

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 09.03.02 «Информационные системы и технологии»
 Отделение школы (НОЦ) – Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы Проектирование и разработка серверной архитектуры веб-приложения «Агенты Будущего»
--

УДК 004.774-0126.12:004.273

Студент

Группа	ФИО	Подпись	Дата
8И8А	Гаскин Марк Вадимович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Демин Антон Юрьевич	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин Виталий Александрович			

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН УОД	Рыжакина Татьяна Гавриловна	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Цапко Ирина Валериевна	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
УК(У)-10	Способен принимать обоснованные экономические решения в различных областях жизнедеятельности
УК(У)-11	Способен формировать нетерпимое отношение к коррупционному поведению
ОПК(У)-1	Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК(У)-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности
ОПК(У)-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-4	Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил
ОПК(У)-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий
ОПК(У)-7	Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем
ОПК(У)-8	Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем
ПК(У)-1	Способен выполнять интеграцию программных модулей и компонент
ПК(У)-2	Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению информационных систем
ПК(У)-3	Способен создавать техническую документацию на продукцию в сфере информационных технологий, управлять технической информацией
ПК(У)-4	Способен выполнять работы по обеспечению функционирования баз данных и обеспечению их информационной безопасности
ПК(У)-5	Способен проводить, оценивать и следить за выполнением концептуального, функционального и логического проектирования систем малого и среднего масштаба и сложности

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 09.03.02 «Информационные системы и технологии»
 Отделение школы (НОЦ) – Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8И8А	Гаскину Марку Вадимовичу

Тема работы:

Проектирование и разработка серверной архитектуры веб-приложения «Агенты Будущего»	
Утверждена приказом директора (дата, номер)	№ 132-1/с от 12.05.2022

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объект исследования – разработка веб-приложений.</p> <p>Предмет исследования – процесс разработки интерфейсов серверного приложения.</p>
---	---

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ul style="list-style-type: none"> – Проектирование интерфейсов серверного приложения; – Реализация авторизации по стандарту OAuth 2.0; – Реализация интерфейсов серверного приложения; – Тестирование возможностей интерфейсов серверного приложения; – Разработка документации; – Внедрение приложения на сервер; – Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; – Социальная ответственность
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Архитектура всей системы, архитектура разрабатываемых интерфейсов, диаграмма базы данных, презентация в формате *.pptx.</p>

<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Рыжакина Т.Г.
Социальная ответственность	Мезенцева И.Л.
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	
--	--

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Демин Антон Юрьевич	к.т.н.		
Ассистент ОИТ	Коровкин В.А.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И8А	Гаскин Марк Вадимович		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
 Направление подготовки – 09.03.02 Информационные системы и технологии
 Уровень образования – Бакалавриат
 Отделение школы (НОЦ) – Отделение информационных технологий
 Период выполнения – весенний семестр 2021/2022 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.06.2022	Основная часть	75
11.05.2022	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
03.05.2022	Социальная ответственность	10

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Демин А.Ю.	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОИТ ИШИТР	Коровкин В.А.			

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Цапко И.В.	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8И8А	Гаскин Марк Вадимович

Школа	Инженерная школа информационных технологий и робототехники	Отделение школы (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах; анкетирование; опрос.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	<i>Проведение предпроектного анализа. Определение целевого рынка и проведение его сегментирования. Выполнение SWOT-анализа проекта</i>
2. <i>Планирование и формирование бюджета научных исследований</i>	<i>Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования</i>
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	<i>Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения</i>

Перечень графического материала (с точным указанием обязательных чертежей):

1. *Оценка конкурентоспособности технических решений*
2. *Матрица SWOT*
3. *Альтернативы проведения НИ*
4. *График проведения и бюджет НИ*
5. *Оценка ресурсной, финансовой и экономической эффективности НИ*

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Рыжакина Татьяна Гавриловна	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И8А	Гаскин Марк Вадимович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8И8А		Гаскин Марк Вадимович	
Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

Тема ВКР:

Проектирование и разработка серверной архитектуры веб-приложения «Агенты Будущего»

Исходные данные к разделу «Социальная ответственность»:

<p>Введение</p> <ul style="list-style-type: none"> – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения 	<p><i>Объект исследования:</i> веб-платформа для создания образовательных треков для школьников. <i>Область применения:</i> дополнительное и профориентационное школьное образование. <i>Рабочая зона:</i> производственное помещение. <i>Размеры помещения:</i> 5*4 м. <i>Количество и наименование оборудования рабочей зоны:</i> рабочий стол с персональным компьютером. <i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> разработка и тестирование сервисов веб-приложения.</p>
---	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения.</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>Трудовой кодекс РФ от 25.02.2002. ГОСТ 12.2.032-78 «Система стандартов безопасности труда».</p>
<p>2. Производственная безопасность при разработке проектного решения</p> <ul style="list-style-type: none"> – Анализ выявленных вредных и опасных производственных факторов 	<p>Вредные факторы:</p> <ol style="list-style-type: none"> 1. Повышенный уровень шума на рабочем месте; 2. Показатели микроклимата воздушной среды на местонахождении работающего: температура и относительная влажность воздуха; 3. Отсутствие или недостаток необходимого искусственного освещения; 4. Монотонность труда, вызывающая монотонию. <p>Опасные факторы:</p> <ol style="list-style-type: none"> 1. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которых попадает работающий. <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: средства отопления и кондиционирования, устройства для вентиляции и очистки воздуха, заземляющие устройства, осветительные приборы, звукоизолирующие устройства.</p>
<p>3. Экологическая безопасность при разработке проектного решения</p>	<p>Воздействие на селитебную зону: не выявлено.</p>

	<p>Воздействие на литосферу: неправильная утилизация отходов при поломке предметов вычислительной техники.</p> <p>Воздействие на гидросферу: не выявлено.</p> <p>Воздействие на атмосферу: не выявлено.</p>
<p>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</p>	<p>Возможные ЧС: Техногенные (пожар, внезапное обрушение здания, аварии на коммунальных системах жизнеобеспечения населения). Биологические (эпидемия, пандемия). Социальные (терроризм, войны). Наиболее типичная ЧС: пожар.</p>

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И8А	Гаскин Марк Вадимович		

РЕФЕРАТ

Выпускная квалификационная работа содержит: 82 страницы, 15 рисунков, 23 таблицы, 16 источников, 1 приложение.

Ключевые слова: веб-приложение, проектирование архитектуры, OAuth, контроллер, интерфейс, документация.

Объектом исследования является разработка веб-приложений. Предметом исследования является процесс разработки интерфейсов серверного приложения.

Цель работы – проектирование и реализация компонентов серверного приложения веб-платформы «Агенты будущего».

В результате исследования были спроектированы интерфейсы веб-приложения согласно выявленным требованиям и архитектуре всей системы.

Степень внедрения: частичная.

Область применения: сфера дополнительного школьного образования.

В будущем планируется разработка всех необходимых интерфейсов и полный перевод платформы на новый способ взаимодействия.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной работе применены следующие обозначения и сокращения:

БД – База Данных;

API – Application Programming Interface;

REST – Representational State Transfer;

MVC – Model-View-Controller;

JSON – JavaScript Object Notation;

ID – Identifier;

HTTP – HyperText Transfer Protocol;

DTO – Data Transfer Object;

OAuth – Open Authorization;

URL – Uniform Resource Locator.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	13
1 Проектирование приложения	15
1.1 Требования к проектируемому приложению	15
1.2 Общая архитектура системы	18
1.3 Архитектура интерфейсов	18
1.4 Сценарии использования	19
1.5 Система управления базами данных	31
2 Реализация приложения	33
2.1 Используемые технологии.....	33
2.2 Реализация OAuth авторизации.....	34
2.3 Реализация интерфейсов.....	36
2.3 Описание программных интерфейсов	39
2.4 Тестирование интерфейсов.....	42
2.5 Документация для интерфейсов	43
2.6 Загрузка файлов на сервер	45
2.7 Система контроля версий	47
3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	48
3.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	48
3.1.1 Потенциальные потребители результатов исследования.....	48
3.1.2 Анализ конкурентных технических решений	48
3.1.3 SWOT-анализ	50
3.2 Определение возможных альтернатив проведения научных исследований	52
3.3 Планирование научно-исследовательских работ	53
3.3.1 Структура работ в рамках научного исследования	53
3.3.2 Определение трудоемкости выполнения работ	54
3.3.3 Разработка графика проведения научного исследования.....	56
3.3.4 Бюджет научно-технического исследования (НТИ)	57
3.3.4.1 Расчет материальных затрат НТИ.....	57
3.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных) работ	58
3.3.4.3 Основная заработная плата исполнителей темы	59
3.3.4.4 Дополнительная заработная плата исполнителей темы	61
3.3.4.5 Отчисления во внебюджетные фонды.....	61
3.3.4.6 Накладные расходы	62

3.3.4.7	Формирование бюджета затрат научно-исследовательского проекта	63
3.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования ..	63
4	Социальная ответственность	66
4.1	Правовые и организационные вопросы обеспечения безопасности ...	67
4.1.1	Правовые нормы трудового законодательства	67
4.1.2	Эргономические требования к правильному расположению и компоновке рабочей зоны	67
4.2	Производственная безопасность	68
4.2.1	Повышенный уровень шума на рабочем месте	69
4.2.2	Показатели микроклимата воздушной среды на местонахождении работающего: температура и относительная влажность воздуха	70
4.2.3	Отсутствие или недостаток необходимого искусственного освещения	71
4.2.4	Монотонность труда, вызывающая монотонию	72
4.2.5	Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которых попадает работающий	73
4.3	Экологическая безопасность	74
4.4	Безопасность в чрезвычайных ситуациях	74
	Вывод по разделу	75
	ЗАКЛЮЧЕНИЕ	77
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	79
	Приложение А. Методы регистрации, авторизации и аутентификации пользователя	81

ВВЕДЕНИЕ

В современном мире постоянно растет потребность в дополнительном образовании как для студентов, так и школьников. С развитием глобальной сети Интернет организовать подобные мероприятия стало гораздо проще. Все чаще открываются различные платформы, позволяющие получить новые знания в виде образовательных курсов. Однако большинство из них направлены на аудиторию студентов и попросту непонятны и неинтересны школьникам. Заполнить этот пробел позволяет игровая веб-платформа на базе Томского Политехнического Университета «Агенты Будущего».

«Агенты будущего» представляют собой платформу для создания и проведения мероприятий по дополнительному образованию для школьников 7-11 классов. Сами мероприятия представляют собой некоторый перечень учебных модулей (миссий) в геймифицированной форме. Миссии бывают разных типов, к примеру:

- «Конструктор»: игроки собирают устройство или тех. систему из элементов;
- «Квест»: игроки исследуют локации и решают многоходовые задачи;
- «Стратегия»: игроки строят свои производства на карте;
- «Проект»: игроки ведут реальный проект и загружают результаты в систему.

За выполнение миссий пользователь получает специальную «валюту», которую сможет потратить на некоторые действия в следующих кампаниях. Также на сайте присутствуют новостной блок, системы уровней и достижений.

Данная работа описывает методы проектирования и реализации DTO-архитектуры на примере платформы «Агенты будущего».

Целью данной работы является проектирование и реализация компонентов серверного приложения веб-платформы «Агенты будущего».

Для решения поставленной цели были выделены следующие задачи:

- составить список функциональных требований, предъявляемых к разрабатываемым компонентам;
- выполнить планирование компонентов системы;
- выполнить разработку интерфейсов для регистрации, авторизации и аутентификации пользователей;
- выполнить разработку интерфейсов для работы с пользователями;
- выполнить разработку интерфейсов для работы с образовательными треками
- выполнить разработку интерфейсов для работы с новостями
- выполнить разработку интерфейсов для работы с миссиями
- провести тестирование реализованных компонентов;
- разработать документацию для интерфейсов;
- внедрить решение на тестовый сервер.

В настоящее время активно ведется работа по переносу веб-платформы на новый стек технологий, который позволит отдельно разрабатывать фронтенд-решения, основанная на javascript фреймворках (VUE), разрабатывать и внедрять контент сторонних разработчиков с помощью API. Сложность состоит в том, что в платформа активно используется для олимпиад и мероприятий для школьников. Ранее для взаимодействия компонентов системы использовалась архитектура «MVC». Теперь же в качестве архитектурного стиля взаимодействия используется «REST API», который позволяет вести более гибкую разработку, что является приоритетной целью команды разработчиков.

1 Проектирование приложения

1.1 Требования к проектируемому приложению

Правильно сформулированные и задокументированные требования необходимы всем участникам проекта для корректного видения концепции и возможностей приложения.

Для разрабатываемого приложения были сформулированы и выделены следующие общие требования:

- модуль должен быть разработан на «PHP» версии не менее 8.0;
- в качестве используемого фреймворка – «Laravel» версии не менее 9.0;
- в качестве базы данных должна использоваться «MySQL» версии 5.7.21;
- серверное приложение должно принимать HTTP-запросы с различными параметрами;
- получаемые данные должны проходить процесс валидации средствами используемого фреймворка;
- в соответствии с запросом должна выполняться выборка данных из БД;
- серверное приложение должно отправлять ответ на HTTP-запрос;
- данные в HTTP-запросах передаются в формате JSON;
- для каждого предоставляемого интерфейса должна быть оформлена документация посредством Swagger;
- формат ответа на HTTP-запрос должен иметь следующий вид:

```
{  
  «success»: true/false,  
  «message»: «тело ответа»  
}
```

В рамках выпускной квалификационной работы реализуются следующие компоненты:

- регистрация, авторизация, аутентификация;
- профиль пользователя;

- новости;
- образовательные треки, в т.ч. олимпиады, как частный вид образовательного трека;
- игровые миссии и задания.

Для описанных выше компонентов был составлен следующий перечень функциональных требований:

- серверное приложение должно обеспечивать возможность регистрации и входа пользователя в аккаунт;
- серверное приложение должно обеспечивать возможность регистрации клиентов (приложений) в системе;
- в качестве протокола авторизации должен использоваться OAuth 2.0;
- серверное приложение должно использовать систему ролевого разграничения прав для предоставления доступа пользователям к методам;
- серверное приложение должно иметь следующие роли:
 1. гость;
 2. зарегистрированный пользователь;
 3. администратор;
 4. администратор мероприятия;
- в качестве системы аутентификации должны использоваться средства «Laravel Passport» и «Laravel Sanctum»;
- в качестве средства аутентификации должен использоваться «Bearer Token»;
- серверное приложение должно обеспечивать следующие возможности работы с пользователями:
 1. возвращать пользователя по его ID;
 2. обновлять пользователя;
 3. возвращать достижения пользователя по его ID;
 4. возвращать список пользователей;

– серверное приложение должно обеспечивать следующие возможности работы с новостями:

1. возвращать список новостей;
2. создавать новость;
3. редактировать новость;
4. удалять новость;
5. возвращать новость по ее ID;

– серверное приложение должно обеспечивать следующие возможности работы с образовательными треками:

1. возвращать список треков;
2. создавать трек;
3. редактировать трек;
4. удалять трек;
5. возвращать трек по его ID;

– серверное приложение должно обеспечивать следующие возможности работы с миссиями:

1. внесение информации о начале пользователем миссии;
2. внесение информации об окончании пользователем миссии;
3. внесение информации о прохождении пользователем

контрольной точки в миссии;

4. для каждого из пунктов 1-3 должна вноситься информация, необходимая для оценки результатов выполнения миссии пользователем, в т. ч. и файлы;

– получаемые файлы должны сохраняться на серверном хранилище;

– информация о принадлежности файлов к сущностям должна храниться в базе данных в виде JSON-файлов.

1.2 Общая архитектура системы

Архитектура всей системы представлена на рисунке 1.

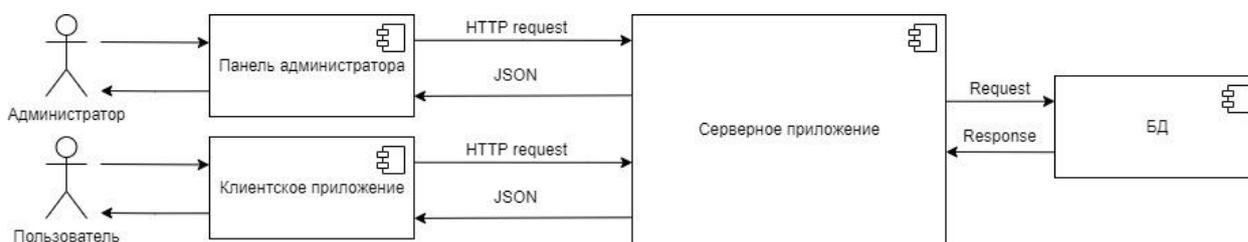


Рисунок 1 – Архитектура системы

Пользователь производит взаимодействие с клиентским приложением или панелью администратора (в зависимости от его роли). При необходимости совершить действия с данными (авторизация пользователя, получение, изменение данных пользователя, новостей и т.п.) происходит отправка HTTP-запроса на сервер, который производит необходимые манипуляции с данными в базе и отправляет ответ обратно на клиентское приложение.

Во время разработки приложение тестируется на локальном сервере Apache. После прохождения всех тестов и написания документации происходит загрузка приложения на веб-сервер nginx, на котором также проверяется работоспособность разработанных интерфейсов.

База данных располагается в сети, что позволяет подключаться к ней как с локального, так и с веб-сервера. Так как количество активных пользователей более 3000, то было принято решение использовать в качестве базы данных полную структурную копию базы, используемой на сайте, однако данные сгенерировать случайным образом и обезличить.

1.3 Архитектура интерфейсов

Для разработки большинства интерфейсов, было решено использовать следующую структуру [1] (рисунок 2).

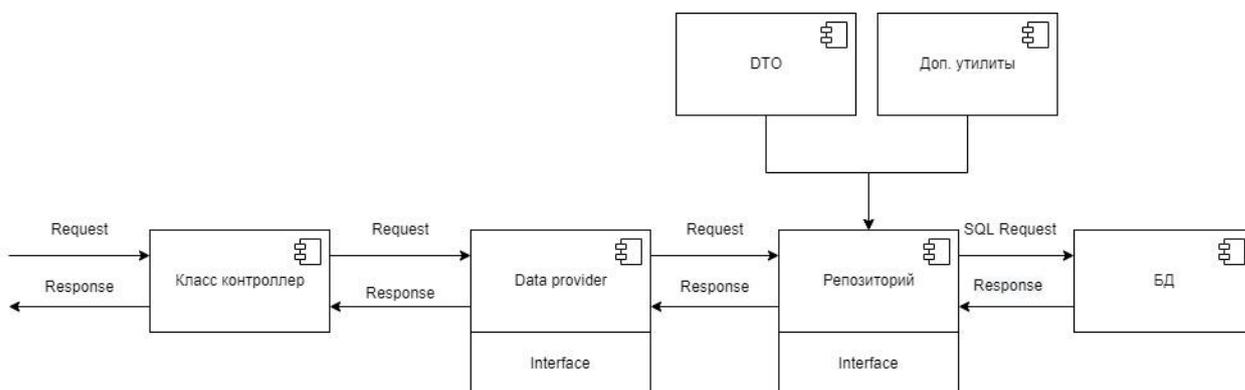


Рисунок 2 – Архитектура интерфейсов

Класс контроллер служит точкой входа для поступающих данных и содержит методы, выполняющие обращения к доступным методам провайдера данных. Информацию о доступных методах провайдера можно получить из его интерфейса.

В класс провайдера передается экземпляр класса репозитория, реализующего собственный интерфейс. Методы провайдера обращаются к необходимым методам репозитория.

Репозиторий реализует операцию посредством работы с БД и возвращает необходимый ответ. В случае если необходимо вернуть сущность используется специальный класс DTO (Data transfer object) для создания объектов. При необходимости возвращаемые данные также могут быть обработаны с помощью дополнительных утилит.

1.4 Сценарии использования

Сценарий использования – описание поведения системы при взаимодействии ее с кем-либо и чем-либо из внешней среды.

Наличие эффективных сценариев использования позволяет значительно упростить процесс последующей реализации приложения. Необходимые для реализации интерфейсов сценарии представлены в таблице 1.

Таблица 1 – Сценарии использования

Сценарий 1. Регистрация пользователя	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Создать учетную запись пользователя
Предусловие	Пользователь зашел в пункт «Регистрация» на сайте
Успешный сценарий:	
<ol style="list-style-type: none"> 1. Пользователь открывает форму регистрации и заполняет необходимые поля. 2. Клиентское приложение отправляет данные на сервер 3. Серверное приложение проверяет отправленные данные, записывает их в базу данных, отправляет на почту пользователя сообщение для подтверждения учетной записи и возвращает приложению токен для аутентификации пользователя. 	
Результат	Учетная запись пользователя успешно создана
Альтернативные сценарии:	
3.1.1 Введенные данные пользователя не проходят валидацию	
3.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
3.2.1 Нет доступа к базе данных	
3.2.2 Серверное приложение возвращает сообщение о неудачной попытке создания записи и записывает данные об ошибке в файл журнала.	
Результат	Учетная запись пользователя не создана
Сценарий 2. Авторизация пользователя	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: войти на платформу и получить доступ к ее возможностям. Серверное приложение: идентифицировать пользователя и его права
Предусловие	Пользователь зашел на сайт
Успешный сценарий:	
<ol style="list-style-type: none"> 1. Пользователь открывает форму входа и заполняет необходимые поля. 2. Клиентское приложение отправляет данные на сервер 3. Серверное приложение проверяет отправленные данные и возвращает приложению токен для аутентификации пользователя. 	

Результат	Пользователь успешно авторизован и может работать с системой
Альтернативные сценарии:	
3.1.1 Введенные данные пользователя не проходят валидацию	
3.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
3.2.1 Нет доступа к базе данных	
3.2.2 Серверное приложение возвращает сообщение о неудачной попытке создания записи и записывает данные об ошибке в файл журнала.	
Результат	Учетная запись пользователя не создана
Сценарий 3. Выход пользователя из аккаунта	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: выйти из аккаунта на платформе Серверное приложение: обновить информацию о пользователе
Предусловие	Пользователь инициировал выход из платформы
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер	
2. Серверное приложение удаляет все токены аутентификации пользователя и возвращает сообщение об успешности операции	
Результат	Пользователь успешно вышел из аккаунта, его токены аутентификации удалены
Альтернативные сценарии:	
2.2.1 Нет доступа к базе данных	
3.2.2 Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.	
Результат	Пользователь не вышел из аккаунта
Сценарий 4. Получение списка пользователей	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить список пользователей
Предусловие	Пользователь инициировал получение списка пользователей
Успешный сценарий:	

<p>1. Клиентское приложение отправляет запрос на сервер с параметрами о количестве записей и их смещении.</p> <p>2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимый список пользователей</p>	
Результат	Список пользователей отправлен на клиентское приложение
Альтернативные сценарии:	
2.1.1 Введенные данные пользователя не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.	
Результат	Список пользователей не отправлен
Сценарий 5. Получение данных пользователя по его идентификатору	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить информацию о заданном пользователе
Предусловие	Пользователь инициировал получение информации о выбранном пользователе
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с идентификатором пользователя	
2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимого пользователя	
Результат	Информация о пользователе отправлена на клиентское приложение
Альтернативные сценарии:	
2.1.1 Отправленные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения данных и записывает информацию об ошибке в файл журнала.	
Результат	Информация о пользователе не отправлена
Сценарий 6. Обновление аватара пользователя	

Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: обновить аватар в профиле Серверное приложение: обновить информацию о фотографии пользователя
Предусловие	Пользователь инициировал обновление аватара
Успешный сценарий:	
<ol style="list-style-type: none"> Клиентское приложение отправляет запрос на сервер с фотографией пользователя Серверное приложение проверяет отправленные данные, сохраняет фото в хранилище, записывает в базу данных ссылку на файл и возвращает сообщение об успешной операции 	
Результат	Аватар пользователя успешно обновлен
Альтернативные сценарии:	
2.1.1 Введенные данные пользователя не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.	
Результат	Аватар пользователя не обновлен
Сценарий 7. Обновление данных пользователя	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: обновить персональные данные в профиле Серверное приложение: обновить информацию о данных пользователя
Предусловие	Пользователь инициировал обновление аватара
Успешный сценарий:	
<ol style="list-style-type: none"> Клиентское приложение отправляет запрос на сервер с персональными данными пользователя Серверное приложение проверяет отправленные данные, записывает их в базу данных и возвращает сообщение об успешной операции 	
Результат	Информация пользователя успешно обновлена
Альтернативные сценарии:	
2.1.1 Введенные данные пользователя не проходят валидацию	

2.1.2	Серверное приложение возвращает сообщение о причине провала валидации
2.2.1	Нет доступа к базе данных
2.2.2	Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.
Результат	Информация пользователя не обновлена
Сценарий 8. Получение достижений пользователя	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить список достижений выбранного пользователя
Предусловие	Пользователь инициировал получение списка достижений
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с идентификатором пользователя	
2. Серверное приложение проверяет отправленные данные и возвращает сообщение о достижениях пользователя	
Результат	Достижения пользователя успешно получены
Альтернативные сценарии:	
2.1.1	Введенные данные пользователя не проходят валидацию
2.1.2	Серверное приложение возвращает сообщение о причине провала валидации
2.2.1	Нет доступа к базе данных
2.2.2	Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.
Результат	Достижения пользователя не получены
Сценарий 9. Получение списка новостей	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Получить некоторое количество новостей
Предусловие	Пользователь инициировал получение списка новостей
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с параметрами о количестве записей и их смещении.	
2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимый список новостей	

Результат	Список новостей успешно возвращен
Альтернативные сценарии:	
2.1.1 Введенные данные пользователя не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения данных и записывает информацию об ошибке в файл журнала.	
Результат	Список новостей не был возвращен
Сценарий 10. Получение подробной информации о новости по ее идентификатору	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить информацию о выбранной новости
Предусловие	Пользователь инициировал получение информации о выбранной новости
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с идентификатором новости	
2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимую информацию о новости	
Результат	Информация о новости отправлена на клиентское приложение
Альтернативные сценарии:	
2.1.1 Отправленные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения данных и записывает информацию об ошибке в файл журнала.	
Результат	Информация о новости не отправлена
Сценарий 11. Обновление данных новости	
Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: обновить информацию о новости
Предусловие	Администратор инициировал обновление информации о новости
Успешный сценарий:	

1.	«Панель администратора» отправляет запрос на сервер с данными о новости
2.	Серверное приложение проверяет отправленные данные, записывает их в базу данных и возвращает сообщение об успешной операции
Результат	Информация о новости успешно обновлена
Альтернативные сценарии:	
2.1.1	Введенные данные не проходят валидацию
2.1.2	Серверное приложение возвращает сообщение о причине провала валидации
2.2.1	Нет доступа к базе данных
2.2.2	Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.
Результат	Информация о новости не обновлена
Сценарий 12. Создание новости	
Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: создать новость на сайте
Предусловие	Администратор инициировал создание новости
Успешный сценарий:	
1.	«Панель администратора» отправляет запрос на сервер с данными о новости
2.	Серверное приложение проверяет отправленные данные, записывает их в базу данных и возвращает сообщение об успешной операции
Результат	Новость успешно создана
Альтернативные сценарии:	
2.1.1	Введенные данные не проходят валидацию
2.1.2	Серверное приложение возвращает сообщение о причине провала валидации
2.2.1	Нет доступа к базе данных
2.2.2	Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.
Результат	Новость не создана
Сценарий 13. Удаление новости	
Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: удалить новость с сайта
Предусловие	Администратор инициировал удаление новости
Успешный сценарий:	

1. «Панель администратора» отправляет запрос на сервер с идентификатором новости	
2. Серверное приложение проверяет полученный идентификатор, удаляет выбранную запись и возвращает сообщение об успешности операции	
Результат	Новость удалена
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке удаления данных и записывает информацию об ошибке в файл журнала.	
Результат	Новость не удалена
Сценарий 14. Получение списка образовательных треков	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Получить некоторое количество образовательных треков
Предусловие	Пользователь инициировал получение списка треков
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с параметрами о количестве записей и их смещении.	
2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимый список треков	
Результат	Список треков успешно возвращен
Альтернативные сценарии:	
2.1.1 Введенные данные пользователя не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения данных и записывает информацию об ошибке в файл журнала.	
Результат	Список треков не был возвращен
Сценарий 15. Получение подробной информации о треке по его идентификатору	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить информацию о выбранном треке

Предусловие	Пользователь инициировал получение информации о выбранном треке
Успешный сценарий:	
<ol style="list-style-type: none"> 1. Клиентское приложение отправляет запрос на сервер с идентификатором трека 2. Серверное приложение проверяет отправленные данные и возвращает клиентскому приложению необходимую информацию о треке 	
Результат	Информация о треке отправлена на клиентское приложение
Альтернативные сценарии:	
2.1.1 Отправленные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения данных и записывает информацию об ошибке в файл журнала.	
Результат	Информация о треке не отправлена
Сценарий 16. Обновление данных трека	
Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: обновить информацию о треке
Предусловие	Администратор инициировал обновление информации о треке
Успешный сценарий:	
<ol style="list-style-type: none"> 1. «Панель администратора» отправляет запрос на сервер с данными о треке 2. Серверное приложение проверяет отправленные данные, записывает их в базу данных и возвращает сообщение об успешной операции 	
Результат	Информация о треке успешно обновлена
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.	
Результат	Информация о треке не обновлена
Сценарий 17. Создание трека	

Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: создать новый трек на сайте
Предусловие	Администратор инициировал создание трека
Успешный сценарий:	
<ol style="list-style-type: none"> «Панель администратора» отправляет запрос на сервер с данными о треке Серверное приложение проверяет отправленные данные, записывает их в базу данных и возвращает сообщение об успешной операции 	
Результат	Трек успешно создан
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке изменения данных и записывает информацию об ошибке в файл журнала.	
Результат	Трек не создан
Сценарий 18. Удаление трека	
Действующие лица	Администратор, приложение «Панель администратора», серверное приложение
Цель	Администратор: удалить трек с сайта
Предусловие	Администратор инициировал удаление трека
Успешный сценарий:	
<ol style="list-style-type: none"> «Панель администратора» отправляет запрос на сервер с идентификатором трека Серверное приложение проверяет полученный идентификатор, удаляет выбранную запись и возвращает сообщение об успешности операции 	
Результат	Трек удален
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке удаления данных и записывает информацию об ошибке в файл журнала.	
Результат	Трек не удален
Сценарий 19. Начало миссии пользователем	

Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: начать выполнения миссии
Предусловие	Пользователь инициировал старт миссии
Успешный сценарий:	
<ol style="list-style-type: none"> 1. Клиентское приложение отправляет запрос на сервер с идентификатором миссии 2. Серверное приложение получает авторизованного пользователя и информацию от клиентского приложения и вносит их в базу данных 	
Результат	Миссия успешна начата
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке записи информации о миссии и записывает информацию об ошибке в файл журнала.	
Результат	Миссия не начата
Сценарий 20. Обновление информации о миссии	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: пройти этап миссии
Предусловие	Пользователь прошел точку сохранения во время выполнения миссии
Успешный сценарий:	
<ol style="list-style-type: none"> 1. Клиентское приложение отправляет запрос на сервер с данными пользователя о миссии (включая файлы). 2. Серверное приложение получает авторизованного пользователя и информацию от клиентского приложения и вносит их в базу данных, файлы сохраняются в директорию миссии пользователя 	
Результат	Информация о прохождении миссии успешно обновлена
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	

2.2.2 Серверное приложение возвращает сообщение о неудачной попытке записи информации о миссии и записывает информацию об ошибке в файл журнала.	
Результат	Данные не обновлены
Сценарий 21. Получение файлов миссии	
Действующие лица	Пользователь, клиентское приложение, серверное приложение
Цель	Пользователь: получить файлы, предоставляющие информацию о результате выполнения миссии
Предусловие	Пользователь инициировал получение файлов миссии
Успешный сценарий:	
1. Клиентское приложение отправляет запрос на сервер с информацией о лабораторной, количестве возвращаемых данных и параметрах их сортировки	
2. Серверное приложение получает авторизованного пользователя и информацию от клиентского приложения, после чего получает необходимую запись в базе данных и из нее уже получает названия всех файлов, после чего возвращает пользователю ссылки на необходимые файлы	
Результат	Данные успешно получены
Альтернативные сценарии:	
2.1.1 Введенные данные не проходят валидацию	
2.1.2 Серверное приложение возвращает сообщение о причине провала валидации	
2.2.1 Нет доступа к базе данных	
2.2.2 Серверное приложение возвращает сообщение о неудачной попытке получения информации о миссии и записывает информацию об ошибке в файл журнала.	
Результат	Данные не получены

1.5 Система управления базами данных

Для хранения данных использовалась СУБД MySQL, так как она является одной из самых наиболее приспособленных для применения в среде веб-разработки. Всего в базе содержится 51 таблица, однако в рамках данной работы использовалось только 14. Фрагмент базы данных, который использовался во время разработки представлен на рисунке 3.

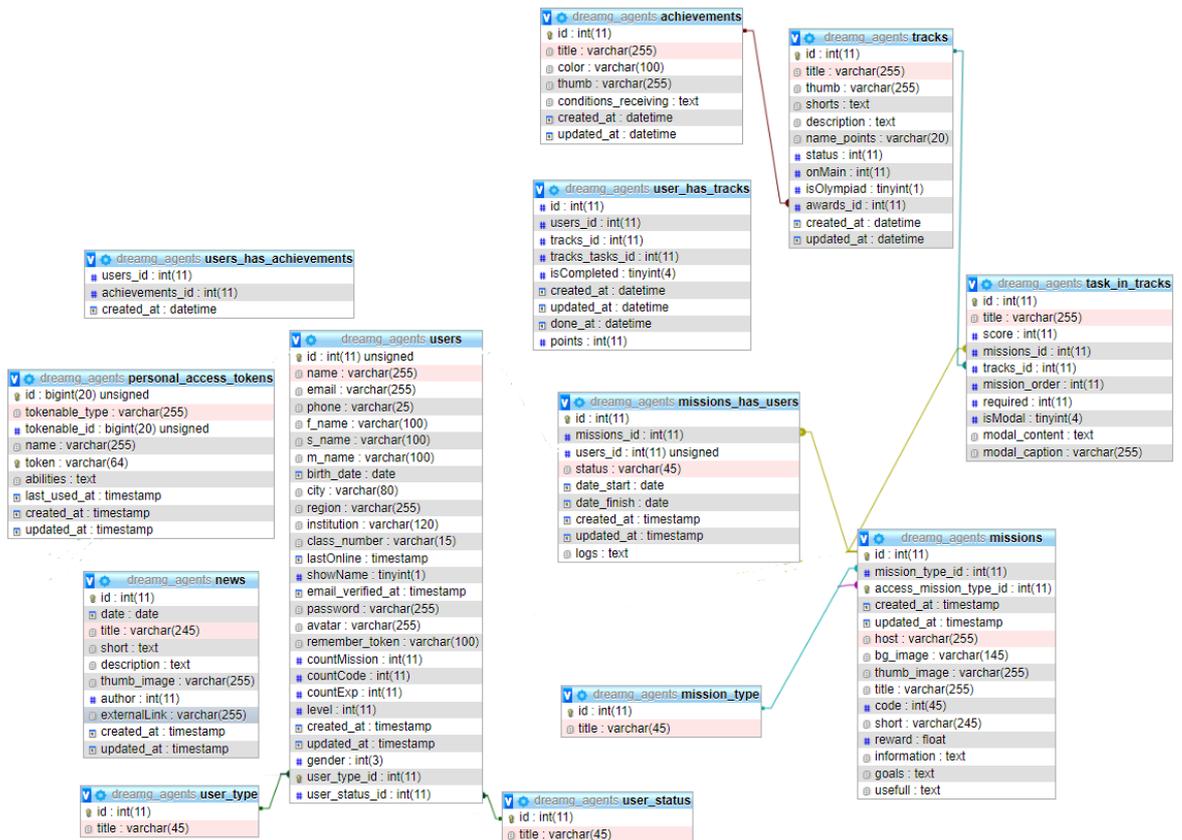


Рисунок 3 – Фрагмент физической модели базы данных

Данный фрагмент имеет следующие сущности:

- users – пользователи и их персональные данные;
- users_type – справочник типа пользователя;
- users_status – справочник статуса пользователя;
- personal_access_tokens – токены доступа;
- news – новости платформы;
- tracks – образовательные треки;
- missions – задания (миссии) платформы;
- task_in_tracks – миссии входящие в образовательные треки;
- missions_type – справочник типа миссий;
- missions_has_users – информация о прогрессе прохождений миссий пользователями;
- achievements – достижения;
- users_has_achievements – информация о достижениях пользователя.

2 Реализация приложения

2.1 Используемые технологии

Работа была выполнена с использованием следующих технологий:

- **PHP** – скриптовый язык программирования для разработки веб-приложений. Поддерживается большинством современных хостинг-провайдеров и является лидером среди языков, применяющихся для создания динамических веб-сайтов

- **Laravel** – бесплатный веб-фреймворк для разработки.

Предоставляет удобные инструменты для аутентификации, маршрутизации, работы с сессиями, кэшированием, базой данных. На данный момент является одним из лидеров среди фреймворков для разработки динамических сайтов.

- **TPU Gitlab** – веб-инструмент для системы управления репозиториями кода на базе Томского Политехнического Университета.

- **Postman** – инструмент для тестирования работы API. Позволяет провести быстрое тестирование интерфейса во время разработки.

- **Swagger** – набор программных инструментов для спецификации RESTful API. Благодаря использованию данной технологии появляется возможность обеспечить удобную систему документирования и тестирования. Также имеет специальный плагин для Laravel, что упрощает его внедрение и использование.

- **FileZilla** – свободный многоязычный FTP-клиент. Необходим для обмена файлами между сервером и ПК разработчика.

- **PuTTY** – свободно-распространяемый клиент для протоколов удаленного доступа. Используется для настройки платформы на стороне сервера.

2.2 Реализация OAuth авторизации

В настоящее время события, связанные с киберпреступностью, не являются чем-то уникальным и редким. Чтобы уменьшить потери от подобных действий, необходимо иметь надежные методы обеспечения безопасности. Среди множества существующих стратегий одной из самых лучших считается организация авторизации с помощью токенов.

Токен – это некий ключ, возвращаемый сервером при успешной авторизации. В данной работе использовался тип токена «Bearer Token». Его основным отличием является возможность использования предъявителем без доказательства его владения, то есть любое приложение, имея токен, получает доступ к ресурсам. Выбор данного типа связан с большим количеством сервисов и приложений, взаимодействующих в рамках системы.

Так как платформа планирует в будущем использовать несколько приложений для работы, было принято решение реализовать систему авторизации клиентов с третьей стороны с помощью протокола авторизации OAuth 2.0 [2]. Система работы аналогична авторизациям Google, VK, Facebook и т.п. и позволяет выдавать ограниченный набор прав приложению.

Порядок работы системы следующий: в начале разработчики приложения с третьей стороны должны зарегистрировать свое приложение в системе, создав в нем «клиента». Для этого необходимо указать имя своего приложения и URL-адрес, на который ваше приложение может перенаправить после того, как пользователи одобряют свой запрос на авторизацию. Для этого в библиотеке Laravel Passport присутствуют все необходимые API интерфейсы, соответственно, к ним необходимо просто обратиться, передав нужные параметры [3].

Если приложение авторизовано в системе, то пользователь может получить токен доступа для последующего взаимодействия. Для этого с приложения, которому необходима авторизация, необходимо отправить запрос следующего вида (рисунок 4):

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> grant_type	password			
<input checked="" type="checkbox"/> client_id	18			
<input checked="" type="checkbox"/> client_secret	DaihyOHaAl2Qz7I7fE9Rq9uCsI5ueUUKfh7MDjL4			
<input checked="" type="checkbox"/> username	test@test.com			
<input checked="" type="checkbox"/> password	password			
<input checked="" type="checkbox"/> scope	*			

Рисунок 4 – Пример запроса

Где:

- grant_type (обязательный параметр) — Параметр по которому проводится идентификация.
- client_id (обязательный параметр) – Идентификатор вашего приложения.
- client_secret (обязательный параметр) – «Секрет» вашего приложения.
- username (обязательный параметр) –Логин пользователя.
- password (обязательный параметр) – Пароль пользователя.
- scope – Битовая маска настроек доступа приложения, которые необходимо проверить при авторизации пользователя и запросить, в случае отсутствия необходимых.

Если все данные прошли проверку, то возвращается токен доступа и токен обновления, которые действуют 1 день. (рисунок 5).

```

"token_type": "Bearer",
"expires_in": 86400,
"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIiXOCIsImp0aSI6IjE2NTM3NDU3NjAuOTAzODcwMTA1NzQzNDU4MjAzMTI1LlZjZDZlIiwiaXNjaWkiOiIiX2VhbnQzIiwiaWF0Ij0iLCJpc0g9k8GPPG-Qp4dYIdJj1iE3_Rf4ZX--Nl9D5qD0kExKWJXc317QkCjvt4e7noNSjRXvh79Km7UPRTA6G4Fby mmdlryQW0pGzhQK9ZHkz3WKTgiRsdSHHmr1-TLuFQE7kWh1zc6swRdiMyTlU7Q8Ittdt3Vplax31fmdBW-DrRs4t6Xarvj4Ddrd55jLJdYL0d0Mgf2aS zQLbT3CNC4R7VnFv_LCQMOhHcVdcFHxr3bBwXVqX0nT86eRMSV6isq9rGJa3dQk9L5_dbG8e1aZr3xhN22jDN9yClEVSYu-Va4iX4bxIcRwyeUaTUYdA0N yWNud-sL2a62aa6KAHrxSRj9dXi9ZD3pjrIUNVDCYq5ndartE1nt76A8rE0lrHww7DdNaKuoD2xHbdZyc6eHxSW2qS9wsST1k",
"refresh_token":
"def502001659c56f61c1b357002aab5f307b8787c964269951de402abda290cf2e53ebdedd62e79c1e1492da5f0099e4f9d28002179621b78ab4 e5c1733420a254ed3ec9131812b2f74511085c11578bea5803b5d00501908d53fd1f53a4134a8e4c0695b969d9564967b8431e4d4cf5292007f07 39f3e31eef7f0029863c7294e05bb3aede0136c413b13c84eb3d772bcc1f8ddcbdf31786f8a14bad51cc18d4aa6b53f4f23ca3c4caf461710c32b 3382eb1e1c84c80802ba28d491c12977af9825afdaf03894aeb0f2a64d45f6a5989e5264e8e7909c588b01a3c1d91ec418ddcc260e19b934c45fb 30810f60e942c77e2c51e2e89b5151b2293952e5a79826ffa9b8fc47157f8dad59bb07991e2e14eb4c5de86a7d079878adf8edb08476918403d8a bf77ff8b706266125219b30eaaaf73a8ede9088983ca2a4d02b0062c0df7cc79ebcaf8b5cadcc93bbaabb07fc7c43399bf1045f7c0eb5a3def44ddb cb0b1a503867b"

```

Рисунок 5 – Пример возвращаемых данных

Данный токен предоставляет доступ к защищенным методам приложения и позволяет взаимодействовать с необходимыми данными.

2.3 Реализация интерфейсов

Согласно архитектуре, представленной на рисунке 2, для методов каждой сущности использовалась следующая структура папок (рисунок 6).

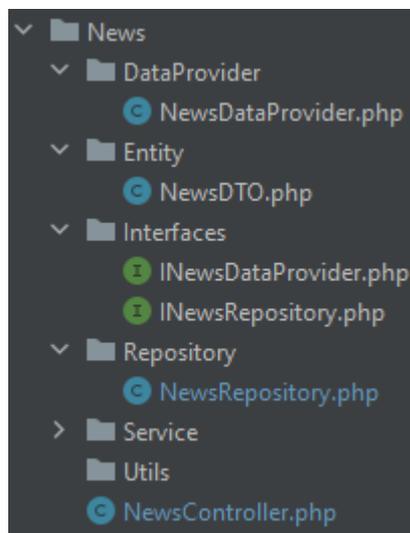


Рисунок 6 – Структура папок для методов

Рассмотрим подробнее класс контроллер (NewsController.php). В нем содержатся контроллеры подобного типа, ответственные за обращение к доступным методам провайдера данных (NewsDataProvider).

```

public function showSomeNumNews(?Request $request): JsonResponse
{
    $news = $this->newsDataProvider->showSomeNumNewsByOffset($request);
}

```

```

    if ($news === null) {
        return response()->json(['message' => 'News not found.', 'success' =>
false], 404);
    }
    return response()->json(['news' => $news, 'success' => true,], 200);
}

```

`INewsDataProvider` является интерфейсом реализующим класс провайдера данных. Подобное архитектурное решение позволяет взаимодействовать с классами, не зная их внутренней реализации.

```

interface INewsDataProvider
{
    public function showNewsById(int $id): ?NewsDTO;
    public function showSomeNumNewsByOffset (Request $request);
    public function updateNews (Request $request, int $id):bool;
    public function createNews (Request $request):bool;
    public function deleteNews (int $id):bool;
}

```

Сам класс провайдера содержит экземпляр класса репозитория и выполняет обращение к его методам с помощью аналогичного собственному интерфейса, реализующего класс репозитория:

```

class NewsDataProvider implements INewsDataProvider
{
    private INewsRepository $repository;

    public function __construct(INewsRepository $repository)
    {
        $this->repository = $repository;
    }

    public function showNewsById(int $id): ?NewsDTO
    {
        return $this->repository->getNewsById($id);
    }
}

```

Попадая в метод репозитория, данные проходят процесс валидации с помощью встроенных средств фреймворка:

```

$validated = $request->validate([
    'limit' => 'bail|required|numeric',
    'offset' => 'bail|required|numeric',
]);

```

Затем проходит преобразование данных (по необходимости) и запрос к базе данных с помощью встроенного в фреймворк фасада DB:

```

$news = DB::table('news')->orderByDesc('news.updated_at')->
>limit($validated['limit'])->offset($validated['offset'])->
join('users', 'author', '=', 'users.id')->get();

```

Если необходимо вернуть сущность или массив сущностей (к примеру, несколько новостей), данные преобразуются в DTO:

```
$data = [];  
foreach ($news as $_news) {  
    $itemNews = new NewsDTO($_news);  
    $data[] = $itemNews;  
}
```

Для отлавливания ошибок в приложении используется блок try-catch, который позволяет при сбое записать сообщение об ошибке в журнальный файл и вернуть необходимый в класс необходимый ответ:

```
catch (\Exception $e) {  
    Log::error($e->getMessage(), ['trace' => $e->getTraceAsString()]);  
    return null;  
}
```

2.3 Описание программных интерфейсов

В соответствии со сценариями использования были разработаны программные интерфейсы для следующих компонентов:

1. классическая регистрация, авторизация, аутентификация (см. Приложение А);
2. профиль пользователя;
3. новости;
4. образовательные треки;
5. задания (миссии).

Подробная информация о всех реализованных методах представлена в таблице 2. Также в списке литературы [4] представлена ссылка на Git-репозиторий, перейдя по которой можно изучить реализацию всех методов

Таблица 2 – Реализованные методы

Программный интерфейс	Метод	Входные данные	Возвращаемый код	Возвращаемые данные
Регистрация, авторизация, аутентификация	register	name	201	Токен
		email password	500	Описание ошибки
	login	name email	201	Токен
			401	Некорректные данные
			501	Описание ошибки
	logout	-	200	ОК
			500	Описание ошибки
	Профиль пользователя	show	id	200
404				Пользователь не найден
400				Плохой запрос

	update	Данные пользователя	200	ОК	
	updatePhoto		400	Плохой запрос	
	showAchievements	id	200	Данные пользователя	
			404	Описание ошибки	
	showSomeNumUsers	limit offset	200	Список пользователей	
			400	Плохой запрос	
			404	Пользователи не найдены	
	Новости	show	id	200	Данные новости
				400	Плохой запрос
404				Новость не найдена	
showSomeNumNews		limit offset	200	Список новостей	
			404	Новости не найжены	
create		Данные новости	200	ОК	
			400	Плохой запрос	
update		Данные новости id	200	ОК	
			400	Плохой запрос	
delete		id	200	ОК	
			400	Плохой запрос	
Треки		show	id	200	Данные трека

			400	Плохой запрос	
			404	Трек не найден	
	showSomeNumTracks	limit offset	200	Список треков	
			404	Треки не найдены	
	create	Данные трека	200	ОК	
			400	Плохой запрос	
	update	Данные трека id	200	ОК	
			400	Плохой запрос	
	delete	id	200	ОК	
			400	Плохой запрос	
	Задания (миссии)	setMissionData	Данные о прогрессе миссии	200	ОК
				400	Плохой запрос
getFiles		id миссии, количество и порядок сортировки файлов	200	Список ссылок на файлы	
			400	Плохой запрос	
setFiles		id миссии, файлы	200	ОК	
			400	Плохой запрос	

2.4 Тестирование интерфейсов

Во время разработки интерфейсов очень часто необходимо тестировать его работоспособность при различных сценариях. В данной работе большинство тестов проводилось с помощью инструмента Postman [5]. Для каждой коллекции методов была создана директория, в которой хранились запросы для тестирования каждого метода (рисунок 7).



Рисунок 7 – Запросы для тестирования (новости)

Само тело запроса представлено на рисунке 8. Сверху указывается HTTP-метод, по которому выполняется запрос и URL-адрес. В центре расположены данные, отправляемые на сервер. Формат данных – JSON. Также для проверки защиты методов присутствует вкладка авторизации, на которой указывается токен аутентификации.

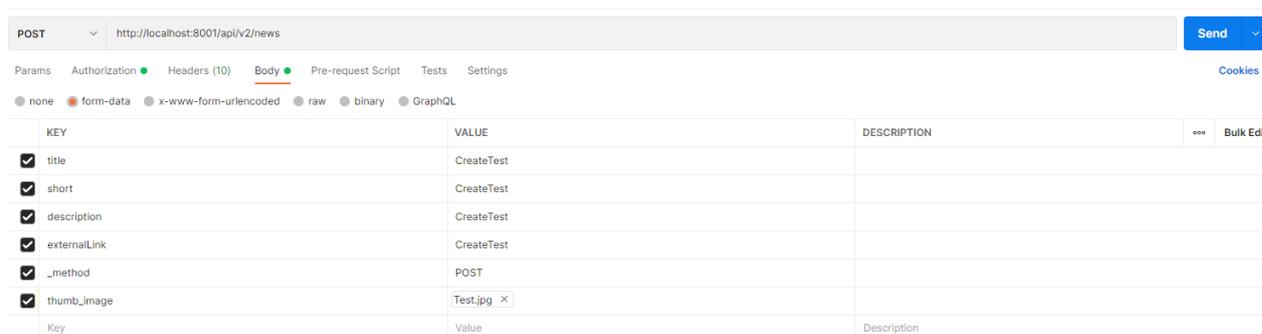


Рисунок 8 – Тело запроса

При отправке запроса внизу приложения можно посмотреть ответ сервера (рисунок 9).

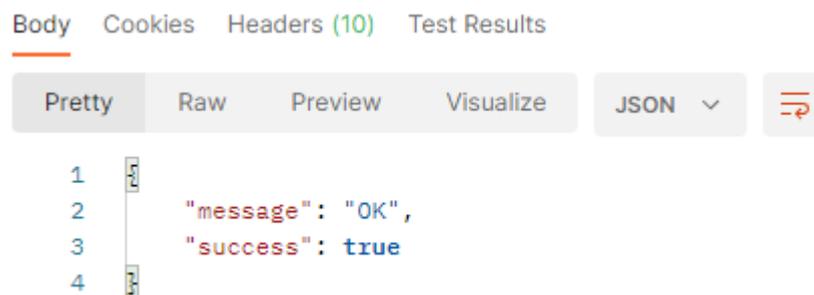


Рисунок 9 – Пример ответа от сервера

2.5 Документация для интерфейсов

Для каждого запроса необходимо составлять документацию [6]. Это необходимо для комфортного взаимодействия между членами команды. В данном проекте для генерации документации использовался специальный пакет [7] для интеграции Swagger [8] в среду Laravel. Его преимущества заключаются в возможности быстрого создания документации прямо по ходу разработки с помощью комментариев методов:

```
/**
 * @OA\Post (
 *   path="/news/show",
 *   operationId="showSomeNumNews",
 *   tags={"News"},
 *   summary="Show some amount of news depending on limit and offset
properties",
 *   @OA\Response (
 *     response="200",
 *     description="OK",
 *     @OA\JsonContent ()
 *   ),
 *   @OA\RequestBody (
 *     required=true,
 *     @OA\JsonContent (ref="#/components/schemas/LimitOffsetReq")
 *   )
 * )
 */
```

Составив таким образом комментарии к каждому методу и настроив маршрут, на котором будет располагаться страница, получается реализовать документацию в следующем формате (рисунок 10).

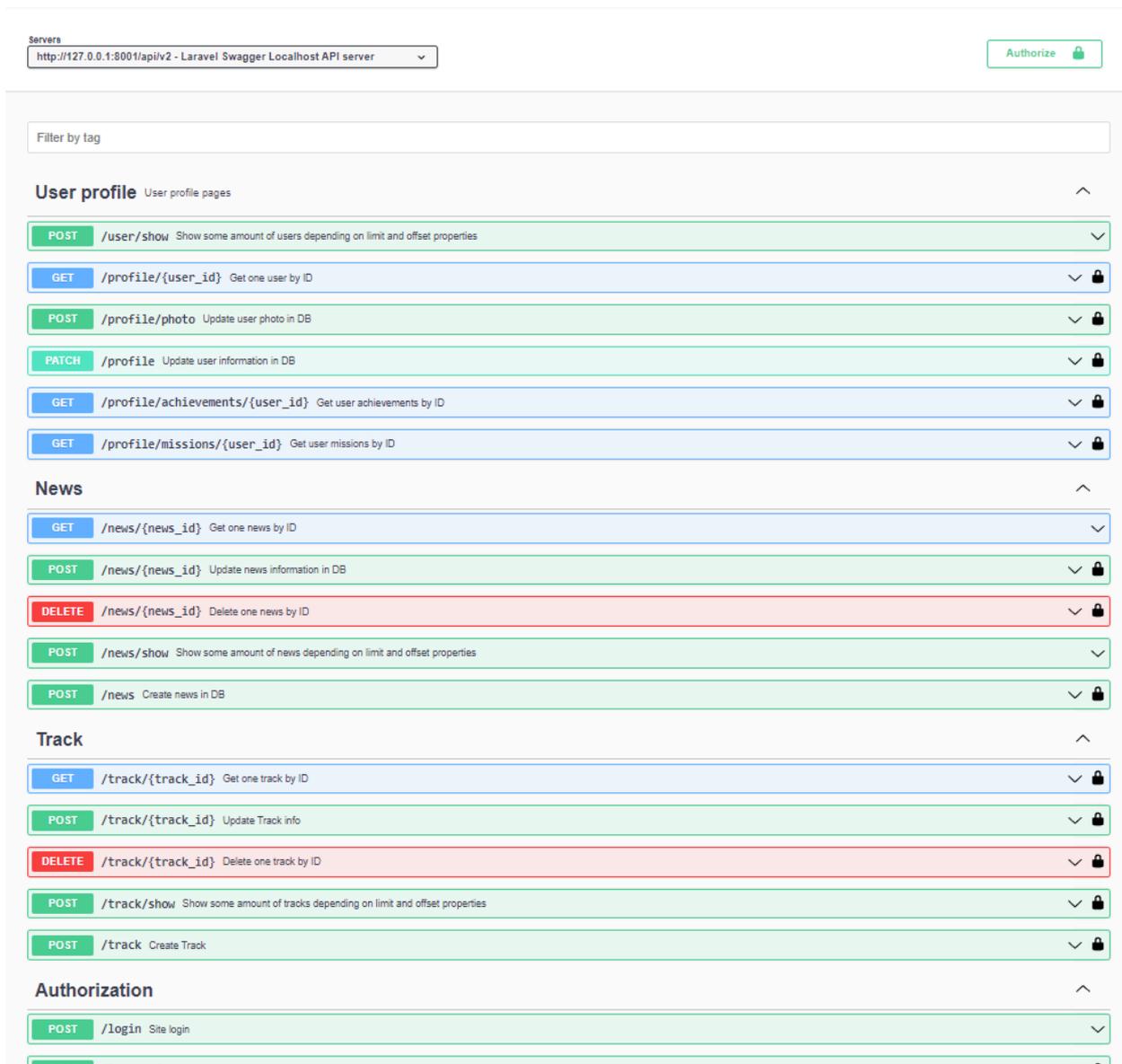


Рисунок 10 – Общий вид документации

В верхней части страницы располагаются поля, ответственные за сервер, на котором производится тестирование и поле авторизации, куда необходимо ввести токен аутентификации для доступа к защищенным методам.

При раскрытии метода появляется более подробная информация о запросе, а также появляется возможность протестировать работу метода аналогично Postman (рисунок 11).

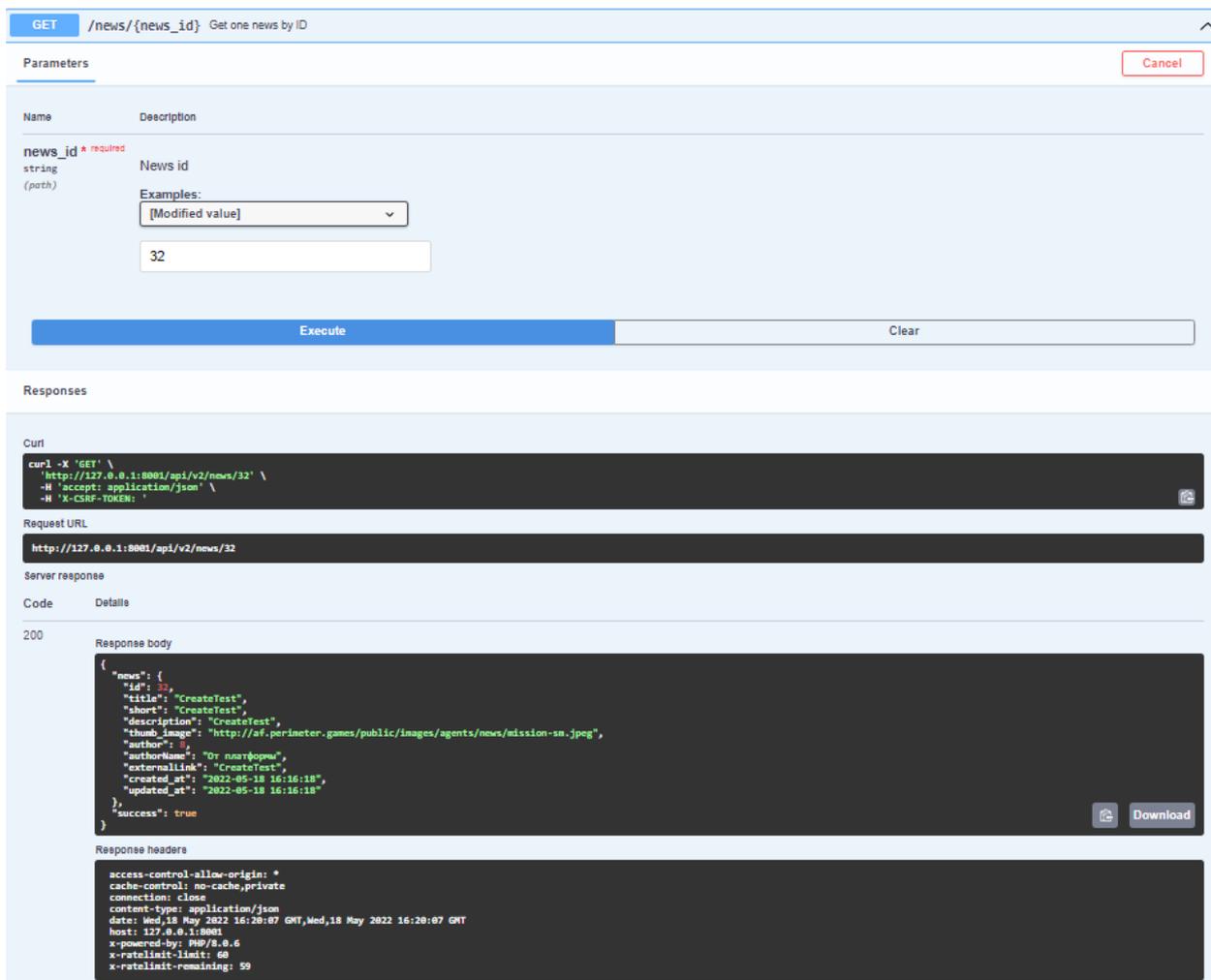


Рисунок 11 – Подробная информация о методе

2.6 Загрузка файлов на сервер

После прохождения всех тестов на локальном сервере, необходимо выполнить загрузку файлов на тестовый сервер в сети. Для этого в работе используется FTP-клиент FileZilla, который позволяет достаточно быстро и удобно выполнить подключение к серверу и перенести на него необходимые данные (рисунок 12).

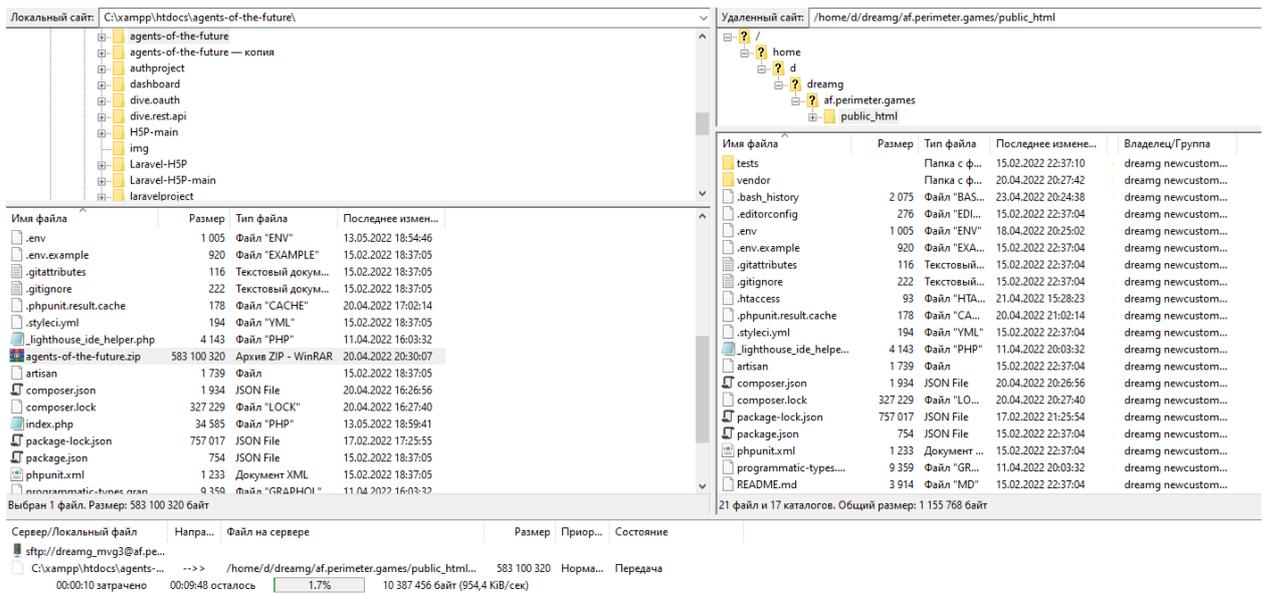


Рисунок 12 – Отправка файлов на удаленный сервер

После отправки файлов необходимо настроить работу проекта на сервере. Для подключения к удаленному серверу по сетевому протоколу SSH использовался клиент для удаленного доступа PuTTY. Для успешной работы проекта необходимо выполнить следующие команды:

- «php artisan view: clear» – очистка кэша представлений;
- «php artisan route: clear» – очистка кэша маршрутов;
- «php artisan cache: clear» – очистка кэша приложения;
- «php artisan config: cache» – создание кэш файла [9].

Реализация данных команд представлена на рисунке 13.

```
af.perimeter.games - PuTTY
login as: dreamg_mv33
Pre-authentication banner message from server:
| Welcome to LTD BeGet SSH Server 'rayman'
End of banner message from server
dreamg_mv33@af.perimeter.games's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 5.10.109-1-beget-acl x86_64)

* Documentation: https://help.ubuntu.com/
Last login: Sat Apr 23 09:17:00 2022 from 85.143.78.82
dreamg_mv33@rayman:~ [0] $ php8.0 artisan view:clear
Compiled views cleared successfully.
dreamg_mv33@rayman:~ [0] $ php8.0 artisan route:clear
Route cache cleared successfully.
dreamg_mv33@rayman:~ [0] $ php8.0 artisan cache:clear
Application cache cleared successfully.
dreamg_mv33@rayman:~ [0] $ php8.0 artisan config:cache
Configuration cache cleared successfully.
Configuration cached successfully.
dreamg_mv33@rayman:~ [0] $
```

Рисунок 13 – Публикация проекта на сервере

После сборки проекта на сервере все тесты, выполняемые на локальном сервере, проводятся и на удаленном.

2.7 Система контроля версий

После прохождения всех тестов как на локальной, так и на виртуальной машине код проекта публиковался в системе управления репозиториями TPU GitLab [10]. Для своего пользователя была создана отдельная ветвь в проекте, что позволяет вести разработку приложения параллельно с другими разработчиками без возникновения конфликтных ситуаций (рисунок 14).

19 Apr, 2022 1 commit		
Bug fixes, Add News and Track controllers gaskinMV authored 4 weeks ago	e1898e92	
12 Apr, 2022 1 commit		
Added GraphQL gaskinMV authored 1 month ago	b87b1057	
11 Apr, 2022 1 commit		
Added swagger documentation gaskinMV authored 1 month ago	8f6631ed	
02 Apr, 2022 1 commit		
Added User profile controllers gaskinMV authored 1 month ago	1e0c6a66	
17 Feb, 2022 1 commit		
API authorization works gaskinMV authored 2 months ago	8051a023	

Рисунок 14 – Список коммитов

3 Финансовый менеджмент, ресурсоэффективность и

ресурсосбережение

3.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

3.1.1 Потенциальные потребители результатов исследования

Выпускная квалификационная работа посвящена разработке серверного приложения для веб-платформы «Агенты будущего». «Агенты будущего» представляют собой платформу для создания и проведения мероприятий по дополнительному образованию. Также на платформе представлены проверочные задания в геймифицированной форме. На сайте регулярно публикуются новости, присутствуют системы уровней и достижений.

Потенциальными потребителями продукта являются школьники, заинтересованные в получении дополнительного образования и профессиональной ориентации по техническому направлению, а также образовательные учреждения, желающие привлечь своих учеников к дополнительным источникам получения знаний.

3.1.2 Анализ конкурентных технических решений

Проведение детального анализа конкурирующих разработок, существующих на рынке, необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Подобный анализ помогает вовремя вносить изменения в исследование, чтобы иметь возможность эффективно существовать на рынке.

Разрабатываемая платформа не имеет прямых аналогов, однако с точки зрения косвенных аналогов можно рассмотреть несколько наиболее популярных решений:

- российская образовательная платформа «Stepik»;

– российская система электронного онлайн-образования
«Универсариум»;

Сравнение технических и экономических характеристик данных продуктов с разрабатываемым решением представлено в таблице 3.

Таблица 3 – Оценочная карта сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Удобство в эксплуатации	0,2	5	4	5	1	0,8	0,8
2. Функциональные возможности	0,3	5	4	3	1,5	1,2	0,9
3. Быстродействие	0,2	3	4	5	0,8	0,8	1
4. Надежность	0,1	4	4	4	0,4	0,4	0,4
Экономические критерии оценки эффективности							
5. Цена	0,1	5	4	3	0,5	0,4	0,3
6. Уровень проникновения на рынок	0,1	2	3	4	0,2	0,3	0,4
Итого:	1	24	23	24	4,4	3,9	3,8

Анализ конкурентных решений определяется по формуле:

$$K = \sum V_i * B_i,$$

где K – конкурентоспособность научной разработки или конкурента;

V_i – вес показателя (в долях единицы);

B_i – балл *i*-го показателя.

Из оценочной карты сравнений можно сделать выводы что разрабатываемая система имеет преимущества:

- удобство в эксплуатации;
- функциональные возможности;
- надежность;
- цена.

Недостатками системы являются:

- быстроедействие;
- уровень проникновения на рынок.

Также исходя из таблицы 3 можно сделать вывод, что конкурентоспособность системы выше, чем у аналогичных решений, соответственно, проведение разработки данного приложения целесообразно.

3.1.3 SWOT-анализ

Успешность проекта зависит от того, насколько он устойчив к различным изменениям извне. Для анализа сильных и слабых сторон проекта был решено использовать SWOT-анализ. Его сущность заключается в анализе внутренних и внешних факторов, оценке рисков и конкурентоспособности товара в отрасли.

В таблице 4 приведены результаты анализа в виде матрицы.

Таблица 4 – Матрица SWOT-анализа

Сильные стороны	Возможности во внешней среде
С1. Удобство и простота использования С2. Выполнение в веб-браузере С3. Актуальность разработки С4. Постоянная поддержка проекта	В1. Адаптация под мобильные устройства В2. Расширение функционала В3. Появление дополнительного спроса на продукт В4. Повышение стоимости конкурентных разработок
Слабые стороны	Угрозы внешней среды
Сл1. Небольшой штат разработчиков Сл2. Слабая известность продукта Сл3. Полная поддержка только одной платформы – ПК	У1. Переход используемых средств разработки на платную основу У2. Развитие конкурирующих разработок У3. Слабая заинтересованность целевой аудитории

Далее необходимо выявить соответствие сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Для этого используется интерактивная матрица проекта. Ее применение позволяет выявить различные комбинации взаимосвязей областей матрицы SWOT. Каждый фактор помечается либо знаком «+» (означает сильное соответствие), либо знаком «-» (означает слабое соответствие). Символ «0»

ставится в том случае, если есть сомнения в выборе между первыми двумя вариантами. Интерактивная матрица проекта представлена в таблицах 5-6.

Таблица 5 – Интерактивная матрица сильных и слабых сторон и возможностей

Возможности проекта	Сильные стороны				Слабые стороны		
		C1	C2	C3	C4	Сл1	Сл2
B1	+	0	+	+	+	-	+
B2	+	-	+	+	+	+	-
B3	+	+	+	+	0	+	+
B4	+	-	+	+	-	0	+

Таблица 6 – Интерактивная матрица сильных сторон и слабых сторон и угроз

Угрозы проекта	Сильные стороны				Слабые стороны		
		C1	C2	C3	C4	Сл1	Сл2
У1	-	-	-	+	+	-	0
У2	0	0	-	+	+	+	+
У3	+	+	+	+	-	+	+

В результате анализа интерактивных таблиц можно выделить следующие сильно коррелирующие сильные и слабые стороны с условиями внешней среды:

- B1B2B3C1C3C4; B3C2;
- B1B2Сл1; B2B3Сл2; B1B3B4Сл3;
- У1У2У3С4; У3С1С2С3;
- У1У2Сл1; У2У3Сл2Сл3.

Самой вероятной и большой угрозой является переход используемых средств разработки на платную основу. Это приведет к невозможности обновления ресурса и перевода сайта на другие технологии, что является очень затратным по материальным ресурсам.

В заключение данного этапа составляется итоговая матрица SWOT-анализа, представленная в таблице 7.

Таблица 7 – Итоговая матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта:	Слабые стороны научно-исследовательского проекта:
--	---	--

	С1. Удобство и простота использования С2. Выполнение в веб-браузере С3. Актуальность разработки С4. Постоянная поддержка проекта	Сл1. Небольшой штат разработчиков Сл2. Слабая известность продукта Сл3. Полная поддержка только одной платформы – ПК
Возможности: В1. Адаптация под мобильные устройства В2. Расширение функционала В3. Появление дополнительного спроса на продукт В4. Повышение стоимости конкурентных разработок	Постоянная поддержка проекта в условиях доминирования на рынке позволит повысить его конкурентоспособность и расширить функциональные возможности, а, соответственно, и сферу влияния.	Привлечение дополнительных специалистов позволит ускорить сроки работы над проектом. При появлении дополнительного спроса на продукт, появится больше экономических ресурсов для его последующего развития.
Угрозы: У1. Переход используемых средств разработки на платную основу У2. Развитие конкурирующих разработок У3. Слабая заинтересованность целевой аудитории	За счет постоянной поддержки проекта имеется возможность перевода ресурса на другой стек технологий и развитие для возможности сохранения дальнейшей конкурентоспособности.	Отсутствие спроса на товар и необходимость его изменения приведет к потере конкурентоспособности и потенциальных клиентов.

3.2 Определение возможных альтернатив проведения научных исследований

Для выявления возможных альтернатив разработки проекта был использован морфологический подход. Он основан на подборе возможных решений для отдельных частей задачи и последующем систематизированном получении их сочетаний. В таблице 8 в виде матрицы представлены возможные варианты реализации разработки.

Таблица 8 – Морфологическая таблица

	1	2
А. Способ организации кода	MVC	REST API

Б. Фреймворк для реализации	Laravel	Yii
В. Среда разработки	PhpStorm	NetBeans
Г. Используемый сервер	Сервер ТПУ	Арендуемый сервер

Путем комбинации различных параметров были определены три наиболее оптимальных варианта исполнения:

- А1Б2В2Г2;
- А2Б1В1Г1;
- А2Б2В1Г2.

Вариантом, используемым при разработке проекта, является А2Б1В1Г1. Это связано с наличием бесплатной студенческой лицензии у среды разработки «PhpStorm», а также особенностями уже существующих компонентов сайта.

3.3 Планирование научно-исследовательских работ

3.3.1 Структура работ в рамках научного исследования

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может изменяться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей.

В данной исследовательской работе количество участников равно трем: руководитель практики, консультант и бакалавр. Перечень этапов и работ, распределение по данным видам работ приведены в таблице 9.

Таблица 9 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель
Выбор направления исследований	2	Определение целей исследования	Студент
	3	Подбор и изучение материалов по теме	Студент
	4	Составление календарного плана	Консультант
Реализация	5	Определение средств разработки	Студент, консультант
	6	Проектирование архитектуры	Студент, консультант
	7	Разработка приложения	Студент
	8	Тестирование	Студент
	9	Разработка документации	Студент
Внедрение приложения	10	Развертывание приложения в сети	Студент
Оформление отчета по ВКР	11	Составление пояснительной записки	Студент

3.3.2 Определение трудоемкости выполнения работ

Трудовые затраты являются одной из самых больших статей расходов исследования, поэтому важным моментом является точное определение трудоемкости работ каждого участника исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожи}$ используется следующая формула:

$$t_{ожи} = \frac{3t_{\min i} + 2t_{\max i}}{5},$$

где $t_{ожи}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.

После получения значений трудоемкости необходимо определить продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65%. Для расчета используется следующая формула:

$$T_{pi} = \frac{t_{ожi}}{Ч_i}$$

где T_{pi} – продолжительность одной работы, раб. дн.;

$t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Для составления календарного план-графика также выполняется расчет длительности в календарных днях по следующей формуле:

$$T_{ki} = T_{pi} * k,$$

где T_{pi} – продолжительность одной работы, раб. дн.;

k – коэффициент календарности (1,43).

Результаты расчетов трудоемкости работ представлены в таблице 8.

Таблица 10 – Временные показатели проведения научного исследования

Работы	Исполнитель	Трудоемкость			Длительность работ в рабочих днях T_{pi}	Длительность работ в календарных днях T_{ki}
		tmin	tmax	toжi		
Составление и утверждение технического задания	Руководитель	2	4	2,8	3	5
Определение целей исследования	Студент	1	3	1,8	2	3
Подбор и изучение материалов по теме	Студент	5	10	7	7	10

Составление календарного плана	Студент , консультант	1	3	1,8	1	2
Определение средств разработки	Студент , консультант	1	3	1,8	1	2
Проектирование архитектуры	Студент , консультант	5	10	7	4	6
Разработка приложения	Студент	22	33	26,4	27	38
Тестирование	Студент	5	7	5,8	6	9
Разработка документации	Студент	5	7	5,8	6	9
Развертывание приложения в сети	Студент	5	7	5,8	6	9
Составление пояснительной записки	Студент	7	12	9	9	13

3.3.3 Разработка графика проведения научного исследования

Одним из самых популярных методов представления графика научных работ за счет своего удобства и наглядности является диаграмма Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

С помощью значений, полученных в таблице 10, было выполнено построение диаграммы Ганта, представленной на рисунке 15.



Рисунок 15 – Календарный план-график проведения НИОКР

3.3.4 Бюджет научно-технического исследования (НТИ)

При планировании бюджета научно-технического исследования необходимо предоставить полную информацию о всех видах расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- материальные затраты НТИ;
- затраты на специальное оборудование для научных (экспериментальных работ);
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

3.3.4.1 Расчет материальных затрат НТИ

Расчет материальных затрат осуществляется по формуле:

$$Z_M = (1 + k_T) \cdot \sum_{i=1}^m C_i \cdot N_{расх\ i} ,$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

C_i – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Результаты расчетов материальных затрат представлены в таблице 11.

Таблица 11 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.	Затраты на материалы, (Зм), руб.		
		Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3
Тетрадь для записей	Шт.	1	1	1	50	50	50	50
Ручка	Шт.	1	1	1	30	30	30	30
Электроэнергия	кВт*ч	318	346	412	3,85	1225	1332	1587
Итого, руб.						1305	1412	1667

Итого общие материальные затраты составили 1305 руб.

3.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных) работ

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ по конкретной теме. Определение стоимости спецоборудования производится по действующим прейскурантам, а в ряде случаев по договорной цене.

Все используемое программное обеспечение предоставляется на бесплатной основе либо по бесплатной студенческой лицензии, поэтому в статью затрат включено не было.

Результат расчета затрат по данной статье представлен в таблице 12.

Таблица 12 – Бюджет затрат на приобретение спецоборудования для научных работ.

№ п/п	Наименование оборудования	Кол-во единиц оборудования			Цена единицы оборудования тыс. руб.			Общая стоимость оборудования, тыс. руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
1	Персональный компьютер	1	1	1	87500	87500	87500	87500	87500	87500
2	Мышь	1	1	1	5500	5500	5500	5500	5500	5500
3	Клавиатура	1	1	1	1000	1000	1000	1000	1000	1000
4	Роутер	1	1	1	1500	1500	1500	1500	1500	1500
5	Аренда сервера (12 месяцев)	0	1	1	0	100800	100800	0	100800	108000
Итого:								95500	196300	196300

3.3.4.3 Основная заработная плата исполнителей темы

Данная статья расходов включает основную заработную плату с учетом премий и доплат для исполнителей проекта: студента, консультанта и научного руководителя. Основная заработная плата ($Z_{\text{осн}}$) руководителя (лаборанта, инженера) рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} * T_{\text{р}},$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника;

$T_{\text{р}}$ – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} * M}{F_{\text{д}}},$$

где $Z_{\text{м}}$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года:

- при отпуске в 24 раб. дня $M = 11,2$ месяца, 5-дневная неделя;
- при отпуске в 48 раб. дней $M = 10,4$ месяца, 6-дневная неделя;

F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (таблица 13).

Таблица 13 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Консультант	Студент
Календарное число дней	365	365	365
Количество нерабочих дней:			
- выходные дни	118	118	118
- праздничные дни			
Потери рабочего времени:			
- отпуск	24	24	24
- невыходы по болезни			
Действительный годовой фонд рабочего времени	223	223	223

Месячный должностной оклад работника рассчитывается по следующей формуле:

$$Z_m = Z_{тс} * (1 + k_{пр} + k_d) * k_p,$$

где $Z_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30% от $Z_{тс}$);

k_d – коэффициент доплат и надбавок равный приблизительно 0,2;

k_p – районный коэффициент, равный 1,3 (для Томска).

Результаты расчетов основной заработной платы представлены в таблице 14.

Таблица 14 – Основная заработная плата

Исполнители	$Z_{тс}$	$k_{пр}$	k_d	Z_m	$Z_{дн}$	Тр, раб. дни			Зосн, руб.		
						Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Научный руководитель	30940	0,3	0,2	60333	3153	3	3	3	9459	9459	9459
Консультант	17030	0,3	0,2	33209	1736	6	6	6	10416	10416	10416
Студент	13890	0,3	0	23474	1227	69	69	69	84663	84663	84663
Итого:									104538	104538	104538

3.3.4.4 Дополнительная заработная плата исполнителей темы

Данная статья расходов включает заработную плату, начисленную рабочим и служащим не за фактически выполненные работы или проработанное время, а в соответствии с действующим законодательством, в том числе оплата очередных отпусков рабочих, времени, связанного с выполнением государственных и общественных обязанностей. Зная основную заработную плату, можно рассчитать дополнительную заработную плату в размере 13% от основной по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}},$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы;

$Z_{\text{осн}}$ – основная заработная плата.

Результаты расчетов дополнительной заработной платы представлены в таблице 15.

Таблица 15 – Дополнительная заработная плата

Исполнитель	Основная заработная плата, руб.			Коэффициент дополнительной заработной платы	Дополнительная заработная плата, руб.		
	Исп. 1	Исп. 2	Исп. 3		Исп.1	Исп.2	Исп.3
Научный руководитель	9459	9459	9459	0,13	1230	1230	1230
Консультант	10416	10416	10416		1354	1354	1354
Студент	84663	84663	84663		11006	11006	11006
Итого:					13590	13590	13590

3.3.4.5 Отчисления во внебюджетные фонды

Данная статья расходов отражает обязательные отчисления, по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Сумма отчисления определяет по следующей формуле:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}),$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды, в соответствии с Федеральным законом для учреждений,

осуществляющих образовательную и научную деятельность, используется пониженная ставка – 30%;

$Z_{\text{осн}}$ – основная заработная плата;

$Z_{\text{доп}}$ – дополнительная заработная плата.

Результаты расчетов отчислений во внебюджетные фонды представлены в таблице 16.

Таблица 16 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Научный руководитель	9459	9459	9459	1230	1230	1230
Консультант	10416	10416	10416	1354	1354	1354
Студент	84663	84663	84663	11006	11006	11006
Коэффициент отчислений во внебюджетные фонды	0,3					
Итого:						
Исполнение 1	35438					
Исполнение 2	35438					
Исполнение 3	35438					

3.3.4.6 Накладные расходы

Данная статья расходов включает прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется согласно следующей формуле:

$$Z_{\text{нак}} = k_{\text{нр}} * \sum \text{статей},$$

где $k_{\text{нр}}$ – коэффициент накладных расходов, принятый за 16%.

Накладные расходы для исполнения 1 составили:

$$Z_{\text{нак}} = (1305 + 95500 + 104538 + 13590 + 35438) * 0,16 = 40060 \text{ рублей}$$

Накладные расходы для исполнения 2 составили:

$$Z_{\text{нак}} = (1412 + 196300 + 104538 + 13590 + 35438) * 0,16 = 56204 \text{ рублей}$$

Накладные расходы для исполнения 3 составили:

$$Z_{\text{нак}} = (1667 + 196300 + 104538 + 13590 + 35438) * 0,16 = 56459 \text{ рублей}$$

3.3.4.7 Формирование бюджета затрат научно-исследовательского проекта

Расчитанные величины затрат научно-исследовательской работы являются основой для формирования бюджета затрат проекта. Результаты составления итогового бюджета разработки представлены в таблице 17.

Таблица 17 – Итоговый бюджет разработки

Наименование статьи	Сумма, руб.		
	Исп.1	Исп.2	Исп.3
Материальные затраты НТИ	1305	1412	1667
Затраты на специальное оборудование для научных (экспериментальных) работ	95500	196300	196300
Затраты по основной заработной плате исполнителей темы	104538	104538	104538
Затраты по дополнительной заработной плате исполнителей темы	13590	13590	13590
Отчисления во внебюджетные фонды	35438	35438	35438
Накладные расходы	40060	56204	56459
Бюджет затрат НТИ	290430	407482	407992

3.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Для определения эффективности НТИ необходимо рассчитать интегральный показатель финансовой эффективности и интегральный показатель эффективности.

Интегральный финансовый показатель определяются по следующей формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}},$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Для исполнения 1: $I_{\text{финр}} = 290430/407992 = 0,71$.

Для исполнения 2: $I_{\text{финр}} = 407482/407992 = 0,99$.

Для исполнения 3: $I_{\text{финр}} = 407992/407992 = 1$.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i * b_i,$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Результаты расчетов интегрального показателя ресурсоэффективности представлены в таблице 18.

Таблица 18 – Сравнительная оценка характеристик вариантов исполнения

Объект исследования / Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Функциональные возможности	0,3	5	4	4
2. Быстродействие	0,3	3	3	4
3. Удобство в эксплуатации	0,2	5	3	3
4. Надежность	0,2	4	4	4
Итого:	1	4,2	3,5	3,8

$$I_{p\text{-исп1}} = 5 * 0,3 + 3 * 0,3 + 5 * 0,2 + 4 * 0,2 = 4,2;$$

$$I_{p\text{-исп2}} = 4 * 0,3 + 3 * 0,3 + 3 * 0,2 + 4 * 0,2 = 3,5;$$

$$I_{p\text{-исп3}} = 4 * 0,3 + 4 * 0,3 + 3 * 0,2 + 4 * 0,2 = 3,8.$$

Интегральный показатель эффективности вариантов исполнения проекта определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{исп1}} = \frac{I_{p\text{-исп1}}}{I_{\text{финр}}^{\text{исп1}}}, I_{\text{исп2}} = \frac{I_{p\text{-исп2}}}{I_{\text{финр}}^{\text{исп2}}}, I_{\text{исп3}} = \frac{I_{p\text{-исп3}}}{I_{\text{финр}}^{\text{исп3}}}$$

Таким образом:

$$I_{\text{исп1}} = 4,2/0,71 = 5,92;$$

$$I_{\text{исп2}} = 3,5/0,99 = 3,5;$$

$$I_{\text{исп3}} = 3,8/1 = 3,8;$$

Для определения самого выгодного варианта с позиции финансовой и ресурсной эффективности необходимо найти сравнительную эффективность исполнений разработки по следующей формуле:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп1}}}{I_{\text{исп2}}}$$

Результаты расчетов сравнительной эффективности разработки представлены в таблице 19.

Таблица 19 – Сравнительная эффективность разработки

№	Показатели	Исп. 1	Исп. 2	Исп.3
1	Интегральный финансовый показатель разработки	0,71	0,99	1
2	Интегральный показатель ресурсоэффективности разработки	4,2	3,5	3,8
3	Интегральный показатель эффективности	5,92	3,5	3,8
4	Сравнительная эффективность вариантов исполнения	1	0,59	0,64

Таким образом, сравнив значения интегральных показателей эффективности, можно сделать вывод, что самым эффективным исполнением с позиции ресурсоэффективности и финансовой эффективности является первое исполнение.

4 Социальная ответственность

В приведенной главе приводится описание вопросов санитарных норм и правил процесса эксплуатации разрабатываемой веб-платформы. Исследуются вредные и опасные факторы среды. Рассматриваются вопросы охраны окружающей среды. При написании работы было проведено исследование возможных чрезвычайных ситуации и действий, выполняемых сотрудником при возникновении ЧС.

Разрабатываемая платформа используется физическими и юридическими лицами, заинтересованными в создании и проведении мероприятий по дополнительному школьному образованию. Доступ к платформе осуществляется с любого устройства поддерживающего использование веб-браузера. В качестве места выполнения выступает офисное помещение с рабочим местом, включающее в себя стол со стулом и персональный компьютер с клавиатурой и мышью. Осуществляемые рабочие процессы – разработка и тестирование серверного веб-приложения.

Вредными факторами были определены:

- повышенный уровень шума на рабочем месте;
- показатели микроклимата воздушной среды на местонахождении работающего: температурой и относительной влажностью воздуха;
- отсутствие или недостаток необходимого искусственного освещения;
- монотонность труда, вызывающая монотонию.

К опасным факторам относятся производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которых попадает работающий.

4.1 Правовые и организационные вопросы обеспечения безопасности

4.1.1 Правовые нормы трудового законодательства

При организации рабочего места с персональным компьютером обязательно должны учитываться требования безопасности, эргономики, технической эстетики и промышленных санитарных норм.

В трудовом кодексе РФ 197-ФЗ [11] дается следующая характеристика организации труда в течении смены:

- продолжительность рабочего времени не должна превышать 40 часов в неделю;
- длительность рабочей смены не должна превышать 8 часов;
- установка обеденного перерыва, для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается.

Продолжительность непрерывной работы за компьютерным устройством, без регламентированного перерыва, не должна превышать 2 часов. Длительность регламентированных перерывов составляет 20 минут (после 1,5 – 2,0 часа от начала рабочей смены и обеденного перерыва).

Во время регламентированных перерывов целесообразно выполнять комплексы упражнений и осуществлять проветривание помещения.

4.1.2 Эргономические требования к правильному расположению и компоновке рабочей зоны

В соответствии с ГОСТ 12.2.032-78. «Система стандартов безопасности труда». Рабочее место при выполнении работ сидя» рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы [12].

Выполнение требований на данном рабочем месте отражено ниже в таблице 20, согласно ГОСТ 12.2.032-78.

Таблица 20 – Требования к организации рабочего места при работе с ПЭВМ

Требование	Требуемое значение	Значение параметров в помещении
Высота рабочей поверхности стола	Регулируемая высота (680-800мм) Нерегулируемая высота (725мм)	Нерегулируемая высота (750 мм)
Рабочий стул	Подъемно-поворотный, регулируемый по высоте и углу наклона спинки	Не соответствует
Расположение монитора от глаз пользователя	600-700мм	Соответствует

На момент разработки выпускной квалификационной работы в качестве рабочего стула выступает стул с мягкой обивкой. Для соответствия требованиям нормативных актов по организации рабочего места компании следует приобрести рабочий стул, который будет обеспечивать функции регулировки по высоте и углу наклона спинки стула.

4.2 Производственная безопасность

Производственные факторы согласно ГОСТ 12.0.003-2015 подразделяются на опасные и вредные. Опасным производственным фактором называется фактор, воздействие которого приводит к травме или резкому ухудшению здоровья. Вредным производственным фактором является фактор, воздействие которого приводит к заболеванию или снижению работоспособности. Возможные опасные и вредные факторы представлены в таблице 21.

Таблица 21 – Опасные и вредные производственные факторы на рабочем месте с ПК

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Вредные факторы	
Повышенный уровень шума на рабочем месте.	ГОСТ 12.1.003-2014. Система стандартов безопасности труда. Шум. Общие требования безопасности.
Показатели микроклимата воздушной среды на местонахождении работающего: температурой и относительной влажностью воздуха	ГОСТ 30494-2011 Здания жилые и общественные. Параметры микроклимата в помещениях.
Отсутствие или недостаток необходимого искусственного освещения;	СП 52.13330.2016. Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий.
Монотонность труда, вызывающая монотонию	ГОСТ Р ИСО 10075-1-2019 «Эргономические принципы обеспечения адекватности умственной нагрузки»
Опасные факторы	
Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которых попадает работающий.	ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов

4.2.1 Повышенный уровень шума на рабочем месте

Источниками шума, на рабочем месте разработчика, являются принтеры, сканеры, вентиляторы, системы охлаждения. Повышенный уровень шума затрудняет разборчивость речи, снижает работоспособность, повышает утомляемость. Длительное воздействие интенсивного шума на слух человека приводит к его частичной или полной потере. Уровень шума на рабочих местах разработчика не должен превышать значений, установленных ГОСТ 12.1.003-2014 и составлять не более 60 дБА [13].

Для защиты от шума используют звукоизоляцию помещений. К индивидуальным средствам защиты от шума относятся специальные звукоизолирующие устройства.

Проведя исследования уровня шума в помещении было получено среднее значение в 48 дБ, что является ниже допустимого уровня шума в 60 дБ, соответственно внедрение систем защиты от шума является нецелесообразной.

4.2.2 Показатели микроклимата воздушной среды на местонахождении работающего: температура и относительная влажность воздуха

Использование персональных компьютеров может привести к повышению температуры и снижению относительной влажности в рабочем помещении, то есть к изменению микроклимата. Отклонение показателей микроклимата ведет к возникновению общих или локальных ощущений теплового дискомфорта, напряжению механизмов терморегуляции, ухудшению самочувствия и понижению работоспособности человека.

Нормативные показатели микроклимата регламентируются ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны».

В таблице 22 приведены оптимальные величины показателей микроклимата на рабочих местах для оператора ЭВМ для категории тяжести работ 1а.

Таблица 22 – Оптимальные параметры микроклимата на рабочем месте

Период года	Температура воздуха, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	22-24	40-60	0,1
Тёплый	23-25	40-60	0,2

Для поддержания оптимальных значений микроклимата используется система отопления и кондиционирования воздуха. Для повышения влажности воздуха в помещении следует применять увлажнители воздуха

4.2.3 Отсутствие или недостаток необходимого искусственного освещения

Недостаточная освещенность приводит к понижению работоспособности, а также может вызвать проблемы со здоровьем, а именно может повлиять на качество зрения работника.

Для комфортной работы сотрудника необходимо отсутствие пульсации света, обеспечение достаточной контрастности в цветопередаче монитора, отсутствие бликов на поверхностях офисного оборудования, а также соответствующее направление светового потока и его спектр.

Согласно СП 52.13330.2016 [14] зрительную работу инженера-программиста можно характеризовать как работу разряда Б – высокой точности, потому необходимо чтобы параметры освещенности рабочего места соответствовали требованиям, представленным в таблице 23.

Таблица 23 – Требования к освещению производственных помещений при зрительной работе высокой точности

Искусственное освещение			
Освещенность на рабочей поверхности при системе общего освещения	Освещенность на рабочей поверхности при системе комбинированного освещения	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, Кп, %, не более
400	1000	25	15

Для снижения влияния фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего места и яркость дисплея персонального компьютера были приблизительно одинаковыми. При недостаточной освещенности помещения может помочь увеличение количества световых приборов.

4.2.4 Монотонность труда, вызывающая монотонию

Длительный процесс разработки программного обеспечения является монотонным трудом, способным вызвать монотонию.

Согласно ГОСТ Р ИСО 10075-1-2019 «Эргономические принципы обеспечения адекватности умственной нагрузки» [15] монотония – медленно развивающееся состояние пониженной активизации, возникающее при длительном выполнении однообразных, повторяющихся заданий, в основном проявляющееся в виде сонливости, утомления, снижения или колебания работоспособности, снижения адаптируемости и восприимчивости, а также сопровождающееся повышением изменчивости частоты сердечных сокращений.

Для уменьшения монотонности труда необходимо осуществлять следующие процедуры:

- ротация рабочих мест;
- повышение разнообразия работы;
- добавление в задачи познавательных элементов для достижения их разнообразия;
- расширение поля внимания, например, с помощью более сложных задач;
- обеспечение возможностей для изменения задач;
- обеспечение возможностей для физической активности;
- соответствующее проектирование климатических условий;
- уменьшение шума и однородного акустического воздействия;

- обеспечение соответствующего освещения;
- обеспечение доступа к общению с коллегами по работе;
- введение перерывов для отдыха.

4.2.5 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которых попадает работающий

Персональный компьютер предоставляет для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться комплектующих компьютера, находящихся под напряжением. При работе с компьютером возможно поражение электрическим током, что ведет к появлению ожогов, нагреву сосудов, механическим повреждениям тканей и сосудов, раздражающим воздействиям на ткани.

Общие требования по электробезопасности представлены в ГОСТ 12.1.038-82 ССБТ.

Для того, чтобы защититься от поражения электрическим током, необходимо:

- обеспечить недоступность токоведущих частей от случайных прикосновений;
- электрическое разделение цепи;
- устранить опасности поражения при проявлении напряжения на разных частях.

Помещение, где разрабатывалось серверное приложение принадлежит к категории помещений без повышенной опасности по степени вероятности поражения электрическим током.

4.3 Экологическая безопасность

Процесс разработки серверного приложения не оказывает влияния на окружающую среду, так как разрабатывается и используется внутри ЭВМ. Соответственно, загрязнение селитебной зоны, гидросферы и атмосферы не происходит. Однако при разработке приложения источниками загрязнения литосферы могут стать неправильно утилизированные печатные бумажные материалы, расходные части печатающих устройств, вышедшие из строя комплектующие ПК, периферийные устройства и их комплектующие.

Утилизация компьютерной и организационной техники ограничена законодательно, так как в производстве такой техники используется большое количество материалов, способных нанести большой вред окружающей среде. Главными нормативными актами, регулирующими вопрос утилизации ноутбуков, являются федеральные законы РФ «Об охране окружающей среды» и «Об отходах производства и потребления». Утилизация компьютерного оборудования происходит через обязательное извлечение компонентов, их сортировку и последующую отправку для повторного использования. Такая утилизация происходит обязательно с привлечением квалифицированного персонала.

Утилизация мусорных отходов, таких как бумажная макулатура, отходы от канцелярских принадлежностей производится через сбор, сортировку и утилизацию. Отходы, которые можно использовать повторно после сортировки отправляют на переработку специализированным компаниям.

4.4 Безопасность в чрезвычайных ситуациях

Возможными чрезвычайными ситуациями могут быть:

- техногенные (пожар, внезапное обрушение здания, аварии на коммунальных системах жизнеобеспечения населения);
- биологические (эпидемия, пандемия и т.п.);

- социальные (терроризм, войны).

Наиболее вероятная чрезвычайная ситуация – это пожар. Распространенными причинами возникновения пожара в помещениях с ПК являются:

- короткие замыкания;
- использование неисправного электрооборудования;
- курение в неположенных местах;
- неправильное обращение с оборудованием и др.

Чтобы не допустить возникновения данной чрезвычайной ситуации, необходима периодическая диагностика по обнаружению неисправностей, а также соблюдение персоналом норм пожарной безопасности.

В случае возникновения пожара работникам предприятия необходимо выполнить следующие действия:

- немедленно сообщить по телефону о пожаре в пожарную охрану;
- оповестить о пожаре всех работников;
- принять меры по эвакуации людей, тушению пожара,

сохранению материальных ценностей.

Вывод по разделу

В результате работы по разделу «Социальная ответственность» были выявлены основные нормативные акты для обеспечения безопасности жизнедеятельности на рабочем месте. На основании нормативных документов были установлены необходимые параметры освещения, микроклимата, уровня шума на рабочем месте. Соблюдение данных параметров, позволит сохранить хорошую работоспособность в течение всего рабочего дня и повысить продуктивность работы.

Рабочее место, использованное при разработке веб-приложения, удовлетворяет всем требованиям безопасности, правилам и нормам,

необходимым работнику категории 1а. Требуемое освещение обеспечивается за счет нескольких энергосберегающих ламп. Уровень шума находится в допустимом диапазоне. Микроклиматические условия соблюдаются за счет системы отопления в холодное время и проветривания в теплое время. Защита от воздействия электрического тока обеспечивается путем проверки состояния ПК и соблюдения правил безопасности при работе с ним, соответственно, согласно правилам устройства электроустановок [16], данное помещение входит в категорию безопасных. Само помещение относится к категории В по пожарной опасности, однако имеет все необходимые компоненты для обеспечения безопасности.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была выполнена разработка программных интерфейсов серверного приложения веб-платформы «Агенты будущего». Каждый метод был проверен на локальном сервере, после чего загружен на удаленный тестовый сервер и протестирован там, что позволяет использовать их для разработчиков клиентской части платформы. Для более удобного понимания принципов работы для каждого интерфейса была составлена интерактивная документация с возможностью проведения тестов.

Система разрабатывалась в команде, что дало дополнительный опыт работы в команде при разработке системы. Личным вкладом автора работы являются все проектные решения, связанные с серверным приложением, начиная с разработки требований, заканчивая полной работоспособностью на удаленном сервере.

В процессе разработки был выполнен следующий перечень задач:

Составлен список функциональных требований, предъявляемых к разрабатываемым компонентам, который позволили обобщить и сгруппировать видение проектных решений.

Выполнено планирование архитектуры компонентов системы, которая позволила внедрить решение в уже существующий проект.

Реализована разработка интерфейсов для регистрации, авторизации и аутентификации пользователей необходимая для защиты от несанкционированного доступа и кибератак.

Выполнена разработка интерфейсов для работы с пользователями, которые позволили выполнять операции с профилем необходимые для функционирования платформы.

Реализованы интерфейсы для работы с образовательными треками с помощью которых можно управлять заданиями на платформе.

Разработаны интерфейсы для работы с новостями, которые позволяют выводить актуальные новости на главной странице платформы, а также управлять ими из панели администрирования.

Реализованы интерфейсы для работы с миссиями, которые сохраняют и предоставляют всю информацию о прогрессе пользователей при их выполнении.

Все реализованные компоненты были протестированы на предмет работы с запросами разных видов, найденные ошибки были устранены.

Для каждого запроса была разработана документации для удобства и понимания принципов их работы всей командой разработчиков.

Полученное решение было внедрено на удаленный сервер, где также успешно прошло все тесты и показало свою готовность к работе в реальных условиях платформы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архитектура Laravel приложения// Zanin [Электронный ресурс]. – Режим доступа: <https://zanin.online/laravel-8/laravel-8-solid/> Дата обращения: 08.03.2022.
2. OAuth 2.0 простым и понятным языком // Habr.com [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/vk/blog/115163/> Дата обращения: 08.05.2022.
3. Документация Laravel Passport на русском языке// laravel.ru [Электронный ресурс]. – Режим доступа: <https://laravel.ru/docs/8.x/passport>
4. Git-репозиторий // gitlab.tpu.ru [Электронный ресурс]. – Режим доступа: <https://gitlab.tpu.ru/agentsoffuture/agents-of-the-future>
5. Postman documentation // postman.com [Электронный ресурс]. – Режим доступа: <https://learning.postman.com/docs>. Дата обращения: 10.03.2022.
6. Курс по документированию API // Learnapidoc-ru [Электронный ресурс]. – Режим доступа: <https://starkovden.github.io/index.html>. Дата обращения: 12.03.2022.
7. DarkaOnLine/L5-Swagger // github.com [Электронный ресурс]. – Режим доступа: <https://github.com/DarkaOnLine/L5-Swagger> Дата обращения: 14.03.2022.
8. Swagger documentation // Swagger [Электронный ресурс]. – Режим доступа: <https://swagger.io/docs/>. Дата обращения: 18.03.2022.
9. Ubuntu documentation // Canonical.com [Электронный ресурс]. – Режим доступа: <https://ubuntu.com/server/docs>. Дата обращения 21.03.2022.
10. GitLab documentation // gitlab.com [Электронный ресурс] – Режим доступа – <https://docs.gitlab.com/ee/user/>. Дата обращения 01.04.2022.
11. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) (с изм. и доп., вступ. в силу с 01.03.2022).
12. ГОСТ 12.2.032-78 «Рабочее место при выполнении работ сидя. Общие эргономические требования».

13. ГОСТ 12.1.003-2014. «Система стандартов безопасности труда. Шум. Общие требования безопасности».
14. СП 52.13330.2016. «Естественное и искусственное освещение».
15. ГОСТ Р ИСО 10075-1-2019 «Эргономические принципы обеспечения адекватности умственной нагрузки».
16. Правила устройства электроустановок. Седьмое издание.

Приложение А. Методы регистрации, авторизации и аутентификации

ПОЛЬЗОВАТЕЛЯ

Метод login:

```
public function login(LoginUserRequest $request)
{
    try {
        //сделать апдейт по времени последнего пребывания
        if (Auth::attempt($request->validated())) {
            $token = Auth::user()->createToken('appToken')->plainTextToken;
            $user = User::where('email', $request['email'])->first();
            $user->lastOnline = Carbon::now();
            $user->save();
            return response()->json(['token' => $token , 'token_type' =>
'Bearer', 'success' => true], 201);
        }
        return response()->json(['message' => 'Incorrect username or
password', 'success' => false], 401);
    }
    catch (\Exception $e)
    {
        Log::error($e->getMessage(), ['trace' => $e->getTraceAsString()]);
        return response()->json(['message' => $e->getMessage(), 'success' =>
false], 500);
    }
}
```

Метод logout:

```
public function logout()
{
    try{
        $user = Auth::user();
        if(!$user){
            return response([
                'message' => 'Something went wrong'
            ], 401);
        }
        $user->tokens()->delete();
        return response([
            'message'=>'Logged out',
            'success' => true
        ]);
    }
    catch (\Exception $e)
    {
        Log::error($e->getMessage(), ['trace' => $e->getTraceAsString()]);
        return response()->json(['message' => $e->getMessage(), 'success' =>
false], 500);
    }
}
```

Метод register:

```
public function register(RegisterUserRequest $request): JsonResponse
{
    try {
        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
```

```
        'password' => Hash::make($request->password)
    });
    $token = $user->createToken('appToken')->plainTextToken;
    event(new Registered($user));
    return response()->json(['token' => $token, 'token_type' => 'Bearer',
'success' => true], 201);
    } catch (\Exception $e) {
        Log::error($e->getMessage(), ['trace' => $e->getTraceAsString()]);
        return response()->json(['message' => $e->getMessage(), 'success' =>
false], 500);
    }
}
```