**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ АСТРАХАНСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение**

**Астраханской области «Астраханский колледж вычислительной техники»**

**Допустить к защите**

**«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.**

**Зам. директора по УМВР**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.В. Расторгуева**

**ДИПЛОМНЫЙ ПРОЕКТ**

**РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ СОВМЕСТНОГО ПРОГРАММИРОВАНИЯ DEVLINK**

**Пояснительная записка**

**АКВТ.09.02.07.ДП00.07ПЗ.2025**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ***Нормоконтролер*** |  |  | ***Разработчик, студент гр. РП-41*** |  |
|  | Ю.А. Кондратенко |  |  | К.С. Зятчин |

|  |  |  |
| --- | --- | --- |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  |  | ***Руководитель*** |
|  |  | Ю.А. Кондратенко |

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

**2025**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Министерство образования и науки Астраханской области**  **Государственное бюджетное профессиональное образовательное учреждение**  **Астраханской области «Астраханский колледж вычислительной техники»**   |  | | --- | | **УТВЕРЖДАЮ** | | **Председатель ЦК 09.02.07 Информационные системы и программирование** | | **\_\_\_\_\_\_\_\_\_\_\_\_ Ю.С. Андрианова** | | **«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.** | | | | | | | | | | | | |
|  | | | **Специальность** | | | 09.02.07 Информационные системы и программирование | | | | |  |
|  | | | **Отделение** | | | Очное | | | | |  |
|  | | | **Группа** | | | РП - 41 | | | | |  |
|  | | |  | | |  | | | | |  |
| **задание** | | | | | | | | | | | |
| **на выполнение дипломного проекта** | | | | | | | | | | | |
|  | | | | | | | | | | | |
| Зятчина Кирилла Сергеевича | | | | | | | | | | | |
| **1** | | **Тема ДП** | | | Разработка платформы для совместного программирования DevLink | | | | | | |
| **утверждена приказом по колледжу №** | | | | | | | 66 | | от | «17» апреля 2025 г. | |
| **2** | | **Срок сдачи студентом законченной ДП** | | | | | | | | «\_\_» июня 2025 г. | |
| **3** | | **Исходные данные к ДП** | | | | |  | | | | |
| Актуальность развития области программирования | | | | | | | | | | | |
| **4** | **Содержание пояснительной записки** | | | | | | | | | | |
| Введение | | | | | | | | | | | |
| 4.1 Исследовательский раздел | | | | | | | | | | | |
| 4.2 Технологический раздел | | | | | | | | | | | |
| Заключение | | | | | | | | | | | |
| Список используемых источников | | | | | | | | | | | |
|  | | | | | | | | | | | |
|  | | | | Руководитель | | | | Ю.А. Кондратенко | | | |
|  | | | | Задание принял к исполнению (дата) | | | | | | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г. | |
|  | | | | Разработчик | | | | | | К.С. Зятчин | |

**Реферат**

Целью дипломного проекта является разработка веб-сайта для совместного кодинга.

По результатам этапов проектирования и разработки был создан сайт, который предоставляет пользователям возможность совместного проектирования, постановки задач и написания кода. Веб-ресурс хранит информацию о созданных репозиториях, командах, задачах.

Проект направлен на упрощение взаимодействия разработчиков, предоставляя инструменты для совместного кодирования, организации задач.

Ключевые фразы: СОВМЕСТНАЯ РАЗРАБОТКА, РЕАЛЬНОЕ ВРЕМЯ, КОМАНДНАЯ РАБОТА, КАНБАН-ДОСКА, ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.

Текст пояснительной записки изложен на 40 страницах, включая 32 рисунка.

Содержание

[Введение 5](#_Toc201432218)

[1. Исследовательский раздел 6](#_Toc201432219)

[1.1 Описание предметной области 6](#_Toc201432220)

[1.2 Актуальность темы 7](#_Toc201432221)

[1.3 Конкурентный анализ 9](#_Toc201432222)

[1.4 Описание целевой аудитории 17](#_Toc201432223)

[1.5 Контекстная диаграмма 19](#_Toc201432224)

[1.6 Диаграмма прецедентов 21](#_Toc201432225)

[2 Технологический раздел 23](#_Toc201432226)

[2.1 Структурная карта сайта 23](#_Toc201432227)

[2.2 Карта пользовательского пути 24](#_Toc201432228)

[2.3 Разработка фирменного стиля 25](#_Toc201432229)

[2.4 Разработка прототипа 27](#_Toc201432230)

[2.5 Используемые технологии для разработки веб-сайта 28](#_Toc201432231)

[2.6 Структура базы данных 29](#_Toc201432232)

[2.7 Описание продукта 31](#_Toc201432233)

[Заключение 39](#_Toc201432234)

[Список используемой литературы 40](#_Toc201432235)

# Введение

В современной IT-индустрии совместная разработка программного обеспечения стала неотъемлемой частью рабочего процесса как крупных компаний, так и небольших команд. Однако существующие решения зачастую перегружены функционалом, что усложняет их использование для локальных проектов или начинающих разработчиков. Кроме того, управление задачами и командная работа часто требуют подключения дополнительных сервисов, что приводит к фрагментации рабочего процесса.

Актуальность проекта «DevLink» заключается в создании унифицированной платформы, объединяющей инструменты для совместного кодирования, управления репозиториями и трекинга задач в одном продукте. Это особенно востребовано в образовательной среде и стартапах, где важны простота и скорость взаимодействия между участниками.

Пояснительная записка к проекту состоит из двух ключевых разделов: исследовательского и технологического.

Исследовательский раздел посвящён анализу текущего состояния рынка области программирования, изучению потребностей целевой аудитории и конкурентных решений. В нём рассматриваются преимущества и недостатки существующих платформ, обосновывается выбор технологий для разработки, а также формируется концепция продукта, направленного на упрощение командной работы.

Технологический раздел описывает процесс реализации проекта: архитектуру веб-приложения, используемый стек технологий, проектирование базы данных и пользовательских сценариев. Особое внимание уделено механизмам совместного редактирования кода, системе управления задачами.

Цель дипломного проекта – создание интуитивно понятной и функциональной платформы, которая сократит время настройки рабочих процессов для разработчиков и повысит эффективность командной работы.

Практическая значимость проекта заключается в предоставлении альтернативного решения для малых и средних команд, которым не требуются сложные корпоративные инструменты, но важно удобство и скорость взаимодействия. «DevLink» может быть использован в образовательных проектах, стартапах и фриланс-разработке, где ключевым фактором является простота интеграции и минималистичный интерфейс.

# 1. Исследовательский раздел

## 1.1 Описание предметной области

Современная разработка программного обеспечения переживает период активной трансформации, когда командная работа становится ключевым фактором успеха проектов. Если раньше программисты могли работать изолированно, то сегодня даже небольшие стартапы требуют слаженного взаимодействия между участниками. Особенно это актуально для образовательных проектов, где студентам необходимо быстро осваивать не только языки программирования, но и инструменты совместной работы.

История развития совместного кодинга началась с появления систем контроля версий, однако настоящий прорыв произошел с развитием облачных технологий. Такие платформы, как GitHub и GitLab, стали стандартом де-факто в индустрии, но их сложность и избыточность функционала часто создают барьеры для начинающих разработчиков. При этом исследования показывают, что более половины студентов IT-специальностей испытывают трудности при первом знакомстве с профессиональными инструментами для командной работы.

В этом контексте особенно важна простота и интуитивность интерфейсов. Современные разработчики, особенно представители поколения Z, ожидают от инструментов мгновенной обратной связи, визуальной понятности и минимального времени на освоение. Именно эти потребности и призван удовлетворить проект «DevLink», предлагая облегченную альтернативу существующим решениям.

Особенностью предметной области является необходимость сочетания нескольких ключевых функций:

* редактирование кода в реальном времени с возможностью совместной работы;
* гибкая система управления задачами;
* простое управление доступом для команд разного размера.

При этом важно отметить, что современные тенденции в разработке ПО смещаются в сторону микрокоманд и распределенной работы. По данным Stack Overflow, более 40% разработчиков сейчас работают в командах от 2 до 5 человек, что требует специально адаптированных инструментов.

Психологические аспекты совместной работы также играют важную роль. Исследования показывают, что визуализация прогресса (например, через канбан-доски) повышает мотивацию участников. Кроме того, возможность быстро получать обратную связь от коллег значительно сокращает время на исправление ошибок.

Технологическая составляющая предметной области постоянно развивается. Если еще пять лет назад для реализации совместного редактирования кода требовались сложные решения, то сегодня появление WebSockets и таких библиотек, как Socket.io, сделало эту задачу значительно проще. Однако сохраняется проблема баланса между функциональностью и производительностью, особенно при работе с большими проектами.

Социальный аспект также важен - современные разработчики хотят не просто совместно работать, но и создавать профессиональные сети. Возможность демонстрировать свои навыки через публичные профили, делиться достижениями и находить единомышленников становится неотъемлемой частью рабочего процесса.

Таким образом, предметная область проекта «DevLink» охватывает не только технические аспекты совместной разработки ПО, но и социально-психологические факторы командной работы, тенденции в IT-образовании и эволюцию инструментов разработчика. Это создает уникальную возможность для создания продукта, который будет одновременно простым в освоении и достаточно мощным для решения реальных задач.

## 1.2 Актуальность темы

В эпоху цифровой трансформации разработка программного обеспечения переживает революционные изменения, где ключевым фактором успеха становится эффективная командная работа. Современные реалии IT-индустрии демонстрируют устойчивый рост распределенных команд и проектов с открытым исходным кодом. По данным GitHub, в 2023 году платформа насчитывала более 100 миллионов разработчиков, при этом 83% из них участвовали в совместных проектах.

Особую актуальность приобретают инструменты для коллективной разработки в контексте стремительного роста IT-образования. Ежегодно университеты выпускают тысячи будущих программистов, которые сталкиваются с проблемой освоения сложных профессиональных инструментов. Исследования Stack Overflow показывают, что 67% начинающих разработчиков испытывают трудности при первом знакомстве с системами контроля версий и инструментами совместной работы.

Современный рынок предлагает множество решений для командной разработки, однако они часто оказываются либо избыточно сложными (GitHub, GitLab), либо чрезмерно специализированными (Trello, Jira). Это создает значительный пробел для образовательных проектов, стартапов и небольших команд, которым требуется простое, но функциональное решение.

Особенно остро эта проблема проявляется в условиях роста популярности распределенных команд. По данным исследования Битрикс24, 51% разработчиков сейчас работают в полностью или частично удаленных командах. Такие коллективы нуждаются в инструментах, которые обеспечивают бесшовную интеграцию процессов разработки, прозрачность рабочих процессов и минимальный порог вхождения для новых участников.

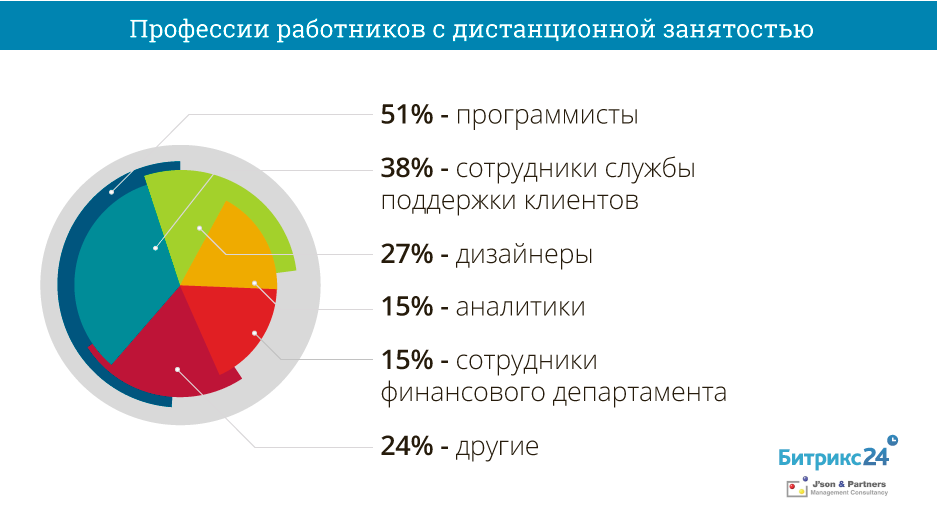


Рисунок 1 – Статистика профессий с дистанционной занятостью

Психологические аспекты совместной работы также играют важную роль. Исследования Калифорнийского университета показывают, что визуализация прогресса (например, через канбан-доски) повышает продуктивность команды на 25-30%. Кроме того, возможность мгновенной обратной связи сокращает время на исправление ошибок в среднем на 40%.

Технологический прогресс последних лет значительно упростил реализацию совместного редактирования кода. Появление WebSockets, развитие облачных технологий и новых фреймворков (например, Operational Transformation для конфликтующих правок) создали благоприятную почву для разработки доступных решений. Однако сохраняется проблема баланса между функциональностью и производительностью, особенно при работе с большими кодовыми базами.

Социальный аспект разработки также претерпел значительные изменения. Современные программисты рассматривают свои профили в профессиональных сетях как цифровое портфолио. Возможность демонстрировать свой вклад в проекты, получать признание коллег и находить единомышленников становится важной составляющей рабочего процесса.

В контексте этих тенденций разработка платформы «DevLink» приобретает особую значимость. Проект предлагает комплексное решение, объединяющее:

* интуитивно понятный интерфейс для командной работы;
* встроенную систему управления задачами;
* возможности для профессионального роста и нетворкинга;
* минимальные требования к технической подготовке пользователей.

На основании анализа рынка и технологических тенденций, экономические факторы однозначно подтверждают стратегическую значимость «DevLink». Устойчивый годовой рост рынка инструментов для разработчиков на 12-15% сочетается с выраженной нишевой возможностью: сегмент решений для образовательных проектов и стартапов демонстрирует наименьшую насыщенность при растущем спросе. Эта конъюнктура создаёт идеальные условия для внедрения специализированного продукта, который решает ключевые боли начинающих разработчиков и малых команд – простоту интеграции, минималистичный интерфейс и консолидацию рабочего процесса.

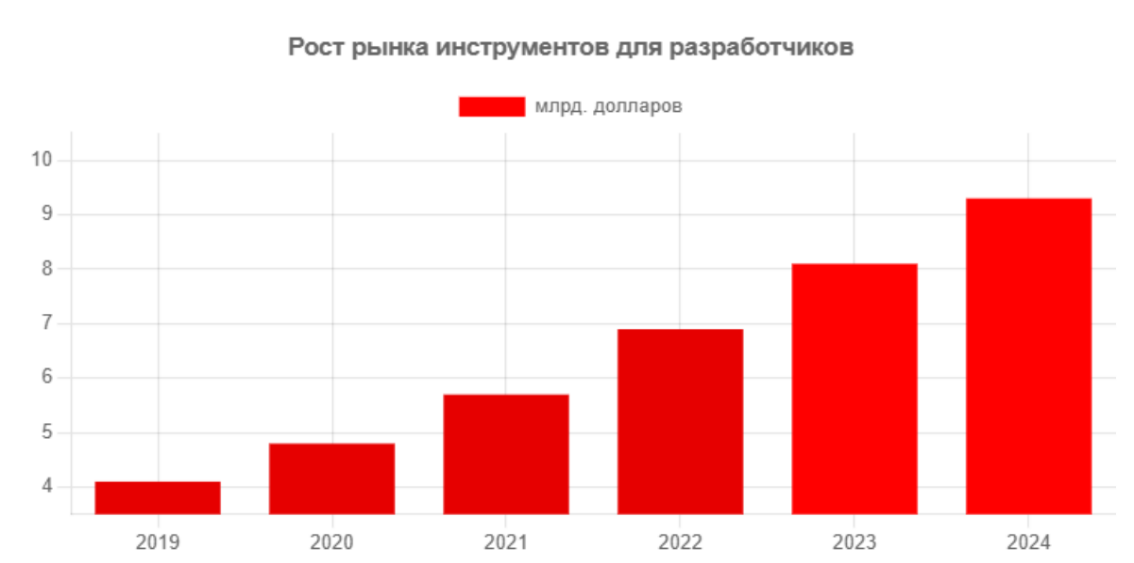


Рисунок 2 – Рост рынка за последние 6 лет

## 1.3 Конкурентный анализ

Конкурентный анализ – это изучение работы конкурентов на рынке. Он подразумевает рассмотрение стратегии конкурентов по ведению бизнеса, преимуществ и недостатков продуктов, объемов продаж, сильных и слабых сторон. Это помогает определить место компании на рынке относительно других фирм.

В рамках анализа была рассмотрена платформа по хостингу IT-проектов и совместной разработки. Сервис насчитывает более 100 миллионов пользователей и 420 миллионов репозиториев. Основу его популярности составляет мощная реализация системы контроля версий Git, дополненная широким набором инструментов для командной работы. Платформа позволяет не только хранить код, но и организовать весь цикл разработки - от постановки задач до автоматизированного тестирования и развертывания. Несмотря на впечатляющие возможности, GitHub остается сложным для освоения инструментом, особенно для начинающих разработчиков и студентов. Интерфейс платформы насыщен технической терминологией и требует понимания множества специфических концепций. Процесс создания нового проекта предполагает выбор лицензии, настройку файла .gitignore и других параметров, которые могут быть непонятны новичкам. Мобильная версия сервиса предлагает урезанный функционал, что ограничивает возможности работы с кодом вне стационарного компьютера.

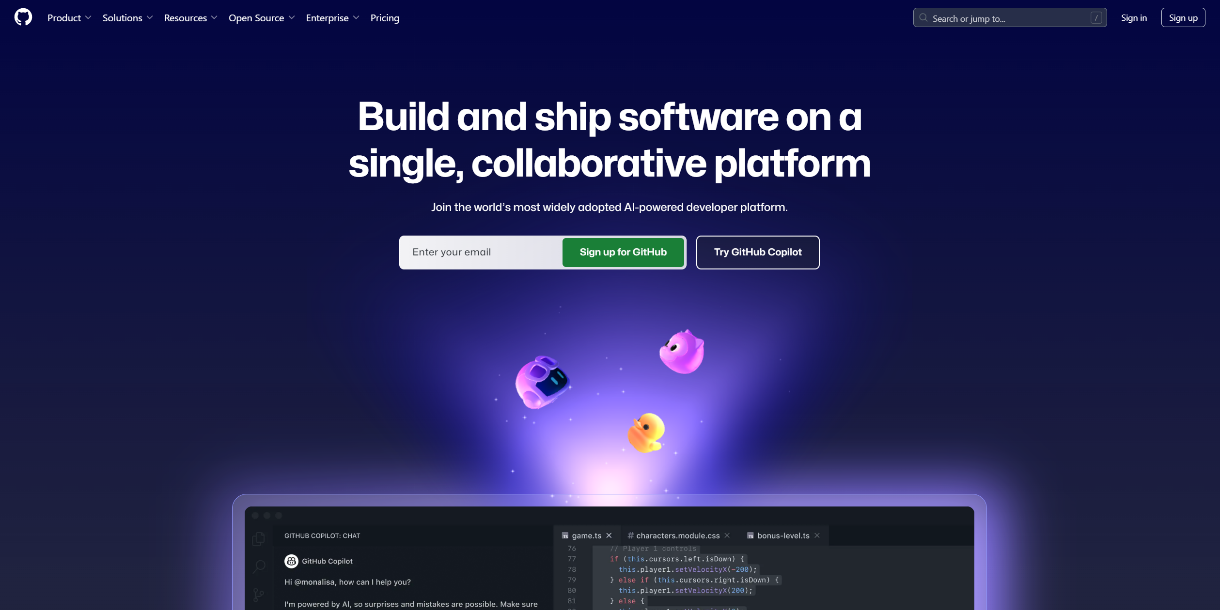


Рисунок 3 – Главная страница ресурса «GitHub»

Визуальная простота обманчива: за минималистичным оформлением скрывается сложная система взаимодействий. Основные элементы интерфейса – списки файлов, история коммитов и окно кода организованы по принципу максимальной информационной плотности. Дизайн мобильной версии существенно упрощен, сохраняя только базовые функции просмотра и управления репозиториями.

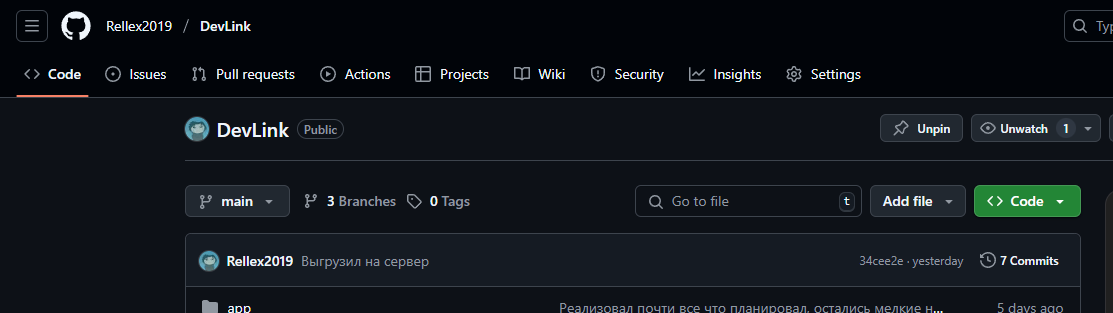


Рисунок 4 – Страница репозитория ресурса «GitHub»

Главный недостаток – визуальная перегруженность для новичков: интерфейс насыщен техническими элементами (хеши коммитов, ветвления, индикаторы статусов), которые профессионалы читают мгновенно, но которые могут отпугнуть начинающих разработчиков. Отсутствие визуальных подсказок и обучающих элементов делает первоначальное знакомство с платформой особенно сложным.

Особенностью GitHub является его фокус на асинхронной модели collaboration. Разработчики работают в своих локальных репозиториях, а затем синхронизируют изменения через систему pull-запросов. Такой подход обеспечивает контроль качества кода, но замедляет процесс совместной работы, особенно в небольших командах, где важна мгновенная обратная связь. Встроенные инструменты управления проектами, такие как Issues и Projects (рисунок 5), предоставляют базовые возможности трекинга задач, но уступают специализированным сервисам по удобству использования.



Рисунок 5 – Страница «Projects» ресурса «GitHub»

Популярность GitHub среди профессионалов неоспорима, но его доминирование на рынке создает пространство для альтернативных решений, ориентированных на специфические потребности. Образовательные учреждения и стартапы на ранних стадиях часто сталкиваются с избыточной сложностью платформы. Это открывает возможности для появления более специализированных сервисов, сочетающих простоту использования с базовыми функциями совместной разработки.

Проведён анализ ещё одного ресурса под названием Jira. Он представляет собой мощную систему управления проектами и задачами, разработанную компанией Atlassian. Платформа широко используется IT-командами для организации рабочих процессов, начиная от Agile-разработки до трекинга багов. Благодаря своей гибкости и интеграции с другими инструментами (Confluence, Bitbucket, Slack), Jira стала стандартом для многих крупных компаний и технологических стартапов.

Дизайн выполнен в корпоративном стиле с преобладанием синих и серых оттенков. Основной упор сделан на списки задач (issues), фильтры и диаграммы прогресса. Содержит множество технических деталей: статусы, приоритеты, метки, связанные коммиты и историю изменений (рисунок 7). Хотя это полезно для разработчиков, менеджерам проектов или дизайнерам такой уровень детализации часто избыточен.

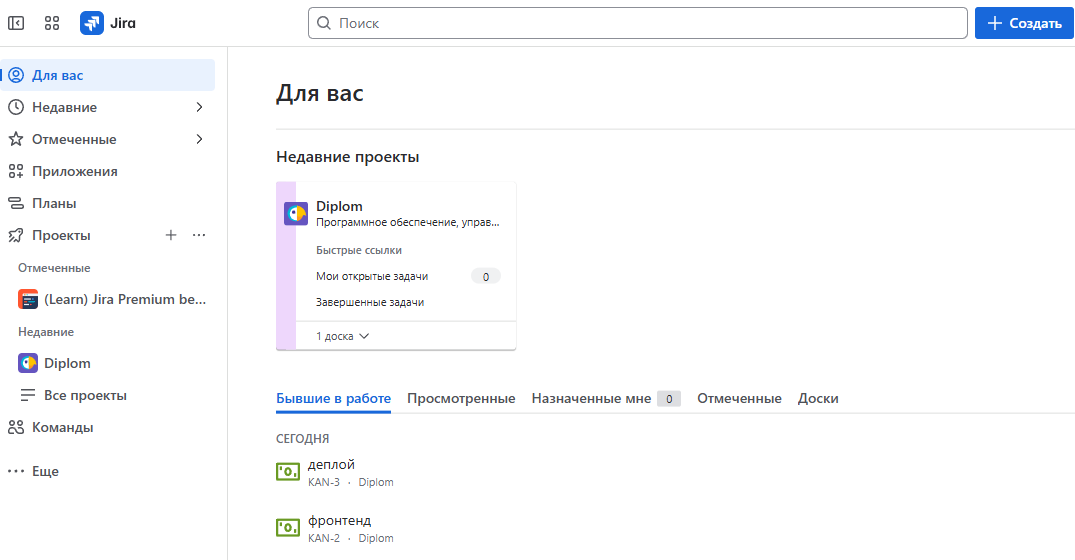


Рисунок 6 – Страница «Kanban» ресурса «Jira»

Несмотря на свою функциональность, Jira отличается высокой сложностью настройки и освоения. Интерфейс перегружен элементами управления (Рисунок 6), что делает его неудобным для небольших команд или новичков. Настройка рабочих процессов требует глубокого понимания системы, а избыточное количество полей и фильтров может замедлять работу. Мобильная версия предлагает ограниченный функционал, что снижает её полезность для удалённых сотрудников.

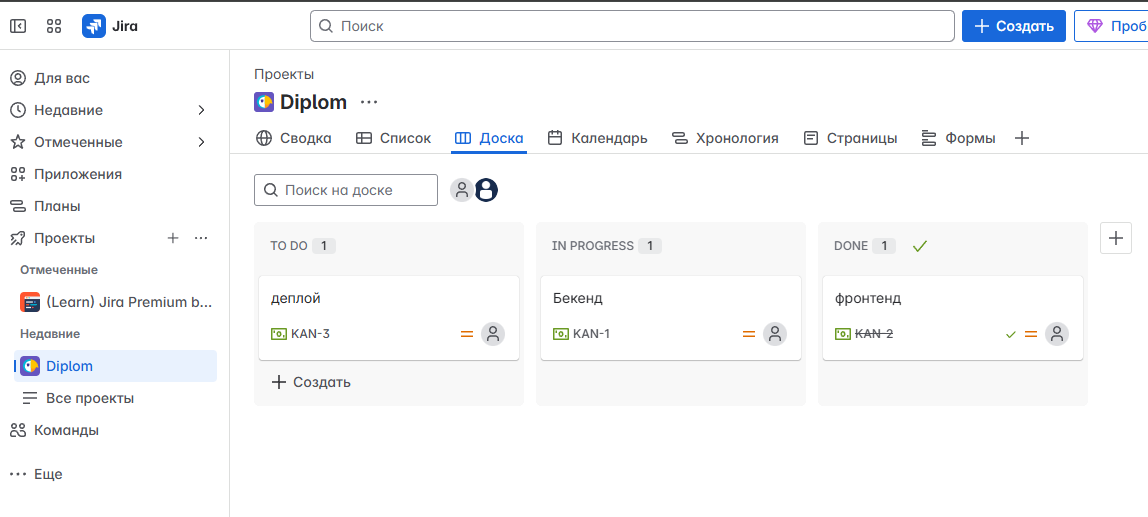


Рисунок 7 – Страница «Kanban» ресурса «Jira»

Несмотря на сложность системы, Jira предоставляет мощные возможности для кастомизации рабочего пространства. Пользователи могут настраивать dashboards под свои нужды, добавляя виджеты для отслеживания ключевых метрик, графиков прогресса или списков задач. Это позволяет каждому члену команды – от разработчика до проджект-менеджера – создать персонализированное рабочее окружение, фокусируясь только на релевантной информации.

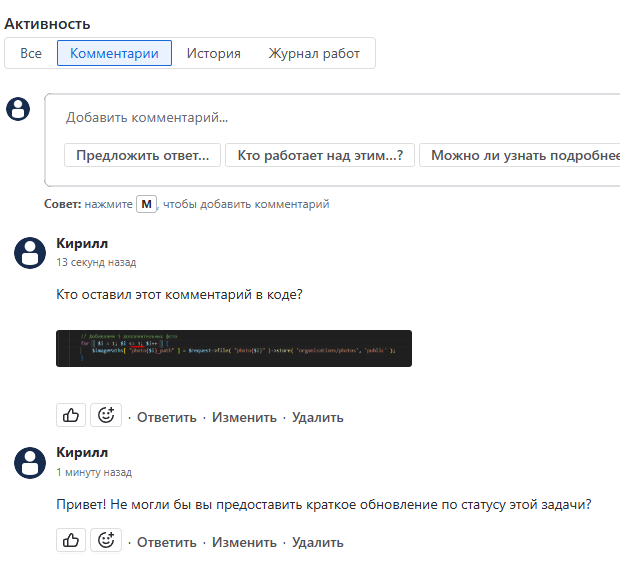


Рисунок 8 – Комментарии к задаче ресурса «Jira»

Командное взаимодействие в Jira организовано через систему комментариев, упоминаний (@mentions) и встроенную интеграцию с Slack. Это обеспечивает прозрачность обсуждений по каждой задаче и сокращает необходимость в дополнительных коммуникационных инструментах. Участники проектов могут оставлять feedback непосредственно в карточках задач, прикреплять скриншоты или логи ошибок, что особенно ценно для распределённых команд (рисунок 7).

Платформа действительно является мощным решением для управления сложными IT-проектами, предоставляя:

* гибкие инструменты для agile-планирования;
* автоматизированные workflow для рутинных процессов;
* глубокую аналитику производительности команды.

Однако вся эта функциональность требует значительных усилий по настройке и обучению. В отличие от более простых альтернатив на подобии Trello, Asana. Jira не предлагает «готовых» решений для небольших команд – каждый процесс приходится настраивать вручную через сложные схемы статусов и переходов. Это делает её избыточной для стартапов или образовательных проектов, где важна простота и скорость внедрения.

Тем не менее, для крупных разработческих команд, особенно работающих по строгим методологиям вроде SAFe, Jira остаётся практически безальтернативным решением. Её экосистема позволяет адаптировать систему под любые, даже самые специфические, процессы разработки.

Так же была рассмотрена облачная среда разработки Replit, которая позиционируется как «браузерная IDE нового поколения». Платформа насчитывает более 20 миллионов пользователей и поддерживает более 50 языков программирования. Основу её популярности составляет концепция кодинга без настройки - возможность начать писать код мгновенно, без необходимости установки дополнительного ПО или настройки среды разработки.

Главное преимущество Replit - минимальный порог входа. В отличие от традиционных IDE и GitHub, здесь не требуется:

* установка компиляторов и интерпретаторов;
* настройка переменных окружения;
* конфигурация систем контроля версий;
* развертывание локальных серверов.

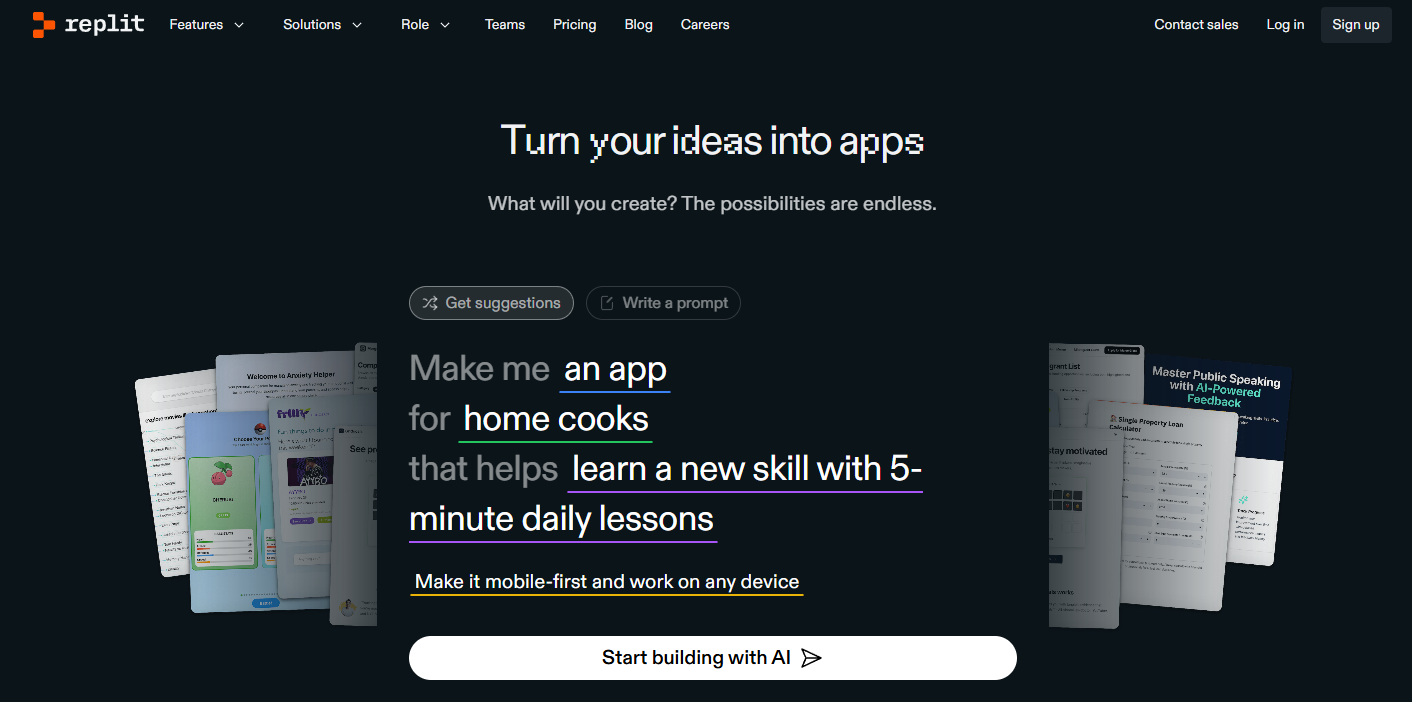


Рисунок 9 – Главная страница ресурса «Replit»

Интерфейс платформы максимально упрощен и интуитивен. Основное рабочее пространство разделено на три ключевые области: редактор кода, консоль выполнения и файловый менеджер (рисунок 10). Такой подход особенно ценен для образовательных целей - студенты могут сосредоточиться на изучении программирования, а не на настройке инструментов.

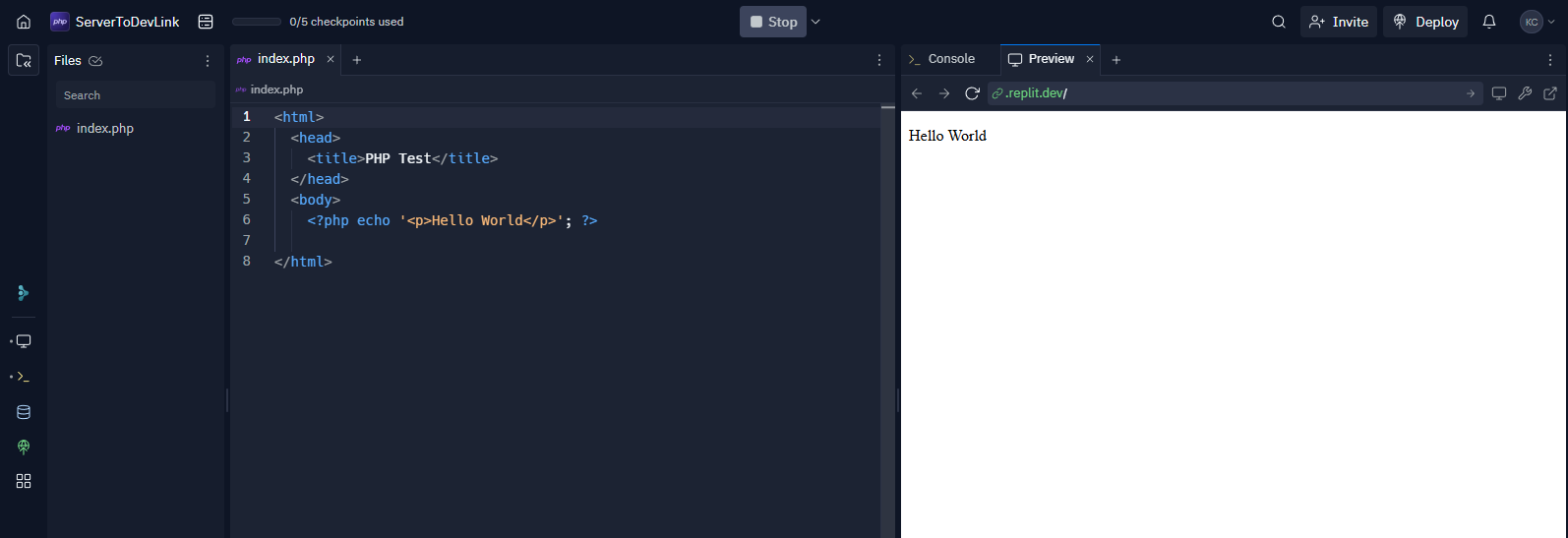


Рисунок 10 – Главная страница ресурса «Replit»

Однако простота имеет и обратную сторону. Replit существенно уступает профессиональным IDE в:

* возможностях рефакторинга кода;
* поддержке сложных проектных структур;
* инструментах отладки;
* производительности при работе с большими проектами;
* нет возможности использовать фреймворк.

Особенностью платформы является встроенная система совместной работы в реальном времени (аналогичная Google Docs). Несколько разработчиков могут одновременно редактировать код, видя изменения друг друга мгновенно. Это выгодно отличает Replit от GitHub с его асинхронной моделью collaboration.

Популярность Replit среди образовательных учреждений и начинающих разработчиков продолжает расти, однако платформа вряд ли сможет заменить профессиональные инструменты для коммерческой разработки. Её ниша – это быстрый старт, обучение программированию и прототипирование идей без сложных настроек. Для серьезных проектов пользователям все равно приходится мигрировать на более мощные решения типа GitHub + VS Code.

В ходе анализа были рассмотрены современные платформы для совместной разработки программного обеспечения. Главной альтернативой на рынке облачных IDE является Replit. Этот сервис не только предоставляет удобную среду для написания кода прямо в браузере, но и активно развивает сообщество начинающих разработчиков, что способствует популяризации программирования среди новичков.

Кроме того, были проанализированы такие платформы, как GitHub и Jira. Эти решения предлагают профессиональные облачные среды разработки, демонстрируя высокую эффективность для коммерческих проектов и обеспечивая разработчиков полным набором инструментов для работы с кодом. Каждая из платформ обладает уникальными техническими особенностями и бизнес-моделями, что позволяет им занимать свои ниши на рынке.

Общая тенденция показывает, что спрос на облачные инструменты разработки продолжает расти. Это подтверждается увеличением числа пользователей, выбирающих подобные решения вместо традиционных локальных IDE. Конкуренция в этом сегменте стимулирует платформы совершенствовать свои продукты: улучшать производительность, расширять поддерживаемые языки программирования, добавлять новые функции для командной работы.

Особенно заметен прогресс в области совместного редактирования кода в реальном времени. Современные платформы предлагают все более совершенные механизмы синхронизации изменений, разрешения конфликтов и коммуникации между разработчиками. Эти инновации делают распределенную командную работу более эффективной и комфортной.

Этот технологический прогресс не только упрощает процесс разработки программного обеспечения, но и способствует демократизации доступа к инструментам создания кода. Теперь для начала программирования достаточно иметь лишь браузер и подключение к интернету, что особенно важно для студентов и разработчиков из стран с ограниченными ресурсами.

В условиях высокой конкуренции и быстрого развития технологий, платформы для совместной разработки должны постоянно эволюционировать, чтобы соответствовать растущим требованиям пользователей. Ключевыми направлениями развития становятся: искусственный интеллект в разработке, улучшенная аналитика кода, глубокие интеграции с популярными сервисами и экосистемами разработчиков.

## 1.4 Описание целевой аудитории

Анализ целевой аудитории нужен для определения потребностей и предпочтений. Он позволяет понять, что именно необходимо вашей целевой группе, и адаптировать продукты или услуги так, чтобы они отвечали этим требованиям. Более того, на основе анализа можно создать более целенаправленные и эффективные маркетинговые кампании, что позволяет сэкономить средства и увеличить рентабельность вложений. Кроме того, знание эмоций и ощущений, которые важны вашей аудитории, позволяет создать продукт или услугу, обеспечивающую максимально положительный клиентский опыт, что способствует росту лояльности клиентов.

Анализ целевой аудитории также помогает в сегментации рынка. Это позволяет разбить аудиторию на разные сегменты и разрабатывать более персонализированные стратегии для каждого из них. Понимание потребностей и предпочтений целевой аудитории может значительно улучшить вашу позицию на рынке в сравнении с конкурентами и создать уникальное предложение (УТП), выделяющее ваш продукт среди аналогов. Второй важный аспект – это уменьшение рисков. Анализ помогает минимизировать риски, связанные с запуском новых продуктов или услуг

Целевая аудитория платформы «DevLink» формируется вокруг начинающих разработчиков и небольших команд, которые ищут простые и эффективные решения для совместной работы над кодом. В отличие от профессиональных сред разработки, ориентированных на опытных программистов, «DevLink» привлекает тех, кто только начинает свой путь в IT или работает над учебными и стартап-проектами.

Основной группой пользователей становятся студенты технических специальностей в возрасте 16-25 лет. Для них важна возможность быстро приступить к работе без сложных настроек и длительного обучения. Они ценят интуитивно понятный интерфейс и встроенные обучающие материалы, которые помогают осваивать основы командной разработки.

Значительную часть аудитории составляют начинающие предприниматели в возрасте 25-35 лет. Эти пользователи ищут инструменты, позволяющие быстро организовать совместную работу над проектом без необходимости настраивать сложные системы контроля версий.

Особую категорию представляют преподаватели программирования и наставники в IT-школах. Они рассматривают «DevLink» как удобную платформу для обучения, где можно создавать шаблоны заданий, отслеживать прогресс студентов и оперативно давать обратную связь. Простота организации групповой работы позволяет сосредоточиться на обучении, а не на технических аспектах настройки среды разработки.

Отдельного внимания заслуживают небольшие стартап-команды, которые только начинают свой путь. Им важно иметь доступ к базовым инструментам разработки без необходимости инвестировать в дорогостоящие профессиональные решения. «DevLink» предлагает им оптимальный набор функций для создания MVP и тестирования идей, экономя время и ресурсы на первоначальном этапе.

Всех этих пользователей объединяет потребность в простом и доступном инструменте для совместной работы над кодом. Они ценят скорость запуска проектов, минимальные требования к техническим знаниям и возможность сосредоточиться на программировании, а не на настройке сложных систем. Именно на эти потребности и ориентирована платформа «DevLink», предлагая сбалансированное решение между функциональностью и удобством использования.

Персона 1: начинающий разработчик.

Алексей, 20 лет, студент 2 курса факультета компьютерных наук.

Ценности: профессиональное развитие и карьерный рост, принадлежность к IT-сообществу, практический опыт работы над реальными проектами

Потребности: простой инструмент для совместной работы над учебными проектами, возможность получать обратную связь от сокурсников, доступ к шаблонам типовых учебных заданий.

Поведение: активно участвует в студенческих IT-мероприятиях, ищет онлайн-ресурсы для обучения программированию, часто работает в командах над курсовыми проектами, делится своими достижениями в профессиональных чатах.

Персона 2: преподаватель программирования.

Дмитрий, 42 года, доцент кафедры программирования.

Ценности: качество образования студентов, практическая направленность обучения, прозрачность учебного процесса.

Потребности: централизованная платформа для студенческих работ, инструменты проверки заданий, возможность демонстрации примеров кода, контроль академической честности.

Поведение: использует проверенные образовательные технологии, отдает предпочтение надежным решениям, участвует в методических объединениях, следит за новинками в IT-образовании.

Персона 3: стартап-разработчик.

Михаил, 30 лет, технический сооснователь IT-стартапа.

Ценности: быстрый запуск продукта, гибкость в управлении проектами, экономия ресурсов на начальном этапе.

Потребности: минимальные настройки для начала работы, встроенные инструменты для совместной разработки, интеграция с популярными сервисами, доступная стоимость.

Поведение: тестирует различные инструменты для MVP, предпочитает облачные решения, участвует в стартап-сообществах, оптимизирует рабочие процессы.

## 1.5 Контекстная диаграмма

Контекстная диаграмма (или диаграмма окружения) показывает взаимодействие системы «DevLink» с внешними сущностями (пользователями, сервисами и сторонними API). Она помогает определить границы системы и ключевые точки интеграции.

База данных репозиториев обеспечивает хранение и версионирование проектов, поддерживая интеграцию с распространенными системами контроля версий.

Облачная инфраструктура предоставляет необходимые вычислительные ресурсы для стабильной работы редактора кода.

Платежные подсистемы интегрированы для обеспечения монетизации дополнительных возможностей, таких как премиум-функции редактора. Все финансовые операции осуществляются с соблюдением требований безопасности и защиты персональных данных.



Рисунок 11 – Контекстная диаграмма

Особое внимание уделено соответствию нормативным требованиям, где стандарт ГОСТ Р 52872-2019 определяет базовые принципы организации совместной работы, а технические требования к интеграции регламентируют взаимодействие с внешними сервисами. Система контроля соответствия непрерывно отслеживает соблюдение установленных стандартов качества и безопасности.

Пользовательские роли разделены на две категории. Гости получают ограниченный доступ к функционалу для ознакомления с возможностями платформы. Зарегистрированные пользователи используют полный спектр инструментов для командной разработки, включая доступ к репозиториям и системе задач.

Визуальное представление платформы реализовано через адаптивные веб-страницы, автоматически подстраивающиеся под различные роли пользователей.

Совокупность этих элементов образует целостную экосистему, где каждый компонент играет определенную роль в обеспечении бесперебойного процесса совместной разработки. Диаграмма наглядно демонстрирует информационные потоки между участниками системы и внешними сервисами, подчеркивая комплексный характер решения.

## 1.6 Диаграмма прецедентов

Диаграмма прецедентов - это один из видов диаграмм в UML (Unified Modeling Language), который наглядно изображает взаимодействия между пользователями (акторами) и системой. Основная цель диаграммы прецедентов заключается в выявлении и описании функциональных требований к системе через сценарии ее использования.

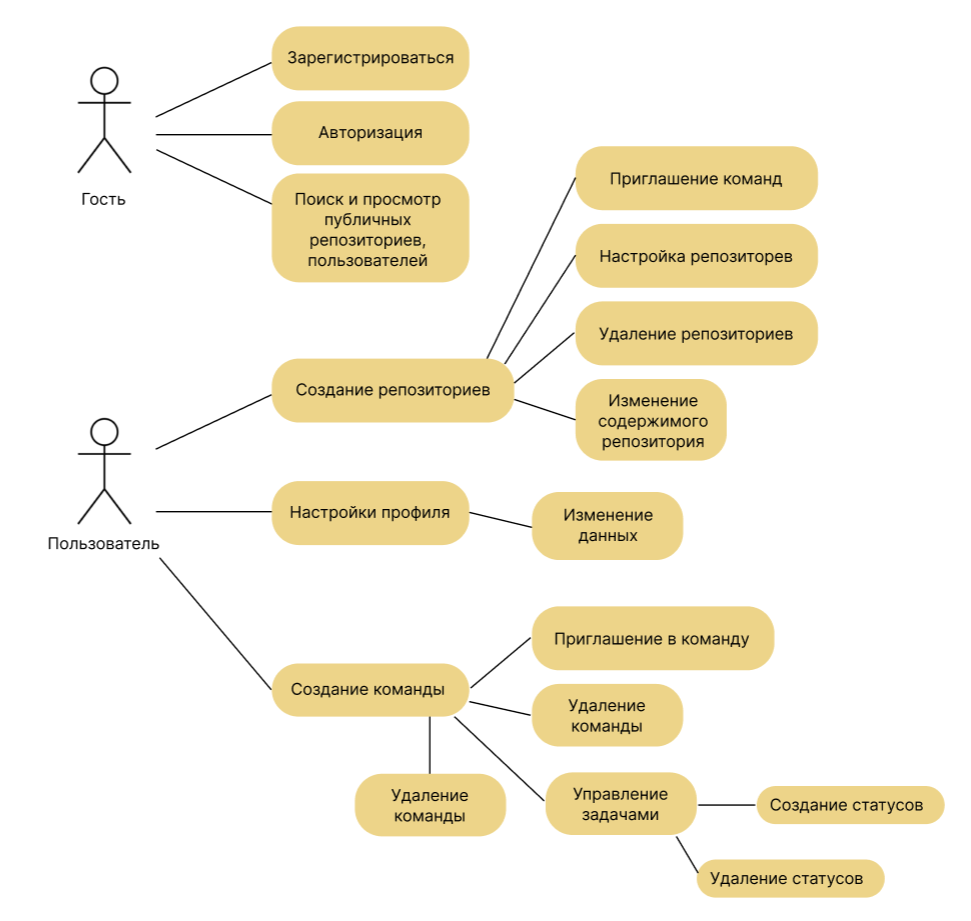


Рисунок 12 – Диаграмма прецедентов

Для новых пользователей, еще не прошедших процедуру регистрации, система предоставляет базовый набор возможностей. Они могут знакомиться с публичными репозиториями и профилями других разработчиков, что позволяет оценить потенциал платформы перед созданием учетной записи. Процесс регистрации и последующей авторизации реализован как отдельный прецедент, обеспечивающий безопасный доступ к расширенному функционалу.

Зарегистрированные пользователи получают полный доступ к инструментарию платформы. Основной акцент сделан на работе с репозиториями - от их создания до тонкой настройки параметров доступа и содержимого. Особое место занимают прецеденты, связанные с командной работой: формирование групп разработчиков, управление членством и распределение прав доступа. Система задач, интегрированная непосредственно в рабочие пространства, позволяет организовать процесс разработки по методологии Kanban через создание и модификацию статусов выполнения.

Все прецеденты связаны между собой через единую систему авторизации и профилирования, что создает целостную экосистему для совместного кодинга. Особенностью модели является глубокое взаимодействие между инструментами управления кодом и задачами, позволяющее отслеживать прогресс разработки в реальном времени.

Такая структура прецедентов отражает современный подход к организации командной работы над программными проектами, где простота взаимодействия сочетается с мощным функционалом для профессиональной разработки.

На основании проведённого исследования проект «DevLink» подтверждает свою стратегическую значимость как специализированное решение для образовательных проектов и малых команд разработчиков. Современный рынок демонстрирует устойчивый рост спроса на инструменты совместной работы (12-15% ежегодно), но сохраняет пробел в сегменте простых и интуитивных платформ. Анализ конкурентов GitHub, Jira, Replit выявил их избыточную сложность для новичков или функциональную ограниченность, что создаёт идеальные условия для внедрения «DevLink».

Ключевое преимущество проекта – сочетание минималистичного интерфейса, консолидации рабочего пространства (совместное редактирование кода плюс управление задачами) и ориентации на потребности начинающих разработчиков. Целевая аудитория (студенты, преподаватели, стартапы) получит инструмент с низким порогом входа, решающий проблемы интеграции и визуальной перегруженности. Технологическая база из WebSockets, облачная инфраструктура обеспечивает реализацию этих задач при соблюдении требований к производительности и безопасности.

# 2 Технологический раздел

## 2.1 Структурная карта сайта

Структурная карта сайта представляет собой наглядную схему организации страниц ресурса по тематическим блокам и подразделам. Она отображает иерархию и связи между различными частями сайта, облегчая пользователям быстрый доступ к нужной информации или конкретной странице.

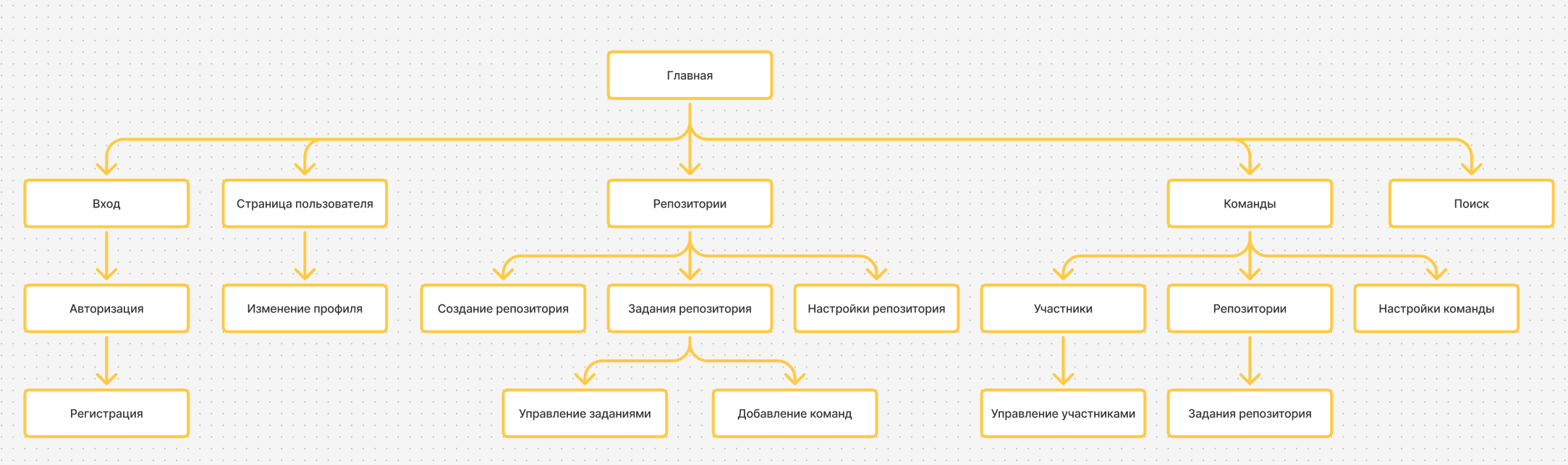


Рисунок 13 – Карта сайта

Платформа «DevLink» предлагает четкую и логичную структуру навигации, разработанную для удобства командной разработки. Главная страница служит центральным узлом, откуда пользователи могут получить доступ ко всем ключевым разделам системы.

Основной функционал сгруппирован в интуитивно понятные разделы. В блоке авторизации реализованы процессы входа в систему и регистрации новых пользователей, что открывает доступ к персонализированным возможностям платформы. После успешной аутентификации разработчики получают доступ к управлению своим профилем, где могут настраивать персональные данные и предпочтения работы.

Центральное место в архитектуре занимает раздел работы с репозиториями. Здесь пользователи могут создавать новые проекты, настраивать их параметры и управлять структурой файлов. Особое внимание уделено инструментам командной работы - отдельный раздел позволяет формировать группы разработчиков, приглашать участников и распределять права доступа.

Система задач интегрирована непосредственно в рабочие пространства репозиториев, обеспечивая seamless-взаимодействие между написанием кода и управлением проектом. Пользователи могут создавать, отслеживать и распределять задания без необходимости переключения между разными интерфейсами.

Поисковая система объединяет все разделы платформы, позволяя быстро находить нужные репозитории, команды или участников. Настройки каждого раздела доступны в соответствующих подменю, что сохраняет чистоту основного интерфейса при сохранении всего необходимого функционала.

Такая структура отражает профессиональный подход к организации рабочего пространства для разработчиков, где простота навигации сочетается с мощными инструментами для совместной работы. Все элементы карты сайта взаимосвязаны, создавая целостную среду для эффективного совместного программирования.

## 2.2 Карта пользовательского пути

Карта пользовательского пути – это визуальное представление процесса, который проходит пользователь при взаимодействии с продуктом или услугой. Эта диаграмма помогает в понимании того, как пользователи движутся по различным этапам, начиная с первого знакомства и заканчивая целевыми действиями, такими как покупка или подписка.

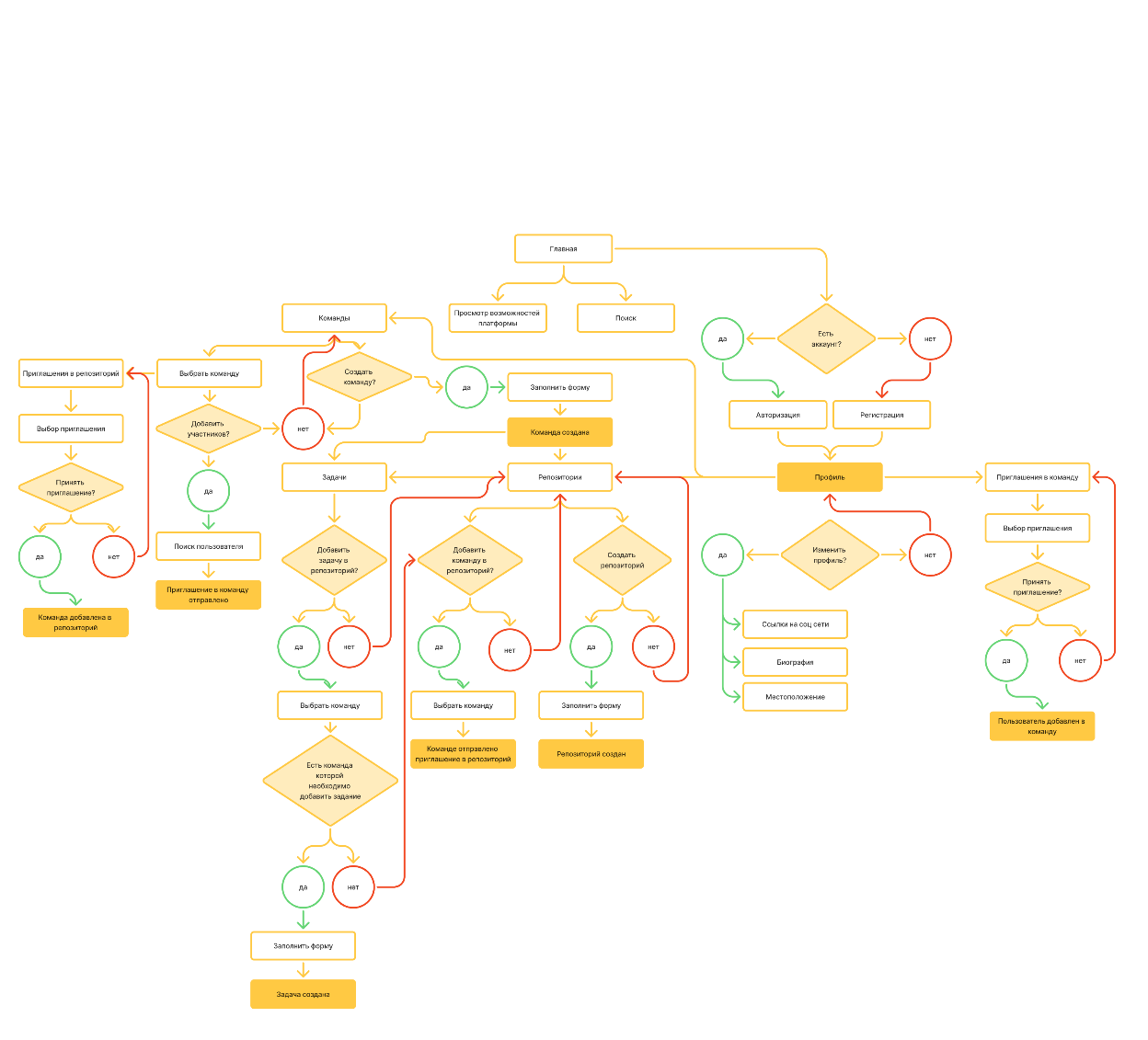


Рисунок 14 – User Flow диаграмма

Путь пользователя на сайте начинается с главной страницы, которая выступает в роли центрального информационного узла. При первом посещении пользователь сталкивается с кратким описанием миссии проекта, основными направлениями работы, которые приглашают участвовать в удобной системе совместной разработки.

Неавторизованному пользователю доступен просмотр предоставляемых преимуществ, а также поиск пользователей, репозиториев.

После авторизации открывается возможность создания репозитория с кодовой базой, добавление в этот репозиторий участников – команды. А также дается возможность управления задачами команд, участвующих в разработке.

Кроме того, можно создать команду и добавлять в нее участников – пользователей. Принимать приглашения от репозиториев и создавать задачи пользуясь удобной Kanban доской.

Появляется доступ к публичному профилю, который можно редактировать, добавляя информацию о себе.

## 2.3 Разработка фирменного стиля

Фирменный стиль представляет собой набор визуальных и текстовых элементов, которые помогают создать уникальное восприятие компании или проекта. Он включает в себя логотип, цветовую палитру, шрифты, графические элементы и общий дизайн, которые формируют образ бренда и помогают отличать его от конкурентов.

Фирменный стиль «DevLink» создает современный технологичный образ платформы для совместной разработки, сочетая профессиональную строгость с дружелюбным подходом к пользователю. Основой визуальной идентичности стал насыщенный солнечно-желтый цвет (#EDB200), который символизирует интеллектуальную энергию, творческий потенциал и ясность мысли - ключевые качества для продуктивной работы с кодом. Этот жизнерадостный оттенок создает ощущение вдохновения и мотивации, что особенно важно для начинающих разработчиков.

Глубокий графитовый цвет (#101112) добавляет в палитру нотки профессионализма и технологичности, ассоциируясь с надежностью и стабильностью работы платформы. Такой контраст теплого и холодного создает динамичный, но сбалансированный визуальный образ, отражающий суть «DevLink» - мощный инструмент для серьезной работы, представленный в доступной форме.

Шрифт Montserrat был выбран за его исключительную читаемость и современную геометричность. Четкие формы букв без засечек идеально передают техническую точность программирования, сохраняя при этом дружелюбный характер. Особое внимание уделено оптимальному кернингу и межстрочному интервалу, что обеспечивает комфортное чтение кода и документации в интерфейсе.

Графические элементы стиля включают плавные переходы между цветами и деликатные тени, создающие эффект глубины интерфейса без перегруженности. Логотип сочетает в себе символическое изображение совместного редактирования с динамичной формой, передающей движение и взаимодействие.

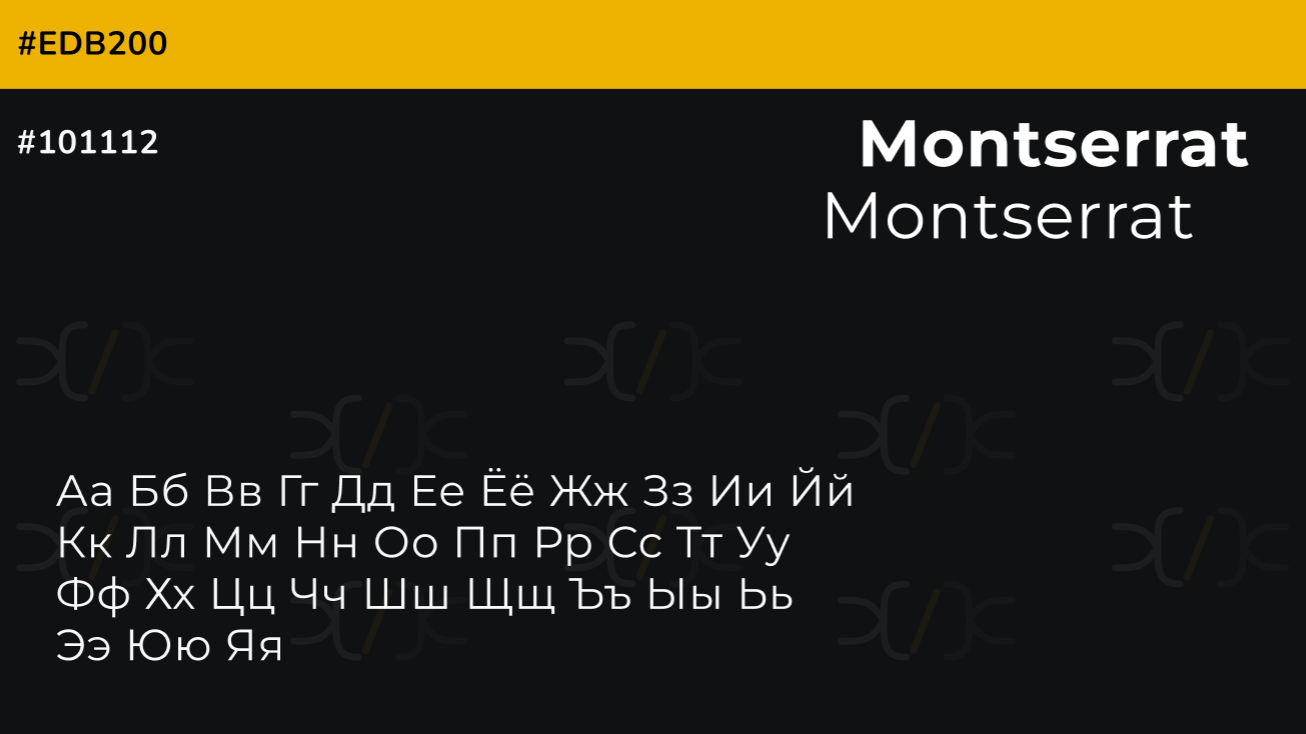


Рисунок 15 – Цветовая палитра и шрифт

Такой фирменный стиль формирует узнаваемый образ платформы, который одновременно вдохновляет на творчество в программировании и внушает доверие как профессиональный инструмент. Визуальная идентичность «DevLink» отражает основные ценности продукта: инновационность, доступность и эффективность командной работы, создавая эмоциональную связь с разработчиками разного уровня подготовки.



Рисунок 16 – Логотип организации

Главным визуальным элементом логотипа являются две стилизованные фигурки, символизирующие двух разработчиков, объединённых символом кода – косой чертой «/». Этот знак выступает метафорой связи и взаимодействия, подчеркивая идею сотрудничества и совместной работы над проектами.

Цветовая гамма логотипа выдержана в контрастных оттенках белого, серого и яркого жёлтого цвета, что придаёт дизайну живость и динамичность. Жёлтый цвет акцентирует внимание на слове «Link» – символизируя связь, коммуникацию и объединение усилий участников платформы.

Логотип гармонично отражает концепцию проекта, ориентированного на создание комфортной среды для совместной работы разработчиков, обмена знаниями и эффективного управления командами и репозиториями. Он легко запоминается и визуально передаёт основную идею платформы – объединение людей через код.

## 2.4 Разработка прототипа

Этап создания прототипа играет ключевую роль в разработке фирменного стиля, так как именно на этом этапе можно визуализировать идею и концепцию бренда. Прототип позволяет проверить визуальное восприятие. Это также упрощает внесение изменений, так как дает возможность быстро корректировать дизайн, экономя время и ресурсы на более поздних этапах разработки. Кроме того, прототипы позволяют оценить, насколько удобно пользователю взаимодействовать с брендом и его элементами, что важно для формирования положительного имиджа.

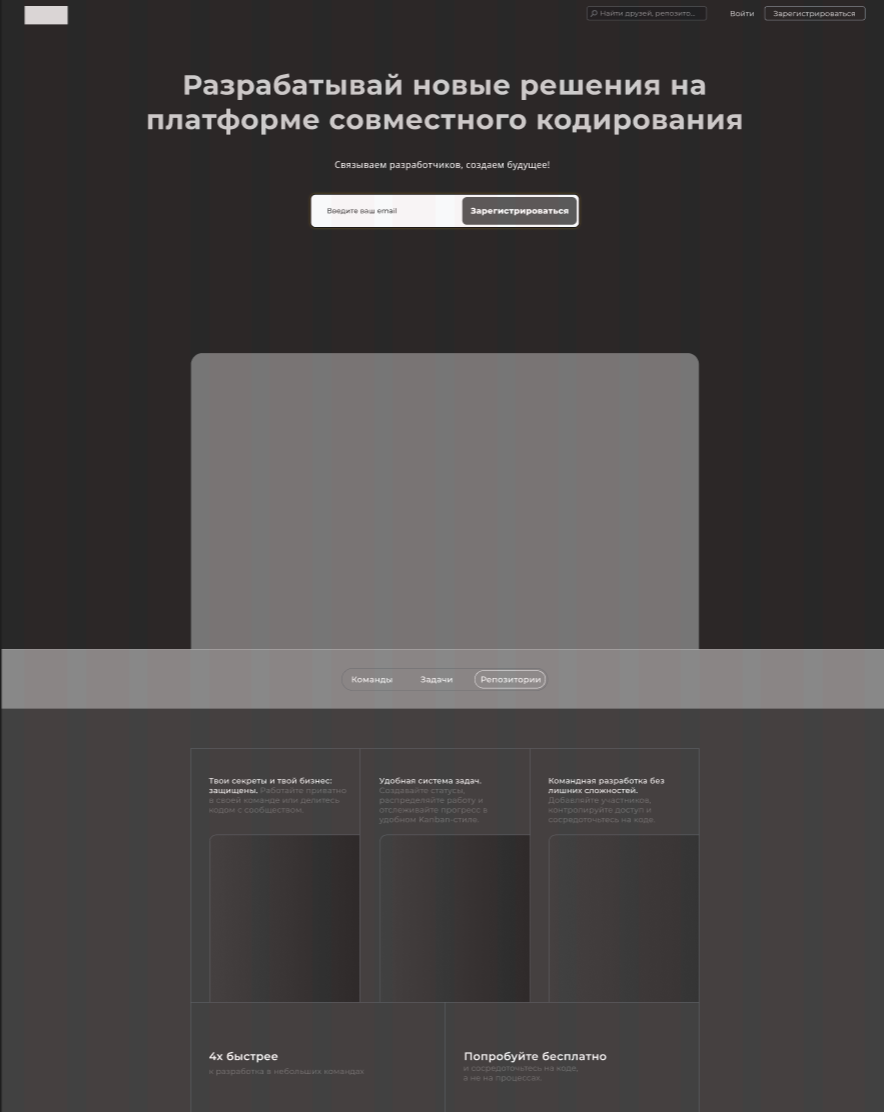


Рисунок 17 – Прототип главной страницы и мобильной версии

## 2.5 Используемые технологии для разработки веб-сайта

Для разработки функциональной и надежной веб-платформы «DevLink» был выбран современный и эффективный технологический стек, сочетающий в себе передовые решения для создания интерактивных и масштабируемых приложений.

В качестве серверной части используется фреймворк Laravel, который обеспечивает мощную и гибкую архитектуру для быстрого создания безопасных и производительных веб-сервисов. Laravel предоставляет удобные инструменты для работы с базой данных, встроенную систему аутентификации, а также продвинутую маршрутизацию, что позволяет реализовать сложную логику приложения с минимальными затратами времени.

Для обеспечения реального времени и мгновенного обмена данными между пользователями в проекте применяется технология WebSocket, что особенно важно для совместного редактирования кода и синхронизации действий участников в репозиториях и командах.

Клиентская часть разработана с использованием современного JavaScript-фреймворка Vue.js, который позволяет создавать отзывчивый, динамичный и интуитивно понятный интерфейс. Vue.js обеспечивает удобную работу с компонентами, двустороннюю привязку данных и эффективное управление состоянием приложения.

Для упрощения развертывания и обеспечения стабильной работы всех компонентов платформы используется Docker – технология контейнеризации, позволяющая создавать изолированные и воспроизводимые окружения для разработки и продакшена.

## 2.6 Структура базы данных

Структура базы данных представляет собой организацию данных, определяющую, как информация представлена, связана и обрабатывается внутри системы управления базами данных.

База данных платформы «DevLink» спроектирована с учетом ключевых аспектов совместной работы разработчиков, управления проектами, командами и файлами. В основе модели лежит комплекс взаимосвязанных сущностей, обеспечивающих полный цикл взаимодействия пользователей с системой.

Центральным элементом является сущность «Пользователи», которая хранит информацию о зарегистрированных участниках, включая их учетные данные, настройки профиля, социальные ссылки и статистику активности. Каждому пользователю соответствует профиль с дополнительными данными, такими как аватар, биография, должность и предпочтения интерфейса.

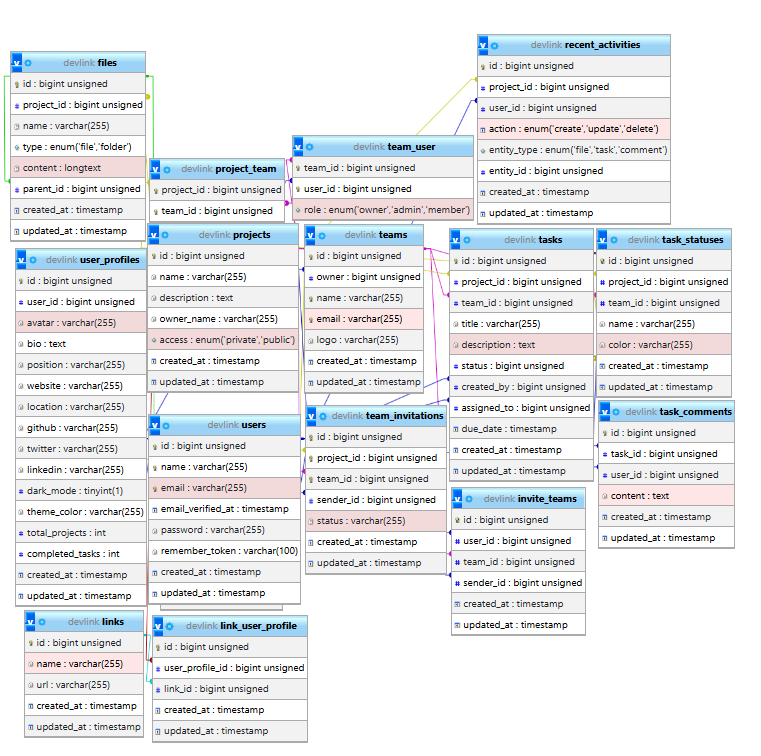


Рисунок 18 – ERM-диаграмма

Система управления проектами представлена сущностями «Проекты» и «Команды». Проекты содержат основную информацию о целях и доступности, а команды – объединения пользователей с определённым владельцем, названием, электронной почтой и логотипом. Взаимодействие между проектами и командами реализовано через промежуточные связи, позволяющие гибко управлять доступом и ролями участников.

Для организации совместной работы внутри команд предусмотрена таблица участников с распределением ролей (владелец, администратор, участник). Также реализована система приглашений для присоединения команд к проектам, что обеспечивает удобный механизм расширения сотрудничества.

Управление рабочими процессами построено вокруг сущностей «Задачи» и «Статусы задач». Задачи связаны с проектами и командами, имеют заголовок, описание, статус, ответственных пользователей и сроки исполнения. Для каждой задачи возможно добавление комментариев, что способствует коммуникации и прозрачности процесса.

Файловая система платформы организована через сущность «Файлы», которая поддерживает иерархическую структуру папок и файлов с возможностью хранения содержимого и версий. История изменений файлов фиксируется отдельной таблицей, что позволяет отслеживать правки и авторов.

Активность пользователей и изменений в системе отражается в журнале недавних действий, фиксирующем тип операции, объект и время. Дополнительно предусмотрены сущности для хранения ссылок и их привязки к профилям пользователей, что расширяет возможности персонализации.

Таким образом, структура базы данных «DevLink» обеспечивает комплексное управление пользователями, командами, проектами, задачами и файлами, создавая надежную и гибкую платформу для совместной разработки и обмена знаниями. Диаграмма данных визуализирует взаимосвязи между сущностями и служит основой для эффективного проектирования и поддержки системы.

## 2.7 Описание продукта

«DevLink» представляет собой инновационную платформу для совместной разработки программного обеспечения, предназначенную для командной работы над проектами различной сложности. Проект разработан с целью упрощения процесса коллективного кодирования и создания комфортной среды для разработчиков всех уровней подготовки. Основной акцент сделан на обеспечении seamless-взаимодействия между участниками команды в режиме реального времени.

Платформа предлагает комплексное решение, объединяющее в себе функционал редактора кода, системы управления задачами и инструментов командной коммуникации. Особенностью «DevLink» является интуитивно понятный интерфейс, который значительно снижает порог входа для начинающих разработчиков, сохраняя при этом все необходимые возможности для профессиональной работы.

Главная страница служит отправной точкой для знакомства с возможностями платформы. Здесь пользователи могут получить общее представление о функционале системы, быстро перейти к процессу регистрации. Четкая визуальная иерархия и продуманная навигация позволяют легко ориентироваться в основных возможностях сервиса.

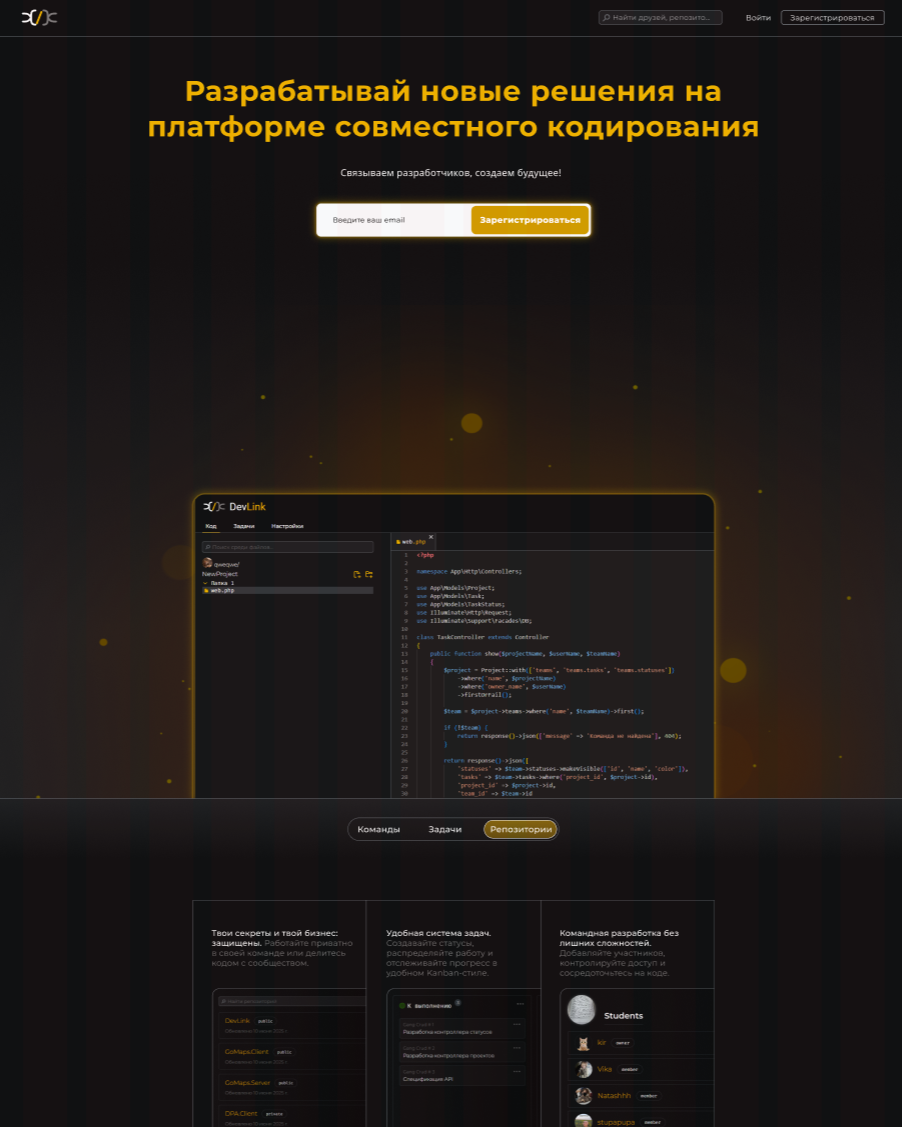


Рисунок 19 – Главная страница «DevLink»

Система авторизации реализована с учетом современных требований к безопасности и удобству. Процесс регистрации занимает минимальное время и требует ввода только базовой информации, что особенно важно для пользователей, желающих быстро приступить к работе. После успешной аутентификации разработчики получают доступ к полному функционалу платформы.

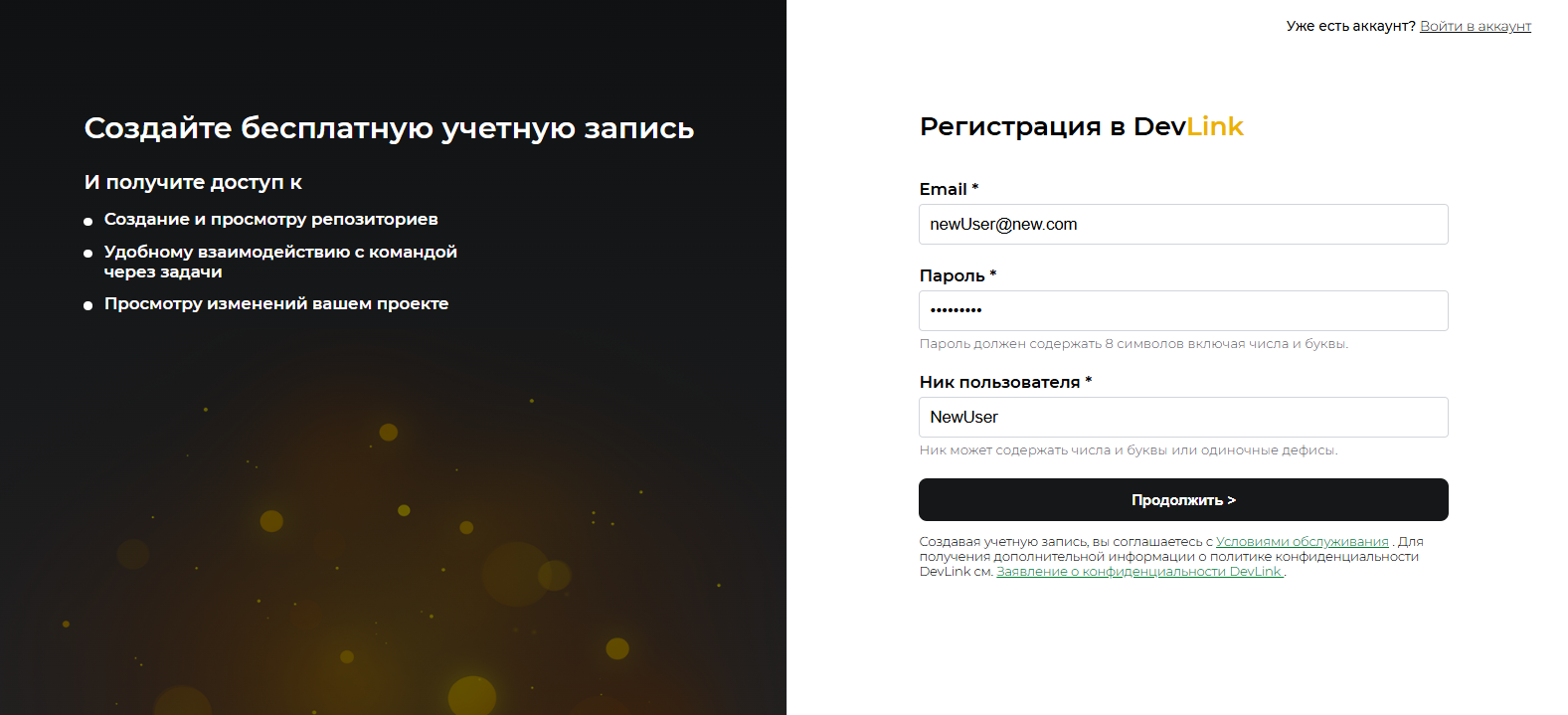


Рисунок 20 – Процесс регистрации нового пользователя

Авторизованный пользователь может редактировать свой профиль после создания. Так же на странице профиля показаны доступные репозитории пользователя, приглашения в команду, информация о пользователе, командах в которых он присутствует.

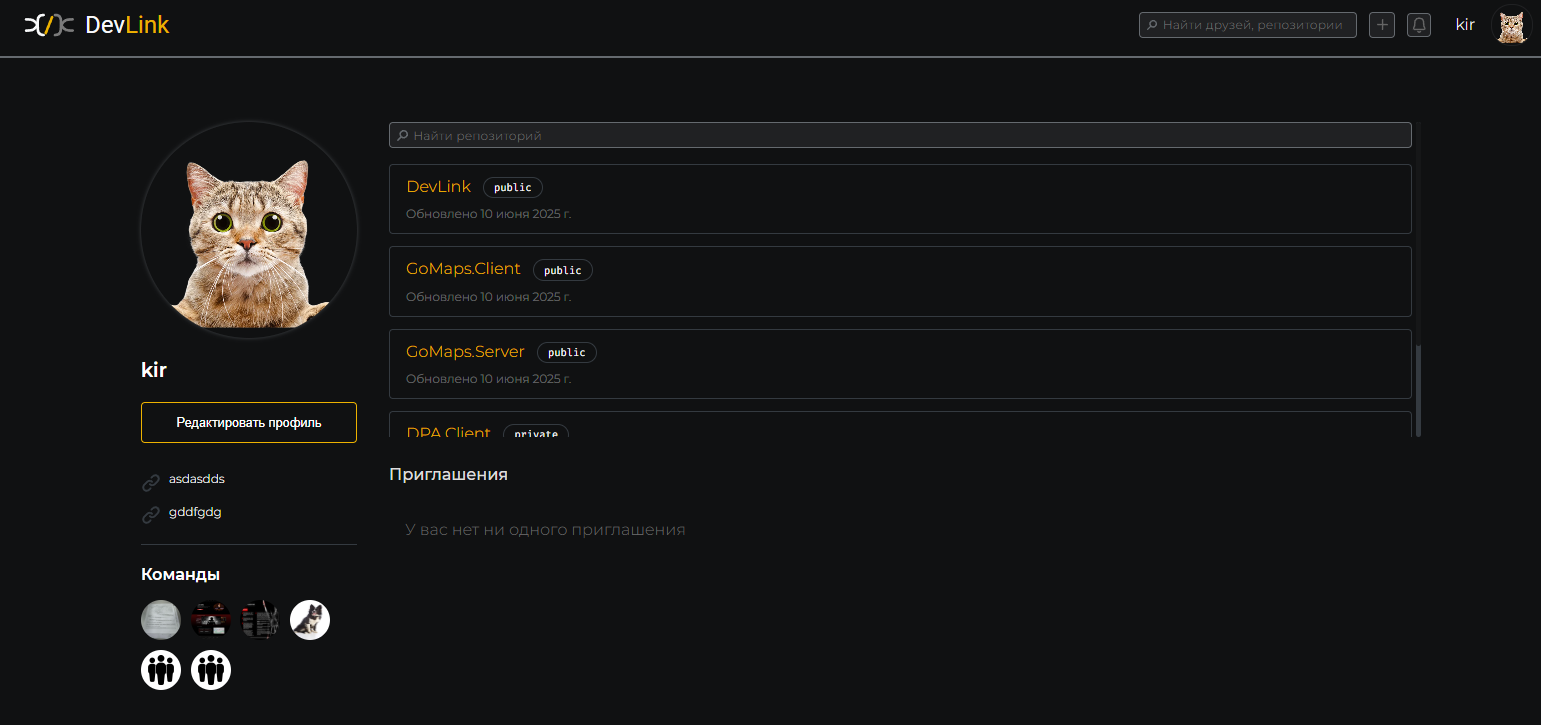


Рисунок 21 – Страница пользователя

Также авторизованному пользователю доступно меню создание репозитория. Где нужно указать адрес, по которому будет доступен проект, описание, и видимость репозитория.

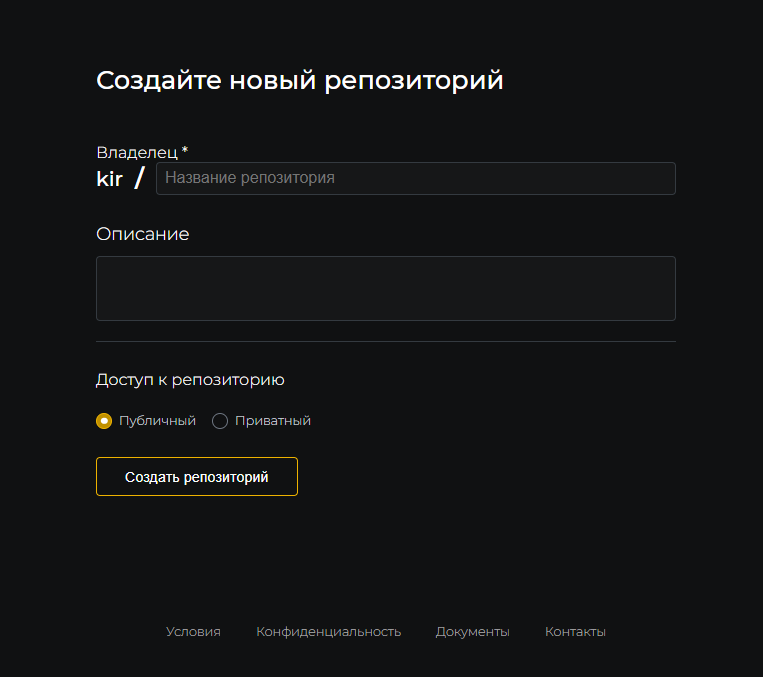


Рисунок 22 – Форма для создания репозитория

Центральным элементом платформы является рабочее пространство, где сосредоточены все инструменты для совместной разработки. Интерфейс разделен на несколько логических зон: редактор кода, система управления задачами и панель командного взаимодействия. Такая организация пространства позволяет эффективно работать над проектом, не переключаясь между различными сервисами.

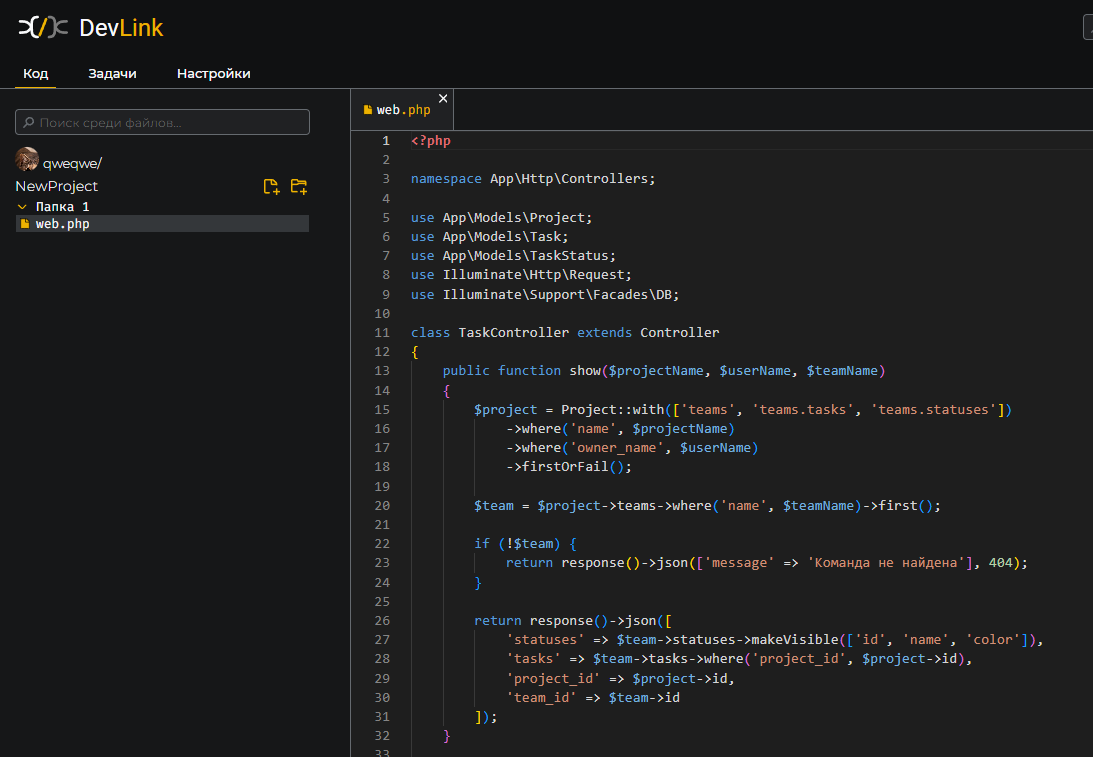


Рисунок 23 – Интерфейс рабочего пространства

Особое внимание уделено системе управления репозиториями. Пользователи могут создавать новые проекты и настраивать параметры доступа.

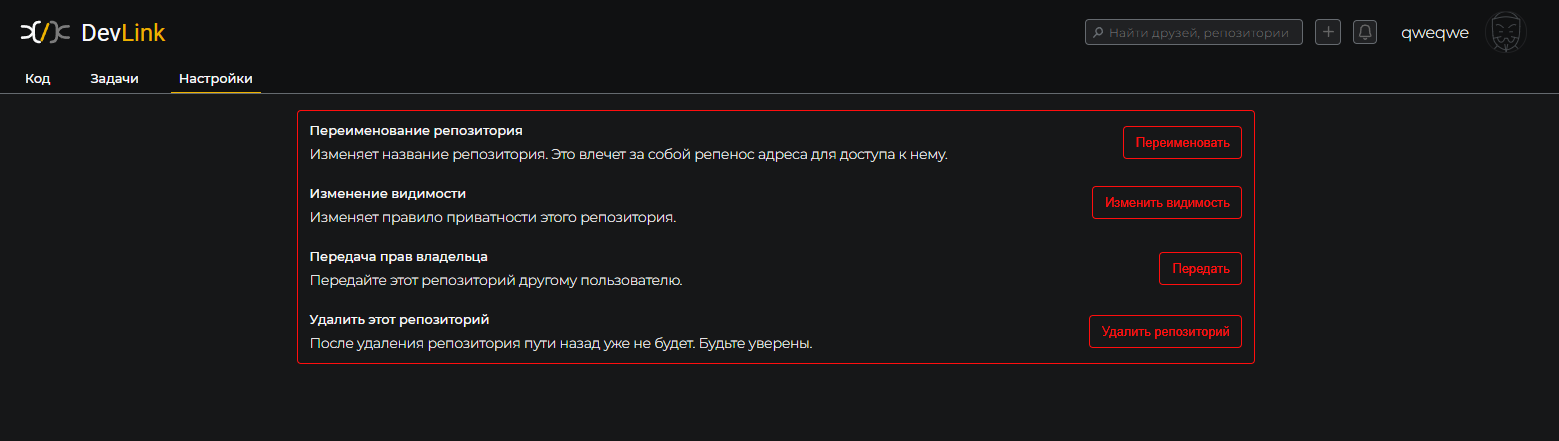


Рисунок 24 – Страница управления репозиториями

Функционал командной работы включает в себя инструменты для приглашения участников, распределения ролей и настройки уровней доступа.

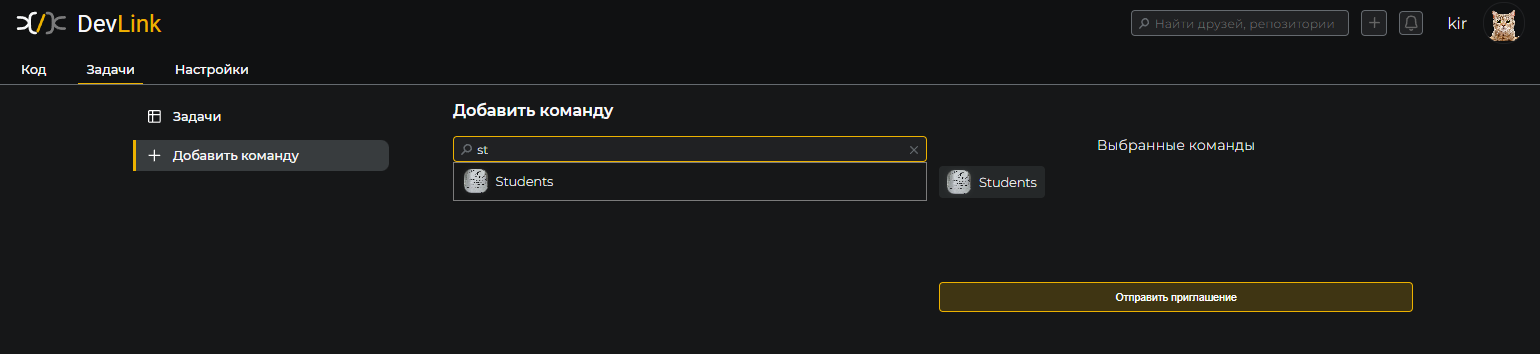


Рисунок 25 – Страница управления репозиториями

Система задач реализована по принципу канбан-досок, что позволяет наглядно отслеживать прогресс разработки. Пользователи могут создавать задачи, а также комментировать выполнение.

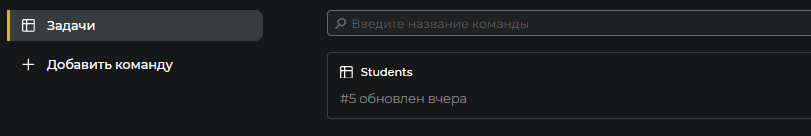


Рисунок 26 – Выбор команды в панели задач

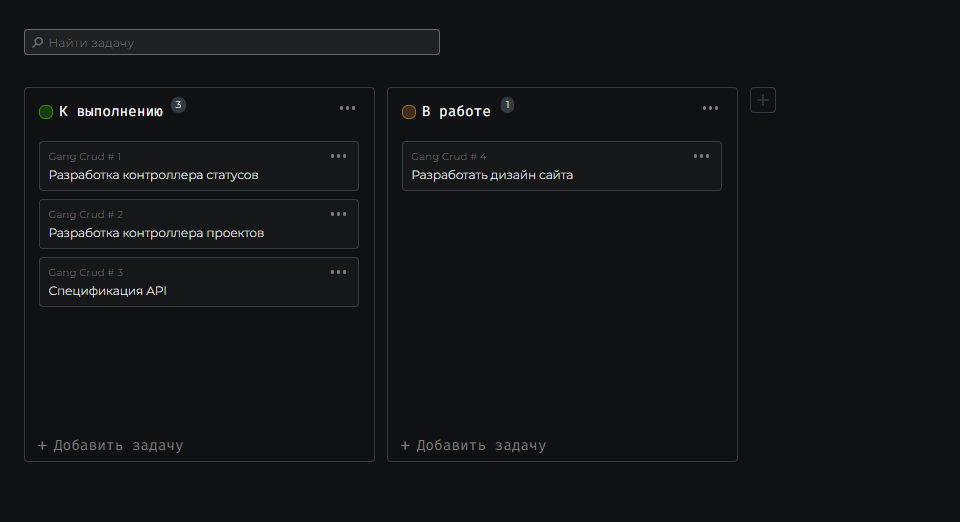


Рисунок 27 – Система kanban-доски

Так же аутентифицированный пользователь может создать команду и пригласить участников в меню создания.

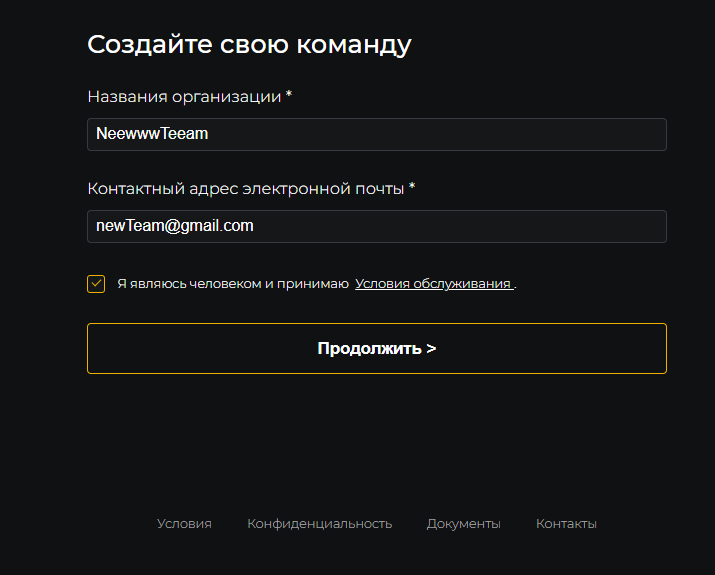
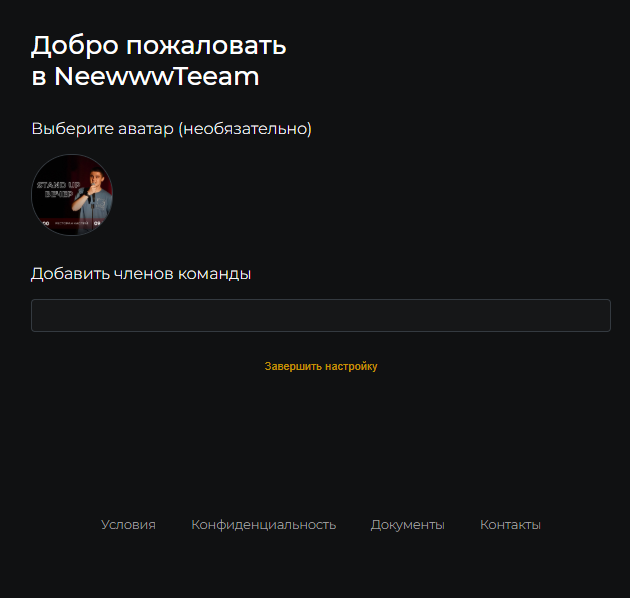
 

Рисунок 28 – Создание команды

Добавлять участников можно и после создания команды, также на странице команды представлена информация он ней, её участники.

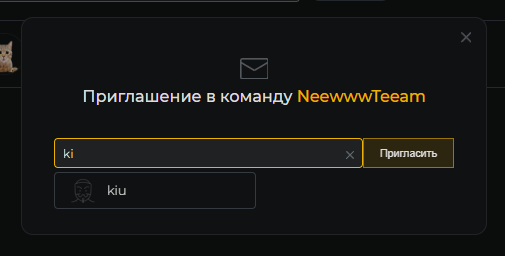


Рисунок 29 – Меню добавления

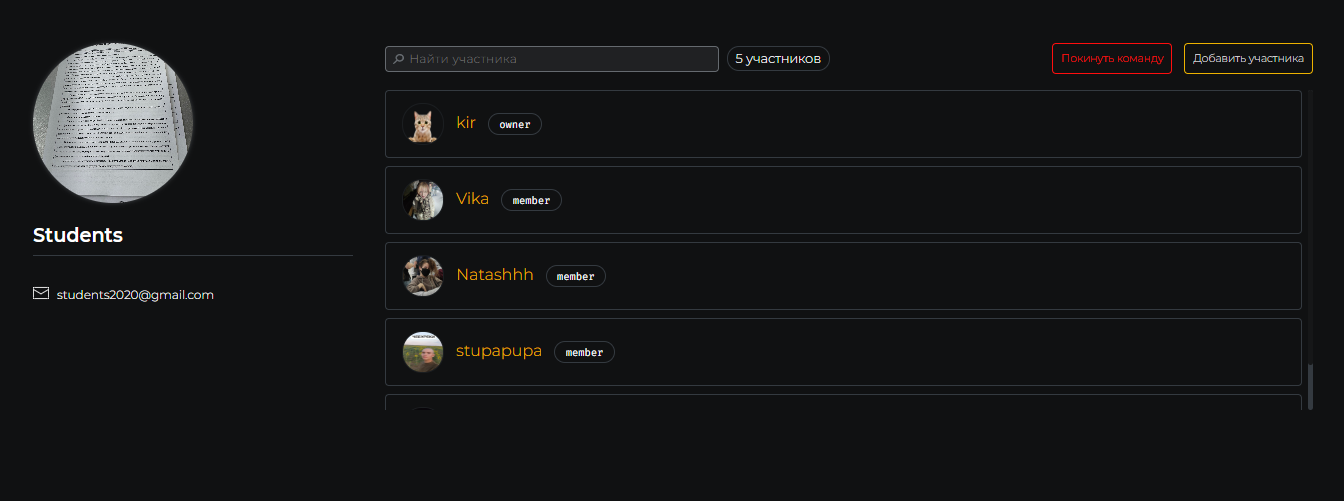


Рисунок 30 – Страница команды

У команды есть возможность посмотреть список доступных для редактирования репозиториев, а также принять приглашение от репозитория.

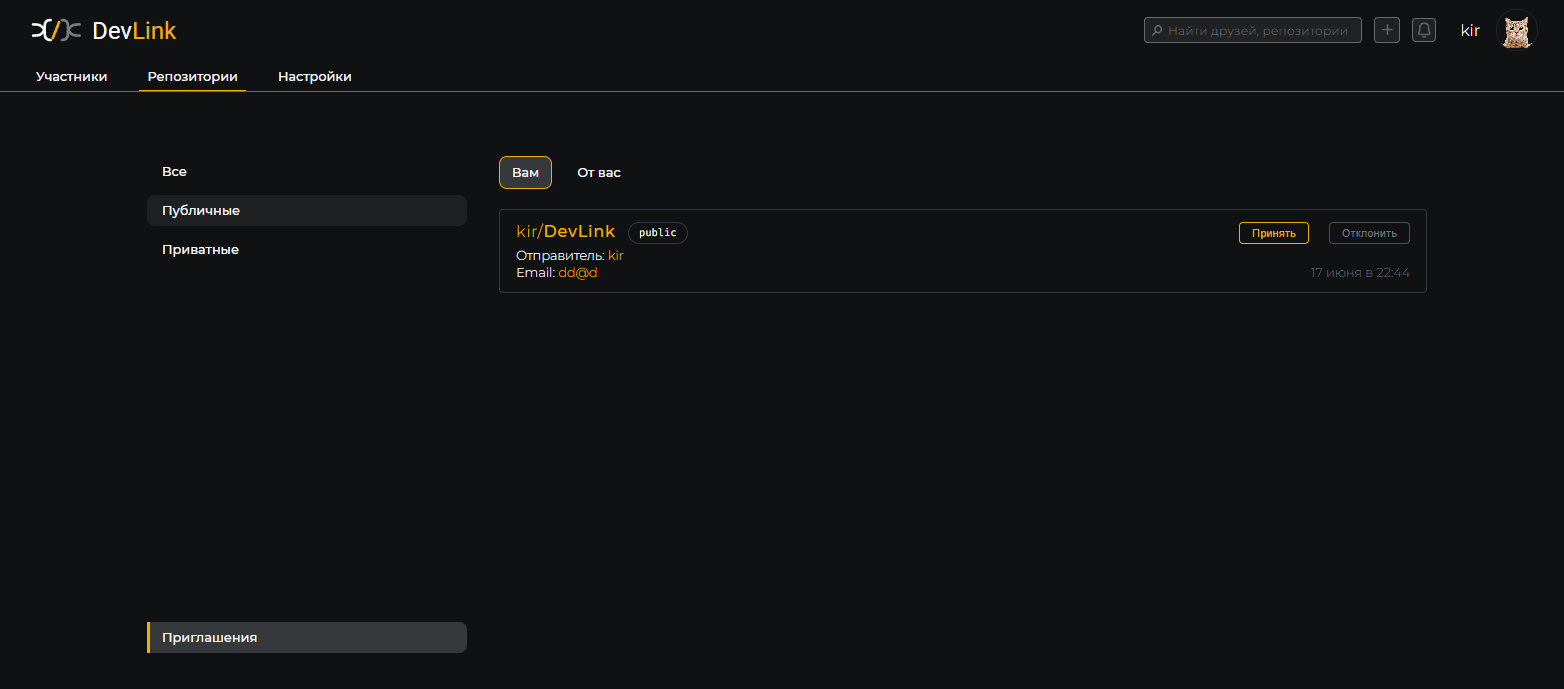


Рисунок 31 – Список приглашений команды

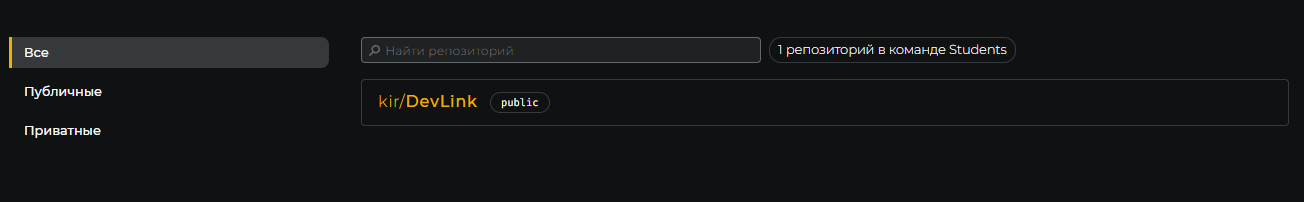


Рисунок 32 – Список доступных репозиториев

«DevLink» представляет собой решение для командной разработки, которое сочетает в себе простоту использования с мощным функционалом. Платформа ориентирована на широкий круг пользователей - от студентов, делающих первые шаги в программировании, до профессиональных команд, работающих над коммерческими проектами. Гибкость системы и возможность кастомизации рабочих процессов делают «DevLink» универсальным инструментом для различных сценариев использования.

Технологическая реализация платформы «DevLink» демонстрирует сбалансированный подход к созданию комплексной среды для командной разработки, где современные инженерные решения органично сочетаются с продуманным пользовательским опытом. Архитектура системы построена вокруг модульной структуры, обеспечивающей интуитивную навигацию между ключевыми функциональными блоками – от управления репозиториями до координации задач через Kanban-доски. Выбор технологического стека (Laravel для бэкенда, Vue.js для динамического интерфейса, WebSockets для синхронизации в реальном времени и Docker для контейнеризации) гарантирует масштабируемость, отказоустойчивость и безопасность, критически важные для инструментов коллективной работы.

Дизайн-система платформы, основанная на контрастной желто-графитовой палитре и геометричном шрифте Montserrat, формирует современный технологичный образ, одновременно снижая когнитивную нагрузку на пользователя. Логотип с символикой соавторства визуально подчеркивает миссию продукта. Прототипы интерфейса подтверждают эффективность подхода: интеграция редактора кода, системы задач и инструментов коммуникации в единое рабочее пространство исключает необходимость переключения на различные ресурсы во время разработки.

# Заключение

В ходе выполнения дипломного проекта была разработана инновационная платформа «DevLink» для совместного кодирования, отвечающая современным требованиям к инструментам коллективной разработки программного обеспечения. Проект успешно решает проблему сложности освоения профессиональных сред разработки, предлагая интуитивно понятное решение для командной работы над кодом. Социальная значимость платформы проявляется в демократизации доступа к инструментам разработки для студентов, начинающих программистов и небольших стартап-команд.

Технологическая ценность платформы заключается в грамотном сочетании современных веб-технологий Laravel, Vue.js, WebSockets для создания производительного и масштабируемого решения. Особое внимание уделено оптимизации пользовательского опыта, что делает «DevLink» доступным инструментом для образовательных учреждений и начинающих разработчиков.

Разработанная платформа не только соответствует современным стандартам веб-разработки, но и вносит значительный вклад в развитие IT-образования и культуры командной работы, демонстрируя эффективность централизованного подхода при создании профессиональных инструментов. «DevLink» открывает новые возможности для распределённых команд и способствует формированию сообщества разработчиков, объединённых вокруг принципов открытости и коллаборации.

# Список используемой литературы

**Основная литература**

1. Корнеев И. В. Современные технологии совместной разработки ПО: от Git до облачных IDE // Издательство «Техносфера», 2024.
2. Петров А. С., Сидоров К. Л. Инструменты для командной работы в IT: анализ и перспективы // Журнал «Программная инженерия», №4, 2023.
3. Учебное пособие по веб-разработке / Под ред. Кузнецова М. А., Васильевой Е. П. – М.: «Инфра-Инженерия», 2025.
4. Григорьев Д. В. Agile и DevOps в малых командах: практическое руководство – СПб.: «Питер», 2022.

**Электронные ресурсы**

1. Официальная документация GitHub [Электронный ресурс]. – URL: https://docs.github.com
2. Исследование Stack Overflow Developer Survey 2024 [Электронный ресурс]. – URL: https://survey.stackoverflow.co/2024/
3. Статья «Будущее облачных IDE» на Habr [Электронный ресурс]. – URL: https://habr.com/ru/articles/789654/
4. Документация Laravel [Электронный ресурс]. – URL: https://laravel.com/docs
5. Руководство по Vue.js [Электронный ресурс]. – URL: https://vuejs.org/guide/
6. Статистика роста IT-рынка (Statista, 2024) [Электронный ресурс]. – URL: https://www.statista.com/topics/1145/developer-tools/