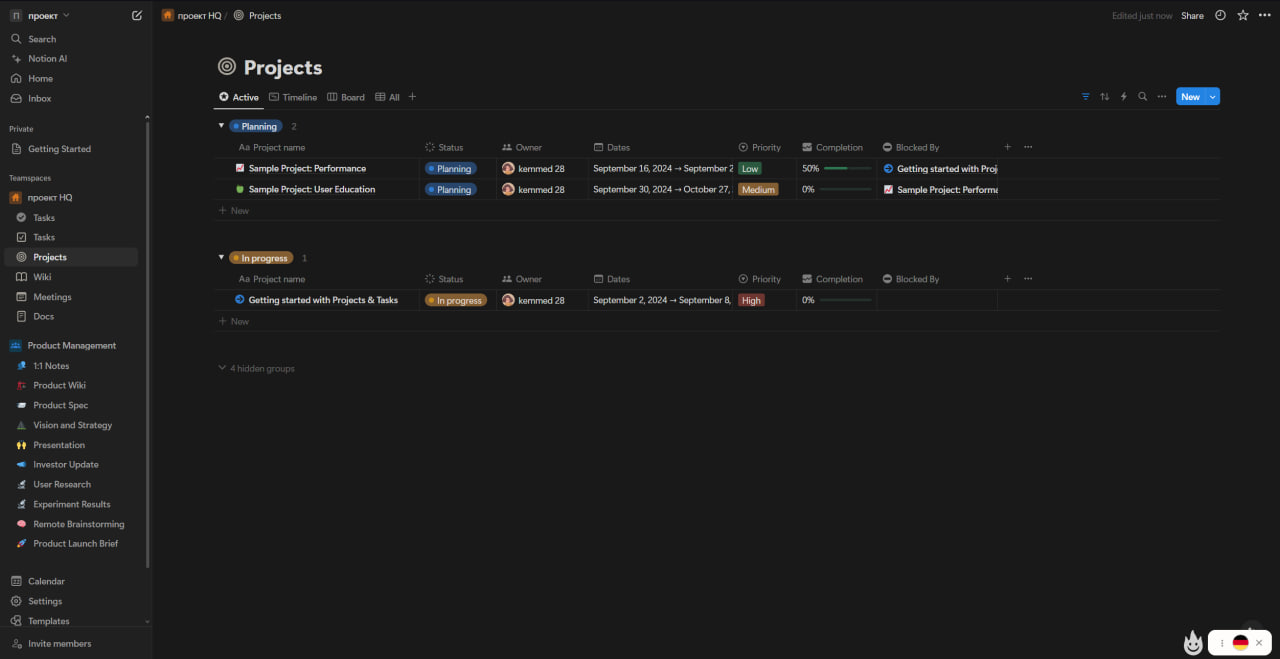


Рисунок 1.8 – просмотр всех проектов Notion

Плюсы:

* Комбинированный подход: позволяет создавать документы, базы данных, таблицы и доски в одном месте.
* Кастомизация: Пользователи могут создавать свои шаблоны и структуры данных.
* Совместная работа: Удобные функции для командной работы, включая комментарии и возможность упоминания пользователей.
* Интуитивно понятный интерфейс: легко адаптируется под разные нужды и стилевые предпочтения команды.
* Доступность в России: не имеет ограничений и доступен для использования.

Минусы:

* Отсутствие оффлайн-доступа: Некоторые функции могут быть недоступны без интернета.
* Ограниченные функции управления проектами: хотя Notion может поддерживать управление задачами, он не всегда подходит для сложных проектов с множеством задач и зависимостей.
* Кривая обучения: поначалу пользователям может быть сложно освоить все возможности программы.

Функционал:

* Создание страниц и связанных баз данных.
* Управление задачами с помощью настраиваемых представлений (доски, списки, календари).
* Документы с поддержкой Markdown.
* Возможность создания диаграмм через интеграции с другими сервисами.

Microsoft To Do — приложение для планирования задач, которое позволяет пользователям управлять своими задачами и напоминаниями. Интерфейс Microsoft To Do представлен на рисунке 1.9.

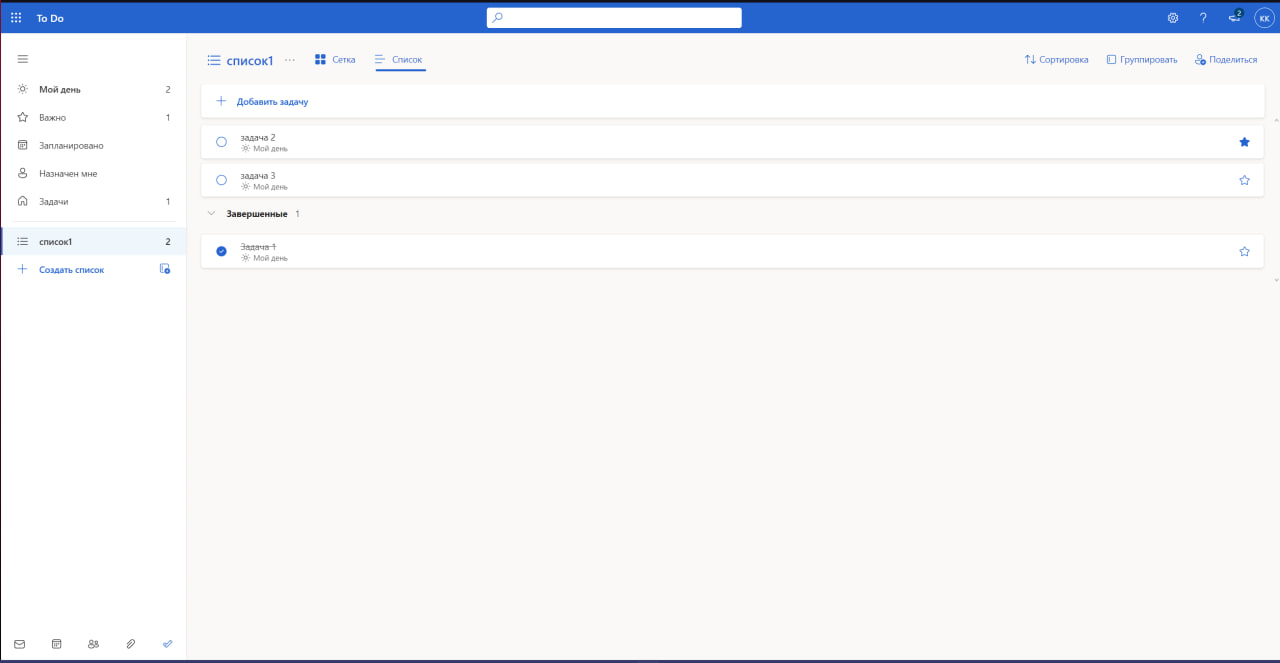


Рисунок 1.9 – доска Microsoft To Do

Плюсы:

* Простота использования: интуитивно понятный интерфейс, идеально подходит для личного использования и небольших команд.
* Интеграция с Microsoft 365: Хорошая интеграция с другими продуктами Microsoft (Outlook, Teams и т.д.).
* Кроссплатформенность: доступно на различных устройствах, включая мобильные платформы.
* Бесплатность: полностью бесплатное приложение без скрытых платежей.

Минусы:

* Ограниченные функции для управления проектами: В основном предназначено для индивидуальных задач, а не для управления командами или проектами.
* Отсутствие поддержки диаграмм и визуальных инструментов: Приложение не предоставляет средств для создания диаграмм Ганта или других визуальных представлений задач.
* Не всегда актуально для больших проектов: для масштабных или многоуровневых проектов может не хватать удобства и гибкости.

Функционал:

* Создание задач с подзадачами и сроками выполнения.
* Организация задач по спискам и категориям.
* Установка напоминаний и сроков.
* Совместная работа над задачами с другими пользователями (в рамках Microsoft 365).

Redmine — это бесплатная система управления проектами и отслеживания ошибок с открытым исходным кодом. Интерфейс Redmine представлен на рисунках 1.10 и 1.11.

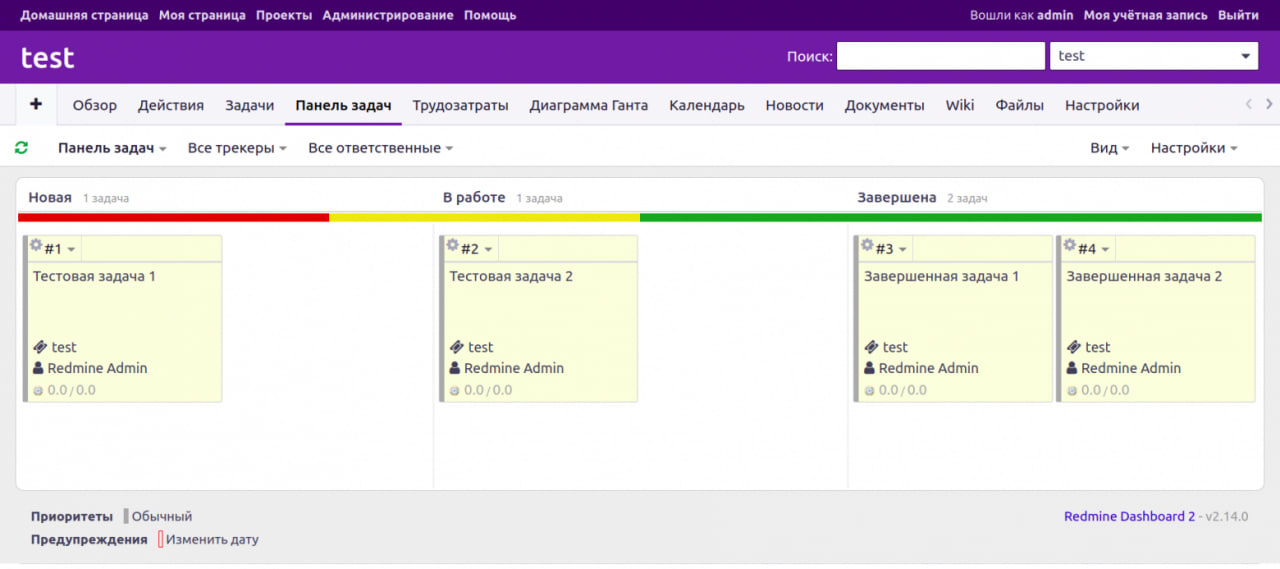


Рисунок 1.10 – панель задач Redmine

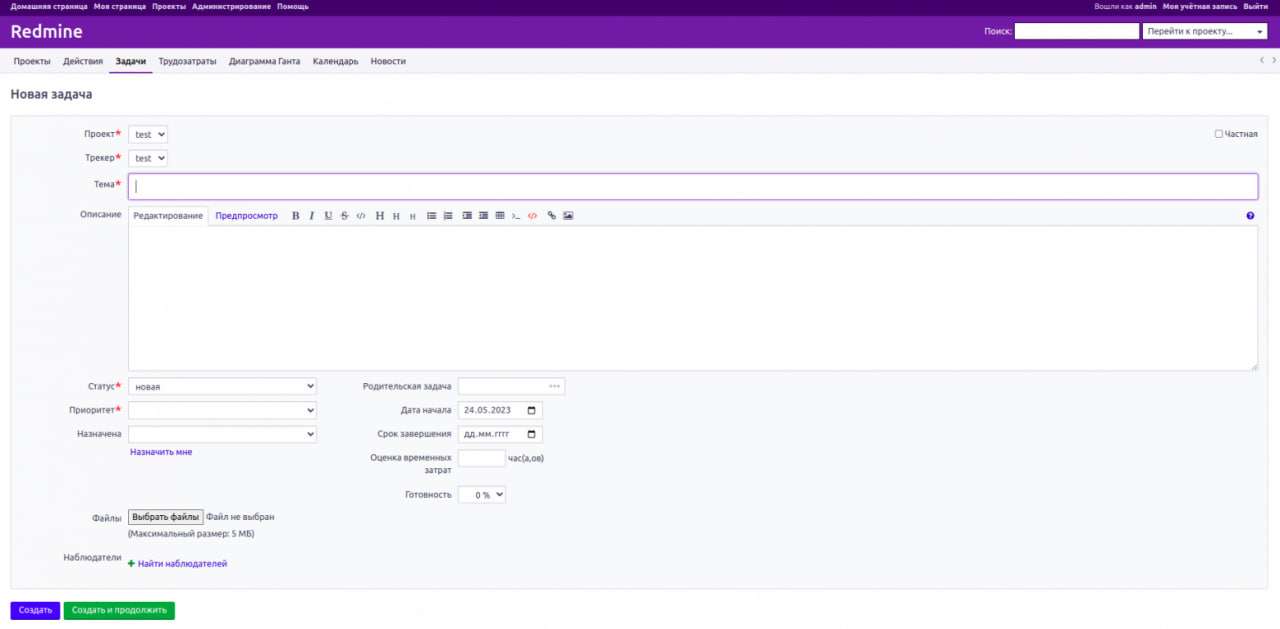


Рисунок 1.11 – задача Redmine

Плюсы:

* Гибкость и расширяемость: позволяет добавлять плагины для расширения функционала.
* Мощные функции управления проектами: Поддержка нескольких проектов, управление задачами, отчетность.
* Поддержка Agile методологий: Возможность работы с задачами в формате Scrum или Kanban.
* Локализация: имеет поддержку русского языка.

Минусы:

* Сложность установки и настройки: требуется сервер для хостинга, что может отпугнуть начинающих пользователей.
* Устаревший интерфейс: Интерфейс может показаться не самым современным по сравнению с конкурентами.
* Требует технических знаний: Настройка и использование могут требовать определенных навыков администрирования.

Функционал:

* Создание и управление проектами и задачами.
* Отслеживание времени и отчетность.
* Возможность создания диаграмм (через сторонние плагины).
* Настройка прав доступа для пользователей и групп.

1.3 Потребности пользователей и заинтересованных лиц

Целевой аудиторией разрабатываемой системы являются:

* Малые команды (до 15 человек).
* Стартапы на начальной стадии развития.
* Фрилансеры, работающие над проектами в составе временных групп.
* Неправительственные организации и учебные заведения, интересующиеся управлением проектами.

В их потребности целевой аудитории входит интуитивно понятный интерфейс, минимально необходимый функционал, позволяющий полноценно работать с задачами, отслеживать их выполнение и иметь возможность анализа производительности команды.

2 Постановка задачи

2.1 Требования к разрабатываемой системе

### 2.1.1 Функциональные требование

Системные характеристики:

СХ-1: разрабатываемая система является веб-приложением.

СХ-2: приложение разрабатывается на языке программирования С# с использованием платформы ASP.Net и модели Razor Pages.

СХ-3: приложение работает с базой данных SQLite.

Пользовательские требования:

ПТ-1: пользователи.

* ПТ-1.1: регистрация в приложении осуществляется с помощью электронной почты с указанием имени, фамилии и пароля.
* ПТ-1.2: при успешной регистрации в приложении на указанный адрес электронной почты приходит письмо с ссылкой для активации аккаунта.
* ПТ-1.3: авторизация в приложении осуществляется с помощью электронной почты и пароля.
* ПТ-1.4: запрос восстановление пароля осуществляется с помощью электронной почты.
* ПТ-1.5: для восстановления пароля необходимо перейти по ссылке, указанной в письме для восстановления пароля, после чего ввести новый пароль и подтвердить его.
* ПТ-1.6: пользователь должен иметь возможность просматривать и изменять информацию об аккаунте в личном кабинете.
* ПТ-1.7: пользователь должен иметь возможность изменять пароль в личном кабинете.
* ПТ-1.8: при успешной авторизации пользователя открывается страница со всеми проектами доступными пользователю.

ПТ-2: проекты.

* ПТ-2.1: приложение позволяет создавать проекты с обязательным указанием названия и срока выполнения. По желанию можно указать описание.
* ПТ-2.2: при создании проекта его статус автоматически устанавливается «В процессе».
* ПТ-2.3: приложение позволяет редактировать уже существующие проекты изменяя их название, описание, статус и срок выполнения.
* ПТ-2.4: в каждом проекте должна быть возможность просмотра добавления задач.
* ПТ-2.5: каждый проект должен иметь журнал действий, архив задач, статистику.
* ПТ-2.6: при просмотре списка всех проектов они должны выделяться цветом в зависимости от их статуса.
* ПТ-2.7: при просмотре списка всех проектов должна быть возможность их фильтрации по статусам.

ПТ-3: категории.

* ПТ-3.1: приложение позволяет создавать категории в проектах.
* ПТ-3.2: приложение позволяет редактировать уже существующие категории в проекте.
* ПТ-3.3: приложение позволяет удалять категории из проекта.
* ПТ-3.4: после удалении категории все задачи с этой категорией станут помечены как задачи без категории.
* ПТ-3.5: приложение позволяет назначать доступные категории пользователям проекта.

ПТ-4: задачи.

* ПТ-4.1: приложение позволяет создавать задачи в проектах с обязательным указанием названия, приоритета, ответственного и сроков выполнения. По желанию можно указать описание и категорию.
* ПТ-4.2: при создании задачи ее статус автоматически устанавливается «В процессе».
* ПТ-4.3: приложение позволяет редактировать уже существующие задачи изменяя их название, описание, статус, приоритет, категорию, ответственного и срок выполнения.
* ПТ-4.3: приложение позволяет удалять уже существующие задачи. После удаления задача перемещается в архив.
* ПТ-4.5: при создании и редактировании задач доступный список ответственных должен фильтроваться по выбранной категории задачи и сортироваться по возрастанию по количеству уже назначенных задач пользователям и дате их завершения.
* ПТ-4.6: приложение позволяет перетаскивать задачи между столбцами. При перетаскивании статус задачи автоматически изменяется.
* ПТ-4.7: при просмотре всех задач на доске они должны выделяться цветом в зависимости от их приоритета.
* ПТ-4.8: при просмотре всех задач на доске должна быть возможность их фильтрации по категориям пользователя и по задачам, назначенным пользователю.

ПТ-5: логирование действий.

* ПТ-5.1: каждое действие в рамках проекта должно сохраняться с указанием пользователя, выполнившего действие и временем выполнения.
* ПТ-5.2: приложение позволяет сохранять всю историю действий в формате .xlsx.

ПТ-6: восстановление задач.

* ПТ-6.1: все удаленные задачи проекта хранятся в архиве.
* ПТ-6.2: приложение позволяет восстановить выбранные задачи из архива.

ПТ-7: пользователи в проекте.

* ПТ-7.1: приглашение пользователя осуществляется с помощью ввода электронной почты пользователя, которого необходимо пригласить.
* ПТ-7.2: при успешной отправке приглашения, на электронную почту приглашенного пользователя приходит письмо с ссылкой для вступления в проект.
* ПТ-7.3: при присоединении к проекту пользователю присваивается роль «Сотрудник».
* ПТ-7.4: в проекте отображаются все состоящие в нем пользователи с указанием их имени и фамилии, категорий и роли.
* ПТ-7.5: в проекте отображаются все пользователи, которые еще не подтвердили присоединение к проекту, и их пригласительная ссылка еще не просрочена.
* ПТ-7.6: приложение позволяет изменять пользователям роли на «Сотрудник» и «Менеджер».
* ПТ-7.7: приложение позволяет передавать права администратора на проект пользователю. При это предыдущий администратор получает роль «Сотрудник».

ПТ-8: статистика.

* ПТ-8.1: приложение позволяет посмотреть процент выполненных задач, а также количество задач по всем статусам и приоритетам.
* ПТ-8.2: приложение позволяет просматривать статистику проекта, в которой указаны созданные и выполненные задачи за определенный период по дням. По умолчанию статистика показывается за текущую неделю.
* ПТ-8.3: приложение позволяет указывать период для просмотра статистики.
* ПТ-8.4: приложение позволяет фильтровать задачи по указанным категориям в статистике.
* ПТ-8.5: приложение позволяет просматривать рейтинг сотрудников, в котором указывается имя и фамилия сотрудника, а также сколько задач было выполнено им в срок и сколько просрочено.

Ограничения:

О-1: пользователи.

* О-1.1: на один адрес электронной почты может быть зарегистрирован только один аккаунт.
* О-1.2: пароль должен быть не менее 8 символов и содержать цифры, заглавные буквы и специальные символы.
* О-1.3: время действия всех ссылок подтверждения действий, приходящих на электронную почту, составляет 24 часа.

О-2: проекты.

* О-2.1: изменять информацию о проекте может только администратор.
* О-2.2: просматривать статистику, журнал действий и архив задач может только администратор проекта.
* О-2.3: администратор не может покинуть проект, пока не передаст права на администрирование другому пользователю.

О-3: категории.

* О-3.1: добавлять, изменять и удалять категории может только администратор и менеджеры проекта.
* О-3.2: назначать категории пользователям может только администратор проекта.

О-4: задачи.

* О-4.1: изменять и удалять задачи может только пользователь, создавший задачу.
* О-4.2: назначать категории пользователям может только администратор проекта.

ПТ-5: пользователи в проекте.

* О-5.1: приглашать, изменять роли, передавать права администрирования и исключать из проекта пользователей может только администратор.
* О-5.2: пригласить в проект можно только тех пользователей, которые зарегистрированы в системе.
* О-5.3: при приглашении пользователя в проект необходимо указывать ту почту, с помощью которой он регистрировался в системе.
* О-5.4: в проекте может быть только один администратор.

### 2.1.2 Нефункциональные требование

В ходе исследования предметной области были выявлены следующие нефункциональные требования:

* Должны быть реализованы различные уровни доступа для различных ролей пользователей (администратор, менеджер, сотрудник).
* Интерфейс должен быть интуитивно понятным и удобным для пользователей с различным уровнем технической подготовки.
* Приложение должно корректно работать в современных браузерах (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).
* Система должна минимизировать вероятность сбоев и свести к минимуму вероятность аварийного завершения работы.
* Пароли должны соответствовать минимальным требованиям сложности.
* Пароли не должны храниться в открытом виде.
* Использование уникальных сессионных токенов для каждой сессии пользователя.
* Сессии должны автоматически истекать через 30 минут неактивности.
* Должна быть возможность принудительного завершения сессии с любой страницы приложения.

2.2 Обоснование выбора инструментальных средств

### 2.2.1 Выбор языка программирования

Для разработки приложения был выбран язык C#.

C# — это мощный, многофункциональный язык программирования, который идеально подходит для разработки веб-приложений. Он поддерживает объектно-ориентированное программирование и обладает богатой экосистемой библиотек и фреймворков, таких как ASP.NET. C# предлагает оптимальное сочетание производительности, простоты использования и поддержки со стороны Microsoft, что делает его идеальным выбором для этого проекта.

Сравнительный анализ альтернатив:

* JavaScript: хорошо подходит для клиентской части, но для серверной разработки требует дополнительных фреймворков, что может усложнить архитектуру.
* Python: производительность может быть ниже по сравнению с C# в определённых сценариях.

### 2.2.2 Выбор СУБД

Для разработки приложения была выбрана система управления базами данных SQLite.

SQLite — легковесная встроенная СУБД, которая не требует установки. Она проста в использовании и предоставляет достаточно функционала для большинства приложений. Для данного проекта SQLite обеспечивает нужную функциональность без лишних накладных расходов.

Сравнительный анализ альтернатив:

* PostgreSQL: более мощная и функциональная, включая поддержку сложных запросов и транзакций, но требует настройки сервера и более высоких затрат на администрирование.
* MySQL: более сложная и менее удобная в настройке и управлении по сравнению с SQLite для небольших проектов.
* MongoDB: подходит для NoSQL-проектов, но может быть избыточной для простых CRUD операций и увеличивает сложность архитектуры.

### 2.2.3 Выбор среды разработки

Для разработки приложения была выбрана среда разработки Visual Studio.

Visual Studio — это мощная среда разработки, специально созданная для работы с C#. Она включает в себя множество инструментов для отладки, тестирования и развертывания приложений, а также обеспечивает полный набор функций для удобной разработки на C#, что делает её лучшим выбором.

Сравнительный анализ альтернатив:

* JetBrains Rider: современная и удобная, но платная.
* VS Code: требует дополнительного конфигурирования для полноценной работы с C# и ASP.NET.

### 2.2.4. Выбор системы контроля версий

Для разработки приложения была выбрана система контроля версий GitHub.

GitHub — универсальная и широко используемая платформа для контроля версий, которая предлагает удобный интерфейс, обширное сообщество и ресурсы для обучения, что делает его предпочтительным выбором для проекта.

Сравнительный анализ альтернатив:

* GitLab: в бесплатной версии ограничивает некоторые функции CI/CD, которые могут быть важны для стартапов.
* Bitbucket: хорошо интегрируется с Jira и другими Atlassian продуктами, но может быть менее популярным среди разработчиков.
* SourceForge: Устаревшая платформа, не поддерживает современные подходы к работе с репозиториями и потеряла популярность.

2.3 Определение этапов реализации проекта

Для разработки дипломного проекта работу необходимо поделить на несколько этапов. Этапы выполнения дипломного проекта представлены в таблице 2.1.

Таблица 2.1 – Этапы реализации проекта

|  |  |
| --- | --- |
| Этап работы | Задачи |
| Подготовительный этап | Определить цели и задачи дипломного проекта  Провести анализ литературы по теме разработки приложений для управления проектами.  Изучить существующие решения и провести их сравнительный анализ.  Составить подробное техническое задание (ТЗ). |
| Проектирование | Разработать архитектуру приложения  Создать прототип пользовательского интерфейса (UI/UX).  Согласовать прототип с научным руководителем.  Спроектировать базу данных и структуры хранения данных. |
| Разработка фронтенда | Реализовать компоненты пользовательского интерфейса.  Настроить маршрутизацию и взаимодействие компонентов.  Имплементировать адаптивный дизайн.  Подключить внешние библиотеки по необходимости. |
| Разработка бэкенда | Разработать контроллеры для взаимодействия с фронтендом.  Организовать обработку запросов и управление данными.  Обеспечить безопасность и авторизацию пользователей. |
| Тестирование и отладка | Провести функциональное тестирование приложения.  Исправить выявленные ошибки и недостатки. |
| Завершение работы | Внести финальные правки в приложение.  Подготовить отчет о выполненной работе.  Защитить дипломную работу перед комиссией. |

Таким образом, каждый этап проекта имеет свои конкретные задачи и цели, что позволяет эффективно подходить к разработке приложения. Соблюдение этих этапов поможет обеспечить последовательное выполнение всех задач и своевременное завершение дипломного проекта.

3 Проектирование приложения

3.1 Архитектурное проектирование

3.2 Проектирование базы данных

3.3 Проектирование интерфейса пользователя

3.4 Проектирование модулей и компонентов

3.5 Диаграммы UML

3.5. Диаграммы UML

### 3.5.1 Диаграмма классов

### 3.5.2 Диаграмма компонентов

### 3.5.3 Диаграмма последовательности

### 3.5.4 Диаграмма состояний

### 3.5.5 Диаграмма развертывания

4 Разработка приложения

4.1 Реализация архитектуры

4.2 Реализация базы данных

4.3 Реализация модулей

4.4 Реализация пользовательского интерфейса

5 тестирование

5.1 Виды тестирования

### 5.1.1 Unit-тестирование

### 5.1.2 Тестирование пользовательского интерфейса

5.2 Инструменты и среды тестирования

### 5.2.1 NUnit/xUnit/MSTest

### 5.2.2 Selenium

5.3 Результаты тестирования

### 5.3.1 Таблицы с результатами тестов

### 5.3.2 Анализ выявленных ошибок и их исправлений

список использованных источников

1. Блог Productstar – Что такое Git и зачем нужен контроль версий [Электронный ресурс]. URL: https://blog.productstar.ru/pochemu-razrabotchiki-polzuyutsya-git/ (Дата обращения 16.12.24);
2. Диаграмма прецедентов – Википедия [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Диаграмма\_прецедентов#:~:text=Диаграмма%20вариантов%20использования%20в%20UML,описать%20систему%20на%20концептуальном%20уровне (Дата обращения 16.12.24);
3. Диаграмма деятельности – Википедия [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Диаграмма\_деятельности (Дата обращения 16.12.24);
4. ASP.Net Core в действии / пер. с анг. Д. А. Беликова. – М.: ДМК Пресс, 2021. – 906 с.: ил.

приложение А

Код главной формы