МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Кемеровский государственный университет»

Институт цифры

Кафедра цифровых технологий

**ОТЧЕТ**

**по производственной практике**

**(преддипломная практика)**

студента 4 курса

Раматшоев Эроншо Наврузшоевич

направление подготовки 09.03.03 Прикладная информатика

направленность (профиль) подготовки «Прикладная информатика в экономике»

Руководитель практики:

Солопова А.Н., доцент, к. т. н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“04” февраля 2023г

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2023 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc126415986)

[1 РАЗРАБОТКА СИСТЕМА ДЛЯ ОТСЛЕЖИВАНИЯ ЗАДАЧ И ПРОЕКТОВ 4](#_Toc126415987)

[1.1 Выбор среды разработки 4](#_Toc126415988)

[1.2 Выбор языка программирования 9](#_Toc126415989)

[1.2.1 Выбор систем управления реляционными базами данных 14](#_Toc126415990)

[1.3 Требования к защите информации от несанкционированного доступа 17](#_Toc126415991)

[1.4 Руководство пользователя с ролью «Пользователь» 19](#_Toc126415992)

[1.5 Руководство пользователя с ролью «Модератор» 24](#_Toc126415993)

[2 ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ПРОЕКТА 31](#_Toc126415994)

[2.1 Оценка финансовых затрат 31](#_Toc126415995)

[2.2 Влияние проекта на работу организации 34](#_Toc126415996)

[2.4 Перспективы развития 37](#_Toc126415997)

[ЗАКЛЮЧЕНИЕ 38](#_Toc126415998)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 39](#_Toc126415999)

# ВВЕДЕНИЕ

Преддипломная практика проходила в компании МБУ «Геоинформационный центр» в период с 9.01.2023 г. по 04.02.2023 г. в качестве практиканта в отделе информационных технологий.

**Целью прохождения преддипломной практики** был сбор информации и для написания выпускной квалификационной работы по разработке приложения «Система для отслеживания задач и проектов».

Руководителем практики была поставлена задача исследования автоматизируемого бизнес-процесса и разработка информационной системы.

Перед прохождением практики был пройден инструктаж по технике безопасности и технике пожарной безопасности. Проведено ознакомление: с Политикой информационной безопасности; приказом «Об обеспечении информационной безопасности», в котором разработан перечень мероприятий по обеспечению конфиденциальности информации в МБУ «Геоинформационный центр» с должностной инструкцией инженера*,* своими обязанностями, а также техническим и программным обеспечением отдела АСУ.

**Задачи преддипломной практики**:

* + - изучить методы и средства обеспечения информационной безопасности и защиты информации;
    - выбрать комплекс задач автоматизации
    - обосновать выбор средств разработки;
    - разработать интерфейс будущей информационной системы.

Отчет по преддипломной практике будет взят за основу при написании выпускной квалификационной работы.

Темой выпускной квалификационной работы была выбрана разработка программного модуля для отслеживания задач и проектов.

# 1 РАЗРАБОТКА СИСТЕМА ДЛЯ ОТСЛЕЖИВАНИЯ ЗАДАЧ И ПРОЕКТОВ

## 1.1 Выбор среды разработки

В последние годы во всем мире, и в частности в России, быстрыми темпами развиваются интернет и web-технологии. Большая доля населения имеет доступ к его ресурсам у себя дома и на рабочем месте.

Сейчас интернет является незаменимой частью нашей жизни. Он экономит наше время, дает легкий доступ к огромным знаниям, доступ к нужной информации сегодня можно получить за считанные секунды, всего лишь сделав запрос в браузерную строку. Люди общаются в социальных сетях, совершают быстрые покупки и оплачивают коммунальные услуги, не выходя из дома.

На сегодняшний день уже не встает вопрос: нужен сайт или нет, но очень актуальна тема, как сделать сайт максимально эффективным с точки зрения успешной коммуникации, автоматизации, коммерции и поддержания благоприятного имиджа.

Поэтому создание и поддержка веб-сайтов является актуальной темой для организации сегодня. Для создания веб-сайтов существует множество сред и инструментов разработки и с каждым годом их качества и количества увеличивается. У каждой из них есть свои плюсы и минусы. В качестве основных были выбраны следующие среды разработки:

1. VSCode - Текстовый редактор. VSCode (Visual Studio Code) - это текстовый редактор, который используется для написания кода и управления им. Текстовый редактор относится как к интерфейсному, так и к серверному инструменту. Один из самых популярных текстовых редактор на сегодняшний день, является наиболее используемым текстовым редактором согласно опросу разработчиков, Stack Overflow 2019 года. Он наиболее популярен из-за его бесконечных расширений, которые просто продолжают улучшаться. Его UX ничем не отличается, так как он очень настраиваемый. Предоставляет интеллектуальные дополнения, подсказки и предложения по коду, фрагменты кода, также включает встроенный отладчик, встроенное управление Git, интегрированный терминал с поддержкой нескольких терминалов.

Другими текстовыми редакторами являются Atom, Sublime Text.

1. GitHub - самый популярный и наиболее часто используемый менеджер репозиториев Git. Это так из-за его многочисленных особенностей и доступности.

GitHub - это место, где крупные компании, такие как Facebook, Google, Twitter, Microsoft и другие, размещают свои исходные коды, создают, отправляют и поддерживают свое программное обеспечение.На GitHub есть инструменты для отслеживания проблем, проверки кода, совместной работы, непрерывной интеграции и общего управления кодом. GitHub предназначен как для интерфейсной, так и для серверной веб-разработки. Как уже говорилось, GitHub - это менеджер репозиториев Git. а Git - это дистрибутивная система контроля версий. Система контроля версий, также называемая управлением исходным кодом, представляет собой программную утилиту, которая управляет и отслеживает изменения в вашем исходном коде. Git - наиболее используемая система контроля версий. [12]

Другими менеджерами репозиториев Git являются GitLab, BitBucket.[8]

1. NPM, который расшифровывается как Node Package Manager, представляет собой менеджер пакетов для JavaScript и NodeJS [28]

NodeJS - это среда выполнения JavaScript, которая в основном расширяет JavaScript до языка сценариев на стороне сервера.С помощью NodeJS мы можем выполнять JavaScript вне браузера. И с этим мы можем создавать потрясающие серверы для наших веб-сайтов, используя JavaScript. [30]

NPM - это то место, где мы можем найти пакеты для использования в наших приложениях, как интерфейсных, так и серверных. Эти пакеты также могут быть установлены в наши проекты через NPM CLI. [28]

Другими менеджерами пакетов для JavaScript и NodeJS являются yarn и pnpm. Yarn также является очень популярным менеджером пакетов, и разработчикам часто бывает трудно решить, что использовать между NPM и yarn. [29]

Для разработки серверного части были использовании следующие инструменты разработки:

1. Postman - это инструмент тестирования API, используемый для проверки соответствия API ожиданиям.. Это действительно приносит пользу командам, которые создают API. С такими инструментами вам не нужно создавать интерфейс, чтобы проверить, соответствуют ли ваши API ожиданиям. Инструмент тестирования API - это внутренний инструмент. [20]

Существуют некоторые API, такие как Facebook API и Spurwing API, которые позволяют проводить тестирование в их документации по API. Это может избавить вас от необходимости использовать инструмент тестирования API для тестирования их API. [27]

Кроме того, есть несколько очень хороших инструментов тестирования API, доступных в виде расширений в текстовом редакторе VSCode, таких как Thunder Client. [27]

1. Sequelize - это популярный ORM, созданный для Node.js. Взаимодействие с базами данных является общей задачей для серверных приложений. Обычно это делалось с помощью необработанных SQL-запросов, которые может быть сложно сконструировать, особенно для новичков в SQL или базах данных в целом. В конце концов появились объектно-реляционные картографы (ORM), разработанные для упрощения управления базами данных. Они автоматически отображают объекты (сущности) из нашего кода в реляционной базе данных, как следует из названия. Больше не надо писать необработанные SQL-запросы и выполнять их в базе данных. Предоставляя нам программный способ подключения нашего кода к базе данных и управления сохраненными данными, мы можем больше сосредоточиться на бизнес-логике и меньше на по ошибки SQL. [21]

Объектно-реляционное сопоставление(ORM) - это метод, который сопоставляет программные объекты с таблицами базы данных. Разработчики могут взаимодействовать с объектами вместо того, чтобы на самом деле писать какие-либо запросы к базе данных. Когда объект считывается, создается, обновляется или удаляется, ORM создает и выполняет запрос к базе данных под капотом.Еще одним преимуществом ORM является то, что они поддерживают несколько баз данных: Postgres, MySQL, SQLite и т.д. Если вы пишете приложение, используя необработанные запросы, будет сложно перейти к другой базе данных, потому что многие запросы нужно будет переписать. С ORM переключение баз данных выполняется самой ORM, и обычно все, что вам нужно сделать, это изменить одно или два значения в файле конфигурации. [21

Существует множество ORM узлов, в том числе популярный Bookshelf.js и TypeORM.Oн существует уже давно - с 2011 года. Он имеет тысячи звезд GitHub и используется множеством приложений. Из-за своего возраста и популярности он стабилен и имеет множество документации, доступной в Интернете. [23]

Следует отметить, что Sequelize основан на обещаниях, что упрощает управление асинхронными функциями и исключениями. Он также поддерживает все популярные диалекты SQL: PostgreSQL, MySQL, MariaDB, SQLite и MSSQL.

Для разработки серверного части были использовании следующие инструменты разработки:

React - популярная библиотека JavaScript, которая используется для создания пользовательских интерфейсов (как веб-, так и мобильных) и одностраничных приложений. С React набор предварительно написанных фрагментов кода можно использовать и повторно использовать для выполнения общих функций JavaScript. В частности, React позволяет нам создавать повторно используемые компоненты пользовательского интерфейса. Именно из этих компонентов состоит наш веб-сайт. Поддерживаются вспомогательные пакеты и библиотеки, которые делают возможным создание сложных приложений с помощью React. [1]

Другими библиотеками JavaScript являются Vue.js , AngularJS.

1. Redux, представляет собой библиотеку JavaScript с открытым исходным кодом, которая позиционирует себя как контейнер предсказуемого состояния для приложений JS”, который является централизованным, отлаживаемым и гибким. В деталях, он предназначен для того, чтобы помочь писать приложения, которые ведут себя стабильно и легко тестируются, и должны работать с любым уровнем пользовательского интерфейса, предлагая большой выбор дополнений.

В основе Redux также лежат принципы функционального программирования, в то время как MobX основан на прозрачном функциональном реактивном программировании. По этой причине и из-за различных парадигм, на которые опираются эти подходы, Redux более требователен и сложен в освоении, в то время как MobX просто легче освоить. [1]

И MobX, и Redux имеют свои собственные инструменты разработки. Redux предоставляет обширный инструментарий, который поставляется с потрясающей документацией, охватывает множество вариантов использования и примеров, которые очень полезны в повседневной работе разработчика программного обеспечения, и помогает разработчикам упростить код своих приложений, позволяя пользователю упростить код своего приложения”. [1]

Стоит отметить, что обе части комбинации MobX Redux используются для мониторинга состояния интерфейса, в частности, в приложениях JavaScript. Еще одна очевидная особенность, которую они оба разделяют, заключается в том, что эти универсальные библиотеки являются внешними. Кроме того, они оба были первоначально выпущены в 2015 году. [1]

1. Figma - это инструмент проектирования, который помогает с макетами, прототипированием и тестированием. Хотя Figma, возможно, и не самый лучший инструмент для проектирования, но это лучший вариант для начала. Помимо личного использования, Figma надежна благодаря возможности совместной работы в режиме реального времени. Как команда дизайнеров, вы можете использовать Figma для совместной работы и создания замечательных вещей. [18]

Другие инструменты дизайна, которые могут быть лучшими вариантами для личного и профессионального использования, - это Adobe XD, InVision Studio, Sketch.

1. Благодаря таким инструментам, как Sass (синтаксически потрясающая таблица стилей), простота и удобочитаемость CSS повысились. Sass - это препроцессор CSS, который позволяет нам использовать логику в CSS. [24]

Препроцессоры CSS - это программы, которые добавляют новые функции в CSS, такие как переменные, вложенность, миксины, селектор наследования, функции и математические операции. Другие препроцессоры CSS - это Less и Stylus. [24]

## 1.2 Выбор языка программирования

Для любого веб-разработчика разработка клиентской части начинается с html хот он и не является языком программированием.

HTML - это язык разметки, поэтому он использует простые теги для разметки и форматирования содержимого. Эти теги заключены в угловые скобки типа <html . Почти все теги также имеют свои закрывающие теги. Тег <html> сообщает браузеру, что HTML-документ был запущен, аналогично </html> сообщает конец HTML-документа. Все вещи или код, написанные внутри этих двух тегов, представляются отправленными в браузер. Затем браузеры отображают содержимое, заключенное в тег body. HTML определяет формат отображения веб-элементов. Чтобы увидеть ваш контент, вам нужно сохранить файл с помощью. html расширение, а затем просто запустите его, выбрав опцию для открытия в любом браузере.

Преимущества HTML:

* HTML очень прост в изучении и понимании. HTML - это первый и самый важный язык, который изучает человек, изучающий веб-разработку. Он имеет простые теги, и в HTML нет суеты с учетом регистра. В нем просто есть несколько тегов, которые служат определенной цели, и все. Один может легко понять чужой код и при необходимости внести в него изменения, поскольку в нем не так много всего, что нужно понять. Более того, он не выдает никаких ошибок и не создает никаких проблем;
* HTML поддерживает практически все браузеры по всему миру. Таким образом, нет необходимости беспокоиться о веб-сайте, написанном на HTML для поддержки браузера, поскольку веб-сайт будет легко отображаться во всех браузерах, если программа будет учитывать необходимость оптимизации веб-сайта для разных браузеров. HTML предоставляет веб-разработчикам простой способ оптимизировать веб-сайт в формате HTML в соответствии с браузерами;
* HTML очень прост в редактировании, так как для его редактирования не требуется специального интерфейса или платформы. Он написан в простом блокноте и, следовательно, может быть просто отредактирован в любом текстовом редакторе, таком как Visual Studio Code и т.д;
* HTML может быть легко интегрирован с несколькими языками и не создает никаких проблем в нем. Например, в Javascript, Php, node.js , CSS и многие другие, мы пишем код этих языков между HTML, и он очень легко смешивается с ними;
* Одним из самых больших преимуществ HTML является то, что можно мгновенно увидеть изменения, просто сохранив его и перезагрузив предыдущую HTML-страницу. В отличие от других языков программирования, нет необходимости запускать весь код и выяснять, где ошибка. Например, если вы выделили слово курсивом, оно сразу же появится на странице после сохранения и перезагрузки;
* HTML используется в интерфейсной разработке уже более стольких лет, прежде чем на рынке веб-разработки не осталось других доступных языков. Хотя HTML предоставляет пользователю все теги для добавления всего на веб-страницу, например, таблицы, изображений, гиперссылок и т. Д. были некоторые недостатки, которые были учтены в последней версии HTML, то есть HTML5, которая позволяет пользователю вставлять графические, мультимедийные, семантические элементы для разработки мощных веб-сайтов и постоянного улучшения пользовательского интерфейса. [30]

Для стилизации HTML кода был выбран CSS (каскадные таблицы стилей).

CSS называют языком оформления веб страниц, то есть работу с цветами, шифрами и т.д. Этот язык был создан для разделения разметки содержимого, написанного на HTML. Результатом разметки является улучшенная доступность сайта, гибкость и уменьшение сложности. Каскадные таблицы стилей (CSS): CSS определяется как язык таблиц методов, который предоставляет веб-дизайнерам контроль над тем, как интернет-сайт взаимодействует с веб-браузерами, включая форматирование и отображение их HTML-документов. [7]

CSS или каскадные таблицы стилей может быть текстовым языком стилей, который определяет форматы веб-сайтов и способ взаимодействия сайта с веб-браузерами. Язык позволяет веб-разработчикам регулировать различные элементы стиля и функциональные возможности, такие как макет, цвет, шрифты и, следовательно, форматирование и отображение HTML-документов. Используя CSS, можно управлять цветом текста, дизайном шрифтов, интервалами между абзацами, размером и расположением столбцов и т.д. [7]

CSS инструктирует отображение HTML о том, как веб-сайт будет отображаться в конце пользователя.

Недостатки CSS:

CSS, от CSS 1 до CSS3, приводят к созданию путаницы среди веб-браузеров. С помощью CSS то, что работает с одним браузером, не всегда может работать с другим. Веб-разработчикам необходимо протестировать совместимость, запустив программу в нескольких браузерах;

После внесения изменений нам необходимо подтвердить совместимость, если они появятся. Подобное изменение затрагивает все браузеры;

Мир языка программирования сложен для не-разработчиков и новичков. Различные уровни CSS, то есть CSS, CSS 2, CSS 3, часто довольно запутанны;

Совместимость с браузером (некоторые таблицы стилей поддерживаются, а некоторые нет);

CSS работает по-разному в разных браузерах. IE и Opera поддерживают CSS как разную логику;

Существует несколько уровней, которые создают путаницу для не-разработчиков и новичков;

В качества основного языка программирование как для серверного так и для клиентской части был выбран JavaScript

JavaScript – это язык программирования, который используется в веб-браузерах, на нем можно писать игры и различные приложения для компьютера и мобильный устройств. Многие из них связаны с тем, как JavaScript часто выполняется непосредственно в браузере клиента, обычно используемом в веб-разработке. Изначально он был разработан Netscape как способ отображения динамических и интерактивных элементов на веб-сайтах. На JavaScript влияет Java с синтаксисом, похожим на C. JavaScript соответствует спецификациям ECMAScript, которые были разработаны Sun Microsystems. [7]

JavaScript может быть языком сценариев на стороне клиента, который предполагает, что текстовый файл ASCII обрабатывается браузером клиента, а не на онлайн-сервере. Это может загрузить веб-страницу без связи с основным сервером с помощью JavaScript. Например, функция JavaScript может проверять интернет-форму перед ее отправкой, чтобы убедиться, что все указанные поля заполнены. Код JavaScript может выдавать сообщение об ошибке до того, как какая-либо информация действительно будет передана на сервер.

Подобно серверным языкам сценариев, таким как PHP и ASP, код JavaScript часто вставляется в любое место HTML веб-страницы. Выходные данные серверной части отображаются в формате HTML, но код JavaScript остается видимым в исходном коде веб-страницы. Файл может быть отдельным файлом “.js”, который может отображаться в браузере. JavaScript имеет некоторые преимущества и недостатки

Преимущества JavaScript:

* Независимо от того, где вы размещаете JavaScript, он всегда выполняется в клиентской среде, чтобы сэкономить большую пропускную способность и ускорить процесс выполнения;
* Самым большим преимуществом JavaScript является способность поддерживать все современные браузеры и выдавать эквивалентный результат;
* Глобальные компании поддерживают развитие сообщества, создавая важные проекты. Примером может служить Google (создал Angular framework) или Facebook (создал React.js);
* JavaScript прекрасно сочетается с другими языками и может быть использован в огромном количестве приложений;
* Существует множество проектов с открытым исходным кодом, которые предоставляют разработчикам полезную помощь в добавлении JavaScript.
* Начать работать на JavaScript не сложно. По этой причине многие из нас предпочитают начинать свое приключение в ИТ-секторе с изучения этого языка;
* Это дает возможность создавать богатые интерфейсы.

Недостатки JavaScript:

* Это может быть сложно при разработке больших приложений, хотя вы также будете использовать наложение TypeScript;
* Настройка часто является утомительной задачей из-за количества инструментов, которые необходимо собрать вместе, чтобы создать среду для такого проекта;
* Основная проблема или недостаток JavaScript заключается в том, что код всегда виден всем, любой может просмотреть код JavaScript;
* Независимо от того, какую пропорцию интерпретирует fast JavaScript, JavaScript DOM (объектная модель документа) работает медленно и может быть никогда не быстрым рендерингом с HTML;
* Если ошибка возникает в JavaScript, она может остановить рендеринг всего веб-сайта;
* JavaScript обычно по-разному интерпретируется разными браузерами;
* Хотя некоторые редакторы HTML поддерживают отладку, это не так эффективно, как другие редакторы, такие как редакторы C / C ++;
* Это непрерывное преобразование занимает больше времени, чем преобразование числа в целое число.

Одной из основных недостатков JavaScript можно назвать отсутствие строгой типизации как в остальных языках высокого уровня. Для решение данной проблемы был выбран TypeScript

TypeScript - это надмножество JavaScript. Это означает, что TypeScript - это язык, построенный на JavaScript. Он основан на синтаксисе, конструкциях, преимуществах (и недостатках) JavaScript, но также содержит новые функции, синтаксис и возможности. [14]

TypeScript предоставляет разработчикам некоторые функции, которые помогают писать код на JavaScript проще и безопаснее. Одной из наиболее важных функций, предоставляемых TypeScript, является статическая типизация. В принципе, статическая типизация позволяет нам делать меньше ошибок с типами данных. Например, вы не можете указать некоторое значение в качестве аргумента функции, если это значение не имеет того же типа, что и параметр. Это очень простой материал, но у TypeScript есть и более мощные функции, о которых мы узнаем в следующих сообщениях этой серии. [14]

TypeScript имеет возможность работать с тем же API и средами, что и JavaScript, например, с API браузера или Node. Однако веб-браузер и узел не могут выполнить TypeScript, поскольку они работают только с JavaScript. Итак, как мы можем использовать TypeScript, если мы не можем выполнить код в средах JavaScript, таких как веб-браузер или узел?

TypeScript - это не просто язык, а мощный инструмент, компилятор, который может компилировать (преобразовывать) код, написанный на TypeScript, в код JavaScript. И этот скомпилированный код JavaScript мы можем выполнить в браузере или узле. Компилятор TypeScript преобразует код с функциями, доступными только в TypeScript, в общий код JavaScript. Все типы и другие конструкции TypeScript не будут присутствовать в коде, который вы будете выполнять в среде, потому что они не существуют в JavaScript.

### 1.2.1 Выбор систем управления реляционными базами данных

Реляционные базы данных используются уже давно. Они стали популярными благодаря системам управления, которые очень хорошо реализуют реляционную модель, которая зарекомендовала себя как отличный способ работы с данными [особенно для критически важных приложений].

В качестве основного языка для работы с базами данными был выбран PostgreSQL

PostgreSQL - это продвинутая система управления объектно-реляционными базами данных с открытым исходным кодом, главная цель которой - быть совместимой со стандартами и расширяемой. PostgreSQL, или Postgres, пытается принять стандарты ANSI / ISO SQL вместе с изменениями.

По сравнению с другими СУБД, PostgreSQL отличается своей поддержкой крайне необходимых и неотъемлемых функций, объектно-ориентированных и/или реляционных баз данных, таких как полная поддержка надежных транзакций, т.е. атомарности, согласованности, изоляции, долговечности (ACID). Благодаря мощной базовой технологии Postgres чрезвычайно эффективно справляется со многими задачами. Поддержка параллелизма достигается без блокировок чтения благодаря реализации (MVCC), которая также обеспечивает соответствие ACID. [13]

PostgreSQL является высоко программируемым и, следовательно, расширяемым с помощью пользовательских процедур, которые называются “хранимыми процедурами”. Эти функции могут быть созданы для упрощения выполнения повторяющихся, сложных и часто требуемых операций с базой данных. Хотя эта СУБД не пользуется такой популярностью, как MySQL, существует множество замечательных сторонних инструментов и библиотек, которые предназначены для упрощения работы с PostgreSQL, несмотря на мощный характер этой базы данных. В настоящее время можно легко получить PostgreSQL в виде пакета приложений через менеджер пакетов по умолчанию во многих операционных системах. [13]

Преимущества PostgreSQL

* СУБД, совместимая со стандартом SQL с открытым исходным кодом;
* PostgreSQL поддерживается преданным и опытным сообществом, доступ к которому можно получить через базы знаний и сайты вопросов и ответов 24/7 бесплатно;
* Несмотря на чрезвычайно продвинутые функции, PostgreSQL украшен множеством отличных сторонних инструментов с открытым исходным кодом для проектирования, управления и использования системы управления;
* Можно программно расширить PostgreSQL с помощью хранимых процедур, как и должно быть в продвинутой СУБД.

Недостатки PostgreSQL

* Для простых операций с большим объемом чтения PostgreSQL может быть перегружен и может показаться менее производительным, чем аналоги, такие как MySQL;
* Учитывая природу этого инструмента, ему не хватает популярности, несмотря на очень большое количество развертываний, что может повлиять на то, насколько легко можно получить поддержку;
* Из-за вышеупомянутых факторов сложнее найти хосты или поставщиков услуг, которые предлагают управляемые экземпляры PostgreSQL. [13]

## 1.3 Требования к защите информации от несанкционированного доступа

Большинство современных приложений "Программное обеспечение " предлагают различные роли, в которых каждому ролью присвоен набор операций, при которых пользователи другой роли не может получить доступ к операциям этого роля. Чтобы контролировать это необходимо реализовать авторизацию, которая ограничивает доступ пользователей к несанкционированным ресурсам.

Так же авторизация позволяет получить баланс между безопасностью и прогрессом бизнеса, играет жизненно важную роль в обеспечении безопасности вашего приложения.

Для взаимодействия с приложения "Программное обеспечение" используется Протокол HTTP. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером.

HTTP - это протокол без состояния, который означает, что новый запрос ничего не помнит о предыдущем. Итак, для каждого запроса вам необходимо войти в систему и пройти проверку подлинности Решением для решения этой проблемы является использование так называемого сеанса. Сеанс - это объект, хранящийся на сервере, который помогает пользователю оставаться в системе или сохранять любую ссылку на свою учетную запись.

Сначала пользователь вводит имя пользователя и пароль, которые проходят проверку подлинности на сервере. Если аутентификация прошла успешно, для соответствующего клиента генерируется идентификатор сеанса. Сгенерированный идентификатор сеанса возвращается клиенту и также сохраняется на стороне сервера.

Теперь клиенту просто нужно отправить свой идентификатор сеанса вместе с запросом для аутентификации и получения необходимой информации. Затем сервер проверит, действителен ли идентификатор сеанса или нет. Если сеанс все еще действителен, он ответит запрошенной веб-страницей / данными. И если нет, сервер ответит сообщением об ошибке, в котором указано, что сделанный запрос является несанкционированным.

Масштабируемость: сеансы должны храниться на сервере либо в памяти, либо в базе данных. В нынешнюю эпоху API сервер может получать большое количество запросов, и, следовательно, сервер необходимо масштабировать. Добавление новых ресурсов также может привести к усложнению инфраструктуры.

Сервер должен управлять сеансами, отслеживая все активные, неактивные (истекшие, недействительные) сеансы. Истекшие или недействительные сеансы необходимо удалить из памяти. Производительность: для каждого запроса серверу необходимо выполнить поиск в памяти, чтобы проверить, действителен ли предоставленный объект сеанса или нет. Это может привести к снижению производительности сервера.

Лучшее и эффективное решение является Веб-токен JSON (JWT) так как он не использует сеансы и, следовательно, предотвращает вышеупомянутые проблемы. Когда вы отправляете свои учетные данные на сервер вместо создания сеанса, сервер возвращает веб-токен JSON. Вы можете использовать этот JWT, чтобы делать с сервером все, что захотите. [26]

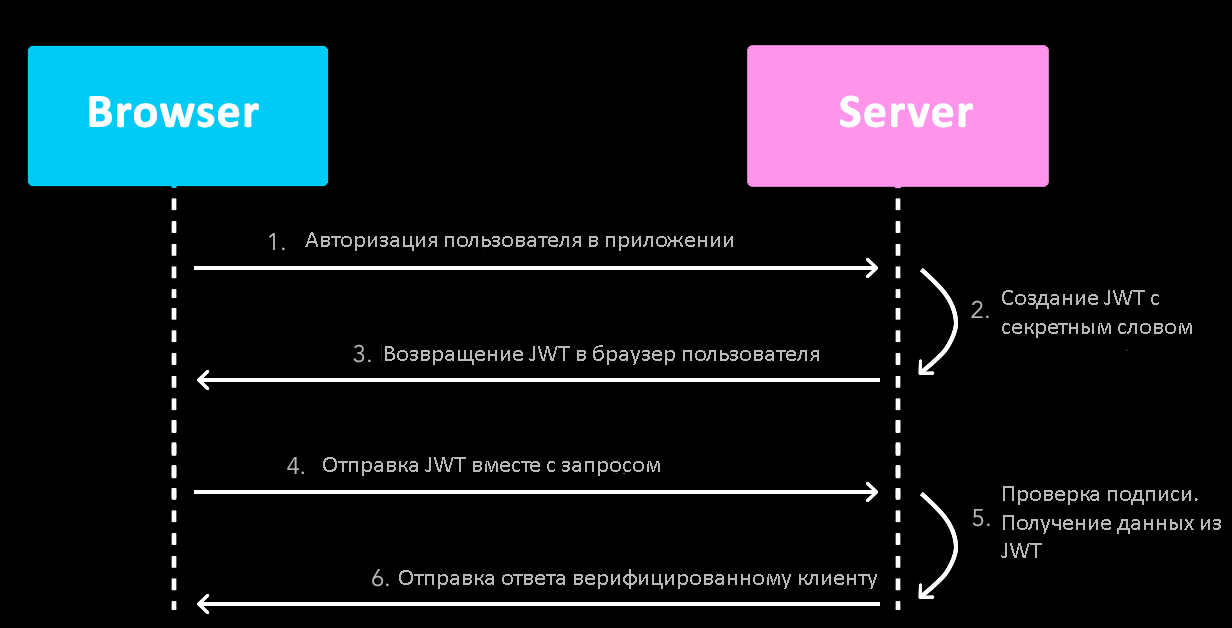


Рисунок 1 – Схема работы JWT

## 1.4 Руководство пользователя с ролью «Пользователь»

Система для отслеживания задач и проектов имеет веб-интерфейс, поэтому запускается в одном из установленных интернет-браузеров, например, Internet Explorer, Mozilla Firefox, Chrome и др. Для наиболее корректной работы программы рекомендуется использовать браузер Mozilla или Chrome. Для запуска программы на компьютере-сервере откройте меню

После запуска системы в браузере появится диалог «Авторизация». пользователю необходима пройти авторизацию. (рис.2).

При первом входе в систему пользователю необходимо ввести логин и пароль, которые необходимо получить при регистрации. При необходимости в дальнейшие использования пароль и логин можно сменить.

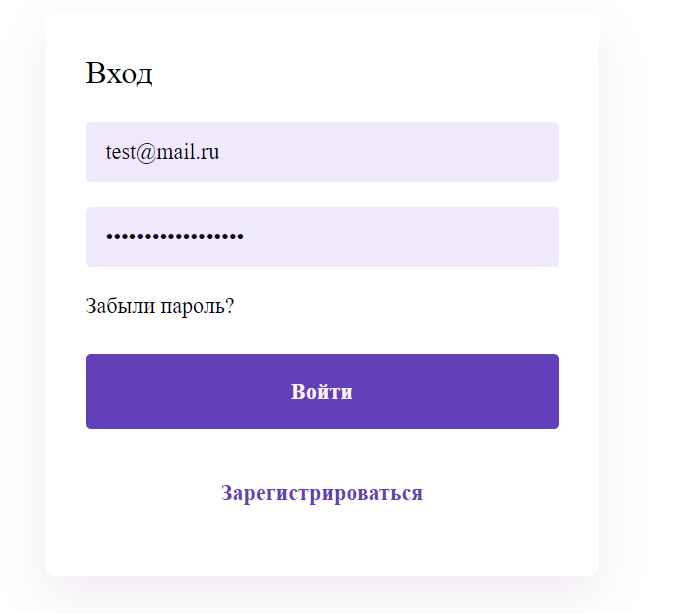


Рисунок 2 – Страница авторизации

Если пользователь входит в систему, собирается входить, то для начала ему необходимо пройти регистрацию (рис. 3).

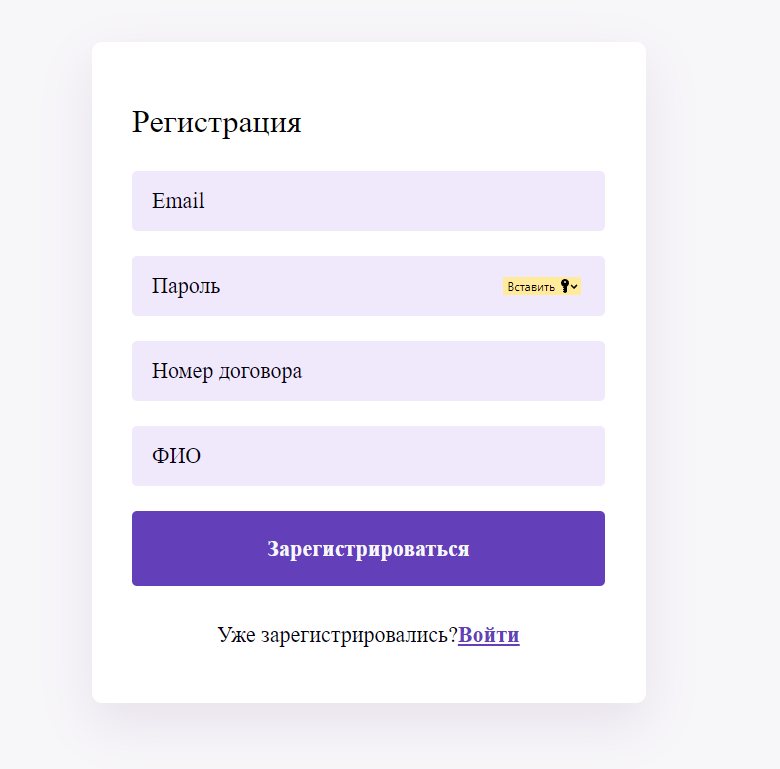


Рисунок 3 – Страница регистрация в роли обычного пользователя

Для успешного прохождения регистрации у пользователя должен быть действующий договор с организацией; договор заключается на определённый период, в течение которого пользователь может пользоваться системой(рис.5).

Договор заключаться на бумажном настели и дублируется модератором в базу данных для дальнейшей работы с организацией. В договоре указывается вся необходимая информация об организации. После прохождения регистрации на почту приходит письмо для подтверждения электронной почты, которая была указана при регистрации. Только в случае успешного подтверждения почты пользователь может зайти в свою учетную запись, используя свой логин и пароль.

Если пользователь забыл пароль, он может его восстановить на странице входа нажав кнопку «Забыли пароль?». Тогда пользователь перейдет на страницу восстановления пароля (рис. 4). Для восстановления пароля пользователю необходимо воспользоваться электронной почтой, привязанной к аккаунту.

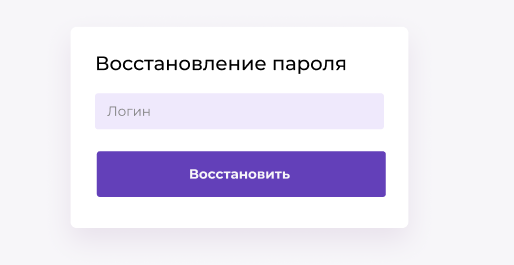


Рисунок 4 – Страница восстановления пароля

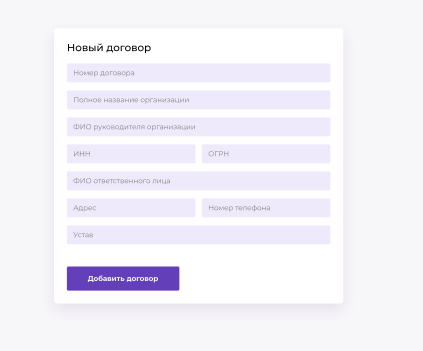


Рисунок 5 – Образец электронного договора

После успешной авторизации пользователь попадает на главную страницу системы. (рис.6)

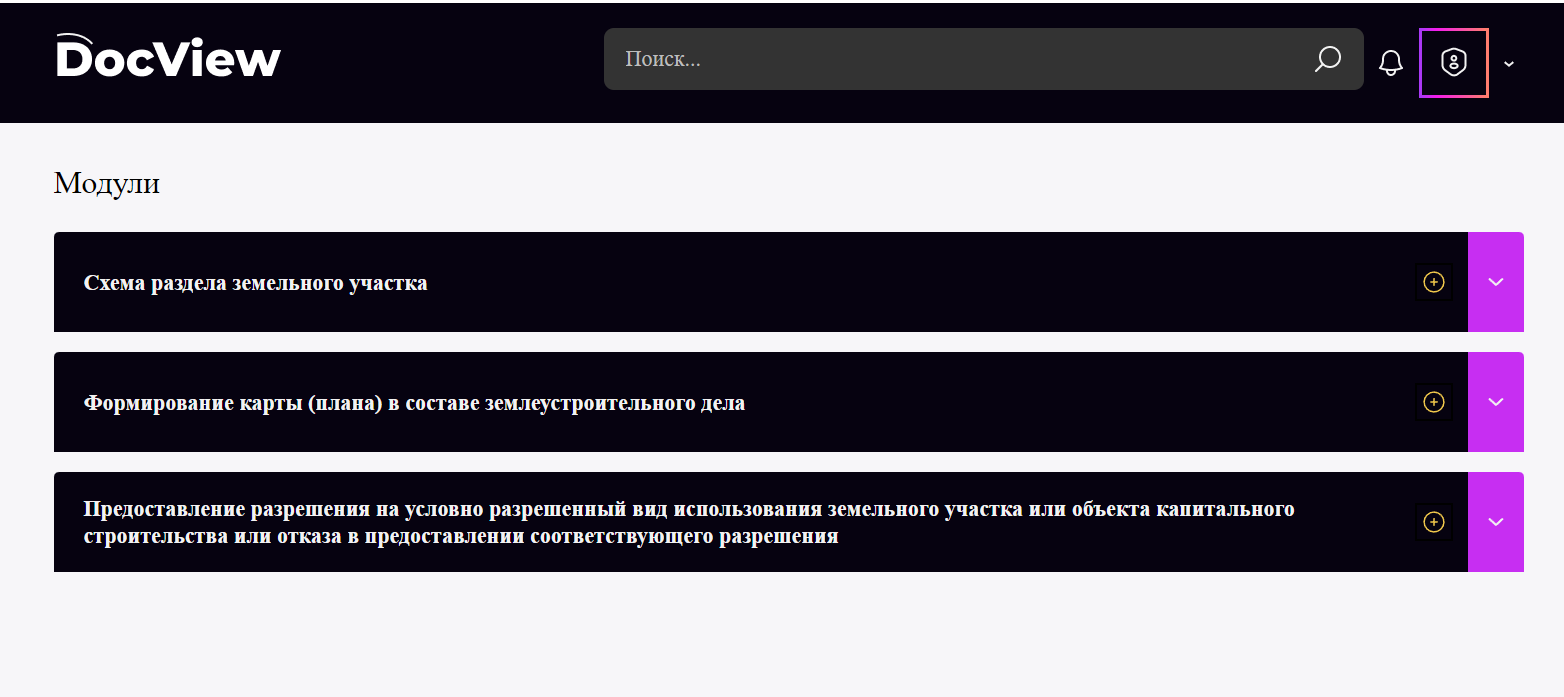


Рисунок 6 – Главная страница

На главные страницы выводятся все доступнее модули, в которые пользователь может добавить свой проект или задачу при нажатии на кнопку «+». После нажатия на кнопку пользователь получает страницу с настройками модуля, в которые он должен ввести определенного формата информацию. (рис.7).

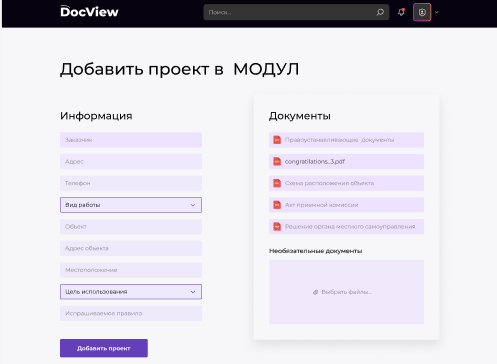


Рисунок 7 – Добавление проекта

Пользователю необходима ввести корректную информация так же добавить обязательные документы нужным форматом, при необходимости может добавить необязательные документы. После заполнение всех форм проект или задача попадает на проверку.

Так же на главной странице пользователю доступен быстрый поиск по модулям (рис.8).

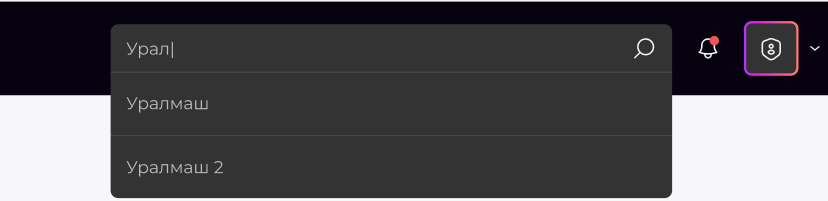


Рисунок 8 – Быстрый поиск

При нажатии на иконку пользователя открывается меню управления (Рис 14), по которому пользователь может перейти в профиль где ему будет доступна информация по уже добавленным проектам (рис.9).

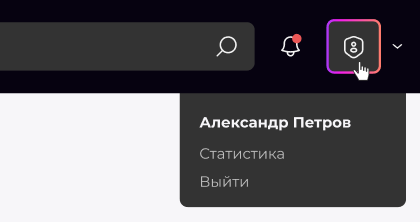


Рисунок 9 – Управление профилям

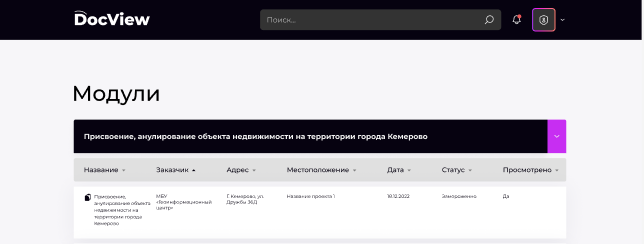


Рисунок 10 – Просмотр проектов в режиме «Модул»

При поступлении нового уведомления в системе отражается значок уведомление при клике на него открывается сами уведомление (рис11)

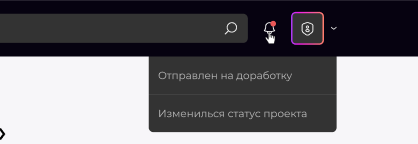


Рисунок 11 – Новые уведомления

При нажатии на сам проект открывается новая страница со всей информацией по проекту. Где можно посмотреть все имевшийся замечание(рис.12).

При клике на кнопку “Изменить” открывается страница добавление проекта, но уже со всеми заполни нами формами (рис 7)

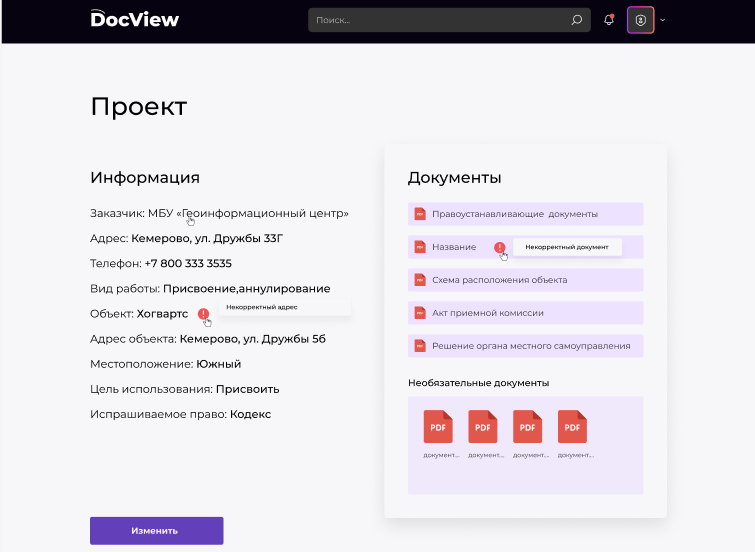


Рисунок 12 – Просмотр проектов

## 1.5 Руководство пользователя с ролью «Модератор»

После авторизации модератору доступно уникальная опция - «добавление новых модулей», в которые пользователи смогут добавлять свои проекты или задачи.

Модераторы добавляют все настройки модуля, которые в последствии будут храниться на сервере в базе данных и при необходимости с помощью запросов к базе данных будут загружаться в клиентскую часть. Для начала нужно указывать название модуля, которое в дальнейшем будет выводиться при добавлении проектов в этот модуль или просмотров уже существующих проектов в этом модуле, а также написать название базы, которая создастся при создании модуля и в которой будут храниться добавленные проекты.

В настройках модуля в зависимости от того, какой способ и тип данных вводимой информации мы ожидаем, необходимо выбрать определённую конфигурацию, а также надо указать название поля, создаваемого в базе данных и заголовке, которое будет вводиться при добавлении проекта у этого поля ввода информации. Сheckbox “Может быть пустим” указывает на то, что допускается оставлять это поле пустыми, по умолчанию находятся в состояние “false”

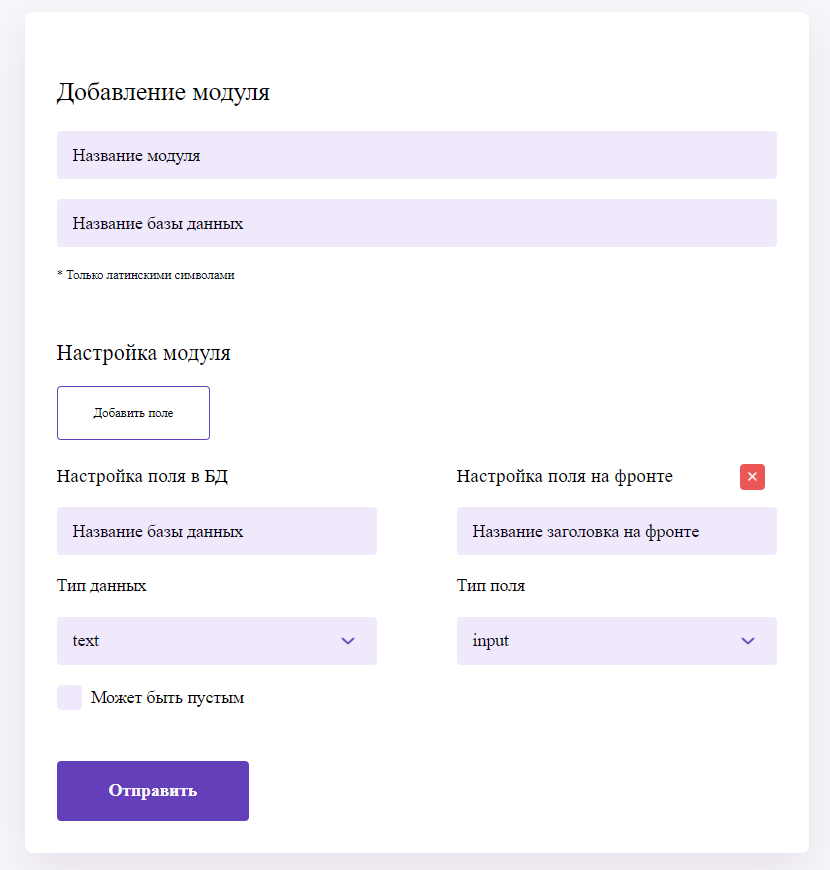


Рисунок 13 – Страница настройка модуля

При выборе тип поля надо учитивать и тип даных, чтобы не было конфликтов сочетаний, например: нельзя выбирать тип поля “ сheckbox” и тип даных “text”, так как нельзя хранить тип “string” в строке ожидаюшей тип “boolean”

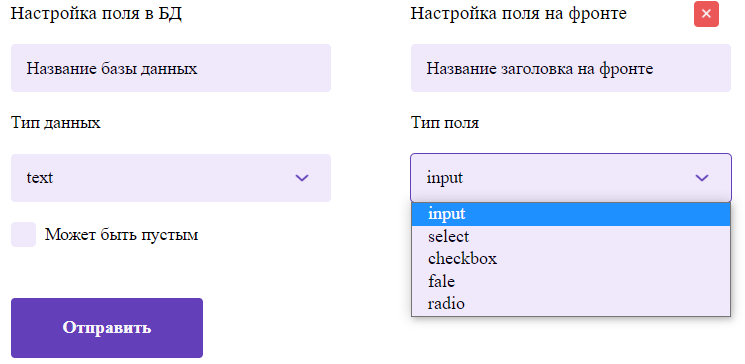


Рисунок 14 – Выбор тип поля

При выборе “select” и “radio” надо дополнительно создать список, из которого будеть предложино выбрать значения. Мы можем выбирать уже из сушетвуюших значений, а так же можем добавить новые значения в базу, а потом их выбрать. (рис.15).

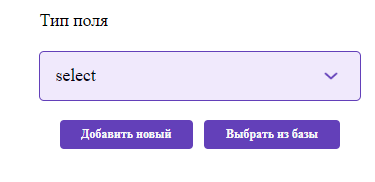


Рисунок 15 –Настройка «select»

Запрешаеться добавлять повторные значения. Ограниечение по количеству значений не стоит.

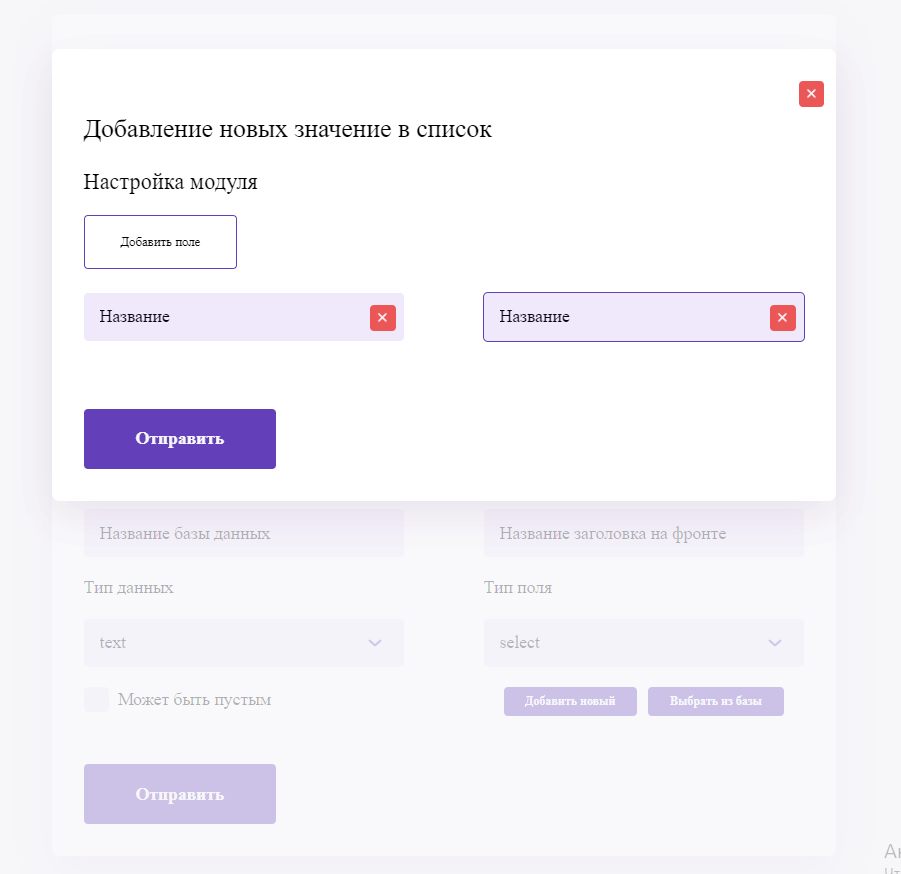


Рисунок 16 – Модальное окно «Добавление новых значение в список»

После добавления и заполнения всех нужных полей модератор добавляет данные в базу.Данные в дальнейшим будуть доступны для выбора новых “SELECT” (Рис17)

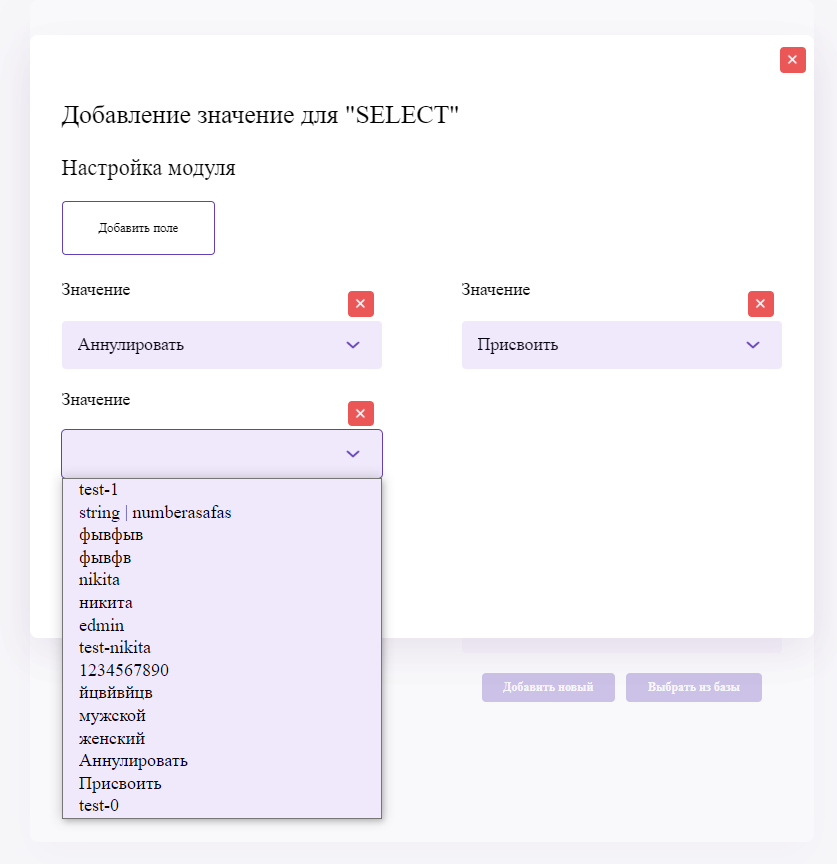
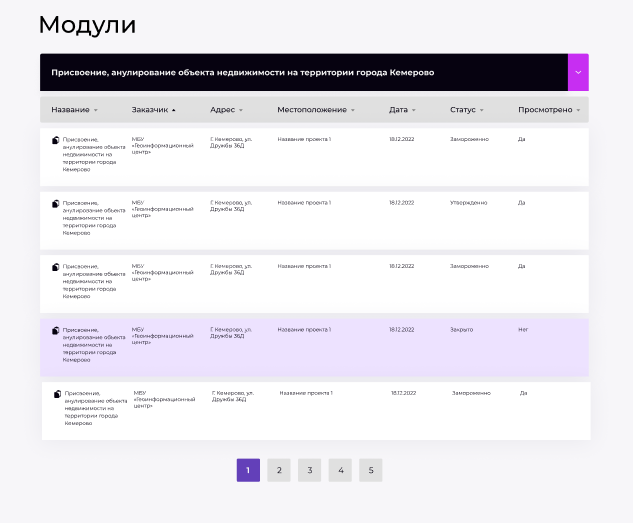


Рисунок 17 – Модальное окно «Добавление значение для SELECT»

На главной странице модератору доступны модули внутри которых находиться информация по проектам и задачам. По каждому из основных заголовков доступны сортировка как по возрастание, так и по убывание. Если модератор не открыл и проверил проект, то у этого проекта будет отличаться фон от других просмотренных проектов(рис.18).

Для открытия и проверки проекта необходима нажать на сам проект, после нажатия проект откроется в новом окне со всей доступной информацией для проверки (рис.19).

 Рисунок 18 – Главная страница для модератора

На странице проекта модератору доступна все информация, а также есть возможность оставлять замечания как по данным, так и по файлам. При отправке проекта если есть какие-то замечания, то проект отправиться на доработку, пользователю направиться уведомление на почту и в систему. Если же замечаний нет, то проект будить ожидать подтверждение у всех модераторов, отвечающих за данный модуль. После получение всех подтверждений будет вынесено решение, о чем пользователь также будет уведомлен как по почте, так и в системе(рис.19).

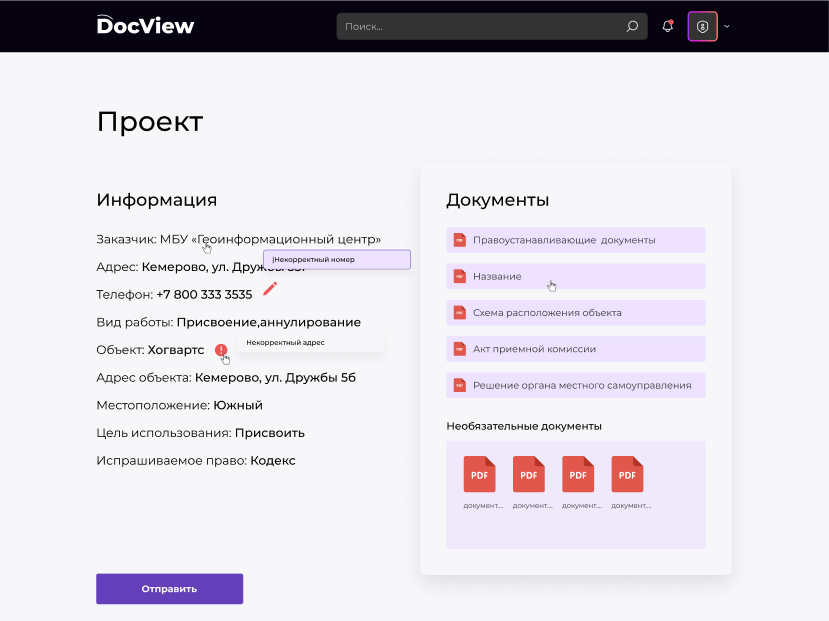


Рисунок 19 – Страница проверки для модератора

.

# 2 ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ ПРОЕКТА

## 2.1 Оценка финансовых затрат

Произведем расчет финансовых затрат на разработку программного модуля анализа посещаемости обучающихся. При определении экономической эффективности рассчитываются разовые инвестиции и оценивается целесообразность разработки нового ПО. [8]

Для этого были определены показатели трудоемкости разработки, представленные в таблице 1.

Таблица 1 - Показатели трудоемкости разработки

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Этап** | **Содержание этапов** | **Временные затраты** |
| 1 | Аналитики | Исследование области и создание четкого ТЗ | 14 дней |
| 2 | Проектирования | Разработка алгоритма решения поставленных задач, обоснование цели разработки и составление расчетов | 10 дней |
| 3 | Построение схем | Формирование диаграмм, которые описывают работу организации до внедрения программного модуля и после. | 7 дней |
| 4 | Разработка дизайн-макетов | Создание дизайн (макет) всех страниц сайта или экранов приложения под все необходимые разрешения и устройства, правила использования всех элементов на них | 16 дней |
| 5 | Программирование: Backend | Реализации всех процессов которые происходят на серверной части сайта в ответ на действия пользователя | 30 дней. |
| 6 | Программирование: frontend | Разработка всего с чем взаимодействует пользователь | 30 дней |
| 7 | Тестирование | Тестируем веб-сервис | 15 дня |
| 8 | Релиз веб-сервиса | Переносим веб-сервис на сервера Передаем инструкцию по использованию | 5 дней |
|  |  | **Итого:** | 127 дней |

На рисунке 20 представлен план по разработке (диаграмма Ганта) программного модуля анализа посещаемости обучающихся.

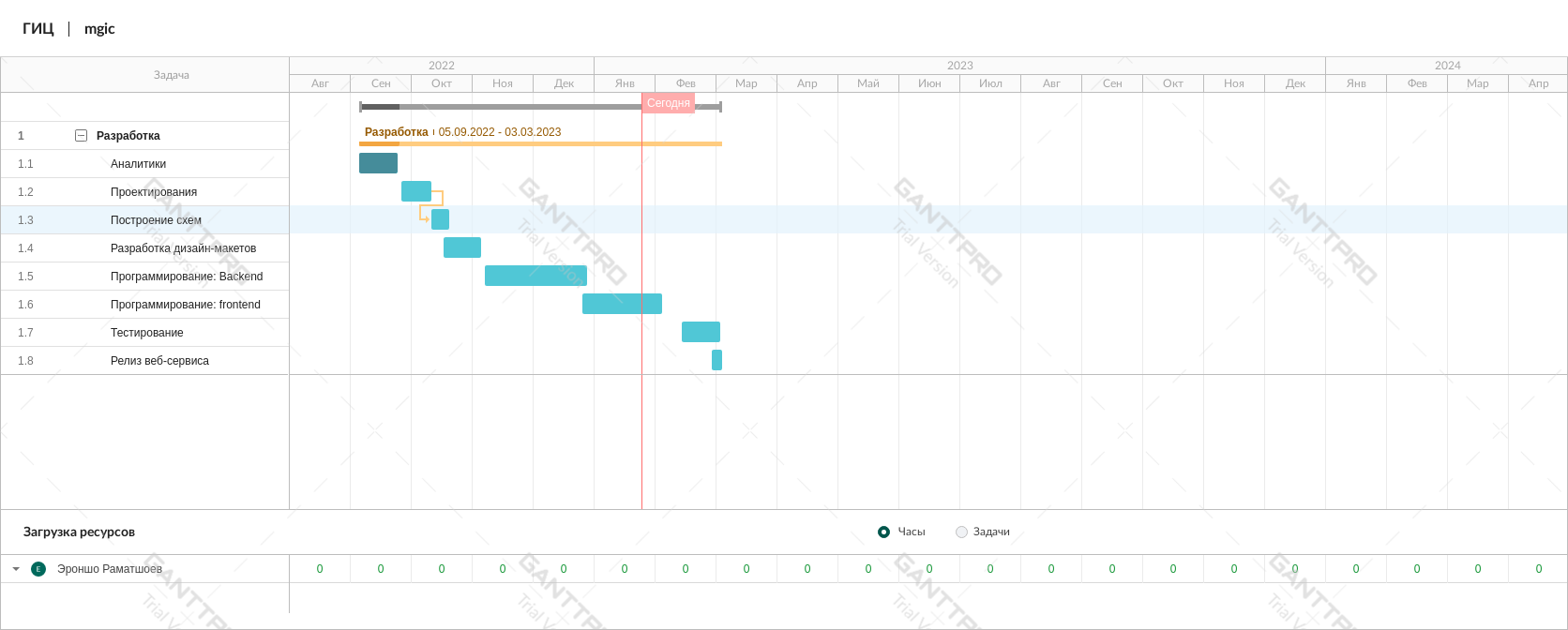


Рисунок 20 – Диаграмма Ганта

Диаграмма Ганта была разработана с учетом выходных и праздничных дней. Выходными днями являлись суббота и воскресенье каждой недели.

Затраты на заработную плату системного аналитика будут состоять: средний тариф, умноженный на количество рабочий часов . То есть 112 рабочих часов по 625 рублей/час . Следовательно, заработная плата планировщика 70000 рублей. [31]

Затраты на заработную плату архитектора будут состоять: средний тариф, умноженный на количество рабочий часов. То есть 80 рабочих часов по 625 рублей/час. Следовательно, заработная плата планировщика 50000 рублей. [31]

Затраты на заработную плату проектировщика будут состоять: средний тариф, умноженный на количество рабочий часов. То есть 56 рабочих часов по 328 рублей/час. Следовательно, заработная плата планировщика 18368 рублей. [31]

Заработная плата веб-дизайнера убудет рассчитываться, как средний тариф умноженный на 128 рабочий часов. При среднем тарифе 750 рублей/час, заработная плата веб-дизайнера за проект будет 96 000 рублей . [31]

Работа Front end разработчика составляет 240 часов. Средний тариф тестировщика составляет 937,5 рублей/час. Исходя из этих данных его заработная плата будет составлять 225 000 рублей. [31]

Работа Back end разработчика составляет 240 часов. Средний тариф т Back end составляет 812,5 рублей/час. Исходя из этих данных его заработная плата будет составлять 195 000 рублей.

Работа тестировщика составляет 120 часов. Средний тариф тестировщика составляет 375 рублей/час. Исходя из этих данных его заработная плата будет составлять 55 000 рублей. [31]

Исходя из расчетов заработной платы по каждому из участников разработке проекта, общая сумма 120 784 рублей.

Потребителем электроэнергии в разработке будет являться компьютер. Время необходимое на разработку 79 рабочий дней по 8 часов в день.

Потребителями электроэнергии при разработке будут: компьютер и сервер. Затраты на электроэнергию рассчитываются по формуле [17]:

где P – мощность оборудования (кВт);

T – время работы оборудования (ч);

– стоимость одного киловатта в час.

Стоимость одного киловатта по заключенному договору об оказании услуг по снабжению электроэнергией составляет 4.15 рубля.

Время использования электроэнергии 1016 часов работы. Мощность компьютера составляет 0.2 кВт, мощность сервера 0.3 кВт. Итого, потребление электроэнергии на разработку 0.5 кВт в час.

Итоговые затраты на электроэнергию будут рассчитывается по формуле:

Стоимость услуг интернет провайдера в месяц составляют 990 рублей в месяц. В период разработки уплачено за пять месяца, т.е. 4950 рублей.

Таблица 2 Стоимость расходных материалов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Расходные материалы: | **Затраты** | | | | |
|  | Рассчитанные на месяц | | | На время разработки | |
|  | Кол-во | Цена | Итого | Всего |
| Бумага | 3шт | 240р | 720р | 3600р |
| Картридж | 1шт | 1500р | 1500р | 7500р |
| Канцтовары |  |  | 700р | 3500р |
| Итого за расчетный период |  |  |  | 14600р |

Данные о общей стоимости проекта представлена на в таблице 4.2.

Таблица 3 Общая стоимость проекта

|  |  |
| --- | --- |
| **Вид затрат** | **Стоимость, руб.** |
| Заработная плата | 709 368р |
| Электроэнергия | р |
| Интернет | 4950р |
| Расходные материалы | 14600р |
| **ИТОГО:** | **731026,2** |

## 2.2 Влияние проекта на работу организации

Для того, чтобы проект пользовался спросом необходимо продемонстрировать его пользу для организации в сравнении эффективности работы компании до и после внедрения программного продукта в работу.

Внедрение программного продукта для отслеживания задач и проектов способно улучшить ряд процессов в работе организации. Время, которое сотрудники тратили на один проект или задачу после внедрения продукта стало значительно уменьшаться по мере внедрение системы.

Для того чтобы застройщик обратился в компанию ему надо заключить договор в регистратуре и получить список информации и документов, которые он должен предоставить, чтобы его проект рассмотрели. После внедрения продукта пользователю достаточно будет иметь действующий договор, чтобы добавить проект в систему.

Раннее на проверку заявлений, данных и всех документов уходило достаточно много времени, так как каждый отдел проверял только относящуюся к нему информацию. Необходимо было затратить большое количество времени как на ожидание проверки информации в каждом отделе, так и на ожидание окончательного результат. А при выявлении неточностей или отсутствии каких – либо обязательных документов необходимо было связаться с клиентом, сообщить о выявленных недостатка, тратить время на устранения выявленных нарушений. У клиентов не всегда бывает возможность посещать организацию, да и сама компания не всегда способна организовать прием посетителей, поскольку с учетом графика посещений и загруженности основного персонала не способна выделить достаточного количества времени. (рис.21).

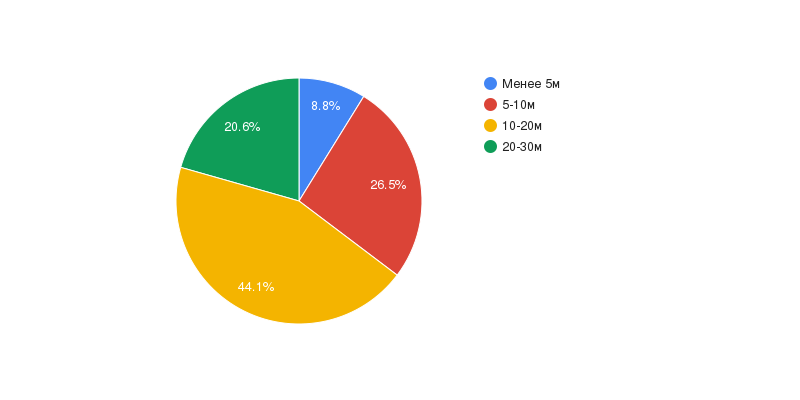


Рисунок 21 – Диаграмма сродной времени на проверку до внедрения

После внедрения программного модуля время необходимое для обработки данных сократилось в среднем до 10 минут (рис.22).

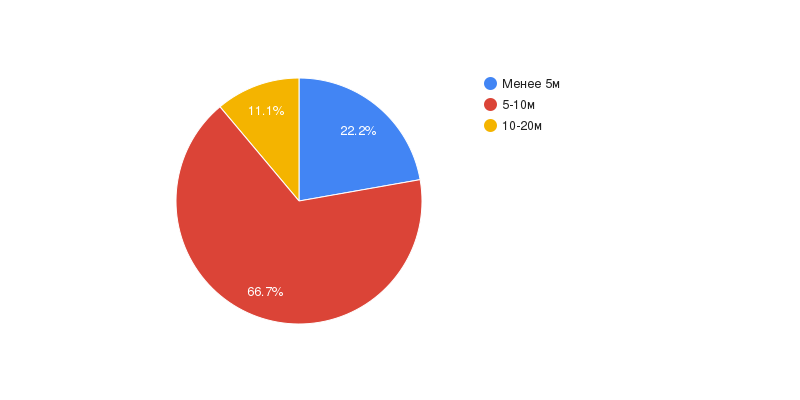


Рисунок 22 – Диаграмма сродной времени на проверку после внедрение

При заключении договора и приеме всей необходимой информации и документов, регистратура тратила достаточно много времени. Но после внедрение продукции организация рабочего процесса наладилась в положительную сторону. Главным образом динамика наблюдается в системе приема и учета входящей документации, которая посредством модернизации приобрела цифровую площадку.

Цифровая площадка позволила сократить время на подачу и прием проектов, упростила систему отслеживания результатов рассмотрения заявок, позволила усовершенствовать порядок устранения допущенных нарушений в режиме онлайн. С ведением данный платформы

Одним из основных преимуществ продукта является полный мониторинг проекта или задачи с момента его формирования и до момента его претворения в жизнь, как со стороны клиента, так и компании, что в свою очередь позволяет: контролировать весь процесс и избежать ошибочных решение

Необходимо отметить, что с учетом внедрение увеличивается масштаб обхватываемых проектов (платформа позволять загрузить неограниченное количество обращений, а персонал уменьшает трудозатратность (на чем делался акцент выше).

## 2.4 Перспективы развития

Перспективой развития является использование данного программного модуля как для пользования компанией, так и для пользование любым частным лицом. При необходимости можно разделить на два подмодуля.

Первый модуль подробно описан выше. Второй модуль позволить хранить все имеющиеся проекте, задачи и результат по их рассмотрению, что в дальнейшем послужит для анализа как проблемных областей в городе, так и для оценки качества работы определённого отдела.

Так же в перспективе развития программного модуля рассматривается добавление дополнительных функций, таких как:

1. Связь с отельными базами данных, в которых хранятся документация, решения, и другая необходимая компании информация, что в итоге позволить уменьшить бумажный объем работы в несколько раз.
2. Создание нового модуля по улучшению рабочего места персонала, что позволит уменьшить время ожидания устранения поломок, и нехватки оборудование и д. т.

Можно также рассматривать вариант, при которым можно будит предоставлять права пользования для других муниципальных компаний по системы подписок.

# ЗАКЛЮЧЕНИЕ

Производственная практика: преддипломная практика проходила в МБУ «Геоинформационный центр» .

За время прохождения практики был решен ряд задач:

* + - изучены методы и средства обеспечения информационной безопасности и защиты информации;
    - выбран комплекс задач автоматизации;
    - обоснован выбор средств разработки;
    - разработан программный модуль.

Поставленная цель достигнута.

Программный модуль включает в себя обработку данных проектов и задач Освободив сотрудников и клиентов от части бумажной работ также от посещение клиентов организацию. Так же в реальном времени позволять следить за ходом проверки или приятие решение по проектом или задачам

Программный модуль имеет ряд функций:

* Авторизацию,
* Модуля,
* Добавление модуля,
* Добавление проект в модуль
* прикрепление документов

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Алекс Бэнкс, Ева Порселло. React и Redux. Функциональная веб-разработка. – СПб.: Питер, 2018. – 336 с.
2. А.Н. Васильев. JavaScript в примерах и задачах. – М.: Эксмо, 2017. – 720 с.
3. Конструктор и решатель дискретных задач оптимального управления («Карма») / Программа для ЭВМ. Свидетельство о регистрации в Роспатенте №2008614387 от 11.09.2008. Правообладатели: А.В. Медведев, П.Н. Победаш, А.В. Смольянинов, М.А. Горбунов.
4. Белайчук А.А Свод знаний по управлению бизнес-процессами. BPM СВООК 3.0: учеб. пособие / А.А. Белайчук. – М.: Альпина Паблишинг, 2019. – 480 с.
5. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 2018 г. Вильямс / Томас Коннолли, Каролин Бегг. 1436 с.
6. Бэрон Шварц, Пётр Зайцев, Вадим Ткаченко. MySQL по максимуму. – СПб.: Питер, 2018. – 864 с.
7. Дэвид Уитни. Программирование для детей. Учимся создавать сайты, приложения и игры. HTML, CSS и JavaScript. – СПб.: Питер, 2018. – 208 с.
8. Назимов А.С. Моделирование и автоматизированная оценка эффективности электронного бизнеса / А.С. Назимов, С.Р. Ли, С.А. Созинов // Международный журнал прикладных и фундаментальных исследований – 2018. - №10.
9. Назимов А.С. Оценка экономической эффективности деятельности оптовых торговых предприятий электронной коммерции / А.С. Назимов, С.Р. Ли, Ю.В. Сусленкова, Е.А. Лисичкина // Фундаментальные исследования – 2019 - № 11 (2).
10. Назимов А.С. Оценка экономической эффективности электронных торговых площадок / А.С. Назимов, С.Р. Ли, Ю.В. Сусленкова, Т.В. Долгина // Фундаментальные исследования. – 2018. – № 1.
11. Раскрутка сайтов: основы, секреты, трюки - Яковлев А. А. / 5-е издание, переработанное и дополненное / СПб.: БХВ-Петербург / 2015 год, - 309 с.
12. С.Чакон, Б.Штрауб. Git для профессионального программиста. – СПб.: Питер, 2016. – 496 с.
13. Саймон Ригс, Ханну Кросинг. Администрирование PostgreSQL 9. Книга рецептов. – М.: ДМК Пресс, 2015. – 364 с.
14. Я.Файн. Angular и TypeScript. Сайтостроение для профессионалов. – СПб.: Питер, 2018. – 464 с.
15. HTML5. Для профессионалов – Гоше Х. Д. / Питер / 2018 год, - 407 с.
16. SEO – искусство раскрутки сайтов - Э. Энж, С. Сненсер, Р. Фишкин, Д. Стрикчиола / 2-е издание / БХВ-Петербург, - 2014, - 211 с.
17. Документация для кода Visual Studio [Электронный ресурс] – Режим доступа: https/code.visualstudio.com/Docs (дата обращения: 01.02.2023)
18. Графический редактор Figma [Электронный ресурс] – Режим доступа: https://practicum.yandex.ru/blog/chto-takoe-figma-dlya-dizainera/ (дата обращения: 01.02.2023)
19. Что такое API / Хабр [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/464261/ (дата обращения: 01.02.2023)
20. Postman API Platform [Электронный ресурс] – Режим доступа: https://www.postman.com/ (дата обращения: 01.02.2023)
21. Sequelize [Электронный ресурс] – Режим доступа: https://sequelize.org/ (дата обращения: 01.02.2023)
22. ORM или как забыть о проектировании БД / Хабр [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/237889/ (дата обращения: 01.02.2023)
23. GitHub - typeorm/typeorm: ORM for TypeScript [Электронный ресурс] – Режим доступа: https://github.com/typeorm/typeorm (дата обращения: 01.02.2023)
24. Sass: Основы Sass [Электронный ресурс] – Режим доступа: https://sass-scss.ru/guide/ (дата обращения: 01.02.2023)
25. TypeScript: JavaScript With Syntax For Types. [Электронный ресурс] – Режим доступа: https://www.typescriptlang.org/ (дата обращения: 01.02.2023)
26. JSON Web Token [Электронный ресурс] – Режим доступа: https://jwt.io/introduction (дата обращения: 01.02.2023)
27. Thunder Client - Rest API Client Extension for VS Code [Электронный ресурс] – Режим доступа: https://www.thunderclient.com/ (дата обращения: 01.02.2023)
28. npm [Электронный ресурс] – Режим доступа: https://www.npmjs.com/ (дата обращения: 01.02.2023)
29. Home | Yarn - Package Manager [Электронный ресурс] – Режим доступа: https://yarnpkg.com/ (дата обращения: 01.02.2023)
30. Node.js [Электронный ресурс]. – Режим доступа: https://nodejs.org/ru/ (дата обращения: 01.02.2023)
31. Зарплаты в IT [Электронный ресурс] – Режим доступа: https://checkroi.ru/blog/zarplaty-v-it-kto-poluchaet-bolshe/ (дата обращения: 01.02.2023)