

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Брянский государственный инженерно-технологический университет»

Кафедра «Информационные технологии»

ОТЧЕТ

по производственной (преддипломной) практике
на ООО «Ноософт»

ОП-02068025-09.03.02-037.22

Студент	_____		С.А. Максимов
Группа	ИСТ-401	№ зачетной книжки	18-2.037
Руководитель от предприятия	_____	Директор ООО «Ноософт»	С.В. Кондратенко
Руководитель от вуза	_____	канд. тех. наук, доц.	А.В. Яковлев
Нормоконтроль	_____	канд. экон. наук, доц.	О.Н. Юркова
Допуск к защите	«17» <u>05</u> 2022 г.	_____ канд. экон. наук, доц.	О.Д. Казаков
Дата защиты	«18» <u>05</u> 2022 г.	Оценка: _____	
Члены комиссии	_____	канд. экон. наук, доц.	О.Д. Казаков
	_____	канд. экон. наук, доц.	О.Н. Юркова
	_____	канд. техн. наук, доц.	А.В. Яковлев

Брянск 2022

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Брянский государственный инженерно-технологический университет»

Индивидуальное задание
Производственной практики (преддипломной)

Обучающийся: Максимов С. А.

Руководитель практики от вуза: Яковлев А. В

Сроки прохождения практики: 18.04.2022-18.05.2022

Место прохождения практики: ООО “Ноософт”

Содержание индивидуального задания:

1 РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ ПРОЕКТАМИ

1.1. Реализация программных компонентов

1.2 Тестирование и апробация программного средства

Дата выдачи задания 18 апреля 2022 г.

Руководитель практики _____ / Яковлев А. В /

Задание принял к исполнению _____ / Максимов С. А. /

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. АНАЛИЗ ЗАДАЧИ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ ПРОЕКТОМ.....	5
1.1 Теоретические аспекты разработки веб-приложения	5
1.2 Обзор существующих программных решений по управлению проектами	7
1.3 Предмет разработки и определение стека инструментов и технологий разработки	12
2. ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА УПРАВЛЕНИЯ ПРОЕКТОМ.....	16
2.1 Разработка функциональных и поведенческих спецификаций.....	16
2.2 Проектирование структуры и компонентов программного обеспечения.....	19
2.3 Разработка информационных спецификаций	22
2.4 Проектирование схемы развертывания разрабатываемого средства.	26
3. РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ	28
3.1 Реализация программных компонентов	28
3.2 Тестирование и апробация программного средства.....	34
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ А	42
ПРИЛОЖЕНИЕ Б.....	44
ПРИЛОЖЕНИЕ В	46
ПРИЛОЖЕНИЕ Г.....	49

					ОП-02068025-09.03.02-037.22							
Изм.	Лист	№ докум.	Подп.	Дата								
Разраб.		Максимов С. А			Отчет по производственной (преддипломной) практике				Лит.	Лист	Листов	
Пров.		Яковлев А. В							у		2	55
									БГИТУ ИСТ-401			
Н.контр.												
Утв.												

ВВЕДЕНИЕ

Сегодня в век информационных технологий у бизнеса и предприятий вырос запрос на различные IT-продукты, начиная от веб-сайта и заканчивая приложениями на мобильных платформах. В такое время «большого спроса» возросла нагрузка на компании по разработке it-продуктов. Обстановка конца первой четверти 21 века не похожа на ее начало, масштабы и объем приложений значительно вырос из-за чего необходимо наращивать штат сотрудников и затраты ресурсов на разработку.

Сейчас, когда у компании большой штат сотрудников и выпускаемый ими продукт сам по себе требует много сил и ресурсов то перед компаниями встал вопрос об управлении ходом разработки, распределение задач по командам и управление этими самими командами. Сегодня уже существуют ПО которое помогает как-то управлять ходом разработки проекта, примерами являются Trello, Битрикс 24 и GitHub. Перечисленное ПО выполняет свои задачи, но проблема заключена в том, что это независимые друг от друга среды, для более качественного и комфортного наблюдения и управления за ходом разработки необходим продукт которые совмещает в себе различные инструменты по управлению и не требовал какой-либо дополнительной установки на машину пользователя. Чаще всего такими средами становятся веб-приложения, сегодня у каждого пользователя сети Интернет есть персональный компьютер, с предустановленным браузером, который не требует от клиента совершать дополнительные манипуляции на своем компьютере.

В данной работе предлагается разработка веб-приложения по управлению проектами.

Актуальность исследования обусловлена тем что на рынке практически нет удобного инструмента по управлению ходом разработки проектов который совмещает в себе различные технологии курирования хода проектирования продукта.

					ОП-02068025-09.03.02-037.22	Лист
						3
Изм.	Лист	№ докум.	Подп.	Дата		

Цель работы – разработать веб-приложение с простым и понятным пользовательским интерфейсом для управления проектом.

Для достижения цели, необходимо решить следующие задачи:

- описать функциональные требования к веб-приложению;
- провести исследование аналогичных разработок;
- спроектировать и разработать удобный интерфейс;
- разработать приложение и описать принцип его работы;
- провести тестирование продукта;

Объектом исследования является функциональное программирование.

Предметом исследования являются инкапсулированные функциональные компоненты с собственным состоянием.

Для написания теоретической части исследования были использованы различные интернет-статьи на тематических формах, учебники и учебные пособия. В процессе исследования использовались различные приемы и методы анализа данных: синтез, анализ, дедуктивный и индуктивный методы.

Практическое значение работы состоит в том, что результаты работы могут быть применены в разработке сторонних продуктов со схожими концепциями, так же материал можно использовать в образовательных целях в университетах.

Структура отчета состоит из трех основных частей.

В аналитической части выполнена работа по обоснованию разработки системы управления проектом, были изучены существующие аналоги, выявлены их достоинства и недостатки. Был выбран и обоснован стек технологий для разработки проекта.

В третьей части работы описана реализация веб-приложения, были приведены примеры реализации функциональных компонент проекта, так же были проведены тестирование и отладка приложения.

1. АНАЛИЗ ЗАДАЧИ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ ПРОЕКТОМ

1.1 Теоретические аспекты разработки веб-приложения

До того, как масштаб проектирования и разработки различного программного обеспечения был не особо велик, менеджеры и разработчики справлялись с разработкой продукта без помощи сторонних средств, максимумом были схемы, нарисованные от руки на листе бумаги и перечень задач для конкретного разработчика.

С повышением объема задач и штатом сотрудников стало очень сложно если невозможно курировать ход разработки продукта, встал вопрос о создании инструментов, которые позволят достаточно легко проектировать, распределять задачи и следить за их выполнением. Так же данные инструменты не должны занимать много места и быть максимально доступны каждому пользователю. В качестве платформы на котором должны были развёрнуты эти инструменты стал обычный браузер которой предустановлен на каждом компьютере, который прост в использовании и не нагружает систему машины.

Браузер – это специальная программа, которая создана для поиска информации в сети Интернет, просматривать сайты, скачивать и загружать файлы. Браузер является посредником между пользователем и глобальной паутиной.

Интернет – это глобальная сеть, которая по различным протоколам объединяет сети различных регионов, государств, организаций. Так же интернет называют «сеть сетей».

Проблема старого подхода проектирования состояла в следующих моментах:

– нет возможности для быстрой коммуникации с командой разработчиков;

					ОП-02068025-09.03.02-037.22	Лист
						5
Изм.	Лист	№ докум.	Подп.	Дата		

– очень сложно распределить и проверять задачи между разработчиками, большая вероятность того что будут допущены ошибки;

– сложно передать всю картину проекта сотрудникам т.к. отсутствует механизм визуализации, «схема и связи» об проекте хранятся лишь в голове лидера и передать их на бумаге очень трудно;

– разработчикам сложно получить оценку выполнения своей работы т.к. менеджеру приходит множество данных и ему очень трудно курировать каждую задачу;

– процесс курирование хода разработки проектов превращается в большой объем сложно структурированных данных которые очень сложно контролировать;

Для решения проблем старого подхода необходимо провести исследование и разработать новые методы и подходы разработки, которые помогут избежать критических ошибок и повысить скорость разработки и оптимально использовать ресурсы.

					ОП-02068025-09.03.02-037.22	Лист
						6
Изм.	Лист	№ докум.	Подп.	Дата		

1.2 Обзор существующих программных решений по управлению проектами

Рассмотрим программные решения, которое уже существуют и выполняют функции по управлению разработки проектов.

Trello – облачная программа для управления проектами небольших групп, разработанная Fog Creek Software.

Trello является одной из самых популярных систем управления проектами в онлайн режиме, которая пользуется популярностью у небольших команд и стартапов. Данная платформа помогает организовывать работу по методологии канбан-досок.

Внешний вид платформы представлен на рисунке 1.

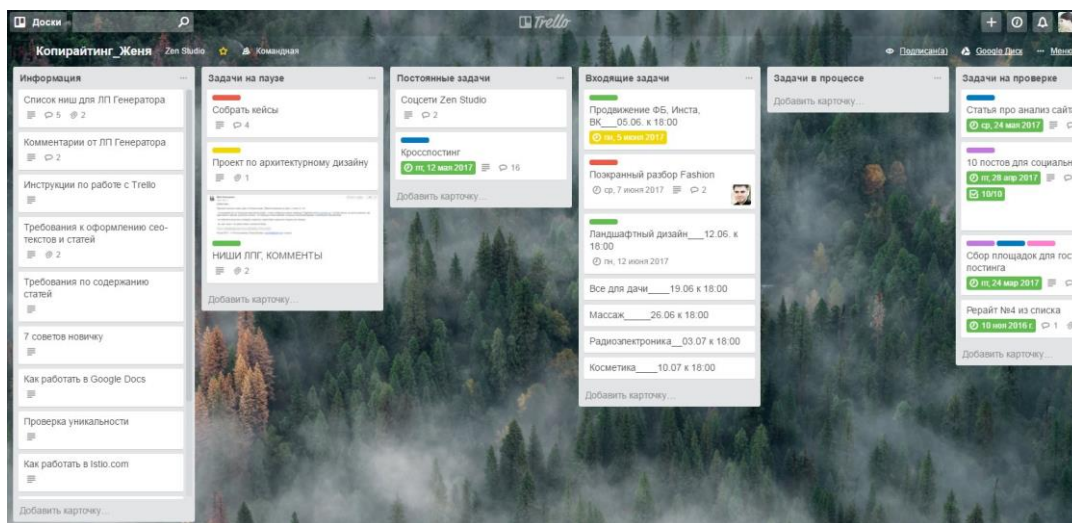


Рисунок 1 – Trello

Платформа была создана Fog Creek Software в 2011 году на базе Backbone.js, MongoDB и Node.js. Можно перечислить следующие преимущества, благодаря которым Trello получила свою популярность:

- простой и понятный интерфейс;
- практически неограниченный бесплатный доступ платформы;
- удобство в работе;

					ОП-02068025-09.03.02-037.22	Лист
						7
Изм.	Лист	№ докум.	Подп.	Дата		

Платформа является максимально простым инструментом, который легко внедрить в любой рабочий процесс. Интерфейс платформы выполнен в виде канбан досок. Данная концепция заключается в том, что существует как минимум три типа задач. Как правило задачи разбивают на:

- новые задачи;
- в разработке;
- выполненные;

Пример канбан доски представлен на рисунке 2.



Рисунок 2 - Канбан доска

Механизм платформы заключается в том, что типы задач представляют собой колонки, а задачи, которые необходимо выполнить сделаны в виде карточек. При выполнении какой-либо задачи, «карточку» перемещают в другую «колонку».

Trello практически бесплатная платформа. У платформы есть платная версия, но в большинстве случаев пользователю для полноценной работы будет хватать и бесплатной версии которая имеет без лимитного тарифа по времени. У платформы открытое API благодаря чему любой программист может настроить свой собственный клиент.

Платформа очень проста в использовании благодаря чему не нужно тратить время на ее изучение. Весь механизм заключается в создании карточек с задачей и перетягивании ее в соответствующую ее текущему статусу колонку.

Trello является одним из удобных инструментов по разработке проектов но у нее есть следующие недостатки:

- нет списка «друзей» для удобного добавления пользователей в рабочее пространство;
- нет инструмента по поиску сотрудников;
- невозможно составить схемы по типу UML;
- нет единого хранилища вложенных файлов, все они хранятся в карточках что не очень удобно;

Битрикс24 – это платформа, помогающая организовать коллективную работу в компании. Благодаря нее можно выставлять и выполнять задачи, планировать рабочее время и общаться со своими коллегами.

Пример битрикс24 представлен на рисунке 3.

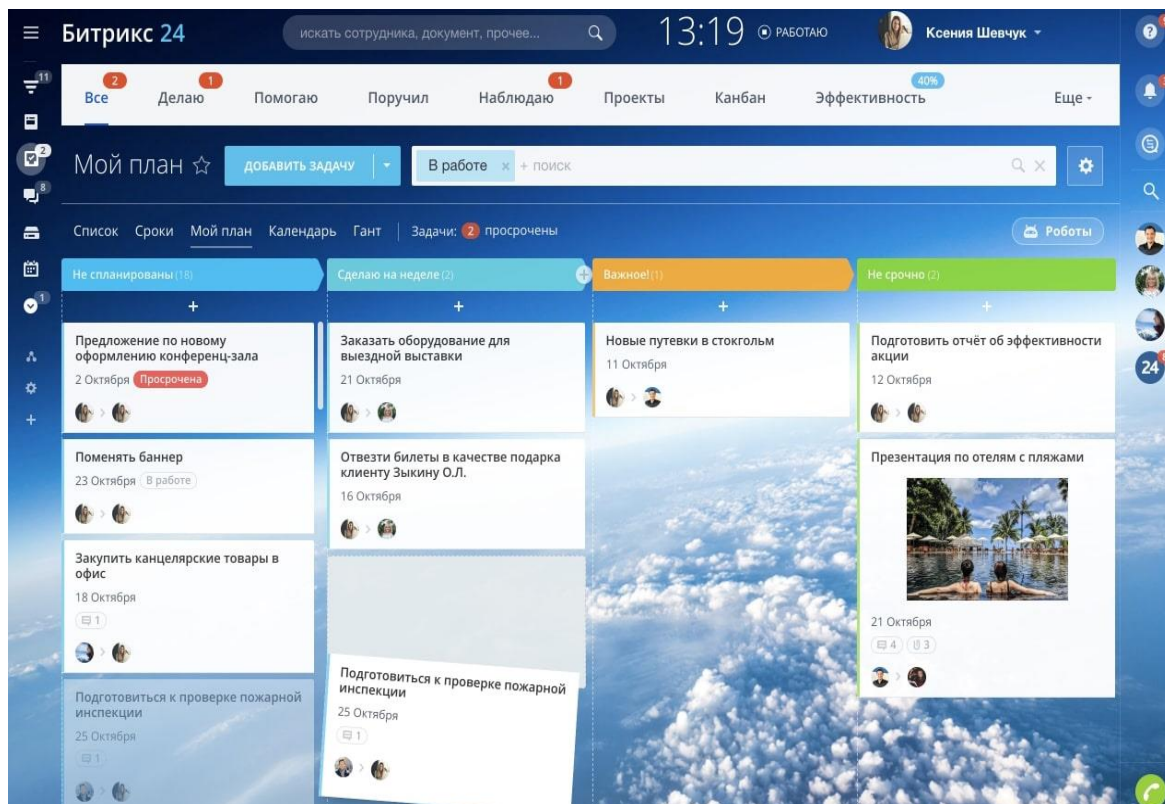


Рисунок 3 - Битрикс24

Битрикс24 является CRM-системой которая позволяет контролировать и координировать работу персонала, анализировать продажи, управлять клиентской базой, общаться с клиентами и многое другое.

Платформа предоставляет большой функционал для администрирования различных проектов. Больше всего она подходит интернет-магазинам, предпринимателям и обычному бизнесу.

Платформа обладает следующими преимуществами:

- множество возможностей;
- интеграция с 1-С;
- возможность общения с коллегами внутри системы;

Битрикс24 это очень большая система, для бизнеса которая из-за своего масштаба обладает следующими недостатками:

- нет возможности построения схем;
- сложная визуализация, длительность привыкания к интерфейсу;

Каждый из сервисов имеет свои достоинства и недостатки, таблица со сравнением представлена в таблице 1.

Таблица 1 – Сравнение сервисов.

Название	Бесплатная версия	Простота освоения	Схемы	Вложения	Канбан доска
Trello	+	+	-	-	+
Битрикс24	+	-	-	+	+

Стоит отметить что существует множество других аналогичных сервисов по управлению проектами. Множество обладает платными и бесплатными версиями. Также стоит помнить, что идеального продукта не существует.

На основе проведенного исследования стало понятно каким должен быть сервис чтобы конкурировать и даже превзойти своих именитых аналогов. Сервис должен быть простым и понятен каждому новому пользователю.

Сервис должен давать возможность создавать канбан доски и возможность построения схем по типу UML.

Так же в каждом «рабочем столе» проекта должна быть вкладка «вложения» для более быстрого поиска прилепленных файлов к проекту.

					ОП-02068025-09.03.02-037.22	Лист
						11
Изм.	Лист	№ докум.	Подп.	Дата		

1.3 Предмет разработки и определение стека инструментов и технологий разработки

Для разработки продукта необходимо изучить требования к функциональным возможностям разрабатываемого приложения.

Веб-приложение должно быть, доступно не смотря на платную и бесплатную версию. Приложение должно иметь возможность авторизации и регистрации пользователя.

Сервис должен обладать поиском пользователей для добавления их в свое рабочее пространство разрабатываемого проекта. Интерфейс должен быть минималистичен и интуитивно понятным, он не должен вызывать трудности у новых пользователей.

Приложение должно совмещать себе возможности создания канбан досок и механизма создания схематического представления работы проекта. Так же сервис должен иметь возможность загрузки файлов любого формата. Для поиска файлов должно быть создано отдельное поле для их хранения – «вложения проекта».

Интерфейс должен быть реализован на русском языке и быть адаптивным под персональные компьютеры различных форматов и планшеты. Так же должен быть выполнен минимальный адаптив под мобильные устройства.

Необходимо рассмотреть стек технологий, который будет использован при разработке веб-приложения под все популярные браузеры.

В прошлом для разработки веб-приложений использовали подход Multi Page Application. Данный подход делил приложение на две части:

- клиентская часть (front-end);
- серверная часть (back-end);

Смысл заключается в том, что сначала разрабатывается клиентская часть при помощи языков разметки и стилей HTML и CSS и языка программирования JavaScript. После реализации клиентской части, шаблоны разбивались на

компоненты и взаимодействовали с базой данных при помощи серверного языка php.

Данный подход, не смотря на свои плюсы устарел и обладает серьезными минусами, такими как нагрузка на сервер и скорость работы на машине пользователя.

Для снижения нагрузки на сервер и повышения скорости работы на клиенте была разработана концепция SPA приложений. Подход заключается в том, что клиенту сервер отправляет JavaScript код который генерирует элементы сайта и меняет свое состояние в зависимости от поведения пользователя. Для разработки SPA приложений была разработана JavaScript- библиотека ReactJS.

ReactJS – это библиотека JavaScript, которая используется для создания пользовательского интерфейса. Первый релиз библиотеки произошел в марте 2013 года. На текущий момент актуальной версией библиотеки является React v18.0.

Первоначально React предназначался для веба, для создания веб-сайтов, однако позже появилась платформа React Native, которая уже предназначалась для мобильных устройств.

React представляется идеальный инструмент для создания масштабируемых веб-приложений (в данном случае речь идет о фронтенде), особенно в тех ситуациях, когда приложение представляет SPA (одностраничное приложение).

React относительно прост в освоении, имеет понятный и лаконичный синтаксис. Так же библиотека поддерживает современный типизированный язык TypeScript.

Библиотека React известна своей простотой и гибкостью благодаря которым на сегодняшний день было разработано различных вспомогательных модулей, благодаря которым разработка пользовательских интерфейсов становится еще быстрее и проще.

Разрабатываемый продукт является публичным инструментом из-за чего разработка на чистом React не подходит, т.к. поисковые роботы не умеют «читать» JavaScript код. Для решения этой проблемы необходимо использовать

					ОП-02068025-09.03.02-037.22	Лист
						13
Изм.	Лист	№ докум.	Подп.	Дата		

фреймворк Next.js. Данная технология совмещает в себе плюсы двух подходов – Multy Page Application и SPA.

Для создания канан досок и mindmap будут использованы библиотеки React Flow и React Beautiful DnD. Данные библиотеки были специально разработаны под React – приложения что делает работу веб-приложения максимально оптимизированным.

Для связи с сервером будет использована JavaScript библиотека – Axios.

Axios — это широко известная JavaScript-библиотека. Она представляет собой HTTP-клиент, основанный на промисах и предназначенный для браузеров и для Node.js.

Веб-приложение является по объему работы с данными средним проектом. Для оптимального управления состоянием приложения необходимо использовать библиотеку Redux.

Redux — это инструмент для управления состоянием данных и пользовательским интерфейсом в приложениях JavaScript с большим количеством сущностей.

Название читается как «Редакс» и составлено из двух слов: reduce и flux. Reduce — это функция, которая приводит большую структуру данных к одному значению. Flux — архитектура приложения, при которой данные передаются в одну сторону. Инструмент основан на этих двух понятиях, поэтому они вынесены в название.

Обычно Redux используется в связке с фреймворками для JavaScript: React, TypeScript, Vue, Angular и другими. Реже он бывает нужен для написания кода на чистом JS. Имеет открытый исходный код и доступен бесплатно. Со всеми зависимостями весит всего около 2 Кб.

Платформами, которые будут использованы для разработки приложения станут IDE WebStorm и браузер Chrome.

Программное обеспечение JetBrains WebStorm представляет собой инструмент для разработки web-сайтов и редактирования HTML, CSS и JavaScript

					ОП-02068025-09.03.02-037.22	Лист
						14
Изм.	Лист	№ докум.	Подп.	Дата		

кода. Решение обеспечивает быструю навигацию по файлам и генерирует уведомления о возникающих проблемах в коде в режиме реального времени. JetBrains WebStorm позволяет добавлять разметку HTML-документов или элементов SQL непосредственно в JavaScript. JetBrains WebStorm осуществляет развертывание и синхронизацию проектов через протокол FTP.

WebStorm обеспечивает отладку кода JavaScript и предоставляет широкий диапазон возможностей: нахождение точки останова в HTML и JavaScript, настройка параметров точки останова, тестирование синтаксиса кода в режиме реального времени и т. д. Продукт поддерживает платформы JQuery, YUI, Prototype, DoJo, MooTools, Qooxdoo и Bindows. WebStorm предусматривает интегрированную проверку текста на тэги, последовательность кода, ошибки в написании и т. д. WebStorm позволяет редактировать файлы и автоматически синхронизировать их по требованию при удаленной работе или хранении.

Продукт поддерживает функцию контроля версий и предварительных вариантов кода и фиксирует все произведенные действия и изменения. Благодаря созданию истории в WebStorm можно восстанавливать кодовые выражения, блоки и даже целые файлы.

					ОП-02068025-09.03.02-037.22	Лист
						15
Изм.	Лист	№ докум.	Подп.	Дата		

2. ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА УПРАВЛЕНИЯ ПРОЕКТОМ

2.1 Разработка функциональных и поведенческих спецификаций

В ходе проектирования веб-приложения была реализована диаграмма потоков данных (DFD).

Диаграммы потоков данных (DFD) являются основным средством моделирования функциональных требований к проектируемой системе. Главная цель построения DFD заключается в том, чтобы сделать требования к системе ясными и понятными на каждом уровне детализации, а также разбить эти требования на части с точно определенными отношениями между ними.

Диаграммы потоков данных позволяют специфицировать как функции разрабатываемого программного обеспечения, так и обрабатываемые им данные. При использовании этой модели систему представляют в виде иерархии диаграмм потоков данных, описывающих асинхронный процесс преобразования информации с момента ввода в систему до выдачи пользователю. На каждом следующем уровне иерархии происходит уточнение процессов, пока очередной процесс не будет признан элементарным.

Для изображения диаграмм потоков данных традиционно используют два вида нотаций: нотации Иордана и Гейна-Сарсона.

Диаграмма потоков данных веб-приложения по управлению проектом представлена на рисунке 4.

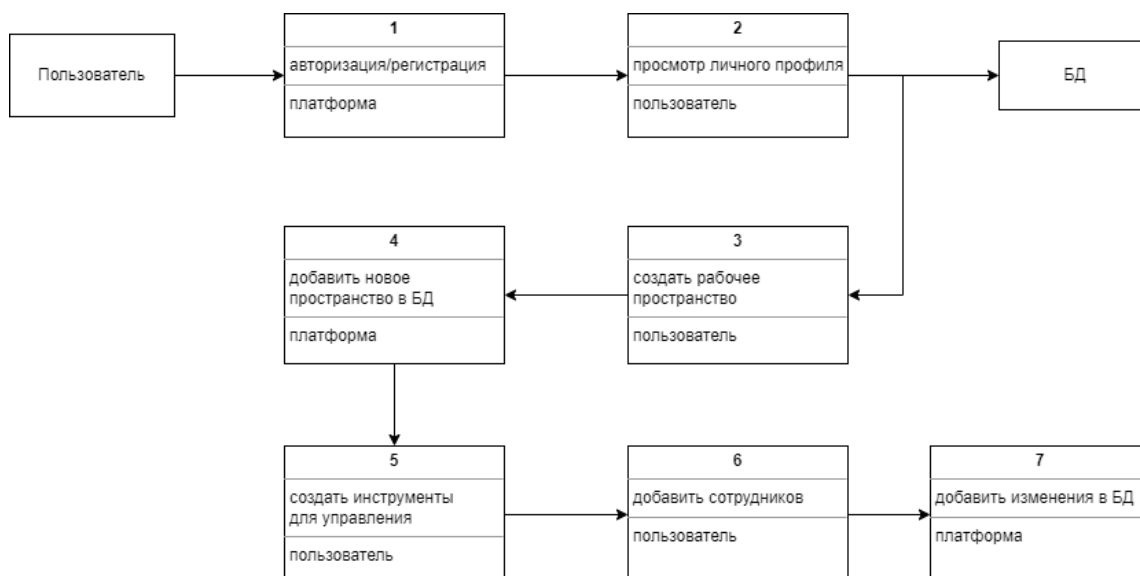


Рисунок 4 - DFD – диаграмма

На диаграмме представлен вход с систему пользователя для управления проектами. Сущностями выступают пользователь и база данных, а процессами весь ход выполнения работы. Из диаграммы видно, что пользователь обменивается информацией с платформой при создании рабочих пространств и изменении его состояния.

Для отображения вариантов использования была разработана диаграмма вариантов использования.

Диаграмма вариантов использования представляет собой описание функциональных возможностей системы. На диаграмме изображены актеры и варианты использования, между которыми существуют отношения. Диаграмма использования позволяет создать список операций, которые выполняются системой. На базе этой диаграммы можно создать список требований к системе и определить выполняемые функции.

Разработанная диаграмма использования представлена на рисунке 5.

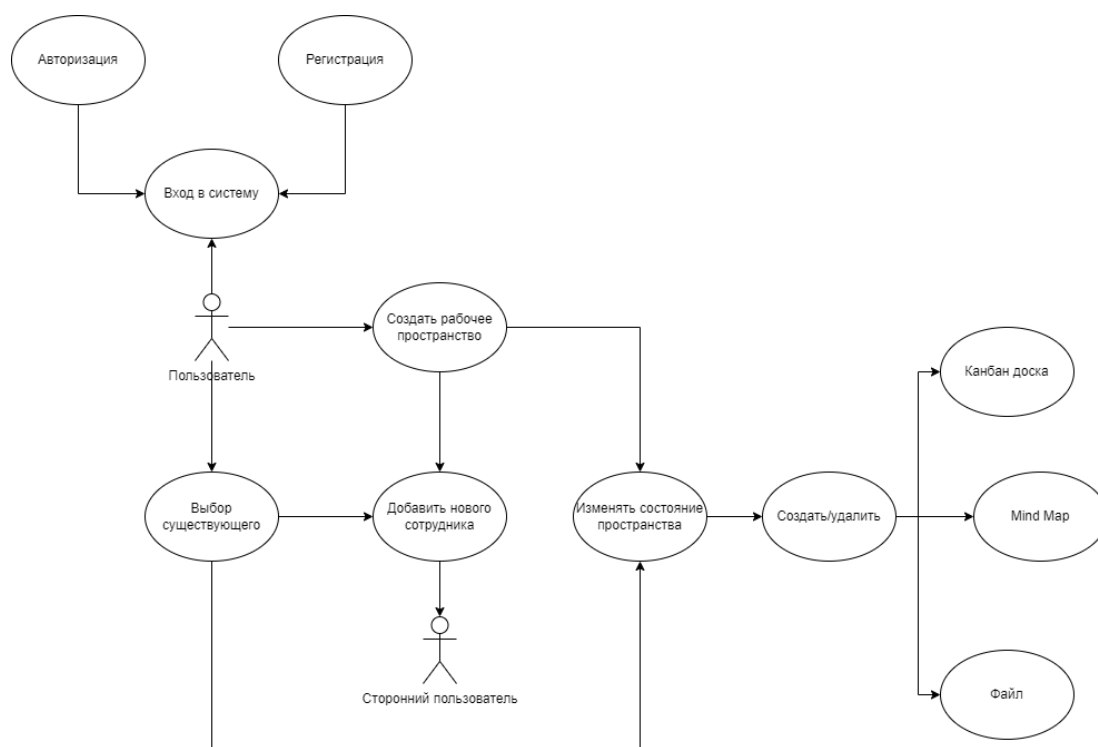


Рисунок 5 - Диаграмма вариантов использования

На разработанной диаграмме виден процесс взаимодействия пользователя с платформой. Алгоритм взаимодействия представляет из себя следующее:

- вход в систему;
- создать новое или использовать существующее рабочее пространство;
- добавить или удалить сотрудника из проекта по необходимости;
- менять состояние пространства – добавить или удалить функциональные элементы;

От правильного представления и построения диаграмм зависит успешность проекта, ведь чем больше сил и внимание будет уделено проектированию проекта, выявлению всех мелких деталей и поиска подводных камней, тем успешнее будет идти процесс разработки веб-приложения.

2.2 Проектирование структуры и компонентов программного обеспечения

Центральное место в ООП занимает разработка логической модели системы в виде диаграммы классов. Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

Классы могут представлять сущности предметной области (на этапе анализа) или элементы программной системы (на этапе проектирования и реализации).

На диаграмме класс изображается в виде прямоугольника, разделенного на три части: имя класса (на английском языке), его атрибуты и его операции. В качестве классов выступают сущности, использованные при построении диаграммы сущность-связь.

Диаграмма классов веб-приложения представлен на рисунке 6.

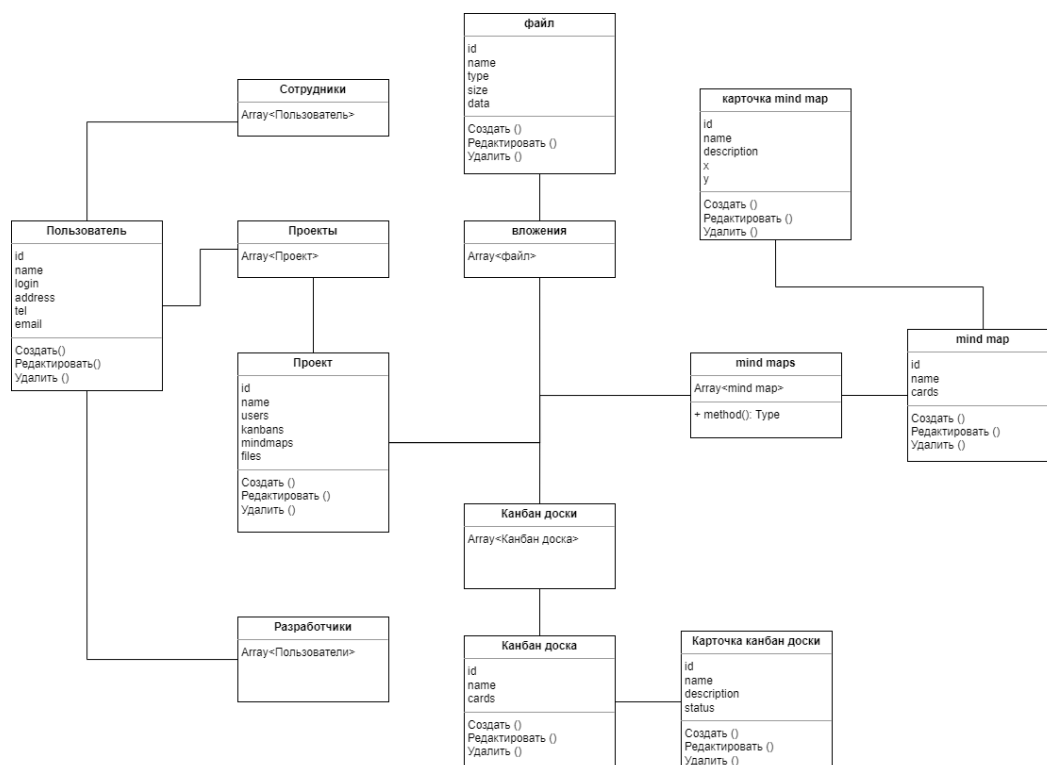


Рисунок 6 - Диаграмма классов

На диаграмме приведены основные классы: Пользователь, Проект, Файл, Канбан доска и Mind Map.

Класс «Пользователь» содержит данные о порядковом номере пользователя, его имя, логин и контактные данные. Данный класс так же используется в других классах в качестве типа свойств. Данный класс в проекте предназначен для описания пользователей в системе.

Класс «Проект» содержит данные о порядковом номере, названии проекта, список инструментов для управления проектом, вложения и список разработчиков которые участвуют в разработке данного проекта. Данный класс так же используется в других классах в качестве типа свойств. Данный класс в проекте предназначен для описания проекта в системе.

Класс «Файл» содержит информацию о порядковом номере, названии, типе файла, его объеме и дате загрузки. Данный класс так же используется в других классах в качестве типа свойств. Данный класс в проекте предназначен для описания файла в системе.

Классы «Канбан» и «Mind Map» описывают информацию об методах для управления проектом. Каждый из этих классов представляет информацию об своем методе управления. Каждый из этих классов имеет подчинённый ему класс «Задачи», который описывает поставленную задачу для разработчика.

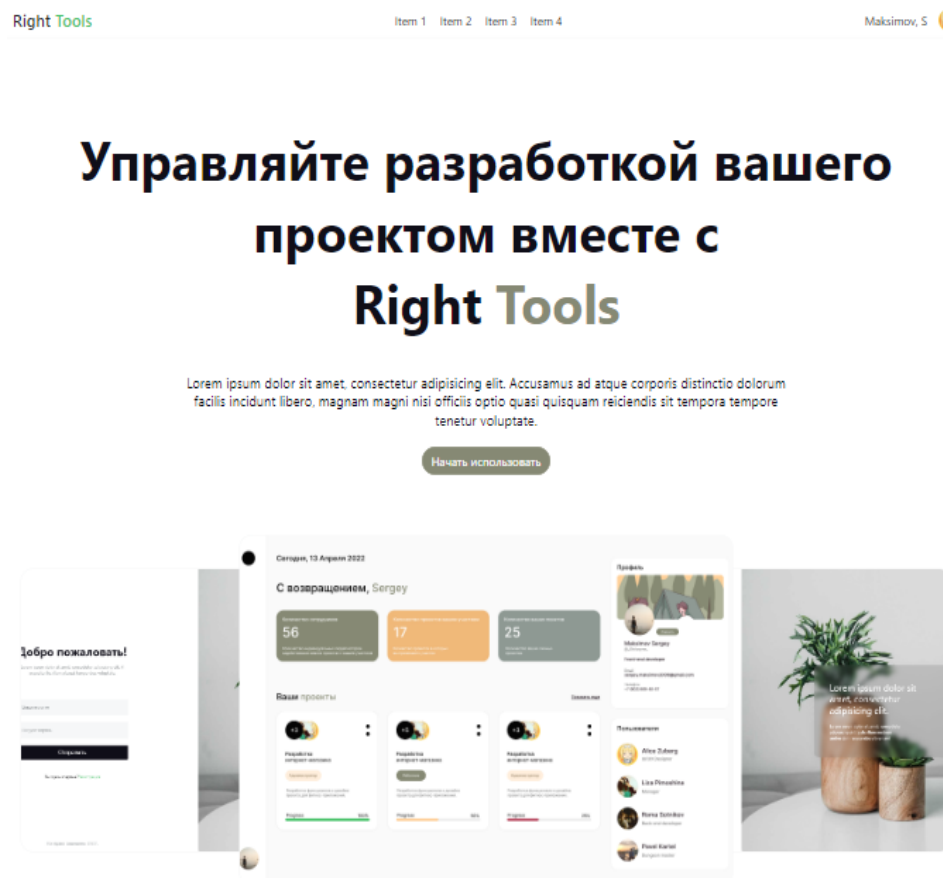
Диаграмма классов описывает структуру, типы и взаимодействия объектов в веб-приложении. Грамотно составленная диаграмма позволит написать front-end и back-end части без дополнительных изменений при финальных тестированиях.

					ОП-02068025-09.03.02-037.22	Лист
						21
Изм.	Лист	№ докум.	Подп.	Дата		

2.3 Разработка информационных спецификаций

Говоря про информационные спецификации, не стоит забывать про внешний вид пользовательского интерфейса. Изучив сторонние решения были спроектированы следующие макеты проекта.

На рисунке 7 продемонстрирован макет главной страницы.



Стало интересно? Немного о нашем проекте.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. A accusantium amet libero pariatur quaerat quam sapiente ut veritatis? A ab adipisci aliquam amet, animi consectetur cumque

Рисунок 7 - макет главной страницы

					ОП-02068025-09.03.02-037.22	Лист
						22
Изм.	Лист	№ докум.	Подп.	Дата		

Если пользователь вышел или не имеет аккаунта во системе, то его перенаправит на страницу входа в систему. Макет страницы входа в систему представлен на рисунке 8.

Right Tools

Добро пожаловать!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. A
expedita illo, illum placet temporibus voluptate.

Введите логин

Введите пароль

Отправить

Были здесь раньше? [Регистрация](#)

Все права защищены 2020.

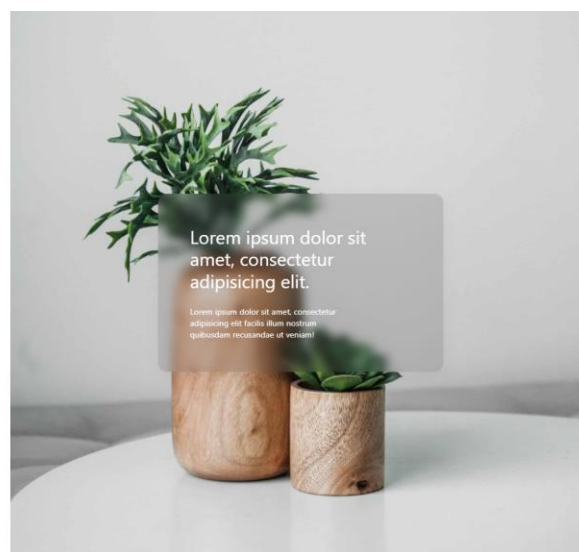


Рисунок 8 - Вход в систему

Когда пользователь авторизовался в системе то он попадает на страницу его профиля. Мак личного кабинета представлен на странице 9.

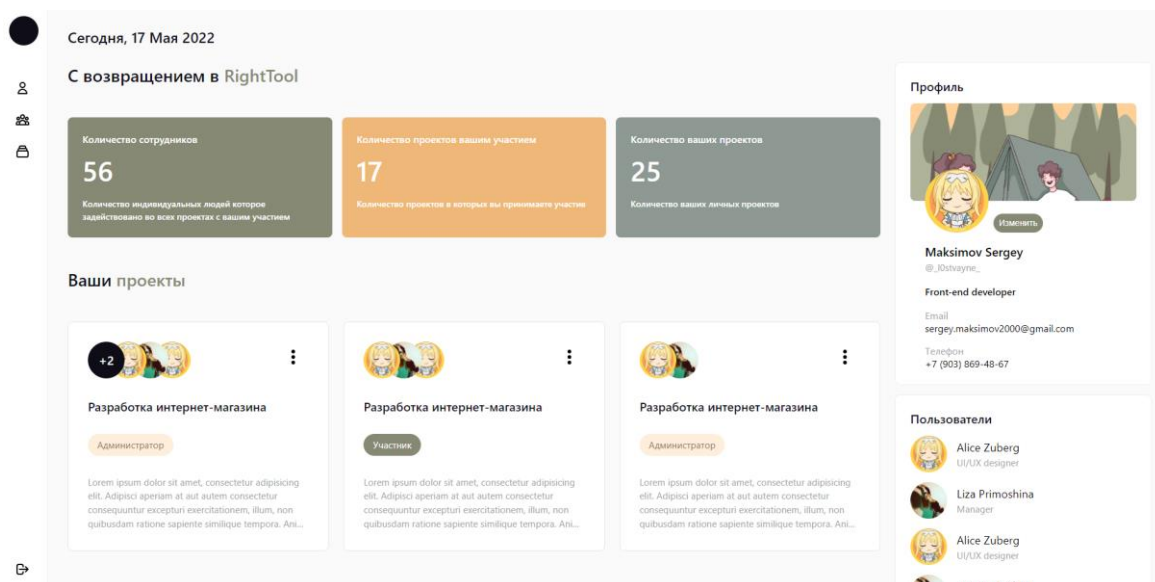


Рисунок 9 - Личный кабинет

У каждого разрабатываемого проекта есть общая информационная страница. Макет страницы проекта представлен на рисунке 10.

					ОП-02068025-09.03.02-037.22	Лист
						23
Изм.	Лист	№ докум.	Подп.	Дата		

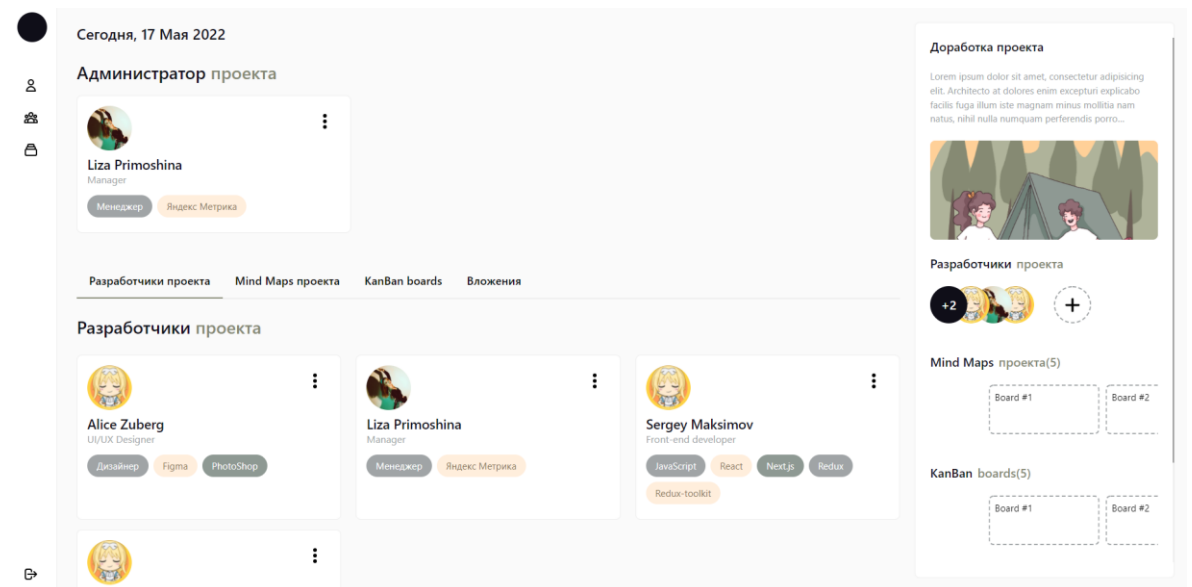


Рисунок 10 - Страница проекта

Каждый проект обладает разделами с Mind Map и канбан досками. Макеты разделов представлены на рисунке 11 и 12.

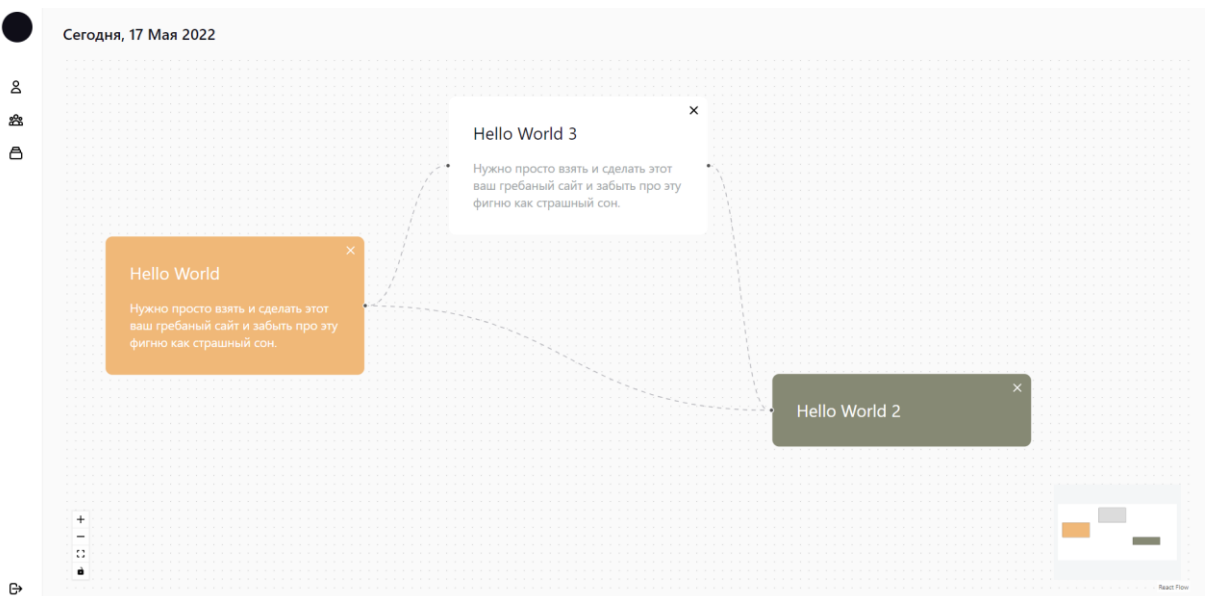


Рисунок 11 - Макет Mind Map

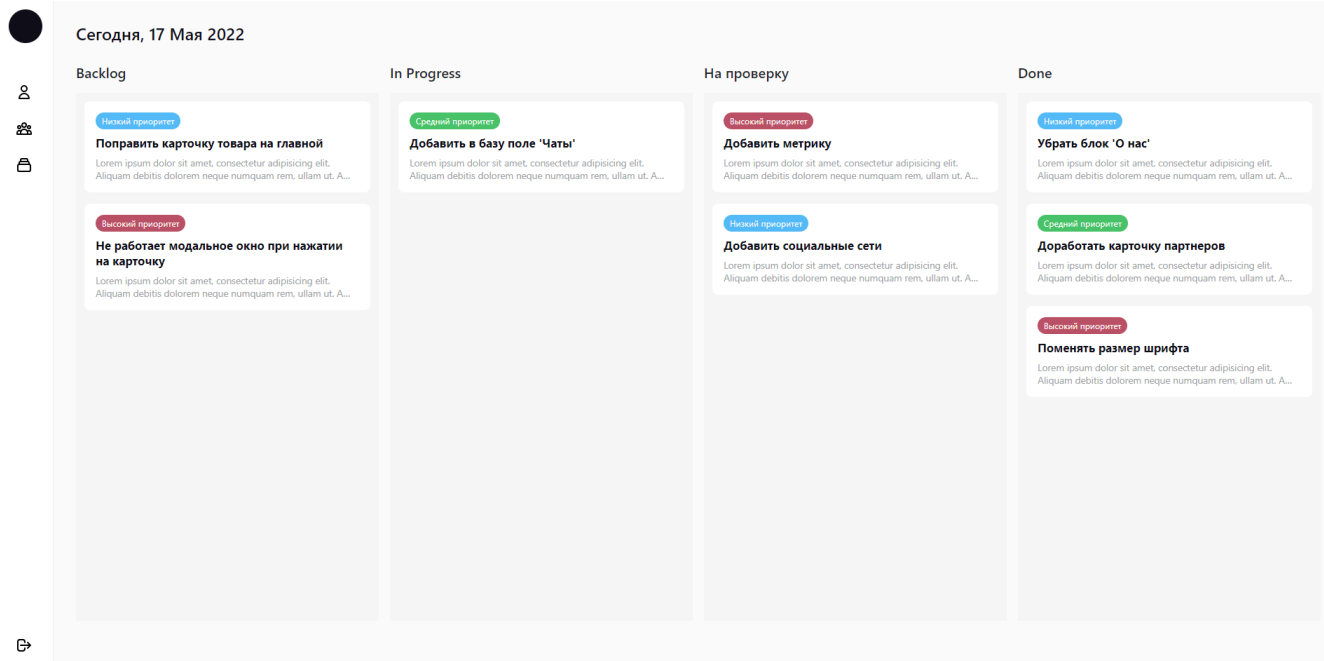


Рисунок 12 - Макет Канбан доски

2.4 Проектирование схемы развертывания разрабатываемого средства.

В данном разделе будет рассмотрен вопрос развёртывания проекта для просмотра и тестирования на наличие ошибок. Самым простым способом развернуть Next проект является Vercel.

Vercel — это платформа для интерфейсных фреймворков и статических сайтов, созданная для интеграции с вашим безголовым контентом, коммерцией или базой данных.

Vercel создан создателями Next.js и имеет первоклассную поддержку Next.js. При развертывании приложения Next.js в Vercel по умолчанию происходит следующее:

- Страницы, использующие статическую генерацию и активы (JS, CSS, изображения, шрифты и т. д.), будут автоматически обслуживаться из невероятно быстрой сети Vercel Edge Network;

- Страницы, использующие рендеринг на стороне сервера и маршруты API, автоматически становятся изолированными бессерверными функциями. Это позволяет бесконечно масштабировать отрисовку страниц и запросы API;

Vercel имеет много других функций, таких как:

- Пользовательские домены: после развертывания на Vercel вы можете назначить личный домен своему приложению Next.js. Взгляните на нашу документацию [здесь](#);

- Переменные среды: вы также можете установить переменные среды в Vercel . Взгляните на нашу документацию [здесь](#). Затем вы можете использовать эти переменные среды в своем приложении Next.js;

- Автоматический HTTPS: HTTPS включен по умолчанию (включая персональные домены) и не требует дополнительной настройки. Мы автоматически продлеваем SSL-сертификаты;

Благодаря интеграции Next приложений и Vercel, все что нам остается это подключиться к депозитарию GitHub.

Пример развертывания приложения представлен на рисунке 13.

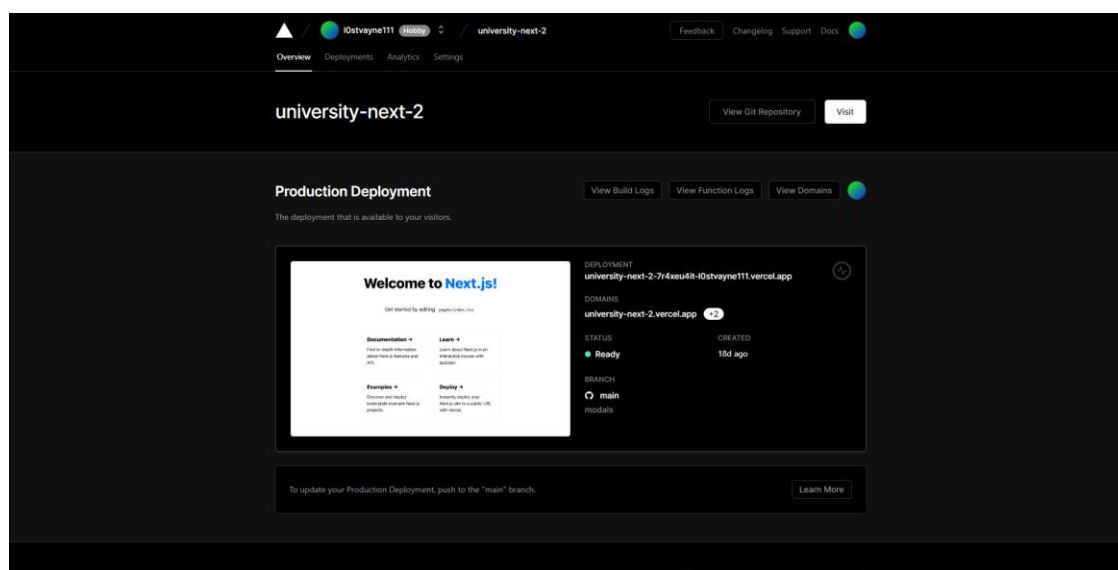


Рисунок 13 - Деплой приложения при помощи Vercel

После того как были установлены все зависимости мы можем открыть приложение с любого устройства для тестирования его на ошибки или работы в целом.

3. РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ

3.1 Реализация программных компонентов

Концепция React – приложений заключается в том, чтобы разбить приложения на как можно большое количество отдельных компонент для их переиспользования и достижения максимальной производительности. Компонент может быть большое количество, но их можно разбить на несколько основных категорий:

- Pages – тут хранятся контейнеры которые представляют собой страницы веб-приложения;
- Components – здесь находятся стандартные компоненты которые чаще всего встречаются в веб-приложении;
- Blocks – тут расположены крупные составляющие которые включают в себя обычные стандартные компоненты;
- Layout – тут расположены основные шаблоны страницы такие как шапка и подвал страницы;

Самым главным элементов приложения является НОС (Компонента высшего порядка) которая принимает и обрабатывает все остальные компоненты, так же он обрабатывает такие операции как переход на другую страницу или закрытие вкладки сайта. НОС представлен на рисунке 14.

```

type NextPageWithLayout = NextPage & {
  getLayout?: (page: ReactElement) => ReactNode,
  auth: boolean,
}

type AppPropsWithLayout = AppProps & {
  Component: NextPageWithLayout,
}

function MyApp({Component, pageProps}: AppPropsWithLayout) {
  const [isLoading, setIsLoading] = useState(false)
  const router = useRouter();

  useEffect(() => {
    const handleRouteChange = () => {
      setIsLoading(true);
      document.body.style.paddingRight = `${window.innerWidth - document.body.offsetWidth}px`;
      document.body.style.overflow = 'hidden';
    };
    const handleRouteComplete = () => {
      setIsLoading(false);
      document.body.style.overflow = 'auto';
      document.body.style.paddingRight = '0px';
    };
    router.events.on('routeChangeStart', handleRouteChange);
    router.events.on('routeChangeComplete', handleRouteComplete);
  }, [router.events]);

  // Use the layout defined at the page level, if available
  const getLayout = Component.getLayout || ((page) => page);
  return (
    <>
      <AnimatePresence exitBeforeEnter>
        {isLoading && <Preloader/>}
        {
          getLayout(<Component {...pageProps} />)
        }
      </AnimatePresence>
    </>
  )
}

export default MyApp

```

Рисунок 14 - НОС приложения

Чтобы была возможность что-то расположить какую-либо информацию на странице необходимо создать компоненты из категории «pages». Все компоненты по механике схожи друг с другом. Страницы представляют собой контейнеры, в которых мы можем использовать другие разработанные нами компоненты. Страница «Проекты» представлена на рисунке 15.

```

const Projects = () => {
  const [query, setQuery] = useState("")
  const handleOnSearch = (value: string) => {
    setQuery(value)
  }
  return (
    <motion.div
      variants={PageTransition}
      initial={initial}
      animate={animate} >
      <div className="row">
        <div className="col">
          <div>
            <Title value={'Список ваши проектов'}/>
          </div>
          <div className={'col-xl-8 my-5'}>
            <Search placeholder={'Поиск ...'} onSubmit={handleOnSearch}/>
          </div>
          <div className="row">
            {
              projects.filter(project => project.name.toLowerCase().includes(query)).map((project, index) => (
                <motion.div
                  variants={fadeUp}
                  initial={initial}
                  animate={animate}
                  custom={index}
                  key={project.id}
                  className="col-xxl-4 col-lg-6 mb-4">
                    <ProjectCard props={project}/>
                  </motion.div>
                ))
            }
          </div>
        </div>
      </div>
    </motion.div>
  );
};
Projects.getLayout = function getLayout(page: ReactElement) {
  return (
    <LayoutPanel title={'Проекты'}>
      {page}
    </LayoutPanel>
  )
}
export default Projects;

```

Рисунок 15 - Страница "Проекты"

Для экономии пространства при отображении информации в приложении был разработан механизм «аккордеон». Данный механизм является большим компонентом из категории «blocks». Данный компонент является переиспользуемым, достаточно лишь указать название кладок и указать контент который должен отображаться. Программный код аккордеона представлен на рисунке 16.

```

type ITabs = {
  tabs: Array<string>,
  children: ReactElement,
}

const Tabs: React.FC<ITabs> = ({tabs, children}) => {
  const [selectedTab, setSelectedTab] = useState<string>(tabs[0]);
  const [activeTab, setActiveTab] = useState<number>(0);
  // @ts-ignore
  const [state, setState] = useState<Array<any>>(React.Children.toArray(children.props.children))
  const handleOnTab = (data: string, index: number) => {
    setSelectedTab(data);
    setActiveTab(index)
  }
  useEffect(() => {
    setState(React.Children.toArray(children.props.children));
  }, [children])
  return (
    <div>
      <div>
        <ul className={style.tabs}>
          {
            tabs.map((tab:string, index: number) => (
              <li
                key={index}
                onClick={() => handleOnTab(tab, index)}>
                {tab}
                {tab === selectedTab && (<motion.div className={style.underline} layoutId="underline"/>)}
              </li>
            ))
          }
        </ul>
      </div>
      <div className={`mt-4`}>
        <AnimatePresence exitBeforeEnter>
          <motion.div
            key={selectedTab ? selectedTab : "empty"}
            animate={{opacity: 1, y: 0}}
            initial={{opacity: 0, y: 20}}
            exit={{opacity: 0, y: -20}}
            transition={{duration: 0.15}}
          >
            {state[activeTab]}
          </motion.div>
        </AnimatePresence>
      </div>
    </div>
  );
};

export default Tabs;

```

Рисунок 16 - Компонент "аккордеон"

Самой крупной по численности категорией компонент является «components» они являются одними из самых важных «кирпичиков» проекта. Примером такого компонента является стилизованная поисковая строка, которая принимает строку и возвращает результат ввода для дальнейшего использования в различных функциях. Программный код представлен на рисунке 17.


```

type ISearch = {
  placeholder: string,
  value?: string,
  onSubmit: (data: string) => void,
}
const Search: React.FC<ISearch> = ({placeholder, value= '', onSubmit}) => {
  const [word, setWord] = useState('');
  const [isDisabled, setIsDisabled] = useState(false)
  const handleOnSubmit = (event: FormEvent) => {
    event.preventDefault();
    onSubmit(word.toLowerCase())
  }
  const handleOnSearch = (event: FormEvent<HTMLInputElement>) => {
    let currencyWord: string = event.currentTarget.value;
    setWord(currencyWord);
  }
  useEffect(() => {
    setWord(value)
  }, [value])
  return (
    <form
      className={style.form}
      onSubmit={event => handleOnSubmit(event)}>
      <button
        disabled={isDisabled}
        type='submit'
        className={style.button}>
        <SearchIcon/>
      </button>
      <input
        onChange={event => handleOnSearch(event)}
        className={style.input}
        value={word}
        type="text"
        placeholder={placeholder}/>
    </form>
  );
};
export default React.memo(Search);

```

Рисунок 17 - Компонент "Поисковая строка"

Немаловажной частью веб-приложения являются файлы, со стилями, которые отвечают за внешний вид и адаптивность интерфейса. Для избежание ошибок связанных с одинаковым именем селектора был использован механизм модульных стилей, который встроен в библиотеку React. Для комфортной и быстрой разработки был использован препроцесс Sass (SCSS). Пример модульных стилей представлен на рисунке 18.

```

|.block {
  height: 100%;
  padding: 1rem;
  border-radius: var(--radius-value);
  border: 1px solid #EEEEEE;
  background: #fff;
  position: relative;
}
.avatar {
  width: 4.5rem;
  height: 4.5rem;
  border-radius: 50%;
  overflow: hidden;
  img {
    width: 100%;
    height: 100%;
    object-fit: cover;
  }
}
.name {
  font-size: 1.35rem;
  color: var(--black);
  font-weight: 500;
  margin-top: .5rem;
}
.status {
  margin-bottom: .95rem;
}
.tag {
  font-size: .95rem;
  padding: .5rem .95rem;
  border-radius: 500px;
  display: inline-block;
  margin-right: .5rem;
  margin-bottom: .5rem;
  &:nth-child(3n+2){
    background: var(--smokeyyellow);
    color: var(--darkwood);
  }
  &:nth-child(3n+1){
    background: var(--gray);
    color: #fff;
  }
  &:nth-child(3n+3){
    background: var(--smokegreen);
    color: #fff;
  }
}
.control {
  position: absolute;
  top: 0;
  right: 0;
  padding: 1rem;
}

```

Рисунок 18 - Модульные стили

3.2 Тестирование и апробация программного средства

В данном разделе будет протестировано разработанное приложение.

При запуске и при переходе на другие разделы веб-приложения открывается заставка. Заставка представляет собой обычные анимированные фигуры. Заставка загрузки экрана представлена на рисунке 19.



Рисунок 19 - Заставка веб-приложения

При запуске приложения пользователь попадает на стартовую страницу с краткой информацией о приложении. Так же через эту страницу возможен вход в профиль пользователя или на форму авторизации и регистрации. Главная страница представлена на рисунке 20.

Управляйте разработкой вашего проектом вместе с Right Tools

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Accusamus ad atque corporis distinctio dolorum facilis incidunt libero, magnam magni nisi officis optio quasi quisquam reiciendis sit tempora tempore tenetur voluptate.

Начать использовать



Рисунок 20 - Главная страница веб-приложения

Если пользователь не имеет аккаунта в системе или срок токена авторизации истек, то пользователя перенаправляет на страницу авторизации/регистрации. Данная страница содержит в себе формы с полями ввода с помощью которых можно войти в систему или зарегистрироваться в ней. Страница авторизации/регистрации представлена на рисунке 21.

Добро пожаловать!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. A expedita illo, illum placeat temporibus voluptate.

Введите логин

Введите пароль

Отправить

Вы здесь впервые? [Регистрация](#)

Все права защищены 2022г.

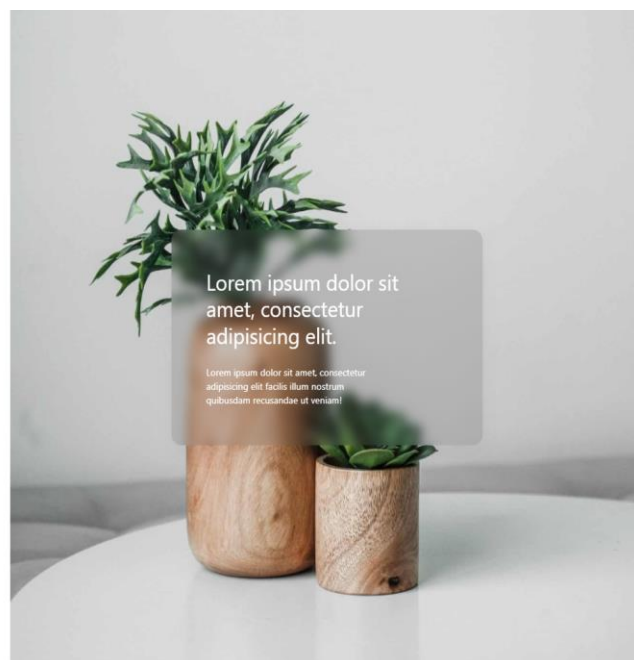


Рисунок 21 - Страница для входа в систему.

Изм.	Лист	№ докум.	Подп.	Дата

ОП-02068025-09.03.02-037.22

Лист

35

После того как пользователь вошел в систему его перенаправляет на страницу профиля пользователя. На данной странице представлена минимальная информация об аккаунте пользователя.

Такой информацией являются:

- информация о пользователе;
- первые три проекта в котором принимает участие пользователь;
- статистика профиля;
- другие пользователи, которые добавлены в коллекцию;

Страница профиля пользователя представлена на рисунке 22.

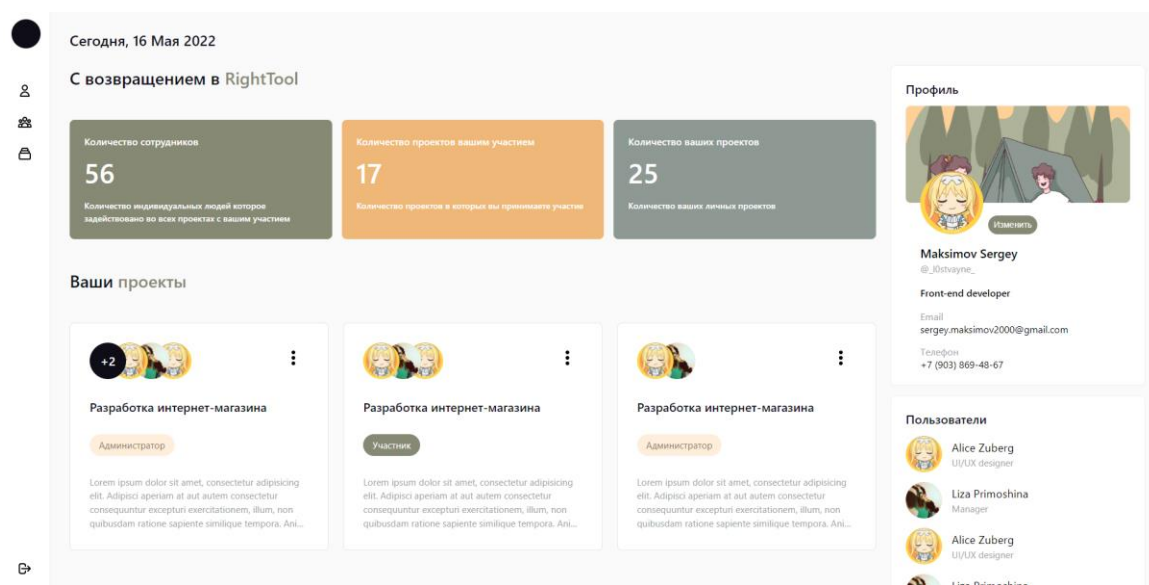


Рисунок 22 - Профиль пользователя

Одним из главных разделов веб-приложения является информационная страница проекта. На данной странице есть два основных поля – полная и быстрая информация.

Быстрая информация – это поле представляет собой боковую панель, на которой расположена компактная информация проекте. Через панель можно прочитать описание проекта, просматривать и создавать канбан и mindmap доски. Так же можно загружать и приглашать в проект файлы и пользователей.

На поле с полной информацией расположена полная информация о проекте. Информация о проекте включает в себя информацию о том, кто менеджер проекта,

список разработчиков, канбан доски, mindmaps и вложения. Для экономии пространства на устройстве вся информация сгруппирована при помощи механизма «аккордеон».

Информационная страница проекта представлена на рисунке 23.

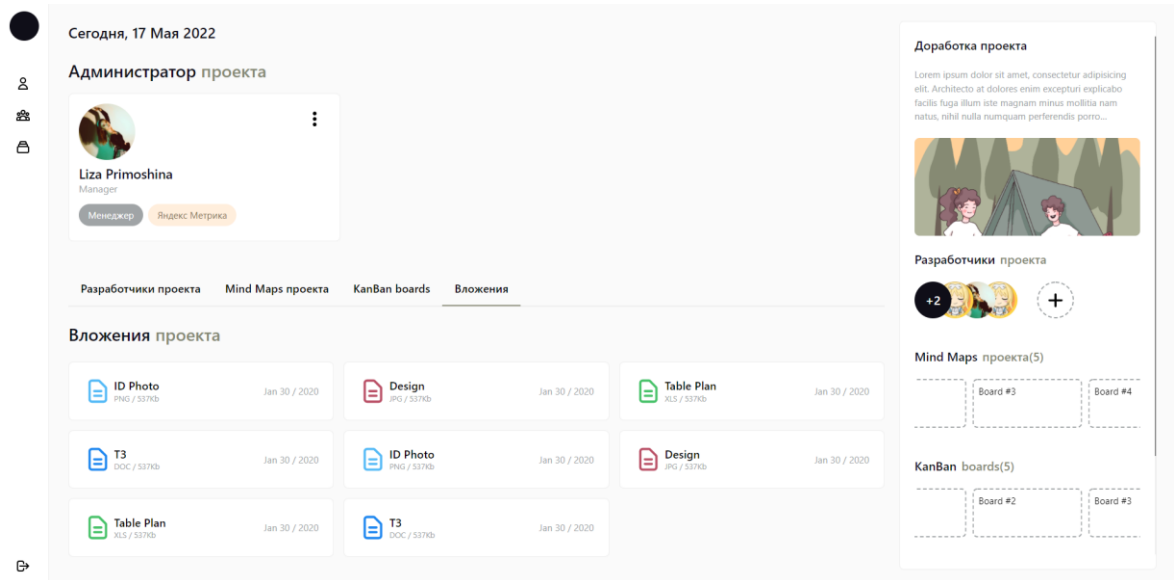


Рисунок 23 - Информационная страница проекта

Раздел о проекте так же содержит два подраздела которые отвечают за работу канбан досок и mindmap.

Раздел «Канбан доска» отвечает за работу канбан доски. Данный раздел представляет собой четыре колонки в которых расположены карточки с задачами. При выполнении разработчиком какого-либо задания разработчик перемещает карточку в определенную колонку с соответствующим статусом.

Страница с канбан доской представлена на рисунке 24.

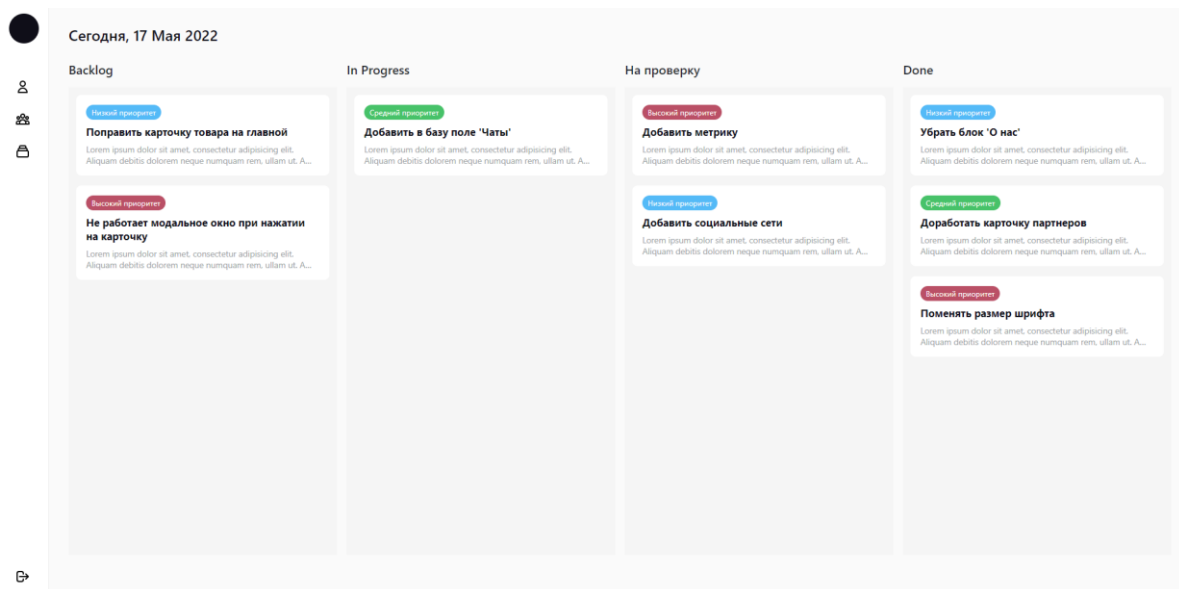


Рисунок 24 - Канбан доска

Раздел «MindMap» представляет собой страницу, на которой можно создавать различные графические схемы при разработке проекта.

Страница для построения схем представлена на рисунке 25.

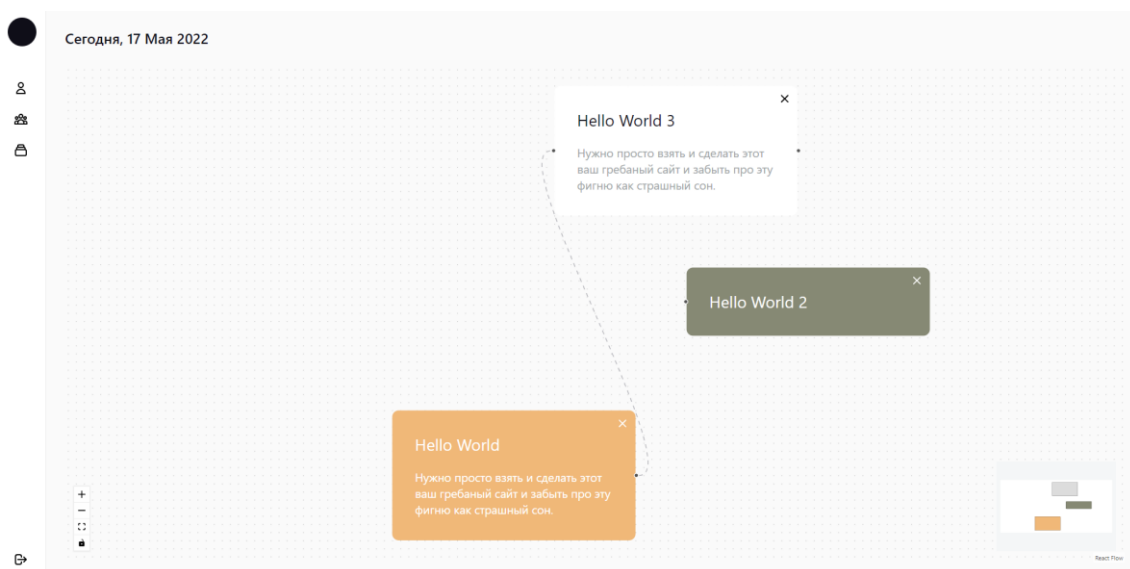


Рисунок 25 - "MindMap"

Тестирование веб-приложения не вывело никаких ошибок.

ЗАКЛЮЧЕНИЕ

При прохождении преддипломной практике не возникло никаких происшествий. Было интересно ознакомиться и примерить на себе роль front-end разработчика.

Во время прохождения практики было разработано веб-приложение по управлению проектами. Разработанное приложение обладает системами управления – канбан доски и mindmaps. Для разработки приложения было проделана большая работа по анализу научной литературы и статей, так же были получены практические навыки по применению полученных данных.

Во время прохождения практики был написан отчет с подробным описанием работы приложения и результатами тестирования на различных устройствах (Mi Book Air 13.3 и Ipad Air 2020).

Полученный опыт будет полезен при написании дипломной работы и разработки других веб-сервисов в будущем.

					ОП-02068025-09.03.02-037.22	Лист
						39
Изм.	Лист	№ докум.	Подп.	Дата		

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 React - JavaScript-библиотека для создания пользовательских интерфейсов: сайт. URL: <https://ru.reactjs.org/> (дата обращения 20.04.2022).
- 2 React Flow: сайт. URL: <https://reactflow.dev/> (дата обращения 20.04.2022).
- 3 freeCodeCamp: сайт. URL: <https://www.freecodecamp.org/news/how-to-add-drag-and-drop-in-react-with-react-beautiful-dnd/> (дата обращения 21.04.2022).
- 4 Next.js: сайт. URL: <https://nextjs.org/> (дата обращения 21.04.2022).
- 5 Redux: сайт. URL: <https://redux.js.org/> (дата обращения 22.04.2022).
- 6 Redux Toolkit: сайт. URL: <https://redux-toolkit.js.org/> (дата обращения 22.04.2022).
- 7 Tproger: сайт. URL: <https://tproger.ru/translations/redux-for-beginners/> (дата обращения 22.04.2022).
- 8 SkillFactory.Блог: сайт. URL: <https://blog.skillfactory.ru/glossary/redux/> (дата обращения 23.04.2022).
- 9 МедиаНетологии: сайт. URL: <https://netology.ru/blog/trello> (дата обращения 23.04.2022).
- 10 Физики и лирики: сайт. URL: <https://fizikiiliriki.ru/bitrix24/index.php> (дата обращения 24.04.2022).
- 11 Kislrod Ecommerce Agency: сайт. URL: <https://o2k.ru/blog/preimushchestva-bitrix> (дата обращения 24.04.2022).
- 12 Framer: сайт. URL: <https://www.framer.com/motion/> (дата обращения 25.04.2022).
- 13 Axios: сайт. URL: <https://axios-http.com/docs/intro> (дата обращения 25.04.2022).
- 14 8Host: сайт. URL: <https://www.8host.com/blog/kak-ispolzovat-axios-v-prilozhenii-javascript> (дата обращения 25.04.2022).

15 Community – Использование Axios с React: сайт. URL: <https://www.digitalocean.com/community/tutorials/react-axios-react-ru> (дата обращения 26.04.2022).

16 NextAuth: сайт. URL: <https://next-auth.js.org/> (дата обращения 26.04.2022).

17 Академия яндекса: сайт. URL: <https://academy.yandex.ru/posts/chto-takoe-react-i-kak-ego-osvoit> (дата обращения 27.04.2022).

18 Skillbox Media: сайт. URL: <https://skillbox.ru/media/code/reactjs-dlya-novichkov-v-programmirovanii-chto-eto-kak-ustroen-i-zachem-nuzhen/> (дата обращения 27.04.2022).

19 Хабр – Как использовать Axios в React: сайт. URL: <https://habr.com/ru/post/521902/> (дата обращения 28.04.2022).

20 TypeScript: сайт. URL: <https://www.typescriptlang.org/> (дата обращения 30.04.2022).

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг файла «_app.tsx»

```
import '../styles/globals.scss'
import '../styles/bootstrap-grid.min.css';
import type {ReactElement, ReactNode} from 'react'
import type {NextPage} from 'next'
import {AnimatePresence} from "framer-motion";
import type {AppProps} from 'next/app'
import { useRouter } from 'next/router';
import {useState, useEffect} from "react";
import Preloader from "../components/Preloader";

type NextPageWithLayout = NextPage & {
  getLayout?: (page: ReactElement) => ReactNode,
  auth: boolean,
}

type AppPropsWithLayout = AppProps & {
  Component: NextPageWithLayout,
}

function MyApp({Component, pageProps}: AppPropsWithLayout) {

  const [isLoading, setIsLoading] = useState(false)
  const router = useRouter();

  useEffect(() => {
    const handleRouteChange = () => {
      setIsLoading(true);
    }
  })
}
```

					ОП-02068025-09.03.02-037.22	Лист
						42
Изм.	Лист	№ докум.	Подп.	Дата		

```

        document.body.style.paddingRight = `${window.innerWidth - document.body.offsetWidth}px`;
        document.body.style.overflow = 'hidden';
    };
    const handleRouteComplete = () => {
        setIsLoading(false);
        document.body.style.overflow = 'auto';
        document.body.style.paddingRight = '0px';
    };
    router.events.on('routeChangeStart', handleRouteChange);
    router.events.on('routeChangeComplete', handleRouteComplete);
}, [router.events]);

// Use the layout defined at the page level, if available
const getLayout = Component.getLayout || ((page) => page);
return (
    <>
        <AnimatePresence exitBeforeEnter>
            {isLoading && <Preloader/>}
            {
                getLayout(<Component {...pageProps} />)
            }
        </AnimatePresence>
    </>
)
}

export default MyApp

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг файла «projects.tsx»

```
import React, {ReactElement, useState} from 'react';
import LayoutPanel from "../../layout/LayoutPanel";
import {motion} from "framer-motion";
import {fadeUp, PageTransition} from "../../motion";
import Title from "../../components/Panel/Title";
import ProjectCard from "../../components/Cards/ProjectCard";
import Search from "../../components/Search";

const Projects = () => {
  const [query, setQuery] = useState("")
  const handleOnSearch = (value: string) => {
    setQuery(value)
  }
  return (
    <motion.div
      variants={PageTransition}
      initial={`initial`}
      animate={`animate`}>
      <div className="row">
        <div className="col">
          <div>
            <Title value={'Список ваши проектов'}/>
          </div>
          <div className={`col-xl-8 my-5`}>
            <Search placeholder={'Поиск ...'} onSub-
mit={handleOnSearch}/>
          </div>
        </div>
      </div>
    </motion.div>
  )
}
```

```

        <div className="row">
            {
                projects.filter(project => project.name.toLowerCase().includes(query)).map((project, index) => (
                    <motion.div
                        variants={fadeUp}
                        initial={`initial`}
                        animate={`animate`}
                        custom={index}
                        key={project.id}
                        className="col-xxl-4 col-lg-6 mb-4">
                            <ProjectCard props={project}/>
                        </motion.div>
                    ))
            }
        </div>
    </div>
</motion.div>
);
};
Projects.getLayout = function getLayout(page: ReactElement) {
    return (
        <LayoutPanel title={'Проекты'}>
            {page}
        </LayoutPanel>
    )
}

export default Projects;

```

ПРИЛОЖЕНИЕ В

(обязательное)

Листинг файла «ProjectCard.tsx»

```
import React from 'react';
import style from './index.module.scss';
import DropDown from "../../DropDown";
import {motion} from "framer-motion";

type IUserCard = {
  id: number,
  avatar: string,
  name: string,
  status: string,
  tags: Array<{ id: number, name: string }>
}

const UserCard: React.FC<{ props: IUserCard }> = ({props}) => {

  const {id, status, name, avatar, tags} = props;

  return (
    <div className={style.block}>
      <motion.div
        whileHover={
          {
            width: '5.75rem',
            height: '5.75rem',
            transformOrigin: 'left top'
          }
        }
        className={style.avatar}>
```

					ОП-02068025-09.03.02-037.22	Лист
						46
Изм.	Лист	№ докум.	Подп.	Дата		

```

        {/* eslint-disable-next-line @next/next/no-img-element */}
        <img src={avatar} alt=""/>
    </motion.div>
    <div className={style.control}>
        <DropDown>
            <ul>
                <li>Подробнее</li>
                <li>Добавить</li>
            </ul>
        </DropDown>
    </div>
    <div className={style.name}>
        <span className="text-black">
            {name}
        </span>
    </div>
    <div className={style.status}>
        <span className={`text-gray`} >
            {status}
        </span>
    </div>
    <div className={`d-flex align-items-center flex-wrap`} >
        {
            tags.map((tag: { id: number, name: string }, index:
number) => (
                <motion.div
                    whileHover={
                        {
                            y: -5,
                            transformOrigin: 'left top'
                        }
                    }
                    key={index} className={style.tag}>
                        {tag.name}
                    </motion.div>

```



```
        ))
    }
    </div>
</div>
);
};

export default UserCard;
```

					ОП-02068025-09.03.02-037.22	Лист
						48
Изм.	Лист	№ докум.	Подп.	Дата		

ПРИЛОЖЕНИЕ Г (обязательное)

Листинг файла «Tabs.tsx»

```
import React, {ReactElement, useState, useEffect} from 'react';
import style from './index.module.scss';
import {motion, AnimatePresence} from "framer-motion";

type ITabs = {
  tabs: Array<string>,
  children: ReactElement,
}

const Tabs: React.FC<ITabs> = ({tabs, children}) => {

  const [selectedTab, setSelectedTab] = useState<string>(tabs[0]);
  const [activeTab, setActiveTab] = useState<number>(0);
  // @ts-ignore
  const [state, setState] = useState<Array<any>>(React.Children.toArray(children.props.children))

  const handleOnTab = (data: string, index: number) => {
    setSelectedTab(data);
    setActiveTab(index)
  }
  useEffect(() => {
    setState(React.Children.toArray(children.props.children));
  }, [children])
  return (
    <div>
      <div>
        <ul className={style.tabs}>
```

```

        {
            tabs.map((tab:string, index: number) => (
                <li
                    key={index}
                    onClick={() => handleOnTab(tab, index)}>
                    {tab}
                    {
                        tab === selectedTab && (<motion.div
className={style.underline} layoutId="underline"/>)
                    }
                </li>
            ))
        }
    </ul>
</div>
<div className={`mt-4`}>
    <AnimatePresence exitBeforeEnter>
        <motion.div
            key={selectedTab ? selectedTab : "empty"}
            animate={{opacity: 1, y: 0}}
            initial={{opacity: 0, y: 20}}
            exit={{opacity: 0, y: -20}}
            transition={{duration: 0.15}}
        >
            {state[activeTab]}
        </motion.div>
    </AnimatePresence>
</div>
</div>
);
};

export default Tabs;

```

ПРИЛОЖЕНИЕ Д

(обязательное)

Листинг файла (PanelInfo/index.module.scss)

```
.container {
  width: 420px;
}

.block {
  position: fixed;

  border-radius: var(--radius-value);
  border: 1px solid #EEEEEE;
  background: #fff;
  top: 2rem;
  right: 2rem;
  bottom: 2rem;
  width: inherit;
  padding: 1.5rem 0;
  & > div {
    padding: 0 1.5rem;
    height: 100%;
    overflow: hidden auto;
  }
}

.title {
  font-size: 1.25rem;
  font-weight: 500;
  color: var(--black);
}
```

					ОП-02068025-09.03.02-037.22	Лист
						51
Изм.	Лист	№ докум.	Подп.	Дата		

```

.description {
  color: var(--gray);
  font-size: .95rem;
  line-height: 1.5;
  overflow: hidden;
  text-overflow: ellipsis;
  display: -moz-box;
  -moz-box-orient: vertical;
  display: -webkit-box;
  -webkit-line-clamp: 4;
  -webkit-box-orient: vertical;
  box-orient: vertical;
}

.banner {
  width: 100%;
  height: 10rem;
  overflow: hidden;
  border-radius: var(--radius-value);
  img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    animation: scale 10s ease-in-out 1s infinite;
  }
}

.avatar,
.count,
.addUser{
  width: 3.75rem;
  height: 3.75rem;
  overflow: hidden;
  border-radius: 50%;
}

```

```
line-height: 1.25rem;
```

```
img {  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}  
}
```

```
.count {  
  background: var(--black);  
  color: #fff;  
  font-size: 1.25rem;  
  font-weight: 500;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  z-index: 4;  
}
```

```
.team {  
  display: flex;  
  align-items: center;
```

```
& > div {  
  &:not(:first-child) {  
    margin-left: -1.5rem;  
  }  
  &:first-child {  
    z-index: 4;  
  }  
  &:nth-child(2){  
    z-index: 3;  
  }  
}
```

					ОП-02068025-09.03.02-037.22	Лист
						53
Изм.	Лист	№ докум.	Подп.	Дата		

```

    &:nth-child(3){
        z-index: 2;
    }
    &:last-child{
        z-index: 1;
    }

}

}

.addUser {
    background: #fff;
    border: 2px dashed var(--gray);
    display: flex;
    align-items: center;
    justify-content: center;
    margin-left: 2rem !important;
    cursor: pointer;
    svg {
        width: 70%;
        height: 70%;
    }
}

.items {
    display: flex;
    overflow: auto hidden;
    flex-wrap: nowrap;
    align-items: center;
    width: 100%;
    padding-bottom: .5rem;
}

```

```

.item {
  height: 5rem;
  border: 2px dashed var(--gray);
  font-size: .95rem;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  border-radius: var(--radius-value);
  cursor: pointer;
  a {
    height: 100%;
    width: 100%;
    display: inline-block;
    padding: .5rem;
  }
}

```

```

@keyframes scale {
  from {
    transform: scale(1);
  }
  50% {
    transform: scale(1.05);
  }
  from {
    transform: scale(1);
  }
}

```