

## Введение

В современном мире технологии играют ключевую роль в различных аспектах жизни, включая образование. Частные школы программирования и робототехники для детей становятся все более популярными, предоставляя уникальные возможности для развития навыков, необходимых в цифровую эпоху. Проект разработки веб-сайта для частной школы программирования и робототехники имеет целью автоматизировать ключевые процессы школы, улучшить взаимодействие между администрацией, преподавателями, родителями и учениками, а также повысить общую эффективность управления образовательным учреждением.

Основной задачей проекта является создание платформы, которая позволит родителям записывать своих детей на оффлайн курсы, отслеживать занятия и управлять своими профилями. Сайт должен быть интуитивно понятным и удобным в использовании, обеспечивая простой доступ к информации о курсах. Преподаватели должны иметь возможность изменять и удалять материалы курсов.

Одним из основных преимуществ предлагаемого решения является автоматизация процессов, связанных с регистрацией на курсы и управлением занятиями. Это даст начало к автоматизации бизнес-процессов частной школы. Кроме того, платформа обеспечит прозрачность и доступность информации, что повысит уровень доверия и удовлетворенности клиентов.

Разработка данного проекта включает создание базы данных для хранения информации о пользователях, курсах и занятиях, а также интеграцию системы аутентификации и авторизации для обеспечения безопасности данных. Использование современных веб-технологий и фреймворков позволит создать надежную и масштабируемую платформу, способную адаптироваться к изменяющимся потребностям школы.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	
Изм.	Лист	№ докум.	Подпись	Дат		3

## 1 Сбор, анализ и формирование требований к программному продукту

Процесс разработки программного обеспечения начинается с этапа сбора, анализа и формирования требований. Это фундаментальный этап, который определяет успех всего проекта. В рамках создания веб-сайта для частной школы программирования и робототехники для детей, необходимо учитывать потребности всех участников образовательного процесса: администраторов, преподавателей, родителей и учеников.

Первым шагом является сбор требований. На данном этапе проводится серия интервью с заказчиком. Также проводится анализ текущих процессов управления и взаимодействия в школе, чтобы выявить слабые места и области, требующие автоматизации. В результате формируется перечень функциональных и нефункциональных требований, которые должны быть реализованы в рамках проекта.

Функциональные требования включают в себя: возможность регистрации и авторизации пользователей, управление профилями, запись на курсы, отслеживание расписания, управление учебными материалами преподавателями, и система уведомлений для всех участников. Также необходимо обеспечить удобный интерфейс для взаимодействия и минимизировать количество шагов, необходимых для выполнения каждой задачи.

Нефункциональные требования включают в себя: безопасность данных, производительность системы, масштабируемость, удобство использования и совместимость с различными устройствами и браузерами. Особое внимание уделяется защите персональных данных пользователей, так как система будет обрабатывать чувствительную информацию о детях.

Следующим этапом является анализ требований. Все собранные данные структурируются и анализируются на предмет полноты, точности и соответствия реальным потребностям школы. На этом этапе выявляются возможные противоречия и пробелы, которые устраняются путем проведения дополнительных консультаций с заинтересованными сторонами.

Формирование требований завершается созданием детализированного технического задания, которое включает в себя описание всех функциональных модулей системы, интерфейсов, архитектурных решений и технологий, которые будут использованы. Техническое задание становится основой для последующих этапов разработки, обеспечивая ясность и согласованность действий команды разработчиков.

### 1.1 Сбор требований к программному продукту

Сбор требований к программному продукту начался с проведения интервью с заказчиком в лице директора школы. Цель интервью состояла в выявлении основных потребностей и проблем, с которыми сталкиваются

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

участники образовательного процесса. Ниже представлена таблица с вопросами и ответами, полученными в ходе интервью. По результатам опроса([https://docs.google.com/document/d/16uKpTe7sZleo3\\_E5xhgMs7O4reoem5\\_6/edit](https://docs.google.com/document/d/16uKpTe7sZleo3_E5xhgMs7O4reoem5_6/edit)) были выявлены следующие требования, представленные в таблице 1:

Таблица 1 – Интервью с заказчиком

Общие вопросы	
Вопрос	Ответ
Какова миссия и основные цели вашей школы программирования?	Миссия нашей школы - предоставить качественное и доступное образование в области программирования для учеников нашей школы. Основные цели включают развитие навыков критического мышления и решения сложных задач с помощью технологий
Какие основные проблемы вы хотите решить с помощью этой платформы?	Нашей главной задачей является автоматизация учебного и бизнес-процессов школы, дать возможность онлайн записи на курс и занятия, следить за расписанием, повысить взаимодействия между родителями, учениками и преподавателям
Какие функциональные возможности вы считаете приоритетными для первой версии продукта?	Для первой версии продукта приоритетными являются регистрация и авторизация пользователей, подача заявок на курсы, форма обратной связи, возможность следить за расписанием занятий
Есть ли у вас предпочтения по технологическому стеку для разработки платформы?	Мы открыты к предложениям
Каков ваш бюджет на разработку платформы?	Наш бюджет ограничен. Важно найти баланс между качеством и стоимостью

Продолжение таблицы 1

Вопрос	Ответ
Каковы сроки проекта? Когда вы хотите запустить платформу?	Наша цель - запустить первую версию платформы в течение следующих 5 месяцев. Мы понимаем, что сроки могут корректироваться в зависимости от сложности проекта, но для нас важно начать эксплуатацию как можно скорее
Есть ли у вас существующие решения, которые вы хотели бы интегрировать с новой платформой?	Нет
Какие языки и регионы должна поддерживать платформа?	Платформа должна изначально поддерживать русский язык
Роли и права доступа	
Вопрос	Ответ
Какие роли пользователей вы хотите внедрить на платформе?	Нам нужны роли для учеников, преподавателей, родителей и администратора
Какие действия должен иметь право выполнять пользователь каждой роли?	Ученик: просмотр расписания в личном кабинете. Преподаватель: управление материалами курса и личным кабинетом. Родитель: запись ребенка на курс, отслеживание расписания, управление личным кабинетом. Администратор: управление заявками, пользователями, курсами, доступом и настройками платформы. Все: заполнение формы обратной связи
Нужна ли система управления правами доступа с возможностью настройки на уровне администратора?	Да

Продолжение таблицы 1

Функционал ролей	
Вопрос	Ответ
Нужна ли интеграция с внешними ресурсами или платформами?	Пока не рассматриваем данные возможности. Готовы к предложениям
Какие элементы геймификации (например, баллы, награды) вы хотели бы включить?	Пока не заинтересованы в создании онлайн обучении и оценивании учеников онлайн, но готовы к предложениям.
Какие инструменты для создания и редактирования учебных материалов вам нужны?	Нам необходимы инструменты для создания ссылок на материалы, текстовых материалов и добавления видео, а также возможность легко обновлять и модифицировать существующие курсы и материалы к ним
Нужен ли инструмент для организации онлайн-занятий через платформу?	Нет
Какие уведомления или отчеты должны отправляться родителям?	Родители видят статус заявки по курсу, по возможности можем обговорить об интеграции системы оповещения
Какие инструменты для управления пользователями вам нужны?	Нам нужны инструменты для создания и управления профилями пользователей, назначения ролей
Нужны ли инструменты для управления расписанием занятий?	Да, необходимы инструменты для создания, редактирования и публикации расписаний занятий, а также для уведомления пользователей о любых изменениях в расписании. В идеале реализовать календарь

Продолжение таблицы 1

Интерфейс и удобство использования	
Вопрос	Ответ
Какие основные принципы дизайна должны быть реализованы в интерфейсе?	Мы хотим, чтобы интерфейс был интуитивно понятным и пользовательский опыт был максимально удобным
Есть ли у вас предпочтения по цветовой схеме или фирменному стилю?	Предпочтений нет, на ваше усмотрение
Какие устройства должны поддерживаться (мобильные, планшеты, ПК)?	Желательно, чтобы доступ к платформе был доступен с мобильного телефона, планшета и ПК. Важно, чтобы пользователи могли легко пользоваться платформой независимо от того, где они находятся
Нужна ли адаптивная версия сайта или мобильное приложение?	Да, нам необходима адаптивная версия сайта для обеспечения удобства использования на мобильных устройствах. Также мы заинтересованы в разработке мобильного приложения в будущем для предоставления более глубокой интеграции с устройствами пользователей
Безопасность и конфиденциальность	
Вопрос	Ответ
Какие меры безопасности должны быть реализованы для защиты данных пользователей?	Нам нужна защита данных пользователей от несанкционированного доступа
Какие требования к конфиденциальности данных пользователей у вас есть?	Конфиденциальность данных учеников и преподавателей важна. Мы ожидаем полное соответствие законодательным требованиям о защите данных, включая возможность пользователей управлять своими данными и их удалением

Продолжение таблицы 1

Нужна ли интеграция с системами идентификации пользователей?	Интеграция с системами идентификации пользователей, для обеспечения безопасного и удобного входа через социальные сети будет плюсом для нашей платформы, но не обязательный критерий
Техническое обслуживание	
Вопрос	Ответ
Какие требования к хостингу и инфраструктуре у вас есть?	Мы ищем надежное решение для хостинга
Нужна ли вам поддержка и обслуживание после запуска платформы?	Скорее всего нам потребуется редкая поддержка платформы
Как часто вы планируете обновлять контент и функционал платформы?	Мы планируем обновлять учебные материалы раз в квартал
Какие каналы поддержки пользователей вы хотели бы иметь?	Мы хотели бы предоставить пользователям несколько каналов поддержки: телефон поддержки, электронную почту или возможность обращения через форму обратной связи на сайте
Обратная связь и улучшения	
Вопрос	Ответ
Как вы планируете собирать обратную связь от пользователей?	Мы планируем использовать форму обратной связи непосредственно на платформе, а также анализировать отзывы через социальные сети и форумы.

## Продолжение таблицы 1

Есть ли у вас предложения по функциям и возможностям для будущих обновлений?	Нам интересно создание платежной системы при покупке занятий, возможность автоматизации расписания системой. Создание наград и бонусов ученикам после прохождения занятий. Создание системы, уведомляющей об изменениях в расписании, а также нам интересна дальнейшая разработка мобильного приложения
--	---

### 1.2 Анализ и формирование требований

На основе собранных требований от заказчика частной школы Start Junior, проведем анализ и структуризацию информации, чтобы выделить наиболее важные и необходимые требования для разработки платформы. Цель этого этапа — предоставить четкий и ограниченный список требований, который будет основой для дальнейшей разработки.

Основная цель проекта: автоматизация учебных процессов, онлайн-запись на курсы.

Приоритетные функциональные возможности:

- регистрация и авторизация пользователей;
- возможность отправления заявки на курс;
- возможность отслеживать расписание занятий;
- управление преподавателями материалами курсов;
- обратная связь на почту администратора.

Технические требования:

- открытость к предложениям по технологическому стеку;
- поддержка русского языка;
- адаптивная версия сайта.

Действия по ролям:

- общий функционал и ученик: авторизация, форма обратной связи, отслеживание расписания;
- преподаватель: управление материалами курсов и личным кабинетом;
- родитель: регистрация, отправление заявки на курс и управление личным кабинетом;
- администратор: управление заявками, пользователями, курсами, настройками платформы.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		



Интерфейс и удобство использования:

– принципы дизайна: интуитивно понятный интерфейс, удобство использования;

– поддержка мобильных устройств, планшетов и ПК;

– адаптивная версия сайта.

Безопасность и конфиденциальность:

– меры безопасности: защита данных пользователей.

– соответствие законодательным требованиям о защите данных.

Техническое обслуживание:

– надежное решение для хостинга;

– потребность в редкой поддержке после запуска платформы;

– каналы поддержки: телефон, электронная почта, форма обратной связи.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 Проектирование и разработка архитектуры программного продукта

### 2.1 Построение диаграммы связей

Для наглядного представления структуры сайта была создана диаграмма связей, которая иллюстрирует графическую карту разрабатываемых продуктов. Этот визуальный инструмент представлен на рисунках 1 и 2.

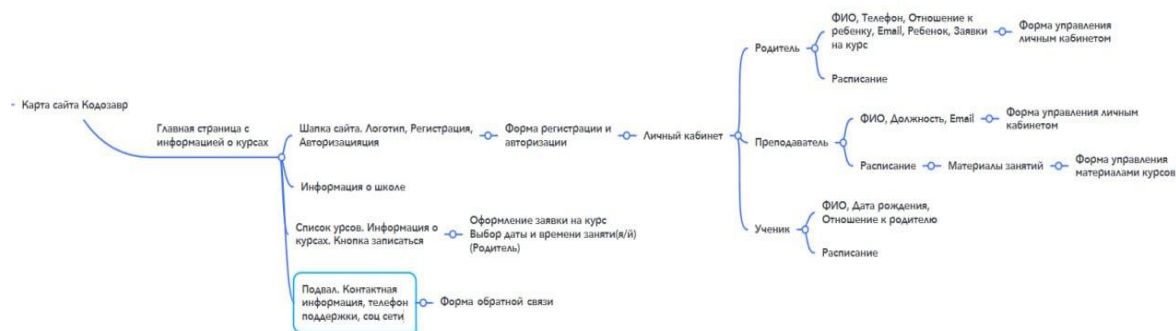


Рисунок 1 – Диаграмма связей клиентской части проекта



Рисунок 2 – Диаграмма связей административной панели

### 2.2 Разработка сценария использования

Для эффективного проектирования и реализации программного продукта необходимо разработать сценарии использования, которые отражают

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

взаимодействие различных ролей пользователей с системой. На представленной диаграмме сценариев использования (Use Case Diagram) показаны основные функции и действия, доступные для учителей, родителей, учеников и администратора. Это позволяет оптимизировать процесс автоматизации школьной деятельности и улучшить пользовательский опыт.

В системе присутствует три типа пользователей:

- администратор;
- родитель;
- ученик;
- учитель.

Каждый из типов пользователей имеет определенный функционал, представленный на рисунке 3.

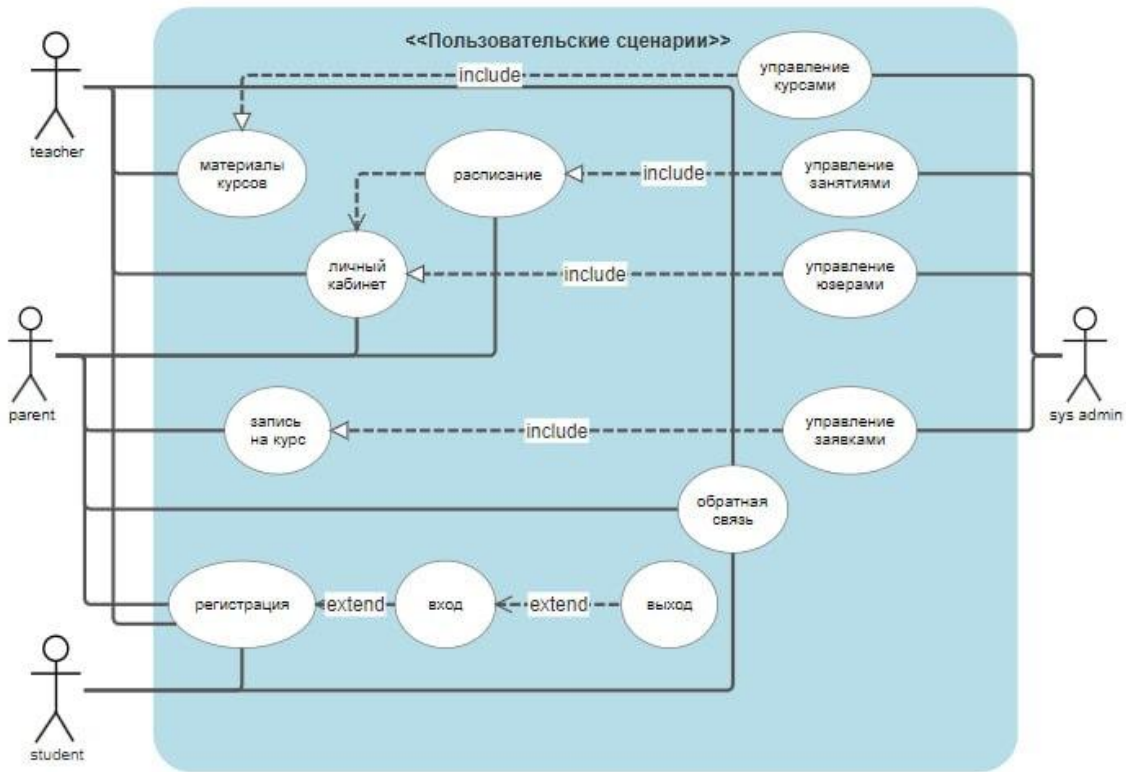


Рисунок 3 – Use Case диаграмма

### 2.3 Прототипирование и дизайн программного продукта

Для разработки продукта прототипирование имеет ключевое значение. Оно позволяет команде получить обратную связь от потенциальных пользователей на ранних этапах разработки, что помогает точнее определить потребности и предпочтения аудитории. Кроме того, прототипирование помогает проверить соответствие созданного программного продукта

спецификациям и требованиям, чтобы убедиться, что разработка идет в правильном направлении. Этот процесс также дает возможность оценить предварительную производительность продукта, что помогает оптимизировать планы разработки и распределение ресурсов. Прототипирование сайта было выполнено в приложении Figma, потому что это мощный и удобный инструмент для создания прототипа и дизайна. Простота использования и множество доступных шаблонов и плагинов позволяют ускорить процесс создания прототипов и дизайна, что особенно важно в условиях ограниченных сроков.

На рисунках 4, 5 и 6 представлен прототип главной страницы, остальной дизайн страниц показан в приложение В.

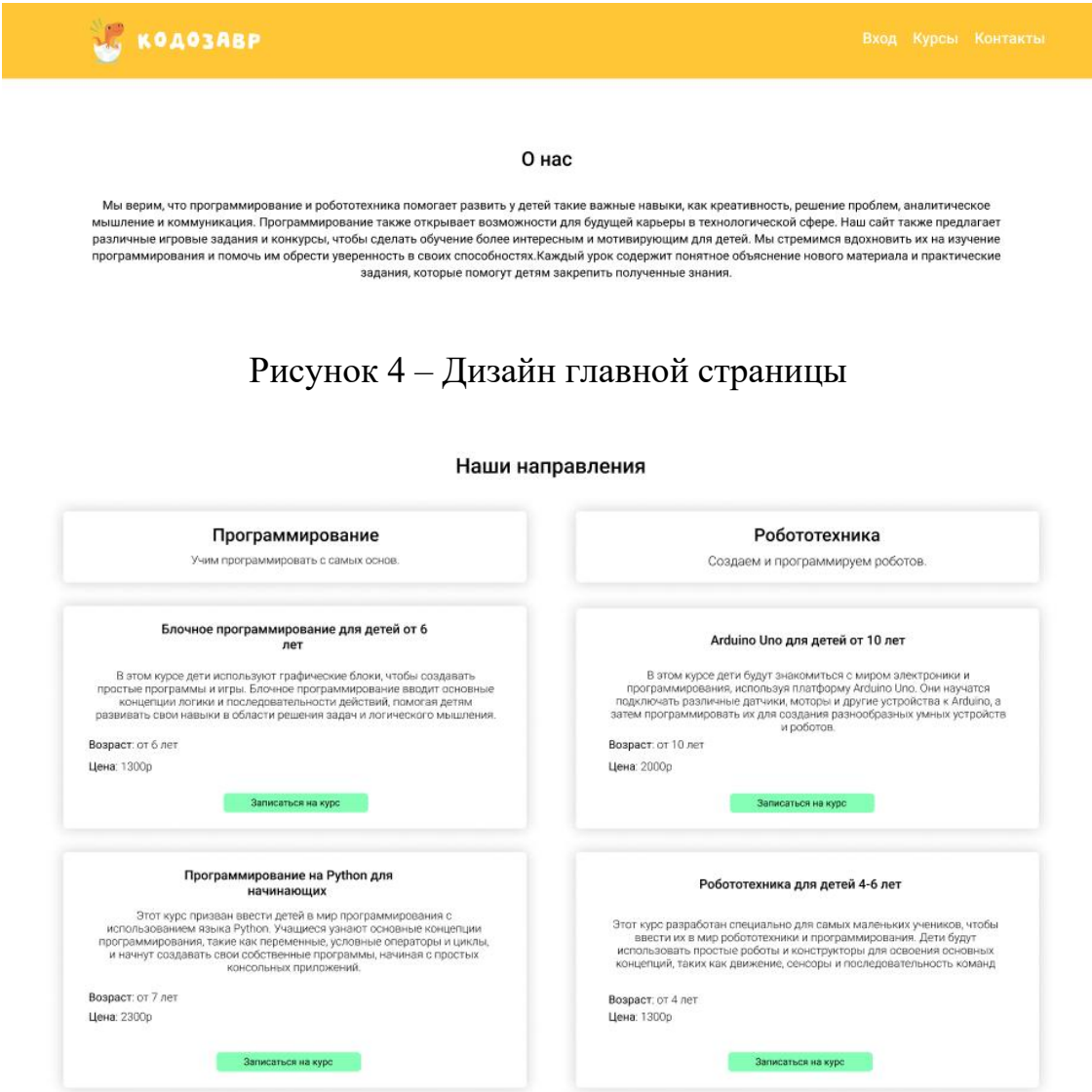
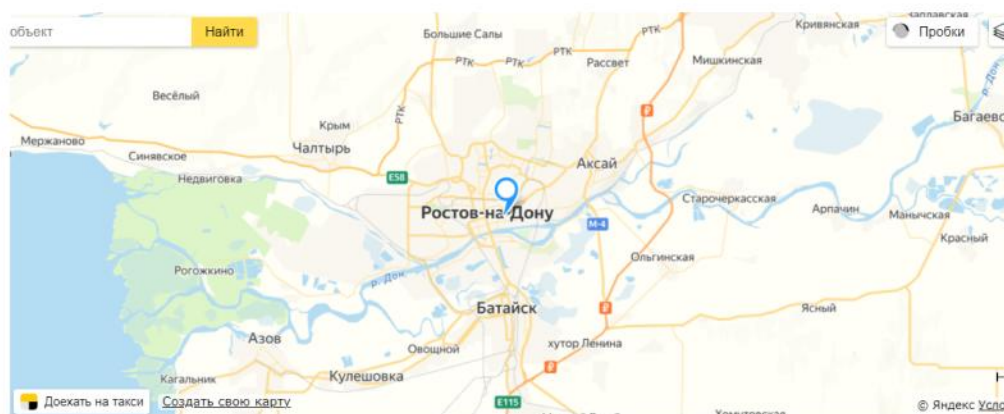


Рисунок 5 – Дизайн главной страницы



Vkontakte YouTube Instagram  
Start Junior  
Телефон: +7 (123) 456-78-90  
Электронная почта: startjunior56@gmail.com



Рисунок 6 – Дизайн главной страницы

## 2.4 Архитектура программного продукта

Архитектура программного продукта, разработанного для частной школы программирования и робототехники, основывается на модели MVT (Model-View-Template), которая является основой фреймворка Django. Этот подход обеспечивает четкое разделение между представлением данных, бизнес-логикой и шаблонами, что упрощает разработку и сопровождение кода. Архитектура представлена на рисунке 7.

**Models:** Модели представляют собой структуру данных и бизнес-логику приложения. В нашем проекте они включают в себя такие сущности, как пользователь (CustomUser), родитель, студент, преподаватель, курс, заявка на курс, обратная связь, тема и урок. Каждая модель описывает структуру таблицы базы данных и методы для работы с данными, что позволяет эффективно управлять информацией и поддерживать целостность данных.

**Views:** Представления отвечают за обработку запросов, выполнение бизнес-логики и взаимодействие с моделями для получения необходимых данных. В нашем проекте используются как классовые представления, так и функциональные, обеспечивающие выполнение различных задач, таких как регистрация пользователей, управление курсами, обработка заявок и предоставление расписания. Представления возвращают ответы в виде HTML-страниц или JSON-данных, которые затем отображаются пользователю.

**Templates:** Шаблоны отвечают за отображение данных, полученных от представлений, в удобной для пользователя форме. В проекте используются шаблоны Django для генерации HTML-страниц, включая формы регистрации,

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

страницы профиля, расписание занятий и другие интерфейсы. Шаблоны позволяют отделить логику представления от логики обработки данных, что делает код более чистым и поддерживаемым.

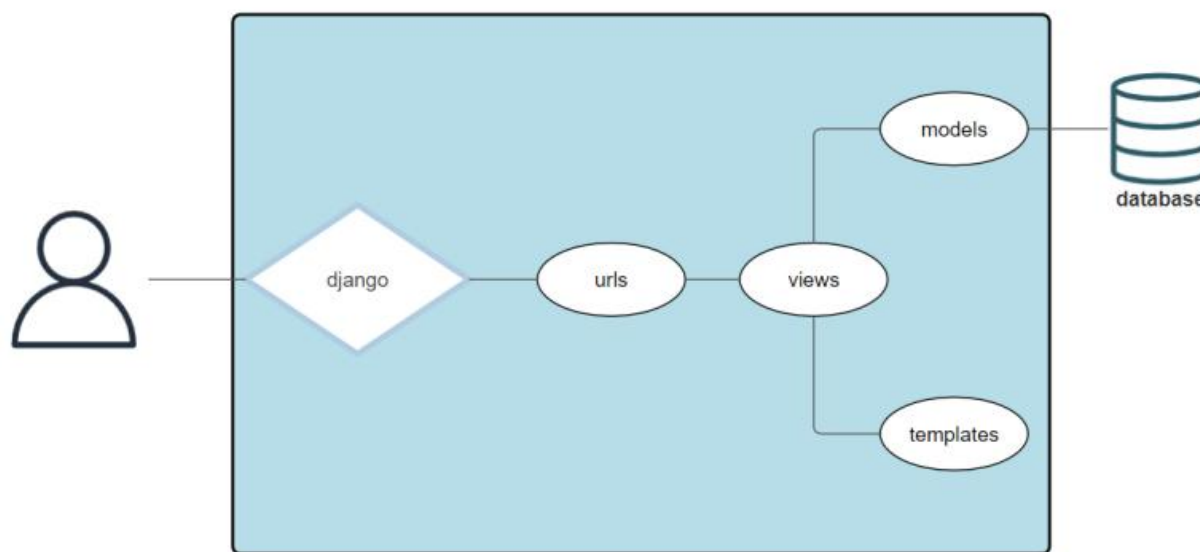


Рисунок 7 – Архитектура MVT

Следующим этапом необходимо спроектировать модель реляционной базы данных. Были сформированы 7 таблиц с атрибутами.

Для целостности БД необходимо связать между собой объекты:

- у родителя может быть несколько детей;
- родитель может оставлять несколько заявок на курс;
- на один курс можно оформить несколько заявок;
- курс имеет несколько тем;
- по одной теме можно проводить несколько уроков;
- учитель проводит несколько уроков.

После формирования логической модели начинаем подвергать её нормализации как со стороны объектов/сущностей, так и со стороны связей. А также стоит посмотреть связи между таблицами на предмет их оптимизации и устранения потенциальных аномалий.

При формировании окончательной ER диаграммы базы данных, расположенной на рисунке 8, было использовано CASE-средство - lucid. Оно просто в использовании и имеет большой функционал

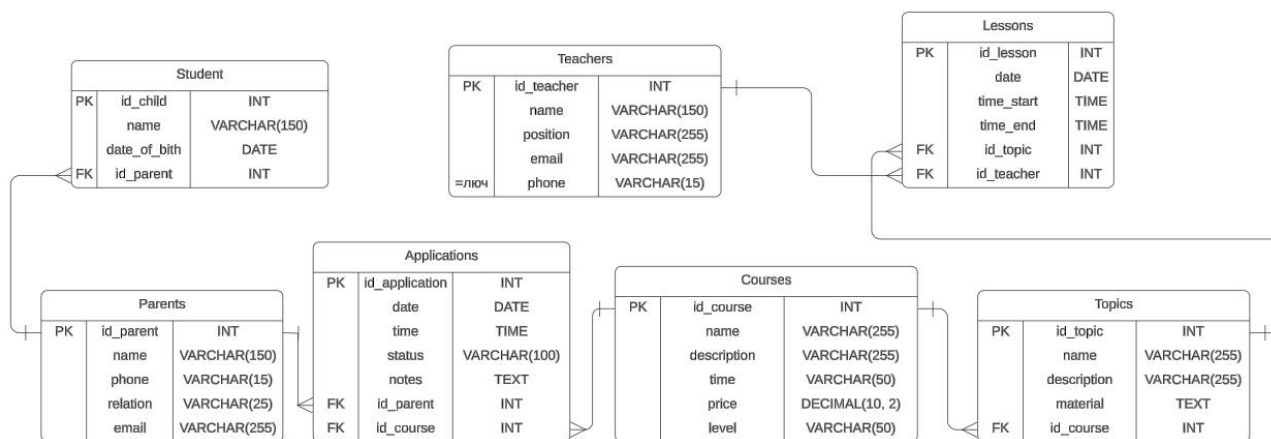


Рисунок 8 – Рисунок ERD codozavr

Описание объектов (сущностей) базы данных с атрибутами:

«Родитель»:

- ФИО;
- телефон;
- кем приходится ученику;
- электронная почта.

«Ученик»:

- ФИО;
- дата рождения;
- почта;
- внешний ключ родителя.

«Заявка на курс»:

- дата;
- время;
- статус;
- внешний ключ родителя;

«Курс»:

- название;
- описание;
- продолжительность;
- Стоимость;
- уровень сложности.

«Тема»:

- название;
- описание;
- ссылки на материал;
- материалы;
- внешний ключ курс.

«Урок»:

- дата;
- время начала;
- время окончания
- внешний ключ учителя;
- внешний ключ темы.

«Учитель»:

- ФИО;
- должность;
- телефон;
- электронная почта.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		



### 3 Разработка программного продукта

#### 3.1 Инструментальные и программные средства разработки

При проектировании программного продукта для частной школы программирования и робототехники были использованы разнообразные средства разработки, способствующие эффективному управлению процессом разработки и визуализации архитектуры проекта.

Прототипирование интерфейсов и пользовательских сценариев осуществлялось с помощью Figma. Этот инструмент позволил создать визуально привлекательные и интерактивные прототипы, что облегчило обсуждение и согласование требований с заказчиком и конечными пользователями.

Для создания диаграмм связей, включая карту сайта и административной панели, использовался MindMeister. Этот инструмент ментальных карт обеспечил наглядное представление структуры проекта и взаимосвязей между различными компонентами системы.

Сценарии использования разрабатывались с помощью SmartDraw. Этот инструмент помог в создании подробных схем, иллюстрирующих взаимодействие пользователей с системой, что позволило выявить и учесть все возможные варианты поведения пользователей.

Архитектура программного продукта, основанная на шаблоне MVT (Model-View-Template), также проектировалась с использованием SmartDraw. Для моделирования базы данных применялся ERD Lab, что позволило четко определить структуру данных и взаимосвязи между сущностями.

Календарный план разработки, представлен в виде диаграммы Ганта, создавался с использованием GanttPRO. Этот инструмент обеспечил детальное планирование всех этапов разработки, что способствовало эффективному управлению временем и ресурсами проекта.

Для управления задачами и координации работы команды использовался Trello. Этот инструмент позволил организовать рабочие процессы, отслеживать прогресс выполнения задач и поддерживать коммуникацию между членами команды.

HTML (HyperText Markup Language) является стандартным языком разметки для создания веб-страниц. Он широко поддерживается всеми веб-браузерами и является основой для структуры веб-страниц. Простота синтаксиса HTML делает его легким для изучения и использования, что особенно важно для начинающих разработчиков. Он поддерживается всеми современными веб-браузерами, что гарантирует корректное отображение контента на различных устройствах и платформах. Это делает HTML идеальным выбором для создания веб-страниц, которые должны работать в различных средах. Сравнение HTML с другими языками разметки описано в таблице 2.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 2 – Сравнение HTML с другими языками разметки

Критерий	HTML	JSX (React)
Простота изучения	Очень высокая	JSX (React)
Скорость разработки	Высокая	Высокая
Поддержка стандартов	Полная	Требуется компиляция
Поддержка браузерами	Все браузеры	Требуется сборка (React)
Читаемость	Высокая	Высокая
Поддержка инструментов разработки	Отличная	Отличная (в рамках React)
Модульность и повторное использование	Средняя	Очень высокая
Сообщество и документация	Очень большое, активное	Большое, активное
Возможности шаблонизации	Ограниченные	Расширенные (в рамках React)
Поддержка динамических данных	Ограниченные	Очень высокая
Лучшие случаи применения	Веб-страницы, прототипирование	Одностраничные приложения (SPA)

CSS (Cascading Style Sheets) является стандартным языком для описания внешнего вида веб-страниц. Его синтаксис прост и легко усваивается, что делает его доступным для начинающих разработчиков. CSS поддерживается всеми современными веб-браузерами, что обеспечивает кроссбраузерную совместимость. Предоставляет механизмы каскадности и наследования, что позволяет легко применять стили к элементам на основе их иерархии и взаимосвязей. Это упрощает процесс создания комплексных стилей для веб-страниц. CSS поддерживает медиа-запросы, что позволяет создавать адаптивные дизайны, которые корректно отображаются на различных устройствах и экранах. Это критически важно для современных веб-сайтов и приложений. Сравнение CSS с другими языками описания внешнего вида описано в таблице 3.

Таблица 3 – Сравнение CSS с другими языками описания внешнего вида

Критерий	CSS	Sass	Less
Простота изучения	Очень высокая	Высокая	Высокая
Скорость разработки	Высокая	Высокая	Высокая
Поддержка браузерами	Полная	Требуется компиляция	Требуется компиляция
Читаемость	Высокая	Высокая	Высокая
Модульность и повторное использование	Ограниченная	Очень высокая	Высокая
Сообщество и документация	Очень большое, активное	Большое, активное	Среднее, активное
Возможности препроцессинга	Нет	Полные возможности	Полные возможности
Поддержка переменных	Ограниченная (CSS Variables)	Полная	Полная
Поддержка вложенности	Нет	Полная	Полная
Инструменты разработки	Отличные	Отличные (с плагинами)	Отличные (с плагинами)
Совместимость с проектами	Полная	Требуется настройка и сборка	Требуется настройка и сборка
Лучшие случаи применения	Любые веб-страницы и приложения	Проекты с большим объемом стилей	Проекты с большим объемом стилей

JavaScript является универсальным языком, который используется как на стороне клиента (фронтенд), так и на стороне сервера (бэкенд) с помощью Node.js. Это позволяет разработчикам использовать один язык для создания всего стека веб-приложения. JavaScript имеет встроенную поддержку асинхронного программирования с помощью колбэков, промисов и `async/await`. Это позволяет эффективно обрабатывать асинхронные операции, такие как запросы к API и взаимодействие с базами данных. JavaScript имеет одно из самых больших и активных сообществ разработчиков. Это означает, что существует множество библиотек, фреймворков и инструментов, которые облегчают разработку веб-приложений и решают многие общие задачи. Сравнение JavaScript с другими языками описано в таблице 4.

Таблица 4 – Сравнение JavaScript с другими языками

Критерий	JavaScript	TypeScript
Простота изучения	Высокая	Средняя
Скорость разработки	Высокая	Высокая
Типизация	Динамическая	Статическая
Поддержка фронтенда	Полная	Полная
Поддержка бэкенда	Полная (Node.js)	Полная (Node.js)
Поддержка асинхронности	Полная	Полная
Производительность	Высокая (Node.js)	Высокая (Node.js)
Сообщество и экосистема	Очень большое, активное	Большое, активное
Инструменты разработки	VSCode, WebStorm	VSCode, WebStorm
Фреймворки	React, Angular, Vue.js, Express.js	React, Angular, Vue.js, NestJS
Масштабируемость	Высокая	Высокая
Соответствие стандартам	Высокое (ES6 и выше)	Высокое
Популярные применения	Фронтенд, серверная часть (Node.js)	Фронтенд, серверная часть (Node.js)

Figma работает в браузере, что означает, что нет необходимости устанавливать дополнительное программное обеспечение. Он предоставляет мощные инструменты для создания интерактивных прототипов, что позволяет дизайнерам и разработчикам тестировать и демонстрировать взаимодействие в интерфейсе до начала разработки. Это облегчает управление проектом и позволяет отслеживать изменения. Он поддерживает создание и использование дизайн-систем, что позволяет стандартизировать элементы интерфейса и ускорить процесс разработки новых экранов и приложений. Сравнение Figma с другими инструментами описано в таблице 5.

Таблица 5 – Сравнение Figma с другими инструментами

Критерий	Figma	Adobe XD	Sketch
Простота изучения	Высокая	Высокая	Средняя

Продолжение таблицы 5

Скорость разработки	Высокая	Высокая	Высокая
Совместная работа	Полная (облачная)	Ограниченная (через файлы)	Ограниченная (только с платными планами)
Поддержка платформ	Веб, Windows, macOS	Windows, macOS	Только macOS
Стоимость	Бесплатная (есть платные планы)	Бесплатная (есть платные планы)	Платная
Поддержка плагинов	Широкая	Ограниченная	Очень широкая
Интерактивные прототипы	Полная	Полная	Ограниченная
Версионность	Встроенная	Ограниченная	Ограниченная
Сообщество и документация	Очень большое, активное	Большое, активное	Среднее, активное
Интеграция с другими инструментами	Хорошая	Хорошая	Ограниченная
Лучшие случаи применения	Совместная работа над дизайном, прототипирование, дизайн-системы	Прототипирование, дизайн интерфейсов	Дизайн интерфейсов, дизайн-системы
Простота изучения	Высокая	Высокая	Средняя

GitHub предоставляет удобный и интуитивно понятный веб-интерфейс, а также десктопные приложения для управления репозиториями и проектами. Это облегчает освоение и использование GitHub даже для начинающих пользователей. Он предоставляет возможность создавать как открытые, так и закрытые репозитории. Бесплатные планы GitHub включают основные функции, необходимые для большинства проектов, а платные планы предлагают дополнительные возможности и улучшенную поддержку. Сравнение GitHub с другими системами контроля версий описано в таблице 6.

Таблица 6 – GitHub с другими системами контроля версий

Простота использования	Высокая	Высокая	Высокая
Стоимость	Бесплатная (есть платные планы)	Бесплатная (есть платные планы)	Бесплатная (есть платные планы)
Критерий	GitHub	GitLab	Bitbucket
Поддержка репозитория	Git	Git	Git, Mercurial
Интерфейс	Веб-интерфейс, десктопное приложение	Веб-интерфейс, десктопное приложение	Веб-интерфейс, десктопное приложение
Инструменты CI/CD	Встроенные (GitHub Actions)	Встроенные (GitLab CI/CD)	Встроенные (Bitbucket Pipelines)
Сообщество	Очень большое, активное	Большое, активное	Среднее, активное
Интеграция с другими инструментами	Очень широкая	Широкая	Широкая
Поддержка документации	Отличная	Очень хорошая	Очень хорошая
Масштабируемость	Очень высокая	Очень высокая	Высокая
Поддержка закрытых репозитория	Полная	Полная	Полная
Управление проектами	Встроенные (Issues, Projects)	Встроенные (Issues, Boards)	Встроенные (Issues, Boards)

Продолжение таблицы 6

Лучшие случаи применения	Открытые и закрытые проекты, совместная разработка, CI/CD	Открытые и закрытые проекты, CI/CD, DevOps	Открытые и закрытые проекты, CI/CD, DevOps
--------------------------	---	--	--

Trello обладает интуитивно понятным и простым интерфейсом, который легко освоить даже для новичков. Доски, списки и карточки позволяют легко организовать задачи и управлять проектами без необходимости долгого обучения. Trello предлагает высокую степень настраиваемости, позволяя пользователям адаптировать доски и карточки под свои нужды. Power-Ups (плагины) расширяют функциональность, добавляя интеграции с другими инструментами и дополнительные возможности. Сравнение Trello с другими инструментами для управления проектом описано в таблице 7.

Таблица 7 - Сравнение Trello с другими инструментами для управления проектом

Критерий	Trello	Jira	Asana
Простота использования	Очень высокая	Средняя	Высокая
Стоимость	Бесплатная (есть платные планы)	Платная (есть бесплатная версия)	Бесплатная (есть платные планы)
Методология	Канбан	Канбан, Скрам, другие	Гибкая
Интеграции	Широкие (с помощью Power-Ups)	Очень широкие (с Atlassian продуктами и др.)	Широкие
Отчетность и аналитика	Ограниченная	Очень хорошая	Хорошая

Продолжение таблицы 7

Управление задачами	Простое и интуитивное	Сложное и мощное	Простое и интуитивное
Настраиваемость	Очень высокая	Очень высокая	Высокая
Совместная работа	Отличная	Отличная	Отличная
Поддержка платформ	Веб, iOS, Android	Веб, iOS, Android	Веб, iOS, Android
Лучшие случаи применения	Простые и средние по сложности проекты, команды любого размера	Средние и крупные проекты, особенно для команд разработчиков	Средние и крупные проекты, команды любого размера
Критерий	Trello	Jira	Asana
Простота использования	Очень высокая	Средняя	Высокая

Для разработки кода использовалась среда разработки Pycharm. Этот мощный IDE поддерживает различные языки программирования и предоставляет инструменты для автоматического завершения кода, отладки, тестирования и управления проектами. Pycharm существенно увеличивает производительность разработчиков, упрощая процесс написания, тестирования и отладки кода. В таблице 8 представлены сравнения с другими IDE.

Таблица 8 - Pycharm

Стек	Преимущества	Недостатки
Pycharm	Мощный IDE с поддержкой множества функций.	Высокая стоимость лицензии для полной версии.
	Поддержка рефакторинга кода и тестирования.	Высокие системные требования.
	Интеграция с VCS (Git, Mercurial и др.).	Затратный порог для новичков.



Продолжение таблицы 8

Visual Studio Code	Бесплатное использование.	Ограниченные возможности по сравнению с Pycharm.
	Множество плагинов и расширений.	Требует больше настроек для аналогичных функций.
Atom	Бесплатное и открытый исходный код.	Медленная работа с большими проектами.
	Простота использования.	Меньше возможностей для Python-разработки.

Основным языком программирования, использованным в проекте, стал Python. Это высокоуровневый язык с простой и понятной синтаксической структурой, что делает его идеальным для быстрой разработки и поддержания кода. Python обладает широкой экосистемой библиотек и модулей, которые позволяют решать самые разнообразные задачи, начиная от веб-разработки и заканчивая анализом данных. Сравнения представлены в таблице 9.

Таблица 9 – Python

Стек	Преимущества	Недостатки
Python	Простой и читаемый синтаксис.	Сравнительно низкая скорость выполнения.
	Огромное количество библиотек и фреймворков.	Динамическая типизация может приводить к ошибкам.
	Широкое сообщество и поддержка.	Ограниченные возможности для мобильной разработки.
JavaScript	Широко используется в веб-разработке.	Сложный синтаксис для новичков.

Продолжение таблицы 9

	Множество фреймворков и библиотек.	Проблемы с производительностью на сервере.
Ruby	Читаемый и простой синтаксис.	Ограниченное количество библиотек.
	Хорош для быстрого прототипирования.	Меньше ресурсов для обучения.

В качестве фреймворка для разработки веб-приложений был выбран Django. Сравнения с другими фреймвоками описаны в таблице 10. Django обеспечивает быстрый и безопасный процесс разработки, предоставляя множество встроенных инструментов и библиотек для работы с базами данных, авторизацией пользователей, обработкой форм и многим другим. Одним из ключевых преимуществ Django является его архитектура MVT (Model-View-Template), которая способствует разделению логики приложения, его представления и данных, что облегчает разработку и поддержку кода.

Таблица 10 – Django

Стек	Преимущества	Недостатки
Django	Высокий уровень безопасности.	Строгая структура может ограничивать гибкость.
	Быстрая разработка благодаря встроенным функциям.	Монотонная настройка для простых проектов.
	Хорошая документация и сообщество.	Высокий порог вхождения для новичков.
Flask	Простота и легкость.	Меньше встроенных функций.
	Гибкость в настройке.	Требует больше времени на разработку.
Ruby on Rails	Быстрая разработка.	Меньше гибкости по сравнению с Django.

Продолжение таблицы 10

	Хорошая документация и сообщество.	Строгая структура.
--	------------------------------------	--------------------

Для создания API использовался Django REST Framework (DRF). Этот мощный инструмент позволяет быстро и удобно разрабатывать RESTful API, обеспечивая все необходимые функции для работы с запросами и ответами, аутентификацией и авторизацией, сериализацией данных и многим другим. DRF значительно упрощает интеграцию фронтенда и бэкенда, делая взаимодействие между клиентом и сервером более эффективным и безопасным. Сравнительная характеристика представлена в таблице 11.

Таблица 11 – Django REST Framework

Стек	Преимущества	Недостатки
Django REST Framework	Легкая интеграция с Django.	Перегруженность для простых API.
	Поддержка аутентификации и авторизации.	Возможны сложности с производительностью.
	Хорошая документация и сообщество.	Высокий порог вхождения для новичков.
Flask-RESTful	Простота использования.	Меньше возможностей.
	Хорош для простых API.	Требует больше настроек для сложных проектов.
Express.js	Высокая производительность.	Меньше ресурсов для обучения.
	Широко используется в веб-разработке.	Требует знания JavaScript.

Для тестирования API использовался Postman. Этот инструмент позволяет отправлять HTTP-запросы к API, проверять ответы и автоматизировать тестирование. Postman поддерживает создание коллекций запросов, тестовых сценариев и автоматических тестов, что помогает обеспечить качество и надежность API.

Swagger был выбран за его способность автоматически генерировать документацию API и поддерживать интерактивное тестирование, что облегчает разработку и тестирование API для всех участников команды. Сравнительная характеристика Swagger и Postman представлена в таблице 12.

Таблица 12 –Swagger и Postman

Стек	Преимущества	Недостатки
Swagger	Автоматическая генерация документации API.	Сложность настройки для сложных API.
	Интерактивное тестирование API.	Возможны проблемы с синхронизацией документации.
	Широкая поддержка форматов данных.	Ограниченная функциональность без дополнительных плагинов.
Postman	Удобство тестирования API.	Высокая стоимость профессиональной версии.
	Интуитивно понятный интерфейс.	Перегруженность интерфейса для простых задач.
Insomnia	Простота использования.	Ограниченная функциональность.
	Бесплатное использование.	Меньше возможностей по сравнению с Swagger.

MySQL был выбран за его высокую производительность и надежность, а также широкую поддержку со стороны сообщества, что делает его отличным выбором для базы данных образовательной платформы. Сравнительная характеристика представлена в таблице 13.

Таблица 13–MySQL

Стек	Преимущества	Недостатки
MySQL	Высокая производительность и надежность.	Ограниченные возможности для масштабирования.
	Широкое сообщество и поддержка.	Ограниченная поддержка NoSQL данных.
	Бесплатный и открытый исходный код.	Сложности с настройкой репликации.
PostgreSQL	Поддержка расширенных функций.	Сложнее в настройке и администрировании.
	Высокая производительность.	Меньше ресурсов для обучения.
SQLite	Простота использования.	Ограниченные возможности масштабирования.
	Встроен в Python.	Меньше возможностей по сравнению с MySQL.

Для локальной разработки и тестирования использовался OpenServer. Этот инструмент позволяет быстро развернуть локальный веб-сервер с поддержкой различных технологий, таких как PHP, MySQL, Apache и многие другие. OpenServer облегчает настройку окружения разработки, предоставляя все необходимые инструменты в одном пакете. Сравнительная характеристика представлена в таблице 14.

Таблица 14 – OpenServer

Стек	Преимущества	Недостатки
OpenServer	Простота установки и настройки.	Ограниченные возможности для масштабирования.

### Продолжение таблицы 14

	Поддержка множества серверных технологий.	Работает только на Windows.
	Бесплатное использование.	Ограниченная поддержка со стороны сообщества.
ХАМРР	Поддержка множества платформ.	Более сложная настройка.
	Простота использования.	Меньше возможностей для настройки.
WampServer	Простота установки и настройки.	Меньше возможностей по сравнению с OpenServer.

### 3.2 Календарный план разработки

Для программного продукта был составлен календарный план разработки, который представлен ниже в таблице 15. В таблице отражены задачи и последовательность их выполнения.

Таблица 15 – Календарный план

№	Виды работ	Даты разработки	Трудозатраты (час)
1	Сбор требований	11.03.24-12.03.24	10
2	Анализ и формирование требований	12.03.24-13.03.24	8
3	Проектирование и разработка архитектуры	13.03.24-14.03.24	13
4	Прототипирование	15.03.24	5
5	Разработка макетов	15.03.24-19.03.24	14
6	Проектирование базы данных	19.03.24-21.03.24	14

Продолжение таблицы 15

7	Согласование дизайна с заказчиком	21.03.24	4
8	Разработка front-end	21.03.24-16.04.24	67
9	Разработка back-end	22.04.24 -16.04.24	80
10	Связывание frontend и backend	17.04.24	4
11	Согласование с заказчиком	17.04.24	3
12	Разработка тест-плана и тест-кейса	17.04.24-18.04.24	4
13	Тестирование	18.04.24-19.04.24	8
14	Доработки и исправление ошибок	19.04.24-22.04.24	8
15	Написание документации	22.04.24	5
16	Внедрение	23.04.24	6

Диаграмма Ганта, разработанная на основе календарного плана, представлена на рисунке 9:

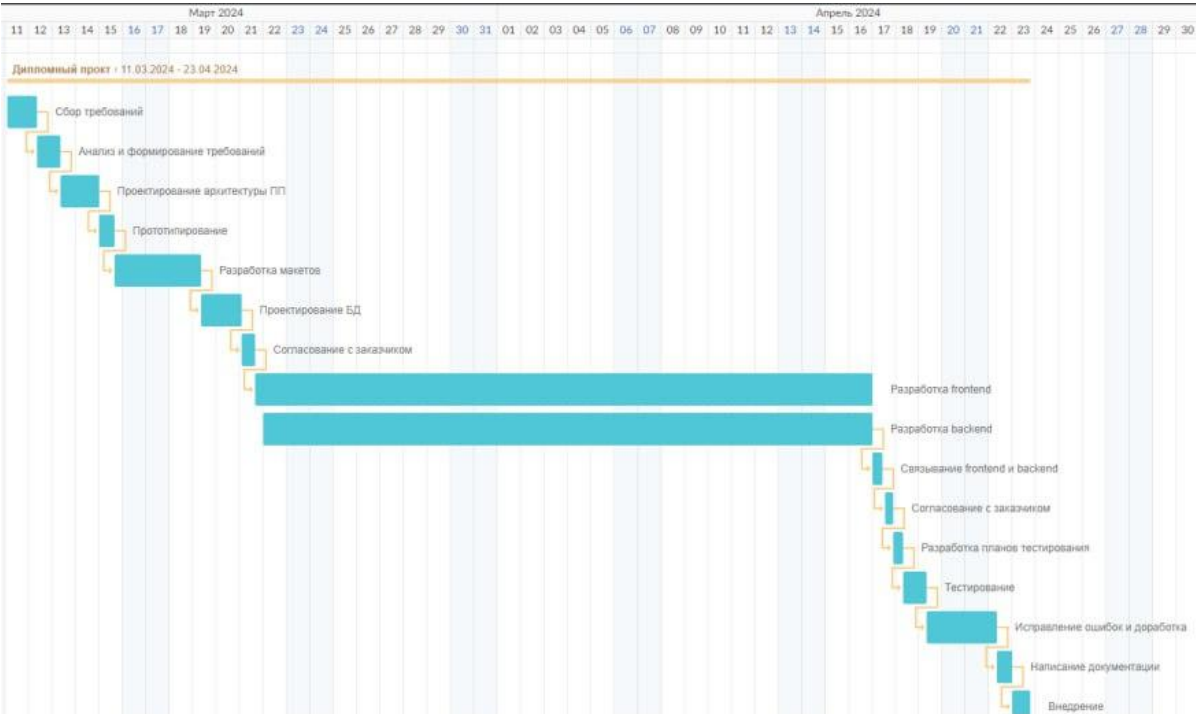


Рисунок 9 – Диаграмма Ганта

## 4 Тестирование программного продукта

Тестирование программного продукта является ключевым этапом разработки, который позволяет выявить и устранить ошибки, обеспечивая стабильную и надежную работу системы. В данной главе представлены методы и результаты ручного и автоматизированного тестирования.

### 4.1 Тест план

Тест план представляет собой документ, описывающий стратегию, цели, объем и методы тестирования программного продукта. В рамках тест плана для нашего проекта были определены ключевые функциональные области, подлежащие проверке, а также конкретные тест кейсы, описывающие шаги выполнения тестов и ожидаемые результаты. План тестирования включает проверку основных функций, таких как регистрация пользователей, запись на курсы и отправка форм обратной связи.

Для тестирования клиентской части сайта было принято решение использовать ручное тестирование. Ручное тестирование позволяет имитировать реальные сценарии использования, что помогает выявить потенциальные проблемы или недочеты, которые могли бы остаться незамеченными при автоматизированном тестировании.

Разработанные тест-планы представлены в таблице 16:

Таблица 16 – План интеграционного тестирования

Тест кейс, №	Название	Описание
1	Тест загрузки страниц информационной системы.	Необходимо протестировать загрузку всех страниц.
2	Тест навигации.	Необходимо протестировать переход между страницами на сайте.

Далее представлены Тест кейсы:

Таблица 17– Тест-кейс навигации

Название	Тест навигации	
Действие	Ожидаемый результат	Результат теста



Продолжение таблицы 17

Успешный переход между страницами сайта при помощи меню	Переход происходит корректно	Положительный
Успешный переход между страницами при помощи кнопок	Переход происходит корректно	Положительный
Результат теста:		
Проверить переход между страницами на сайте	Все страницы загружаются, корректный переход между страницами	Положительный

Таблица 18– Тест-кейс кабинета администратора

Название	Тест корректности работы кабинета администратора	
Действие	Ожидаемый результат	Результат теста
Вход в кабинет администратора с правильными данными	Вход успешно выполнен	Положительный
Корректность работы меню	Переход по всем страницам работает корректно	Положительный
Удаление записей в БД	Удаление успешно выполнено	Положительный
Редактирование записей в БД	Редактирование успешно выполнено	
Вход в кабинет администратора с неправильными данными	Вход не выполнен	Положительный
Выход из кабинета администратора	Выход успешно выполнен	Положительный

Продолжение таблицы 18

Отображение статистики данных на главной странице	Корректное отображение данных	Положительный
Результат теста:		
Проверить работу кабинета администратора	Корректная работа кабинета администратора	Положительный

Таблица 19 – Тест-кейс отображения текста и шрифтов

Название	Тест отображения текста и шрифта	
Действие	Ожидаемый результат	Результат теста
Успешное отображение текста и шрифта на всех страницах	Отображение происходит корректно	Положительный
Результат теста:		
Проверить отображение текста и шрифта	Весь текст отображается корректно, шрифт отображается корректно	Положительный

Таблица 20 – Тест-кейс адаптивность сайта

Название	Тест адаптивности сайта	
Действие	Ожидаемый результат	Результат теста
Успешная адаптация на всех устройствах	Адаптация происходит корректно	Положительный
Результат теста:		
Проверить адаптацию сайта	Адаптация на всех устройствах происходит корректно	Положительный

Таблица 21 – Тест-кейс целостность ссылок

Название	Тест целостности ссылок	
Действие	Ожидаемый результат	Результат теста
Успешный переход в чат в «WhatsApp»	Переход совершился успешно	Положительный

Продолжение таблицы 21

Успешный переход на страницах по ссылкам	Переход совершился успешно	Положительный
Результат теста:		
Проверить адаптацию сайта	Переход по всем ссылкам совершился успешным	Отрицательный

Таблица 22 – Тест-кейс карты

Название	Тест корректности отображения карты	
Действие	Ожидаемый результат	Результат теста
Отображение карты	Корректное отображение карты	Положительный
Результат теста:		
Проверить отображение карты	Корректное отображение карты	Положительный

Функциональные тест-кейсы и баг-репорты представлены в таблицах от 23, до 30:

Таблица 23–Тест-кейс ТС-001: Тестирование API для подачи заявки на курс

Поле	Описание
Идентификатор	ТС-001
Название	Тестирование API для подачи заявки на курс
Описание	Проверка функциональности API для подачи заявки на курс через POST-запрос.
Предусловия	Пользователь должен быть аутентифицирован.
Шаги выполнения	Отправить POST-запрос на endpoint/api/course_applications/ с телом запроса, содержащим данные заявки. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 201. 2. Данные новой заявки в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.

Продолжение таблицы 23

Приоритет	Высокий
Статус	Не пройден

Таблица 24– Баг-репорт BUG-001: Тестирование API для подачи заявки на курс.

Поле	Описание
Идентификатор	BUG-001
Название	Некорректный ответ API для подачи заявки на курс
Описание	При отправке POST-запроса на endpoint /api/course_applications/ возвращается некорректный статус или структура данных.
Шаги воспроизведения	1. Запустить сервер. 2. Отправить POST-запрос на endpoint /api/course_applications/ с телом запроса, содержащим данные заявки. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 201. 2. Данные новой заявки в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Открыт
Решение	Проверьте логику обработки POST-запросов для endpoint /api/course_applications/. Убедитесь, что данные заявки корректно обрабатываются и возвращаются.

Таблица 25– Тест-кейс TC-002: Тестирование API для отправки формы обратной связи

Поле	Описание
Идентификатор	TC-002

Продолжение таблицы 25

Название	Тестирование API для отправки формы обратной связи
Описание	Проверка функциональности API для отправки формы обратной связи через POST-запрос.
Предусловия	Пользователь должен быть зарегистрирован и аутентифицирован.
Шаги выполнения	1. Запустить сервер. 2. Отправить POST-запрос на endpoint /api/feedback/ с телом запроса, содержащим данные формы обратной связи. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 201. 2. Данные новой формы обратной связи в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Не пройден

Таблица 26– Баг-репорт BUG-002: Тестирование API для отправки формы обратной связи

Поле	Описание
Идентификатор	BUG-002
Название	Некорректный ответ API для отправки формы обратной связи
Описание	При отправке POST-запроса на endpoint /api/feedback/ возвращается некорректный статус или структура данных.
Шаги воспроизведения	1. Запустить сервер. 2. Отправить POST-запрос на endpoint /api/feedback/ с телом запроса, содержащим данные формы обратной связи. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 201. 2. Данные новой формы обратной связи в ответе.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение таблицы 26

Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Открыт
Решение	Проверьте логику обработки POST-запросов для endpoint /api/feedback/. Убедитесь, что данные формы обратной связи корректно обрабатываются и возвращаются.

Таблица 27– Тест-кейс ТС-003: Тестирование API для изменения материалов

Поле	Описание
Идентификатор	ТС-003
Название	Тестирование API для изменения материалов
Описание	Проверка функциональности API для изменения материалов через PUT-запрос.
Предусловия	Пользователь должен быть зарегистрирован как преподаватель и аутентифицирован.
Шаги выполнения	1. Запустить сервер. 2. Отправить PUT-запрос на endpoint /api/materials/{id}/ с телом запроса, содержащим измененные данные материалов. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 200. 2. Обновленные данные материалов в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Не пройден

Таблица 28– Баг-репорт BUG-003: Тестирование API для изменения материалов

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Поле	Описание
Идентификатор	BUG-003
Название	Некорректный ответ API для изменения материалов
Описание	При отправке PUT-запроса на endpoint /api/materials/{id}/ возвращается некорректный статус или структура данных.
Шаги воспроизведения	1. Запустить сервер. 2. Отправить PUT-запрос на endpoint /api/materials/{id}/ с телом запроса, содержащим измененные данные материалов. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 200. 2. Обновленные данные материалов в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Открыт
Решение	Проверьте логику обработки PUT-запросов для endpoint /api/materials/{id}/. Убедитесь, что данные материалов корректно обрабатываются и возвращаются.

Таблица 29 –Тест-кейс ТС-004: Тестирование API для регистрации

Поле	Описание
Идентификатор	ТС-004
Название	Тестирование API для регистрации
Описание	Проверка функциональности API для регистрации нового пользователя через POST-запрос.
Предусловия	Нет
Шаги выполнения	1. Запустить сервер. 2. Отправить POST-запрос на endpoint /api/register/ с телом запроса, содержащим данные нового пользователя. 3. Проверить статус ответа и структуру данных.

Продолжение таблицы 29

Ожидаемый результат	Статус ответа 201. 2. Данные нового пользователя в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Не пройден

Таблица 30– Баг-репорт BUG-004: Тестирование API для регистрации

Поле	Описание
Идентификатор	BUG-004
Название	Некорректный ответ API для регистрации
Описание	При отправке POST-запроса на endpoint /api/register/ возвращается некорректный статус или структура данных.
Шаги воспроизведения	1. Запустить сервер. 2. Отправить POST-запрос на endpoint /api/register/ с телом запроса, содержащим данные нового пользователя. 3. Проверить статус ответа и структуру данных.
Ожидаемый результат	Статус ответа 201. 2. Данные нового пользователя в ответе.
Фактический результат	Статус ответа 400. 2. Некорректная структура данных в ответе.
Приоритет	Высокий
Статус	Открыт
Решение	Проверьте логику обработки POST-запросов для endpoint /api/register/. Убедитесь, что данные нового пользователя корректно обрабатываются и возвращаются.



## 5 Разработка документации на программный продукт

### 5.1 Руководство пользователя

На главной странице пользователь может увидеть курсы. Авторизованный пользователь имеет доступ к личному кабинету, где можно просмотреть профиль, расписание занятий, записаться на курсы и отправить форму обратной связи. Родители могут записываться на занятия. Преподаватели имеют возможность редактировать материалы курсов. Полное руководство пользователя и администратора представлено в Приложении Г.

### 5.2 Руководство администратора

Административная панель проекта «codozavrt» предоставляет интерфейс для управления различными аспектами веб-приложения. Доступ к административной панели имеют только пользователи с соответствующими правами (администраторы). В панели администратора можно управлять пользователями, назначать роли, создавать и редактировать курсы, а также обрабатывать заявки на запись. Администратор может также просматривать отчеты о работе системы и анализировать обратную связь, поступающую от пользователей. Полное руководство пользователя и администратора представлено в Приложении Д.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

## 6 Информационная безопасность

Информационная безопасность играет ключевую роль в разработке любого программного продукта, особенно в случае веб-приложений, которые обрабатывают персональные данные пользователей. Наш проект, разработанный для частной школы программирования и робототехники, требует надежных методов защиты информации, чтобы обеспечить безопасность данных учеников, родителей и преподавателей. В этом разделе будут рассмотрены методы, применяемые для обеспечения информационной безопасности в рамках нашего проекта.

Первым и важнейшим аспектом является аутентификация и авторизация пользователей. В нашем проекте используется система токенов JWT (JSON Web Tokens), которая позволяет безопасно передавать данные между клиентом и сервером. Каждый пользователь получает уникальный токен, обеспечивающий доступ к защищенным ресурсам. Это позволяет надежно идентифицировать пользователей и контролировать их доступ к различным функциям системы. Также применяются надежные алгоритмы шифрования для хранения паролей и других чувствительных данных в базе данных, что минимизирует риск компрометации данных в случае взлома.

Еще одним критически важным методом является управление доступом. В системе реализована модель ролевого доступа (RBAC), которая ограничивает доступ пользователей к определенным функциям и данным на основе их ролей. В нашем проекте предусмотрены роли для родителей, учеников, учителей и администраторов, каждая из которых имеет свои права и ограничения. Это позволяет строго контролировать, какие действия могут выполнять пользователи, и предотвращает несанкционированный доступ к важным данным. Встроенные в Django механизмы проверки CSRF-токенов и экранирования пользовательских данных помогают предотвратить выполнение вредоносных запросов и внедрение вредоносного кода.

### 6.1 Требования безопасности при выполнении работ

При разработке и эксплуатации веб-сайта, предназначенного для предоставления информации о нотариальных услугах и тарифах, необходимо учитывать несколько аспектов безопасности. Несмотря на то, что сайт не обрабатывает персональные данные, обеспечение его защищенности от потенциальных угроз и уязвимостей остается критически важным. Поэтому были предприняты следующие меры:

- система входа и выхода для административной панели;
- протокол HTTPS для шифрования данных, передаваемых между пользователями и сервером, что обеспечивает конфиденциальность и целостность данных;

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

- CSRF токены: используются для защиты от подделки межсайтовых запросов, предотвращая выполнение нежелательных действий от имени пользователя;
- ORM (Object-Relational Mapping) для построения запросов к базе данных, что предотвращает возможность SQL-инъекций.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

## 7 Экономический расчет

### 7.1 Расчет стоимости работ, связанных с разработкой программного продукта (сайта)

#### 7.1.1 Расчёт прямых расходов

Разработкой проекта будут заниматься аналитик, дизайнер, front-end разработчик, back-end разработчик, тестировщик.

Заработная плата аналитика составит 45 000 руб. в мес.

Заработная плата дизайнера составит 30 000 руб. в мес.

Заработная плата front-end разработчика составит 50 000 руб. в мес.

Заработная плата back-end разработчика составит 50 000 руб. в мес.

Заработная плата тестировщика составит 35 000 руб. в мес.

Расчёт произведём по формуле:

$$\text{ФОТ} = \text{ЗП} \times n, \quad (1)$$

где

ФОТ – фонд оплаты труда работника, руб.;

ЗП – заработная плата работника, руб.;

n – количество месяцев в году, мес.

ФОТ год аналитик =  $45\,000 \times 12 = 540\,000,00$  руб.

ФОТ год дизайнер =  $30\,000 \times 12 = 360\,000,00$  руб.

ФОТ год front-end =  $50\,000 \times 12 = 600\,000,00$  руб.

ФОТ год back-end =  $50\,000 \times 12 = 600\,000,00$  руб.

ФОТ год тестировщик =  $35\,000 \times 12 = 420\,000,00$  руб.

Определим стоимость трудозатрат за 1 час.

$$\text{Ct}_{\text{час}} = \text{ФОТ} / N_{\text{рв}}, \quad (2)$$

где

Ct<sub>час</sub> – стоимость трудозатрат за 1 час, руб.;

ФОТ – фонд оплаты труда работника, руб.;

N<sub>рв</sub> – норма рабочего времени при 40-ка часовой рабочей неделе, в соответствии с производственным календарём на 2024 год,

$$\text{Ct}_{\text{час аналитик}} = \frac{540\,000}{1979} = 272,87 \text{ руб.}$$

$$\text{Ct}_{\text{час дизайнер}} = \frac{360\,000}{1979} = 181,91 \text{ руб.}$$

$$\text{Ct}_{\text{час front-end}} = \frac{600\,000}{1979} = 303,18 \text{ руб.}$$

$$Ct_{\text{час back-end}} = \frac{600\,000}{1979} = 303,18 \text{ руб.}$$

$$Ct_{\text{час тестировщик}} = \frac{420\,000}{1979} = 212,23 \text{ руб.}$$

Рассчитаем сумму расходов на оплату труда, связанных с разработкой программного продукта (сайта). Расчёт расходов на оплату труда с учётом трудозатрат занесём в таблицу 38.

Таблица 31 – Расчет расходов на оплату труда

Название специалиста	Виды работ	Трудозатраты, час	Стоимость трудозатрат в час, руб.	Общая стоимость работы, руб.
Аналитик	Сбор информации	10	272,87	27228,70
	Анализ и формирование требований	8	272,87	2182,96
	Согласование с заказчиком	7	272,87	1910,09
Итого	-	25	-	6821,75
Дизайнер	Разработка прототипа	5	181,91	909,55
	Разработка макета	14	181,91	2546,74
Итого	-	19	-	3456,29
front-end разработчик	Проектирование архитектуры ПП	6	303,18	1819,08
	Разработка front-end	67	303,18	20313,06
	Связывание front-end и back-end	2	303,18	606,36
	Доработки и исправление ошибок	4	303,18	1212,72
	Написание документации	1	303,18	303,18
	Внедрение	3	303,18	909,54
Итого	-	83	-	25163,94

Продолжение таблицы 31

back-end разработчик	Проектирование и разработка архитектуры	7	303,18	2122,26
	Проектирование БД	14	303,18	4244,52
	Разработка back-end	80	303,18	24254,40
	Связывание front-end и back-end	2	303,18	606,36
	Доработки и исправление ошибок	4	303,18	1212,72
	Написание документации	4	303,18	1212,72
	Внедрение	3	303,18	909,54
Итого	-	114	-	34562,52
Тестировщик	Разработка тест-плана и тест-кейса	4	212,23	848,92
	Тестирование	8	212,23	1697,87
Итого	-	12	-	2546,76
Всего	-	253	-	72551,26

7.1.2 Расчёт страховых взносов во внебюджетные фонды

Расчет страховых взносов произведем по формуле:

$$CB = \frac{POT \cdot T_{CB}}{100\%}, \quad (3)$$

где

CB – сумма страховых взносов, руб.;

POT – расходы на оплату труда, руб.;

T<sub>CB</sub> – тариф страхового взноса, %.

$$CB = \frac{72551,26 \times 30,2\%}{100\%} = 21910,00 \text{ руб.}$$

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

## 7.2 Определим накладные расходы на разработку программного продукта (сайта)

Для расчёта накладных расходов рассчитаем временные сроки выполнения проекта. Расчёт произведём по формуле:

$$СП = t_{\text{общ}} / 8, \quad (4)$$

где

СП – временные сроки выполнения проекта, дн.;  
 $t_{\text{общ}}$  – общая сумма трудозатрат (из таблицы 1), час.;  
8 – стандартный рабочий день при 40-ка часовой рабочей неделе, час.

$$СП = 253 / 8 = 31,7 \text{ дн.}$$

### 7.2.1 Расчёт расходов на услуги связи

Расчёт услуг связи произведём по формуле:

$$P_{\text{у.с.}} = (P_{\text{интернет}} + P_{\text{тел. связь}} + \text{и т.д.}) / \text{Кол-во дней (мес)} \times СП, \quad (5)$$

где

$P_{\text{у.с.}}$  – расходы на услуги связи, руб.;  
 $P_{\text{интернет}} + P_{\text{тел. связь}} + \text{и т.д.}$  – расходы на интернет, мобильную связь и т.д.;  
Кол-во дней (мес) – среднее количество рабочих дней в месяце, дн.;  
СП – временные сроки выполнения проекта, дн.

$$P_{\text{у.с.}} = ((500 + 500) / 20 + (500 + 500) / 21) / 2 \times 31,7 = 1547,28 \text{ руб.}$$

### 7.2.2 Расчёт расходов на коммунальные услуги

Расчёт коммунальных услуг произведём по формуле:

$$P_{\text{коммунальные}} = n_{\text{к.у.}} \times S / \text{Кол-во дней (мес)} \times СП, \quad (6)$$

где

$P_{\text{коммунальные}}$  – расходы на коммунальные услуги, руб.;  
 $n_{\text{к.у.}}$  – средняя рыночная ставка на коммунальные услуги за 1 кв. метр;  
S – площадь помещения в кв.метрах.

$$P_{\text{коммунальные}} = (140 \times 10 / 20 + 140 \times 10 / 21) / 2 \times 31,7 = 2166,38 \text{ руб.}$$

### 7.2.3 Расчёт расходов на рекламу

Сумма расходов на рекламу рассчитывается по формуле

$$P_{\text{реклама}} = \text{РОТ} \times n_{\text{реклама}} / 100\%, \quad (7)$$

где

$P_{\text{реклама}}$  – сумма расходов на рекламу, руб.

$\text{РОТ}$  – расходы на оплату труда, руб.;

$n_{\text{реклама}}$  – норматив расходов на рекламу, %.

$$P_{\text{реклама}} = 72551,26 \times 15\% / 100\% = 10882,69 \text{ руб.}$$

### 7.2.4 Расчёт прочих расходов

Сумма прочих расходов рассчитывается по формуле:

$$P_{\text{прочие}} = \text{РОТ} \times n_{\text{прочие}} / 100\%, \quad (8)$$

где

$P_{\text{прочие}}$  – сумма прочих расходов, руб.

$\text{РОТ}$  – расходы на оплату труда, руб.;

$n_{\text{прочие}}$  – норматив прочих расходов, %.

$$P_{\text{прочие}} = 72551,26 \times 10\% / 100\% = 7255,13 \text{ руб.}$$

### 7.2.5 Расчёт общей суммы накладных расходов

Определим общую сумму накладных расходов по формуле

$$P_{\text{накладные}} = P_{\text{у.с.}} + P_{\text{коммунальные}} + P_{\text{реклама}} + P_{\text{прочие}}, \quad (9)$$

$$P_{\text{накладные}} = 1547,28 + 2166,38 + 10882,69 + 7255,13 = 21851,48 \text{ руб.}$$

## 7.3 Расчёт себестоимости работ по разработке программного продукта (сайта)

Расчёт себестоимости работ по проекту, то есть цену создания программного продукта (сайта) ( $C_p$ ) сведём в таблицу 32.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		



Таблица 32 – Себестоимость работ по созданию программного продукта (сайта)

Статьи расходов	Сумма, руб.
Расходы на оплату труда	7255,26
Страховые взносы	21910,00
Накладные расходы	21851,48
Итого:	116312,74

#### 7.4 Расчет суммы выручки от реализации программного продукта (сайта)

Цена создания и реализации программного продукта (сайта), то есть выручка от реализации проекта определяется по формуле:

$$V_{\text{реал}} = C_p + П, \quad (10)$$

где

$C_p$  – себестоимость работ или затраты на создание программного продукта (сайта);

$П$  – планируемый размер прибыли.

Размер прибыли определяется исходя из нормы рентабельности по формуле:

$$П = \frac{C_p \cdot ЧН_R}{100\%}, \quad (11)$$

где

$Н_R$  – уровень рентабельности проекта.

$$П = \frac{116312,74 \times 15\%}{100\%} = 17446,91 \text{ руб.}$$

$$V_{\text{реал}} = 116312,74 + 17446,91 = 133759,65 \text{ руб.}$$

Основные показатели, учитываемые при расчете цены программного продукта (сайта) приведены в таблице 33.

Таблица 33 – Расчет цены программного продукта (сайта).

Наименование показателя	Сумма, руб.
Затраты на создание программного продукта (сайта)(себестоимость)	116312,74

Продолжение таблицы 33

Прибыль	17446,91
Выручка от реализации проекта	133759,65

7.5 Расчет суммы единого налога при применении упрощенной системы налогообложения

Произведем расчет суммы единого налога по формуле:

$$УСН_{нач} = \frac{Д \cdot Ч \cdot С_{УСН}}{100\%}, \quad (12)$$

где

$УСН_{нач}$  – сумма единого налога начисленная, руб.;

$Д$  – доход, руб.;

$С_{УСН}$  – ставка налога, %.

$$УСН_{нач} = \frac{133759,65 \times 6\%}{100\%} = 8025,00 \text{ руб.}$$

Определим сумму минимального налога по формуле:

$$УСН_{min} = \frac{УСН_{нач} \cdot Ч \cdot 50\%}{100\%}, \quad (13)$$

где

$УСН_{min}$  – минимальная сумма налога, руб.

$$УСН_{min} = \frac{8025 \times 50\%}{100\%} = 4012,00 \text{ руб.}$$

Произведем расчет суммы единого налога, подлежащей перечислению в бюджет по одной из формул:

$$УСН_{бюджет} = УСН_{нач} - УСН_{min}, \quad (14)$$

$$УСН_{бюджет} = УСН_{нач} - \text{Страховые взносы}, \quad (15)$$

$$УСН_{бюджет} = 8025 - 21910 = - 13885 \text{ руб}$$

Сумма единого налога, подлежащего перечислению в бюджет составляет 4012,00 руб.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

## 7.6 Расчет чистой прибыли организации

Рассчитаем сумму чистой прибыли, остающаяся в распоряжении организации, после уплаты единого налога по формуле

$$П_{\text{чистая}} = П - УСН_{\text{бюджет}}, \quad (16)$$

где

П – планируемая сумма прибыли от реализации проекта, руб.;

УСН<sub>бюджет</sub> – единый налог, подлежащий уплате в бюджет, руб.

$$П_{\text{чистая}} = 17446,91 - 4012,00 = 13434,91 \text{ руб.}$$

Расчет чистой прибыли организации от разработки программного продукта (сайта), сведём в таблицу 34.

Таблица 34 – Расчёт чистой прибыли организации от разработки программного продукта (сайта).

Наименование статей	Сумма, руб.
Себестоимость работ по проекту	116312,74
Планируемая сумма прибыли	17446,91
Выручка от реализации проекта	133759,65
Сумма единого налога, подлежащая уплате в бюджет	4012,00
Прибыль, остающаяся в распоряжении организации, после уплаты единого налога при применении УСН	133759,65

## 7.7 Расчет стоимости владения программным продуктом

Стоимость владения программным продуктом включает в себя:

- 1) Эксплуатационные расходы на обслуживание разработанного программного продукта измеряются в часах затраченных на указанные в таблице 35 виды работ.
- 2) Ежемесячные затраты на функционирование программного продукта представлены в таблице 36.

Таблица 35 – Расчет эксплуатационных расходов

Виды затрат	Специалист	Количество часов, час/мес.	Стоимость трудозатрат в час, руб.	Сумма, руб./месяц
Эксплуатационные расходы, (Эр): – обновление информации в БД; – обновление ссылок; – обновление информационного наполнения.	back-end	4	303,18	1212,72
	front-end	2	303,18	606,39
	front-end	3	303,18	909,54
Итого		9	-	2728,62

Таблица 36– Расчет затрат на функционирование

Виды затрат	Специалист	Сумма, руб./месяц
Затраты на функционирование, (Зф): – аренда хостинга, руб./месяц; – стоимость домена;	back-end	440
	back-end	32,5
Итого		472,50

Итого, стоимость владения (Св) рассчитывается как сумма эксплуатационных расходов и затрат на функционирование по формуле:

$$Св = Эр + Зф, \quad (17)$$

где

Св – стоимость владения программным продуктом (сайтом), руб.;

Эр – сумма эксплуатационных расходов, руб

Зф – сумма затрат на функционирование, руб.

$$Св = 2728,62, + 472,50 = 3201,12 \text{ руб.}$$

Стоимость владения разработанным продуктом, составляет 3201,12 руб. в месяц.

#### 7.9 Смета затрат на проект

Смета затрат на проект включает в себя себестоимость проекта, заложенную сумму прибыли разработчика, сумму налоговых платежей. Отдельной строкой необходимо указать стоимость владения программным продуктом (сайтом). Расчёт сметы представлен в таблице 37.

Таблица 37 – Смета затрат на проект.

№ п/п	Статьи сметы затрат на проект	Сумма, руб.
1	Затраты на создание программного продукта	116312,74
	– Расходы на оплату труда	72551,26
	– Страховые взносы	21910,00
	– Накладные расходы	21851,48
2	Чистая прибыль	13434,91
3	Налог по УСН с выручки (расчёт в п. 1.5 проекта)	4012,00
ИТОГО общая стоимость работ по проекту		133759,65
4	Стоимость владения программным продуктом, месяц	3201,12

## 8 ТБ и ОТ

### 8.1 Требования охраны труда перед началом работы

Перед началом работы необходимо обеспечить выполнение всех требований охраны труда, предусмотренных нормативными документами и внутренними инструкциями организации. Все сотрудники должны пройти инструктаж по охране труда, ознакомиться с рабочими инструкциями и получить необходимые средства индивидуальной защиты (СИЗ). Рабочее место должно быть организовано в соответствии с правилами эргономики и безопасности, исключая возможность возникновения травмоопасных ситуаций.

Важно проверить исправность оборудования и инструментов, а также их соответствие установленным требованиям. Рабочие помещения должны быть обеспечены достаточным уровнем освещения и вентиляции, а также оборудованы первичными средствами пожаротушения. Особое внимание следует уделить наличию и исправности аварийных выходов и путей эвакуации.

Сотрудники должны быть ознакомлены с планом действий в аварийных ситуациях, включая порядок эвакуации, использование первичных средств пожаротушения и порядок оказания первой помощи.

### 8.2 Охрана труда во время работы

Во время работы необходимо строго соблюдать все предписания по охране труда и технике безопасности. Сотрудники обязаны использовать СИЗ, предоставленные работодателем, и выполнять работы в соответствии с установленными инструкциями и нормами. Важно следить за соблюдением санитарно-гигиенических норм на рабочем месте, обеспечивая чистоту и порядок.

При работе с электрическим оборудованием и электроинструментами необходимо соблюдать меры предосторожности, избегая контакта с проводами под напряжением и проводящими материалами. В случае обнаружения неисправностей оборудования или нарушений в работе системы электроснабжения следует немедленно сообщить об этом ответственному лицу и прекратить работу до устранения неисправностей.

Особое внимание необходимо уделять соблюдению правил пожарной безопасности. Запрещается курить и использовать открытый огонь в неустановленных местах. Рабочие помещения должны быть оборудованы первичными средствами пожаротушения, доступ к которым должен быть свободным и незагроможденным.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

### 8.3 Действия в аварийных ситуациях

В случае возникновения аварийной ситуации или несчастного случая необходимо действовать быстро и четко, следуя установленному плану действий. Прежде всего, следует немедленно прекратить работу и оповестить всех сотрудников о возникновении опасности. Необходимо принять меры по устранению источника опасности, если это возможно без риска для жизни и здоровья.

В случае возникновения пожара необходимо немедленно сообщить об этом в пожарную службу, используя телефонные номера экстренных служб или внутреннюю систему оповещения. Если пожар только начался и его можно локализовать первичными средствами пожаротушения, необходимо воспользоваться огнетушителями или другими средствами для тушения пожара. Важно помнить, что при работе с огнетушителями следует соблюдать осторожность, избегая вдыхания дыма и продуктов горения.

Если пожар невозможно локализовать, необходимо немедленно организовать эвакуацию сотрудников согласно плану эвакуации. Во время эвакуации следует сохранять спокойствие, избегать паники и действовать в соответствии с указаниями ответственных лиц.

В случае обнаружения пострадавших необходимо оказать им первую помощь и вызвать медицинскую службу. Если пострадавший находится в сознании, важно успокоить его и обеспечить комфортные условия до прибытия медицинской помощи. Если пострадавший находится без сознания, следует проверить его дыхание и пульс, при необходимости провести реанимационные мероприятия (искусственное дыхание, непрямой массаж сердца).

После ликвидации аварийной ситуации необходимо провести детальный анализ причин и обстоятельств произошедшего, разработать меры по предотвращению подобных инцидентов в будущем и внести соответствующие изменения в инструкции по охране труда и технике безопасности.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

## Заключение

В процессе выполнения дипломной работы была разработана веб-платформа для частной школы программирования и робототехники для детей, которая автоматизирует учебные и административные процессы школы. Разработка сайта включала несколько ключевых этапов: проектирование архитектуры, создание пользовательского интерфейса, реализация функционала и обеспечение удобства использования для различных ролей пользователей, включая учащихся, преподавателей, родителей и администраторов.

Одной из главных задач было создание интуитивно понятного интерфейса, который позволял бы пользователям легко ориентироваться на сайте, записываться на курсы, отслеживать расписание занятий и взаимодействовать с образовательными материалами. Важным аспектом также стало обеспечение безопасности и конфиденциальности данных пользователей. Для этого были реализованы функции безопасного хранения и обработки данных, что обеспечивает надежность и защиту информации.

Проект демонстрирует преимущества использования современных технологий веб-разработки, включая Django и Django REST framework, что позволяет обеспечить высокую производительность, гибкость и адаптивность интерфейса. Эти технологии делают сайт доступным на различных устройствах, обеспечивая комфортное использование как на компьютерах, так и на мобильных устройствах. Благодаря использованию структурированных библиотек и фреймворков удалось создать поддерживаемый и масштабируемый код, что существенно упрощает дальнейшее развитие проекта.

Созданный сайт не только автоматизирует административные процессы и улучшает взаимодействие между всеми участниками учебного процесса, но и служит примером эффективной организации и реализации веб-приложений. Платформа стала ценным опытом в области веб-разработки и доказала, что с помощью современных технологий можно создавать высококачественные и функциональные веб-приложения, отвечающие потребностям пользователей.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		



## Список используемых источников

1. Пузыревский И.А., Правила оформления текстовых документов в учебном процессе – Ростовский-на-Дону государственный колледж связи и информатики, 2006.
2. Коммер К. "SQLAlchemy: Database Access Using Python" – 2-е изд. – O'Reilly Media, 2019.
3. Гринфельд Д., Рой У. "Two Scoops of Django 3.x: Best Practices for the Django Web Framework" – 3-е изд. – Two Scoops Press, 2020.
4. [Электронный ресурс]. – Официальная документация Django – Режим доступа: <https://docs.djangoproject.com/en/3.2/>. – Загл. с экрана.
5. [Электронный ресурс]. – Официальная документация Django REST Framework – Режим доступа: <https://www.django-rest-framework.org/>. – Загл. с экрана.
6. [Электронный ресурс]. – Официальная документация Python – Режим доступа: <https://docs.python.org/3/>. – Загл. с экрана.
7. [Электронный ресурс]. – Официальная документация MySQL – Режим доступа: <https://dev.mysql.com/doc/>. – Загл. с экрана.
8. [Электронный ресурс]. – Официальная документация Docker – Режим доступа: <https://docs.docker.com/>. – Загл. с экрана.
9. [Электронный ресурс]. – Официальная документация Swagger – Режим доступа: <https://swagger.io/docs/>. – Загл. с экрана.
10. [Электронный ресурс]. – Официальная документация GitHub – Режим доступа: <https://docs.github.com/en>. – Загл. с экрана.
11. [Электронный ресурс]. – Официальная документация Postman – Режим доступа: <https://learning.postman.com/docs/getting-started/introduction/>. – Загл. с экрана.
12. [Электронный ресурс]. – Руководство по Pycharm – Режим доступа: <https://www.jetbrains.com/pycharm/guide/>. – Загл. с экрана.
13. [Электронный ресурс]. – Учебник по Django – Режим доступа: <https://tutorial.djangogirls.org/ru/>. – Загл. с экрана.
14. [Электронный ресурс]. – Документация OpenServer – Режим доступа: <https://ospanel.io/docs/>. – Загл. с экрана.
15. Гас Г. [Секреты JavaScript]. – Питер, 2018. – Режим доступа: <https://www.piter.com/JavaScriptSecrets>.
16. Пауэлл Т. [HTML и CSS. Исчерпывающее руководство]. – 2-е изд. – Символ-Плюс, 2015. – Режим доступа: <https://www.symbolplus.ru/HTMLCSSComprehensive>.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

## Front-end

## Index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Платформа для оптимизации учебного процесса</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://api-maps.yandex.ru/2.1/?apikey=3f4b1058-72f7-4dc3-9c29-2fd9aab905cf&lang=ru_RU" type="text/javascript"></script>

  <script type="text/javascript">
    ymaps.ready(init);
    function init() {
      var myMap = new ymaps.Map("map", {
        center: [47.230686, 39.739772],
        zoom: 10
      });

      var myPlacemark = new ymaps.Placemark([47.230686, 39.739772], {
        balloonContent: 'Это точка на карте'
      });

      myMap.geoObjects.add(myPlacemark);
    }
  </script>
</head>

<body>
  <header>
    <div class="container">
      <nav>
        <h1>Кодозавр</h1>
        <div class="burger-menu">
          <div class="burger-icon" id="burgerIcon">&#9776;</div>
          <div class="burger-dropdown" id="burgerDropdown">
            <ul class="menu" id="menu">
              <li><a href="#" id="loginBtn">Вход</a></li>
              <li><a href="#" id="cours">Курсы</a></li>
            </ul>
          </div>
        </div>
      </nav>
    </div>
  </header>
</body>

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <li><a href="#">Контакты</a></li>
    </ul>
</div>
</div>
</nav>
</div>
</header>

<section id="about">
    <div class="container">
        <h2>Добро пожаловать!</h2>
        <p>Мы рады приветствовать тебя на нашей платформе, созданной
для оптимизации учебного процесса и повышения твоей продуктивности. Здесь
ты сможешь записаться на курсы, отслеживать свой прогресс и управлять
расписанием занятий.</p>
    </div>
</section>

<section id="branches2">
    <div class="container2">
        <h2>Наши направления</h2>
        <div class="branches-main">
            <div class="column">
                <div class="branch2">
                    <h3>Робототехника</h3>
                </div>
                <div class="sub-branches">
                    <div class="sub-branch">
                        <h3 id="course-name">Arduino Uno для детей от 10
лет</h3>
                        <p>В этом курсе дети будут знакомиться с миром
электроники и программирования, используя платформу Arduino Uno. Они
научатся подключать различные датчики, моторы и другие устройства к
Arduino, а затем программировать их для создания разнообразных умных
устройств и роботов.</p>
                        <p><b>Возраст:</b> от 10 лет</p>
                        <p><b>Цена:</b> 1000p</p>
                        <div class="button-container">
                            <button onclick="showFeedbackForm('Arduino Uno
для детей от 10 лет')">Записаться на курс</button>
                        </div>
                    </div>
                </div>
            </div>
            <div class="sub-branch">
                <h3>Робототехника для детей 4-6 лет</h3>
            </div>
        </div>
    </div>
</section>

```

<p>Этот курс разработан специально для самых маленьких учеников, чтобы ввести их в мир робототехники и программирования. Дети будут использовать простые роботы и конструкторы для освоения основных концепций, таких как движение, сенсоры и последовательность команд.</p>

<p><b>Возраст:</b> от 4 лет</p>

<p><b>Цена:</b> 2000p</p>

<div class="button-container">

<button onclick="showFeedbackForm('Робототехника для детей 4-6 лет')">Записаться на курс</button>

</div>

</div>

<div class="sub-branch">

<h3>Робототехника для детей старше 7 лет</h3>

<p>Курс робототехники для детей от 7 лет - увлекательное погружение в мир технологий. Дети изучают механику, электронику и программирование, создавая и управляя роботами. Уроки структурированы для постепенного развития навыков, используют интерактивные методы и поддержку инструкторов.</p>

<p><b>Возраст:</b> от 7 лет</p>

<p><b>Цена:</b> 2500p</p>

<div class="button-container">

<button onclick="showFeedbackForm('Робототехника для детей старше 7 лет')">Записаться на курс</button>

</div>

</div>

</div>

</div>

<div class="column">

<div class="branch2">

<h3>Программирование</h3>

</div>

<div class="sub-branches">

<div class="sub-branch">

<h3>Блочное программирование для детей от 6 лет</h3>

<p>В этом курсе дети используют графические блоки, чтобы создавать простые программы и игры. Блочное программирование вводит основные концепции логики и последовательности действий, помогая детям развивать свои навыки в области решения задач и логического мышления.</p>

<p><b>Возраст:</b> от 6 лет</p>

<p><b>Цена:</b> 2500p</p>

<div class="button-container">

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <button onclick="showFeedbackForm('Блочное
программирование для детей от 6 лет')">Записаться на курс</button>
    </div>
</div>
<div class="sub-branch">
    <h3>Программирование на Python для
начинающих</h3>
    <p>Этот курс призван ввести детей в мир
программирования с использованием языка Python. Учащиеся узнают основные
концепции программирования, такие как переменные, условные операторы и
циклы, и начнут создавать свои собственные программы, начиная с простых
консольных приложений.</p>
    <p><b>Возраст:</b> от 7 лет</p>
    <p><b>Цена:</b> 1500p</p>
    <div class="button-container">
        <button
onclick="showFeedbackForm('Программирование на Python для
начинающих')">Записаться на курс</button>
    </div>
</div>
<div class="sub-branch">
    <h3>Программирование на Python для
продвинутых</h3>
    <p>Этот курс призван ввести детей в мир
программирования с использованием языка Python. Учащиеся узнают основные
концепции программирования, такие как переменные, условные операторы и
циклы, и начнут создавать свои собственные программы, начиная с простых
консольных приложений. Этот курс предназначен для детей, уже знакомых с
основами Python или имеющих некоторый опыт программирования. В нем
будут рассматриваться более сложные концепции. Дети также начнут работать
над проектами, включающими создание веб-приложений, игр или даже
машинного обучения.</p>
    <p><b>Возраст:</b> от 10 лет</p>
    <p><b>Цена:</b> 2500p</p>
    <div class="button-container">
        <button
onclick="showFeedbackForm('Программирование на Python для
продвинутых')">Записаться на курс</button>
    </div>
</div>
<div class="sub-branch">
    <h3>Программирование на JavaScript для
начинающих</h3>
    <p>Этот курс предлагает введение в
программирование с использованием языка JavaScript, который широко

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

используется для создания интерактивных веб-сайтов и приложений. Дети изучат основы JavaScript, такие как переменные, операторы и функции, и начнут создавать простые веб-приложения и игры.</p>

<p><b>Возраст:</b> от 7 лет</p>

<p><b>Цена:</b> 2000p</p>

<div class="button-container">

<button

onclick="showFeedbackForm('Программирование на JavaScript для начинающих')">Записаться на курс</button>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<section id="mapSection">

<div class="container">

<h2>Наше местоположение</h2>

<div id="map" style="width: 100%; height: 400px;"></div>

</div>

</section>

<footer>

<div class="container">

<div class="footer-info">

<div class="social-media">

<a href="https://vk.com/startjunior" target="\_blank">VKontakte</a>

<a

href="https://www.youtube.com/channel/UCDBBxFZ\_exYNyAxQiTwbVcA" target="\_blank">YouTube</a>

<a href="https://www.instagram.com/sj\_kazan/"

target="\_blank">Instagram</a>

</div>

<div class="contact-info">

<a href="http://www.startjunior.ru/" target="\_blank">Start Junior</a>

<p>Телефон: +7 (123) 456-78-90</p>

<p>Электронная почта: startjunior56@gmail.com</p>

</div>

</div>

</div>

</footer>

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		64

```

<div id="loginModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    
    <div class="tab">
      <button class="tablinks" id="defaultOpen" onclick="openTab(event,
'Login')">Войти</button>
      <button class="tablinks" onclick="openTab(event,
'Register')">Регистрация</button>
    </div>
    <div id="Login" class="tabcontent">
      <form>
        <label for="username">Имя пользователя</label>
        <input type="text" id="username" name="username" required>
        <label for="password">Пароль</label>
        <input type="password" id="password" name="password"
required>
        <label>
          <input type="checkbox"
onclick="togglePasswordVisibility('password')"> Показать пароль
        </label>
        <button type="submit">Войти</button>
      </form>
    </div>
    <div id="Register" class="tabcontent">
      <form>
        <label for="newUsername">Имя пользователя</label>
        <input type="text" id="newUsername" name="newUsername"
required>
        <label for="newPassword">Пароль</label>
        <input type="password" id="newPassword"
name="newPassword" required>
        <label for="confirmPassword">Подтвердите пароль</label>
        <input type="password" id="confirmPassword"
name="confirmPassword" required>
        <label>
          <input type="checkbox"
onclick="togglePasswordVisibility('newPassword', 'confirmPassword')"> Показать
пароли
        </label>
        <button type="submit">Зарегистрироваться</button>
      </form>
    </div>
  </div>
</div>

```

```

</div>

<div id="feedbackModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    
    <div id="feedback-form" class="feedback-form">
      <h3 id="course-title">Форма обратной связи</h3>
      <form id="feedbackForm">
        <label for="name">Имя:</label>
        <input type="text" id="name" name="name">
        <div id="name-error" class="error-message"></div>

        <label for="surname">Фамилия:</label>
        <input type="text" id="surname" name="surname">
        <div id="surname-error" class="error-message"></div>

        <label for="age">Возраст ребенка:</label>
        <input type="number" id="age" name="age">
        <div id="age-error" class="error-message"></div>

        <label for="phone">Телефон:</label>
        <input type="text" id="phone" name="phone">
        <div id="phone-error" class="error-message"></div>

        <div class="button-container">
          <button type="button"
onclick="submitForm()">Отправить</button>
        </div>
      </form>
    </div>
  </div>
</div>
<script src="script.js"></script>
</body>
</html>

```

Style.css

```

/* Стили для главной стр */
html {
  scroll-behavior: smooth;
}
.adaptive-img-v {
  position: absolute;

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		



```

    left: 0;
    top: 0;
    width: 100%;
    height: auto;
}

.adaptive-img-fon {
    position: absolute;
    width: 100%;
    height: auto;
}

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 1200px;
    grid-template-columns: repeat(2, 1fr);
    margin: 0 auto;
    padding: 0 20px;
}

header {
    background-color: #ffc736;
    color: #fff;
    padding: 20px 0;
}

/* Стилизация для десктопной версии */
.menu {
    list-style: none;
    padding: 0;
}

.menu li {
    display: inline-block;
    margin-right: 20px;
}

.menu li:last-child {
    margin-right: 0;
}

.menu li a {

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

```

    color: #fff;
    text-decoration: none;
}

.burger-icon {
    display: none;
}

/* Стилизация для мобильных устройств */
@media screen and (max-width: 768px) {
    .burger-menu {
        display: flex;
        align-items: center;
    }

    .burger-icon {
        display: block;
        font-size: 24px;
        cursor: pointer;
        margin-right: 10px;
    }

    .menu {
        display: none;
        transition: all 0.3s ease;
    }

    .menu.open {
        display: flex !important;
        flex-direction: column;
        align-items: center;
        opacity: 1;
    }

    .menu.open li {
        margin-bottom: 10px;
    }
}

nav {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

```

```

nav h1 {
    margin: 0;
    font-size: 32px;
    color: #fff;
    display: flex;
    align-items: center;
}

nav h1 img {
    margin-right: 10px;
    height: 40px;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    display: flex;
}

nav ul li {
    margin-left: 20px;
}

nav ul li:first-child {
    margin-left: 0;
}

nav ul li a {
    color: #fff;
    text-decoration: none;
    font-size: 18px;
}

section {
    padding: 50px 0;
}

#about, #branches, #cta {
    background-color: #f8f8f8;
    text-align: center;
}

#about h2, #branches h2, #cta h2 {
    margin-bottom: 20px;
}

```

```

}

.container2 {
    max-width: 1200px;
    margin: 0 auto;
    padding: 0 20px;
    text-align: center;
}

h2 {
    margin-bottom: 30px;
}

.branches-main {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
}

.column {
    flex-basis: calc(45% - 10px);
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-bottom: 20px;
}

.branch2 {
    width: 100%;
    padding: 20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: left;
}

.sub-branches {
    width: 100%;
    margin-top: 20px;
}

.sub-branch {
    margin-top: 20px;
    padding: 10px;
    background-color: #ffffff;

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

```

border-radius: 5px;
box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
text-align: left;
}

```

```

/* Адаптивность */
@media (max-width: 768px) {
  .column {
    flex-basis: 100%;
    margin-bottom: 20px;
  }
}

```

```

footer p {
  margin: 0;
}

```

```

.btn {
  display: inline-block;
  padding: 10px 20px;
  background-color: #ffc736;
  color: #fff;
  text-decoration: none;
  border-radius: 5px;
  transition: background-color 0.3s;
}

```

```

/*футер*/
body {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

```

```

footer {
  background-color: #ffc736;
  padding: 20px 0;
  text-align: center;
  border-top: 1px solid #ffc736;
}

```

```

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 15px;
}

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

```

}

.footer-info {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 10px;
}

.footer-info .social-media,
.footer-info .contact-info {
    margin-bottom: 10px;
}

.footer-info a {
    margin: 0 5px;
    color: #333;
    text-decoration: none;
    transition: color 0.3s;
}

.footer-info a:hover {
    color: #ffffff;
}

.footer-info p {
    margin: 5px 0;
    color: #333;
}

/* Стили для личного кабинета */
#profile {
    padding: 20px 0;
    background: #fff;
    margin-bottom: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

#profile .profile-info {
    display: flex;
    align-items: center;
}

#profile .profile-info img {

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

```

border-radius: 50%;
width: 150px;
height: 150px;
margin-right: 20px;
}

#profile .info {
  flex: 1;
}

#profile .info h3 {
  margin-top: 0;
  color: #333;
}

#cours {
  padding: 20px 0;
  border-radius: 5px;
}

#courses .course-list {
  display: flex;
  flex-direction: column;
  gap: 20px;
}

#courses .course-item {
  background: #f4f4f4;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
}

#courses .course-item h3 {
  color: #000000;
}

.progress {
  background: #85FFB6;
  height: 20px;
}

.progress-bar {
  background: #ddd;

```

```

border-radius: 5px;
overflow: hidden;
margin: 10px 0;
}

.continue-btn:hover {
    background: #ffc736;
}

/* Стили для прогресса прохождения курсов */
.progress-bar-inner {
    background-color: #ffc736;
border-radius: 10px;
height: 100%;
}

/* Стили для количества купленных курсов */
.course-count {
    font-weight: bold;
}

.avatar {
    width: 100px;
    height: 100px;
border-radius: 50%;
overflow: hidden;
}

.avatar img {
    width: 100%;
    height: auto;
}

.user-info {
    margin-bottom: 20px;
margin-right: 20px;
}

.progress-info {
    margin-bottom: 20px;
margin-left: 20px;
}

/* Стили для модального окна */
.modal {
    display: none;
position: fixed;

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		



```

    z-index: 1;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    overflow: auto;
    background-color: rgba(0,0,0,0.4);
    animation: fadeIn 0.5s;
}

@keyframes fadeIn {
    from {opacity: 0;}
    to {opacity: 1;}
}

.modal-content {
    background-color: #fefefe;
    margin: 10% auto;
    padding: 20px;
    border-radius: 10px;
    border: none;
    width: 80%;
    max-width: 400px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    animation: slideDown 0.5s;
    text-align: center;
}

@keyframes slideDown {
    from {transform: translateY(-50px);}
    to {transform: translateY(0);}
}

.close {
    color: #ffc736;
    float: right;
    font-size: 28px;
    font-weight: bold;
    margin-right: -10px;
}

.close:hover,
.close:focus {
    color: black;
    text-decoration: none;
}

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		

```

        cursor: pointer;
    }

    form {
        display: flex;
        flex-direction: column;
        align-items: flex-start;
    }

    label {
        margin: 10px 0 5px;
        font-weight: bold;
    }

    input[type="text"],
    input[type="password"],
    input[type="number"] {
        padding: 10px;
        margin-bottom: 5px;
        border: none;
        border-bottom: 2px solid #ccc;
        border-radius: 0;
        font-size: 16px;
        width: 100%;
        box-sizing: border-box;
    }

    input[type="text"]:focus,
    input[type="password"]:focus,
    input[type="number"]:focus {
        outline: none;
        border-bottom: 2px solid #ffc736;
    }

    .button-container {
        display: flex;
        justify-content: center;
        margin-top: 20px;
    }

    button {
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        background-color: #85FFB6;
    }

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						76
Изм.	Лист	№ докум.	Подпись	Дата		

```

        color: rgb(0, 0, 0);
        font-size: 16px;
        cursor: pointer;
        transition: background-color 0.3s;
    }

    button:hover {
        background-color: #ffb400;
    }

    .logo {
        width: 100px;
        margin-bottom: 20px;
    }

    .error-message {
        color: red;
        font-size: 12px;
    }

    /* Стили для вкладок */
    .tab button {
        background-color: #20190725;
        border: none;
        outline: none;
        cursor: pointer;
        padding: 10px 20px;
        transition: 0.3s;
        font-size: 14px;
        margin: 0 2px;
        border-radius: 5px 5px 0 0;
    }

    .tab button:hover {
        background-color: #ffc736;
    }

    .tab button.active {
        background-color: #ffc736;
    }

    .tabcontent {
        display: none;
        padding: 20px;
        border-top: none;

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

```

}

/*календарь*/
#calendar-container {
    background-color: #ffffff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
    overflow: hidden;
}

#calendar-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 10px;
    background-color: #ffc736;
    color: #ffffff;
}

#calendar-header h2 {
    margin: 0;
}

#calendar-header button {
    background-color: #ffffff;
    color: #000000;
    border: none;
    padding: 5px 10px;
    cursor: pointer;
    border-radius: 5px;
}

#calendar {
    width: 100%;
    border-collapse: collapse;
}

#calendar th, #calendar td {
    width: 14.28%;
    text-align: center;
    padding: 10px;
    border: 1px solid #ddd;
}

#calendar td {

```

```

    cursor: pointer;
}

#calendar .today {
    background-color: #f0f0f0;
}

#calendar .event {
    background-color: #ffc736;
}

#event-form-container {
    margin-top: 20px;
    padding: 10px;
    background-color: #ffffff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
}

#event-form {
    display: flex;
    flex-direction: column;
}

#event-form label {
    margin-bottom: 5px;
}

#event-form input {
    margin-bottom: 10px;
    padding: 5px;
}

#event-form button {
    padding: 5px 10px;
    background-color: #ffc736;
    color: #ffffff;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}

#event-form button:hover {
    background-color: #ffc736;
}

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						79
Изм.	Лист	№ докум.	Подпись	Дата		

```

@media (max-width: 768px) {
  #schedule-container {
    flex-direction: column;
  }

  #calendar-container {
    margin-right: 0;
    margin-bottom: 20px;
  }

  #event-form-container {
    max-width: 100%;
  }
}

@media (max-width: 480px) {
  #calendar-header {
    flex-direction: column;
    text-align: center;
  }

  #calendar-header button {
    margin: 5px 0;
  }
}

#course-calendar-container {
  background-color: #ffffff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  border-radius: 10px;
  overflow: hidden;
  margin-top: 20px;
}

#course-calendar-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px;
  background-color: #ffc736;
  color: #ffffff;
}

#course-calendar-header h2 {
  margin: 0;

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подпись	Дата		

```

}

#course-calendar-header button {
    background-color: #ffffff;
    color: #000000;
    border: none;
    padding: 5px 10px;
    cursor: pointer;
    border-radius: 5px;
}

#course-calendar {
    width: 100%;
    border-collapse: collapse;
}

#course-calendar th, #course-calendar td {
    width: 14.28%;
    text-align: center;
    padding: 10px;
    border: 1px solid #ddd;
}

#course-calendar td {
    cursor: pointer;
}

#course-calendar .today {
    background-color: #f0f0f0;
}

#course-calendar .course-day {
    background-color: #ffc736;
}

```

Script.js

```

const burgerIcon = document.getElementById('burgerIcon');
const menu = document.getElementById('menu');

// Добавление обработчика события на клик по иконке бургер-меню
document.addEventListener('DOMContentLoaded', () => {
    const burgerIcon = document.getElementById('burgerIcon');
    if (burgerIcon) {

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

```

const menu = document.getElementById('menu');
burgerIcon.addEventListener('click', () => {
    menu.classList.toggle('open');
});
}
});

```

/\*календарь учит\*/

```

document.addEventListener('DOMContentLoaded', () => {
    const events = new Set(['2023-06-08', '2023-06-15', '2023-06-22']);

    const calendar = document.getElementById('calendar-body');
    const monthYear = document.getElementById('month-year');
    const prevMonth = document.getElementById('prev-month');
    const nextMonth = document.getElementById('next-month');
    const eventForm = document.getElementById('event-form');
    const eventDateInput = document.getElementById('event-date');

    let currentMonth = new Date().getMonth();
    let currentYear = new Date().getFullYear();

    const renderCalendar = (month, year) => {
        calendar.innerHTML = "";
        const firstDay = (new Date(year, month, 1).getDay() + 6) % 7;
        const daysInMonth = new Date(year, month + 1, 0).getDate();

        monthYear.textContent = new Date(year, month).toLocaleString('ru-RU',
        { month: 'long', year: 'numeric' });

        let date = 1;
        let rowCount = Math.ceil((firstDay + daysInMonth) / 7);
        for (let i = 0; i < rowCount; i++) {
            const row = document.createElement('tr');

            for (let j = 0; j < 7; j++) {
                const cell = document.createElement('td');

                if (i === 0 && j < firstDay) {
                    cell.innerHTML = "";
                } else if (date > daysInMonth) {
                    cell.innerHTML = "";
                } else {
                    const cellDate = new Date(year, month,
date).toISOString().split('T')[0];

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						82
Изм.	Лист	№ докум.	Подпись	Дата		



```

        cell.innerHTML = date;

        if (events.has(cellDate)) {
            cell.classList.add('event');
        }

        if (cellDate === new Date().toISOString().split('T')[0]) {
            cell.classList.add('today');
        }

        date++;
    }

    row.appendChild(cell);
}

calendar.appendChild(row);
}
};

const updateEvents = (event) => {
    event.preventDefault();
    const newEventDate = eventDateInput.value;
    if (newEventDate && !events.has(newEventDate)) {
        events.add(newEventDate);
        renderCalendar(currentMonth, currentYear);
    }
    eventDateInput.value = "";
};

prevMonth.addEventListener('click', () => {
    currentMonth--;
    if (currentMonth < 0) {
        currentMonth = 11;
        currentYear--;
    }
    renderCalendar(currentMonth, currentYear);
});

nextMonth.addEventListener('click', () => {
    currentMonth++;
    if (currentMonth > 11) {
        currentMonth = 0;
        currentYear++;
    }
}

```

```

        renderCalendar(currentMonth, currentYear);
    });

    eventForm.addEventListener('submit', updateEvents);

    renderCalendar(currentMonth, currentYear);
});

/*обычн календарь*/
document.addEventListener('DOMContentLoaded', () => {
    const courseDays = [2, 5, 10, 15, 20];

    const calendar = document.getElementById('course-calendar-body');
    const monthYear = document.getElementById('course-month-year');
    const prevMonth = document.getElementById('course-prev-month');
    const nextMonth = document.getElementById('course-next-month');

    let currentMonth = new Date().getMonth();
    let currentYear = new Date().getFullYear();

    const renderCalendar = (month, year) => {
        calendar.innerHTML = "";
        const firstDay = (new Date(year, month, 1).getDay() + 6) % 7;
        const daysInMonth = new Date(year, month + 1, 0).getDate();

        monthYear.textContent = new Date(year, month).toLocaleString('ru-RU',
        { month: 'long', year: 'numeric' });

        let date = 1;
        let rowCount = Math.ceil((firstDay + daysInMonth) / 7);
        for (let i = 0; i < rowCount; i++) {
            const row = document.createElement('tr');

            for (let j = 0; j < 7; j++) {
                const cell = document.createElement('td');

                if (i === 0 && j < firstDay) {
                    cell.innerHTML = "";
                } else if (date > daysInMonth) {
                    cell.innerHTML = "";
                } else {
                    const cellDate = new Date(year, month,
                    date).toISOString().split('T')[0];
                    cell.innerHTML = date;
                }
            }
        }
    }

```

```

        if (courseDays.includes(date)) {
            cell.classList.add('course-day');
        }

        date++;
    }

    row.appendChild(cell);
}

calendar.appendChild(row);
}
};

prevMonth.addEventListener('click', () => {
    currentMonth--;
    if (currentMonth < 0) {
        currentMonth = 11;
        currentYear--;
    }
    renderCalendar(currentMonth, currentYear);
});

nextMonth.addEventListener('click', () => {
    currentMonth++;
    if (currentMonth > 11) {
        currentMonth = 0;
        currentYear++;
    }
    renderCalendar(currentMonth, currentYear);
});

renderCalendar(currentMonth, currentYear);
});

/*скроллинг к секции "Курсы"*/
document.getElementById("cours").addEventListener("click", function() {
    document.getElementById("branches2").scrollIntoView({ behavior: 'smooth' });
});

/*окно обр связи */
var feedbackModal = document.getElementById("feedbackModal");
var spanFeedback = document.getElementsByClassName("close")[1];

```

```

spanFeedback.onclick = function() {
    clearForm();
    feedbackModal.style.display = "none";
}

window.onclick = function(event) {
    if (event.target == feedbackModal) {
        clearForm();
        feedbackModal.style.display = "none";
    }
}

function openModal(courseName) {
    var courseTitle = document.getElementById("course-title");
    courseTitle.textContent = "Запись на курс: " + courseName;
    feedbackModal.style.display = "block";
}

function showFeedbackForm(courseName) {
    openModal(courseName);
}

function submitForm() {
    var courseName = document.getElementById("course-name").value;
    var name = document.getElementById("name").value;
    var surname = document.getElementById("surname").value;
    var age = document.getElementById("age").value;
    var phone = document.getElementById("phone").value;

    var nameError = document.getElementById("name-error");
    var surnameError = document.getElementById("surname-error");
    var ageError = document.getElementById("age-error");
    var phoneError = document.getElementById("phone-error");

    /*очистка предыдущих сообщений об ошибках*/
    nameError.textContent = "";
    surnameError.textContent = "";
    ageError.textContent = "";
    phoneError.textContent = "";

    var isValid = true;

    /*Валидация имени (только буквы)*/
    var namePattern = /^[a-zA-Zа-яА-ЯёЁ]+$/;
    if (!name) {

```

```

        nameError.textContent = "Обязательно к заполнению";
        isValid = false;
    } else if (!name.match(namePattern)) {
        nameError.textContent = "Имя должно содержать только буквы";
        isValid = false;
    }

    /*Валидация фамилии (только буквы)*/
    if (!surname) {
        surnameError.textContent = "Обязательно к заполнению";
        isValid = false;
    } else if (!surname.match(namePattern)) {
        surnameError.textContent = "Фамилия должна содержать только буквы";
        isValid = false;
    }

    /*Валидация возраста*/
    if (!age) {
        ageError.textContent = "Обязательно к заполнению";
        isValid = false;
    } else if (isNaN(age) || age <= 0) {
        ageError.textContent = "Возраст должен быть положительным числом";
        isValid = false;
    }

    /*Валидация телефона*/
    var phonePattern = /^+\?d{10,15}$/;
    if (!phone) {
        phoneError.textContent = "Обязательно к заполнению";
        isValid = false;
    } else if (!phone.match(phonePattern)) {
        phoneError.textContent = "Некорректный формат номера телефона";
        isValid = false;
    }

    if (!isValid) {
        return;
    }

    var message = "Имя: " + name + "\n";
    message += "Фамилия: " + surname + "\n";
    message += "Возраст ребенка: " + age + "\n";
    message += "Телефон: " + phone;

```

```

    var whatsappLink = "https://wa.me/+79515022195/?text=" +
    encodeURIComponent("Здравствуйте! Хочу получить подробную
    информацию.\n" + message);
    window.open(whatsappLink);

    closeFeedbackModal();
}

function closeFeedbackModal() {
    clearForm();
    feedbackModal.style.display = "none";
}

function clearForm() {
    document.getElementById("course-name").value = "";
    document.getElementById("name").value = "";
    document.getElementById("surname").value = "";
    document.getElementById("age").value = "";
    document.getElementById("phone").value = "";

    document.getElementById("name-error").textContent = "";
    document.getElementById("surname-error").textContent = "";
    document.getElementById("age-error").textContent = "";
    document.getElementById("phone-error").textContent = "";
}

/*окно авториз и регистр*/
var loginModal = document.getElementById("loginModal");
var btnLogin = document.getElementById("loginBtn");
var spanLogin = document.getElementsByClassName("close")[0];

btnLogin.onclick = function() {
    loginModal.style.display = "block";
}

spanLogin.onclick = function() {
    loginModal.style.display = "none";
}

window.onclick = function(event) {
    if (event.target == loginModal) {
        loginModal.style.display = "none";
    }
}

```

```

function openTab(evt, tabName) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace(" active", "");
    }
    document.getElementById(tabName).style.display = "block";
    evt.currentTarget.className += " active";
}

document.getElementById("defaultOpen").click();

function togglePasswordVisibility(...ids) {
    ids.forEach(id => {
        var input = document.getElementById(id);
        if (input.type === "password") {
            input.type = "text";
        } else {
            input.type = "password";
        }
    });
}

```

Back-end

```
# course/models.py
```

```

from django.db import models
from users.models import Parent, Teacher, Student

```

```

class Course(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    duration = models.PositiveIntegerField(help_text="Duration in hours")
    cost = models.DecimalField(max_digits=10, decimal_places=2)
    level = models.CharField(max_length=50)

    def __str__(self):
        return self.name

```

```

class CourseApplication(models.Model):
    STATUS_CHOICES = (
        ('pending', 'Pending'),
        ('approved', 'Approved'),
        ('rejected', 'Rejected'),
    )
    date = models.DateField(auto_now_add=True)
    time = models.TimeField(auto_now_add=True)
    status = models.CharField(max_length=10, choices=STATUS_CHOICES,
default='pending')
    parent = models.ForeignKey(Parent, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    parent_phone = models.CharField(max_length=15)
    parent_email = models.EmailField()

    def __str__(self):
        return f'{self.parent.user.username} - {self.course.name}'

class FeedbackForm(models.Model):
    date = models.DateField(auto_now_add=True)
    time = models.TimeField(auto_now_add=True)
    description = models.TextField()

    def __str__(self):
        return f'Feedback on {self.date} at {self.time}'

class Topic(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    material_links = models.TextField(blank=True, help_text="Comma separated
links")
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    materials = models.FileField(upload_to='materials/', blank=True, null=True)

    def __str__(self):
        return self.name

class Lesson(models.Model):
    date = models.DateField()
    start_time = models.TimeField()
    end_time = models.TimeField()
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
    topic = models.ForeignKey(Topic, on_delete=models.CASCADE)

    def __str__(self):
        return f'{self.topic.name} - {self.date}'

```

					ПКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						90
Изм.	Лист	№ докум.	Подпись	Дата		



```

# course/signals.py
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.core.mail import send_mail
from .models import FeedbackForm

@receiver(post_save, sender=FeedbackForm)
def send_feedback_notification(sender, instance, created, **kwargs):
    if created:
        send_mail(
            subject='New Feedback Received',
            message=f'New feedback received:\n\n{instance.description}',
            from_email='ovs1ano4ka@yandex.ru',
            recipient_list=['ovs1ano4ka@yandex.ru'],
            fail_silently=False,
        )

# course/admin.py

from django.contrib import admin
from .models import Course, CourseApplication, FeedbackForm, Topic, Lesson

@admin.register(Course)
class CourseAdmin(admin.ModelAdmin):
    list_display = ('name', 'level', 'duration', 'cost')
    search_fields = ('name', 'level')

@admin.register(CourseApplication)
class CourseApplicationAdmin(admin.ModelAdmin):
    list_display = ('parent', 'course', 'date', 'status')
    search_fields = ('parent__user__username', 'course__name')
    list_filter = ('status', 'date')

@admin.register(FeedbackForm)
class FeedbackFormAdmin(admin.ModelAdmin):
    list_display = ('description', 'date', 'time')
    search_fields = ('description',)
    list_filter = ('date',)

@admin.register(Topic)
class TopicAdmin(admin.ModelAdmin):

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						91
Изм.	Лист	№ докум.	Подпись	Дата		

```
list_display = ('name', 'course', 'description')
search_fields = ('name', 'course__name')
```

```
@admin.register(Lesson)
class LessonAdmin(admin.ModelAdmin):
    list_display = ('topic', 'teacher', 'date', 'start_time', 'end_time')
    search_fields = ('topic__name', 'teacher__user__username')
    list_filter = ('date',)
```

```
# users/models.py
from django.contrib.auth.models import AbstractUser, Group, Permission
from django.db import models
from .validators import validate_username

class CustomUser(AbstractUser):
    username = models.CharField(
        max_length=150,
        unique=True,
        validators=[validate_username],
        error_messages={
            'unique': "A user with that username already exists.",
        },
    )
    USER_TYPE_CHOICES = (
        (1, 'parent'),
        (2, 'student'),
        (3, 'teacher'),
    )
    user_type = models.PositiveSmallIntegerField(choices=USER_TYPE_CHOICES)

    groups = models.ManyToManyField(Group, related_name='customuser_set')
    user_permissions = models.ManyToManyField(Permission,
        related_name='customuser_set_permissions')

    def __str__(self):
        return self.username

class Parent(models.Model):
    user = models.OneToOneField(CustomUser, on_delete=models.CASCADE,
        primary_key=True)
    phone = models.CharField(max_length=15)
    email = models.EmailField(unique=True)
```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						92
Изм.	Лист	№ докум.	Подпись	Дата		

```
relation_to_child = models.CharField(max_length=50)
```

```
def __str__(self):  
    return self.user.get_full_name()
```

```
class Student(models.Model):  
    user = models.OneToOneField(CustomUser, on_delete=models.CASCADE,  
primary_key=True)  
    birth_date = models.DateField()  
    email = models.EmailField(blank=True, null=True)  
    parent = models.ForeignKey(Parent, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return self.user.get_full_name()
```

```
class Teacher(models.Model):  
    user = models.OneToOneField(CustomUser, on_delete=models.CASCADE,  
primary_key=True)  
    position = models.CharField(max_length=100)  
    email = models.EmailField(unique=True)  
    phone = models.CharField(max_length=15)  
  
    def __str__(self):  
        return self.user.get_full_name()
```

```
# users/views.py  
from rest_framework import viewsets  
from .models import CustomUser, Parent, Student, Teacher  
from .serializers import CustomUserSerializer, ParentSerializer, StudentSerializer,  
TeacherSerializer
```

```
class CustomUserViewSet(viewsets.ModelViewSet):  
    queryset = CustomUser.objects.all()  
    serializer_class = CustomUserSerializer
```

```
class ParentViewSet(viewsets.ModelViewSet):  
    queryset = Parent.objects.all()  
    serializer_class = ParentSerializer
```

```
class StudentViewSet(viewsets.ModelViewSet):  
    queryset = Student.objects.all()  
    serializer_class = StudentSerializer
```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подпись	Дата		

```

class TeacherViewSet(viewsets.ModelViewSet):
    queryset = Teacher.objects.all()
    serializer_class = TeacherSerializer

# users/admin.py

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import CustomUser, Parent, Student, Teacher
from .forms import CustomUserCreationForm, CustomUserChangeForm,
StudentForm

class CustomUserAdmin(UserAdmin):
    add_form = CustomUserCreationForm
    form = CustomUserChangeForm
    model = CustomUser
    list_display = ['username', 'email', 'user_type', 'is_staff', 'is_active']
    list_filter = ['user_type', 'is_staff', 'is_active']
    fieldsets = (
        (None, {'fields': ('username', 'email', 'password', 'user_type')}),
        ('Permissions', {'fields': ('is_staff', 'is_active')}),
        ('Personal', {'fields': ('first_name', 'last_name')}),
    )
    add_fieldsets = (
        (None, {
            'classes': ('wide',),
            'fields': ('username', 'email', 'password1', 'password2', 'user_type', 'is_staff',
'is_active')}
        ),
    )
    search_fields = ['email', 'username']
    ordering = ['email']

class ParentAdmin(admin.ModelAdmin):
    list_display = ['user_username', 'phone', 'email', 'relation_to_child']

    def user_username(self, obj):
        return obj.user.username

    user_username.short_description = 'Username'

class StudentAdmin(admin.ModelAdmin):

```

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						94
Изм.	Лист	№ докум.	Подпись	Дата		

```

form = StudentForm
list_display = ['user_username', 'birth_date', 'parent_username']

def user_username(self, obj):
    return obj.user.username

user_username.short_description = 'Username'

def parent_username(self, obj):
    return obj.parent.user.get_full_name()

parent_username.short_description = 'Parent Username'

class TeacherAdmin(admin.ModelAdmin):
    list_display = ['user_username', 'position', 'phone', 'email']

    def user_username(self, obj):
        return obj.user.username

    user_username.short_description = 'Username'

admin.site.register(CustomUser, CustomUserAdmin)
admin.site.register(Parent, ParentAdmin)
admin.site.register(Student, StudentAdmin)
admin.site.register(Teacher, TeacherAdmin)

```

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА ВЫПОЛНЕНИЕ РАБОТ  
по созданию информационной системы «платформа для оптимизации  
учебного процесса»

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						96
Изм.	Лист	№ докум.	Подпись	Дата		

## 1 Общие сведения

### 1.1 Краткое описание

Онлайн платформа для автоматизации учебного процесса (далее – ОП) - представляет собой веб-ресурс, цель которого заключается в автоматизации учебных и бизнес-процессов, онлайн-записей на курсы, улучшении взаимодействия между родителями, учениками и преподавателями. Родителям предоставляется возможность записать своих детей на курс, который будет проходить оффлайн в школе. Учителям платформа предоставляет место для хранения материалов по курсам. У ребенка есть возможность отслеживать расписание своих занятий. А администратор управляет наполнением платформы.

### 1.2 Основные функции системы

Функциональность и дизайн системы разрабатываются на основе прототипов, являющихся неотъемлемой частью настоящего технического задания.

Функциональность для всех пользователей системы:

- просмотр главной страницы сайта;
- переходы на ссылки в “шапке” и “подвале” сайта;
- просмотр своих личный кабинетов;
- регистрация;
- вход;
- выход;
- форма обратной связи;
- отслеживание расписания.

Функциональность для преподавателя:

- управление материалами курсов;
- личным кабинетом.

Функциональность для родителя:

- отправление заявки на курс;
- управление личным кабинетом.

Функциональность для администратора:

- управление заявками;
- пользователями;
- курсами;
- настройки платформы.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						97
Изм.	Лист	№ докум.	Подпись	Дата		

Функциональность и дизайн системы разрабатываются на основе прототипов, являющихся неотъемлемой частью настоящего технического задания.

Требования к функциональности, перечень объектов системы (а также их наименования), описанные в настоящем документе, могут быть скорректированы и дополнены в процессе разработки по согласованию сторон.

### 1.3 Требования к дизайну системы

Структура страниц системы определяется дизайном, являющегося неотъемлемой частью настоящего технического задания (<https://www.figma.com/design/nzptcWSoVTxPbz3OAFp7ko/Untitled?node-id=0%3A1&t=d3DPZpH645rlcAGI-1>).

Дизайн страниц системы должен быть адаптивным и корректно отображаться на следующих устройствах и браузерах:

– Персональные компьютеры и ноутбуки: Google Chrome (124.0.6367.203), Яндекс Браузер (24.1.5.803).

– Мобильные телефоны (вертикальное расположение экрана): Safari на iOS (16.5.2), Google Chrome на Android (124.0.6367.171)

### 1.4 Перечень сокращений, терминов и обозначений

Таблица 1 – Перечень сокращений Таблица заполняется исходя из ТЗ

Сокращение	Расшифровка
БД	База данных
ОП	Онлайн платформа
ЛК	Личный кабинет
ПП	Программный продукт
ИС	Информационная система



## 2 Требования к основным функциям системы

### 2.1 Роли пользователей

В ОП требуется предусмотреть следующие роли: родитель, ребенок, учитель, администратор.

### 2.2 Требования к регистрации и авторизации пользователя

#### Регистрация:

Форма регистрации должна содержать следующие поля:

- имя пользователя (обязательно);
- электронная почта (необязательно);
- пароль (обязательно);
- подтверждение пароля (обязательно);
- поля формы должны быть валидированы на стороне клиента и сервера.

#### Валидация полей:

- имя пользователя должно быть уникальным и содержать только алфавитные символы, цифры и знаки подчеркивания;
- электронная почта должна быть в формате email и уникальной;
- пароль должен быть не менее 8 символов, содержать буквы верхнего и нижнего регистра, цифры и специальные символы;
- подтверждение пароля должно совпадать с паролем.

#### Обработка формы:

- при успешной валидации формы данные должны быть отправлены на сервер;
- сервер должен проверить уникальность имени пользователя и электронной почты;
- в случае успешной проверки пользователь должен быть создан в базе данных;
- пользователю должно быть отправлено письмо с подтверждением регистрации на указанный электронный адрес.

#### Подтверждение электронной почты:

- письмо с подтверждением регистрации должно содержать ссылку для активации аккаунта;
- при переходе по ссылке пользователь должен быть перенаправлен на страницу успешной активации аккаунта;
- после активации аккаунта пользователь должен иметь возможность войти в систему.

#### Авторизация пользователя

Форма авторизации должна содержать следующие поля:

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						99
Изм.	Лист	№ докум.	Подпись	Дата		

- имя пользователя или электронная почта;
- пароль (обязательно);
- поля формы должны быть валидированы на стороне клиента и сервера.

Обработка формы:

- при успешной валидации формы данные должны быть отправлены на сервер;
- сервер должен проверить наличие пользователя с указанными именем пользователя или электронной почтой и паролем;
- в случае успешной проверки пользователь должен быть аутентифицирован и перенаправлен в свой личный кабинет.

Управление сессиями:

- после успешной авторизации должна быть создана сессия пользователя;
- сессия должна быть защищена и хранить минимально необходимую информацию для идентификации пользователя.

Сообщения об ошибках:

- в случае некорректного ввода данных, пользователь должен видеть соответствующее сообщение об ошибке (например, "Неверное имя пользователя или пароль").

Восстановление пароля:

- на странице авторизации должна быть ссылка на страницу восстановления пароля;
- на странице восстановления пароля пользователь должен ввести свою электронную почту для получения ссылки на сброс пароля;
- письмо со ссылкой на сброс пароля должно быть отправлено на указанную электронную почту;
- при переходе по ссылке пользователь должен быть перенаправлен на страницу создания нового пароля;
- новый пароль должен быть валидирован аналогично требованиям к паролю при регистрации.

## 2.3 Личный кабинет администратора

Администратор имеет возможность создавать и управлять профилями пользователей, а также назначать роли. В разделе users администратор может:

- создавать и редактировать пользователей;
- назначать пользователям роли: родитель, ученик, преподаватель;
- связывать пользователей по внешним ключам;
- удалять пользователей;

В разделе course applications администратор может обрабатывать заявки от пользователей. Функциональные возможности включают:

- прием заявок;
- изменение статуса заявки;

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подпись	Дата		

- Просмотр информации родителя, отправившего заявку.

Администратор имеет доступ к инструментам для управления курсами. В разделе course администратор может:

- создавать и редактировать курсы;
- создавать и редактировать темы и материалы к курсам;
- создавать и редактировать занятия по курсам.

Администратор имеет возможность видеть обратную связь от пользователей в разделе feedback forms:

- описание проблемы;
- дата обращения;
- время обращения.

## 2.4 Личный кабинет пользователя

ЛК пациента доступен пользователям после авторизации в системе. ЛК пользователя должен позволять выполнять следующий функционал:

С ролью «Родитель»: просмотр купленных курсов и их количество; просмотр расписания; редактирование данных профиля.

С ролью «Учитель»: просмотр материалов курса; удаление, редактирование и добавление новых материалов.

С ролью «Ученик»: просмотр расписания.

## 2.5 Описание основного бизнес-процесса «Регистрация и аутентификация пользователей»

### Бизнес-процесс 1: Регистрация и аутентификация пользователей

Описание основного бизнес-процесса «Регистрация и аутентификация пользователей»

Этот бизнес-процесс описывает действия, необходимые для регистрации новых пользователей на платформе и их последующей аутентификации. Он включает в себя шаги по созданию учетной записи, подтверждению электронной почты и входу в систему.

#### Схема основного бизнес-процесса

1. Пользователь заполняет форму регистрации.
2. Система проверяет введенные данные.
3. Система активирует учетную запись.
4. Пользователь входит в личный кабинет.

#### Участники процесса (роли):

- пользователь (ученик, преподаватель, родитель, администратор);
- система.

#### Описание действий участников

1. Пользователь: заполняет форму регистрации, вводит необходимые данные (имя пользователя, электронная почта, пароль и т.д.).

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		101

Какие сущности ИС используются / создаются в процессе:

- учетная запись пользователя;
- сессия пользователя.

Какие вспомогательные бизнес-процессы связаны с основным:

- восстановление пароля;
- изменение или удаление профиля пользователя;
- управление ролями пользователей.

## 2.6 Описание вспомогательного бизнес-процесса2 «Управление курсами»

### Бизнес-процесс 2: Управление курсами

Этот бизнес-процесс описывает действия по созданию, редактированию и удалению курсов на платформе. Он включает в себя шаги по добавлению информации о курсах, обновлению данных и управлению доступом к материалам курсов.

Схема основного бизнес-процесса

1. Администратор входит в систему.
2. Администратор переходит в раздел управления курсами.
3. Администратор создает новый курс, заполняя форму с информацией о курсе.
4. Система сохраняет данные о новом курсе.
5. Администратор редактирует информацию о существующих курсах при необходимости.
6. Система обновляет данные о курсах.
7. Администратор удаляет курсы при необходимости.
8. Система удаляет данные о курсе.

Участники процесса (роли):

- администратор;
- система.

Какие сущности ИС используются / создаются в процессе:

- курс (название, описание, даты начала и окончания, материалы);
- учетные записи администраторов.

Какие вспомогательные бизнес-процессы связаны с основным:

- управление материалами курсов;
- управление пользователями;
- управление занятиями.

## 2.7 Описание основного бизнес-процесса «Управление занятиями»

### Бизнес-процесс 3: Управление расписанием

Этот бизнес-процесс описывает действия по созданию, редактированию и публикации расписаний занятий на платформе. Он включает в себя шаги по

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						102
Изм.	Лист	№ докум.	Подпись	Дата		

добавлению новых занятий, обновлению информации о занятиях и публикации расписания.

Схема основного бизнес-процесса

1. Администратор входит в систему.
2. Администратор переходит в раздел lessons.
3. Администратор создает новое занятие, заполняя форму с информацией о занятии.
4. Система сохраняет данные о новом занятии.
5. Администратор редактирует информацию о существующих занятиях при необходимости.
6. Система обновляет данные о занятиях.
7. Администратор публикует расписание.
8. Система делает расписание доступным для всех пользователей.

Участники процесса (роли):

- администратор;
- система.

Какие сущности ИС используются / создаются в процессе:

- расписание занятий (даты, время, место, преподаватель);

Какие вспомогательные бизнес-процессы связаны с основным

- управление курсами;
- управление пользователями;

## 2.8 Информационные страницы

В ОП должны быть следующие информационные страницы:

- главная страница;
- личный кабинет.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						103
Изм.	Лист	№ докум.	Подпись	Дата		

### 3 Нефункциональные требования

#### 3.1 Требования к масштабируемости

Система должна поддерживать возможность масштабирования для обработки увеличивающегося числа пользователей и объемов данных. Основные параметры масштабируемости включают:

- горизонтальная масштабируемость: Система должна поддерживать добавление новых серверов для обработки увеличивающейся нагрузки без значительных изменений в архитектуре;
- вертикальная масштабируемость: Система должна позволять увеличивать ресурсы на одном сервере (например, CPU, память) для улучшения производительности.

#### 3.2 Требования к надежности и отказоустойчивости

Система должна быть надежной и устойчивой к сбоям, обеспечивая высокий уровень доступности и возможность восстановления данных. Основные параметры надежности включают:

- доступность;
- резервное копирование данных;
- отказоустойчивость.

#### 3.3 Требования к удобству использования (юзабилити)

Система должна быть интуитивно понятной и удобной для пользователей всех категорий. Основные параметры удобства использования включают:

- пользовательский интерфейс должен быть интуитивно понятным и удобным в использовании для всех категорий пользователей (ученики, преподаватели, родители, администраторы);
- система должна корректно работать на различных устройствах и браузерах (ПК, планшеты, мобильные устройства);
- система должна иметь подробную пользовательскую документацию и инструкции.

#### 3.4 Требования к интеграции

Система должна поддерживать интеграцию с внешними сервисами и системами. Основные параметры интеграции включают:

- система должна предоставлять открытые API для интеграции с внешними системами и сервисами;
- система может быть совместима с основными внешними сервисами, такими как системы оплаты, системы уведомлений и карты. Не критично.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						104
Изм.	Лист	№ докум.	Подпись	Дата		

## 4 Прочие требования к системе

### 4.1 Требования к пользовательским интерфейсам

Требования к пользовательскому интерфейсу Системы:

- пользовательский интерфейс должен быть понятным и удобным, не должен быть перегружен графическими элементами;
- пользовательский интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям;
- все элементы пользовательского интерфейса быть представлены на русском языке;
- навигационные элементы пользовательского интерфейса должны быть выполнены в удобной для пользователя форме;
- задание критериев выбора информации должно производиться путем простой, интуитивно понятной настройки значений параметров фильтров поиска;
- дизайн пользовательского интерфейса должен быть выполнен в едином стиле;
- в случае возникновения ошибочных ситуаций пользователи должны быть уведомлены о них с описанием ошибки на русском языке;
- интерфейс должен корректно отображаться в ландшафтном виде (горизонтальное расположение экрана) с расширением не менее 1024x768. При использовании меньшего разрешения экрана допускается появление полос прокрутки.

### 4.2 Требования к режиму работы

В штатном режиме функционирования Система должна обеспечивать следующий режим работы: доступность функций системы в режиме — 24 часа в день, 7 дней в неделю (24x7).

В целях обеспечения надежного функционирования в Системе должны быть предусмотрены:

- контроль целостности программного обеспечения;
- контроль целостности данных на уровне базы данных;
- сохранение работоспособности программного обеспечения при некорректных действиях пользователя;
- резервное копирование базы данных для восстановления работоспособности Системы в случае ее логического или физического разрушения;

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						105
Изм.	Лист	№ докум.	Подпись	Дата		

– перерывы на администрирование (профилактику) Системы, перенастройку и простои в связи с неисправностью инфраструктуры Системы, которые должны составлять не более 100 часов в год.

#### 4.3 Требования к сохранности информации при авариях

Система должна обеспечивать сохранность информации при аварии. Под аварией следует понимать ситуацию, характеризующуюся:

- полным или частичным прекращением выполнения функциональных задач;
- аномальным (нештатным) режимом работы Системы;
- полной или частичной потерей информации;
- нелегитимным доступом к информации в Системе и преднамеренным ее искажением или уничтожением.

В случае повреждения данных вследствие аварии, должно обеспечиваться восстановление данных Системы из резервных копий, предоставляемых Заказчиком.

Восстановление данных должно осуществляться в соответствии со следующими требованиями: восстановление БД из резервной копии должно осуществляться средствами СУБД.

Для обеспечения защиты данных от разрушений при авариях и сбоях в процессе выполнения пользовательских задач в Системе, Исполнитель предоставляет Заказчику рекомендации по обеспечению: создания резервных копий БД.

Сохранность информации должна обеспечиваться специалистами Заказчика и штатными средствами собственной ИТ-инфраструктуры на основе рекомендаций Исполнителя.

#### 4.4 Требование к хранению данных

Все данные Системы должны храниться в структурированном виде под управлением реляционной СУБД. Исключения составляют мультимедиафайлы, шаблоны и нормативно-правовые документы в форматах Word, Excel, PDF и другие файлы, размещение которых в базе данных может вызвать заметное снижение производительности. Такие файлы размещаются на сервере в файловой системе, а в БД размещаются ссылки на них

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						106
Изм.	Лист	№ докум.	Подпись	Дата		



#### 4.5 Требования к техническому обеспечению

Система и ее компоненты должны быть совместимы с операционными системами семейства Windows и Linux.

Компоненты Системы должны допускать возможность развертывания на виртуальных машинах.

Внутренние испытания Системы Исполнитель проводит на своих серверах.

Приемо-сдаточные испытания Системы проводятся на промышленном сервере.

Аппаратное обеспечение серверной части должно удовлетворять следующим минимальным требованиям:

- операционная система Windows или Linux;
- четырехъядерный процессор 2.5 ГГц;
- 16 Гб ОЗУ;
- не менее 3Тб свободного места на диске;
- система резервного копирования не менее 10 Тб;
- двухъядерный процессор 2.5 ГГц;
- 16 Гб ОЗУ.

1) Серверное оборудование - 1 процессор Intel Xeon, 4 ядра 2.53 ГГц (и выше) ОЗУ 16 Гб, HDD 500 Гб RAID1, ОС на ядре Linux. Система резервного копирования объемом не менее 1 Тб.

2) Клиентское оборудование (стационарный компьютер): Intel Core 2 Duo 2.2 ГГц и выше. Оперативная память от 4 Гб и выше. ОС: Windows 8/MS Windows 10. Соотношение сторон экрана монитора 16:9.

3) Клиентское оборудование (мобильный телефон) – смартфон под управлением ОС Android 6.0 и выше, iOS 9 и выше.

Представленные требования к аппаратной части являются предварительными и могут быть пересмотрены после завершения разработки системы.

#### 4.6 Требования к исходному коду

Исходный код должен разрабатываться в соответствии с требованиями спецификаций соответствующих технологий, а также не допускать запутывание кода.

Исходный код программы должен сопровождаться значимыми комментариями.

Для всех файлов, классов, методов, переменных должны использоваться значимые имена, по имени должно быть понятно, для чего предназначена указанная сущность.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						107
Изм.	Лист	№ докум.	Подпись	Дата		

## 5 Состав и содержание работ по созданию системы


### 5.1 Общие требования к составу и содержанию работ по созданию Системы

Состав и содержанию работ по созданию Системы определяются настоящим Техническим заданием.

В рамках создания Системы Исполнитель должен выполнить следующие работы:

- разработка Системы;
- развертывание Системы на промышленном сервере;
- наполнение Системы контентом;
- проведение испытаний Системы.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						108
Изм.	Лист	№ докум.	Подпись	Дата		



Войти

Регистрация



Имя пользователя

Пароль

☐ Показать пароль

Войти

Рисунок 10 – Окно авторизации



Войти

Регистрация

Имя пользователя


Пароль

Подтвердите пароль

☐ Показать пароли


Зарегистрироваться

Рисунок 11 – Окно регистрации

 Кодозавр

ГлавнаяКурсыКонтакты

Личный кабинет учителя



Калашников Ирина Сергеевна

Электронная почта: user@example.com

Телефон: +7 (800) 123-45-67

Дата регистрации: 01.01.2023

Расписание

Назад

июнь 2024 г.

Вперед

Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Рисунок 12 – Личный кабинет преподавателя

ГлавнаяКурсыКонтакты

Личный кабинет родителя

Клименко Ирина Александровна

Электронная почта: irina@example.com

Телефон: +7 (800) 777-45-67

Дата регистрации: 04.01.2023

Мои курсы

Робототехника для детей 4-6 лет

Прогресс: 60%

Рисунок 13 – Личный кабинет родителя

Расписание курсов

Назад

июнь 2024 г.

Вперед

Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

VkontakteYouTubeInstagram

Start Junior

Телефон: +7 (123) 456-78-90

Электронная почта: startjunior56@gmail.com

Рисунок 14 – Личный кабинет родителя

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ  
Платформы для оптимизации учебного процесса

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						112
Изм.	Лист	№ докум.	Подпись	Дата		

Проект "Codozavr" представляет собой веб-платформу, предназначенную для автоматизации ключевых процессов частной школы программирования и робототехники для детей. Платформа позволяет родителям записывать детей на курсы, детям - отслеживать расписание занятий, а преподавателям — получать доступ к необходимым материалам. Данное руководство предназначено для пользователей всех ролей: родителей, преподавателей и учеников.

## 1. Начало работы

### 1.1 Регистрация и авторизация

Регистрация для родителей:

1. Перейдите на главную страницу сайта.
2. Нажмите кнопку "Регистрация".
3. Заполните форму, указав имя пользователя, email и пароль.
4. Нажмите кнопку "Зарегистрироваться".

После успешной регистрации вы получите подтверждение на указанный email. Для входа в систему используйте кнопку "Вход" и введите свои учетные данные.

Авторизация для преподавателей и учеников:

Логин и пароль для преподавателей и учеников предоставляются администратором. Для входа в систему используйте кнопку "Вход" и введите предоставленные учетные данные.

## 2. Основные функции

### 2.1 Личный кабинет

После входа в систему вы получите доступ к личному кабинету, где сможете следить за расписанием, а также:

- родители: редактировать свой профиль;
- преподаватели: редактировать профиль и материалы по курсам.

### 2.2 Запись на курсы

Запись на курсы доступна только родителям и осуществляется через главную страницу.

Шаги для записи на курс:

1. Перейдите на главную страницу сайта.
2. Ознакомьтесь с представленными курсами.
3. Выберите интересующий курс и нажмите на кнопку "Записаться".
4. Заполните форму заявки, указав необходимые данные.
5. Нажмите кнопку "Отправить".

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						113
Изм.	Лист	№ докум.	Подпись	Дата		

После отправки формы заявка будет обработана администратором, и вы получите уведомление на указанный email или телефон.

## 2.3 Расписание занятий

Просмотр расписания:

1. Войдите в систему под своей учетной записью.
2. Перейдите в личный кабинет.
3. Здесь вы сможете просмотреть все запланированные занятия и мероприятия.

## 2.4 Обратная связь

Форма обратной связи расположена в подвале сайта и доступна всем пользователям.

Отправка формы обратной связи:

1. Перейдите в подвал сайта.
2. Заполните форму, указав свои комментарии или вопросы.
3. Нажмите кнопку "Отправить".

Форма будет отправлена администратору, который обработает ваш запрос и свяжется с вами при необходимости.

## 3. Функциональные возможности для ролей

### 3.1 Родители

Функциональные возможности:

- регистрация и авторизация на платформе;
- просмотр и редактирование профиля;
- запись детей на курсы через главную страницу;
- просмотр расписания занятий;
- отправка формы обратной связи.

### 3.2 Преподаватели

Функциональные возможности:

- авторизация на платформе с использованием данных, предоставленных администратором;
- просмотр курсов и управление материалами;
- просмотр расписания занятий.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						114
Изм.	Лист	№ докум.	Подпись	Дата		



### 3.3 Ученики

Функциональные возможности:

- авторизация на платформе с использованием данных, предоставленных администратором;
- просмотр профиля;
- просмотр расписания занятий;
- отправка формы обратной связи.

## 4. Безопасность и конфиденциальность

### 4.1 Аутентификация и авторизация

В проекте используется система токенов JWT (JSON Web Tokens), обеспечивающая безопасную передачу данных между клиентом и сервером. Каждый пользователь получает уникальный токен, который позволяет идентифицировать его и контролировать доступ к ресурсам.

### 4.2 Управление доступом

Реализована модель ролевого доступа (RBAC), которая ограничивает доступ пользователей к определенным функциям и данным на основе их ролей. Это позволяет строго контролировать действия пользователей и предотвращать несанкционированный доступ.

### 4.3 Защита данных

Все пароли и чувствительные данные хранятся в зашифрованном виде. Для защиты от межсайтовых запросов используется механизм CSRF-токенов. Также применяется экранирование пользовательских данных для предотвращения выполнения вредоносного кода.

## 5. Техническая поддержка

Если у вас возникли вопросы или проблемы при использовании платформы "Codozavr", вы можете обратиться за помощью по следующим каналам:

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						115
Изм.	Лист	№ докум.	Подпись	Дата		

- телефон поддержки: +7 (952) 415-09-72;
- электронная почта: [xilatomil@gmail.com](mailto:xilatomil@gmail.com);
- форма обратной связи на сайте.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						116
Изм.	Лист	№ докум.	Подпись	Дата		

РУКОВОДСТВО АДМИНИСТРАТОРА  
Платформы codozavr

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						117
Изм.	Лист	№ докум.	Подпись	Дата		

Настоящее руководство предназначено для администраторов системы и описывает порядок работы с моделями, представленными на изображении, включая "Course applications", "Courses", "Feedback forms", "Lessons", "Topics", "Parents", "Students", "Teachers" и "Users". Данное руководство соответствует требованиям ГОСТ 34.603-92 "Автоматизированные системы. Руководства пользователя. Требования к содержанию и оформлению".

### Управление заявками на курсы (Course applications)

Цель: Управление заявками на курсы от родителей.

Действия:

1. В меню "Course applications" нажмите "Добавить заявку".
2. Заполните следующие поля:
  - Parent: выберите родителя из выпадающего списка.
  - Course: выберите курс из выпадающего списка.
  - Parent Email: укажите электронную почту родителя.
  - Parent Phone: укажите телефонный номер родителя.
  - Status: выберите статус заявки (Pending, Approved, Rejected).
3. Нажмите "Сохранить".

Замечания: убедитесь, что все обязательные поля заполнены.

### Управление курсами (Courses)

Цель: Управление информацией о курсах.

Действия:

1. В меню "Courses" нажмите "Добавить курс".
2. Заполните следующие поля:
  - Name: укажите название курса.
  - Description: добавьте описание курса.
  - Duration: укажите продолжительность курса в часах.
  - Cost: укажите стоимость курса.
  - Level: укажите уровень курса (начальный, средний, продвинутый).
3. Нажмите "Сохранить".

Замечания: Описание курса должно быть подробным и точным.

### Управление формами обратной связи (Feedback forms)

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						118
Изм.	Лист	№ докум.	Подпись	Дата		

Цель: Управление отзывами и обратной связью.

Действия:

1. В меню "Feedback forms" нажмите "Добавить форму".
2. Заполните следующие поля:
  - Description: введите описание отзыва.
3. Нажмите "Сохранить".

Замечания: Описание отзыва должно быть информативным.

### Управление уроками (Lessons)

Цель: Управление расписанием и содержанием уроков.

Действия:

1. В меню "Lessons" нажмите "Добавить урок".
2. Заполните следующие поля:
  - Date: укажите дату проведения урока.
  - Start Time: укажите время начала урока.
  - End Time: укажите время окончания урока.
  - Teacher: выберите преподавателя из выпадающего списка.
  - Topic: выберите тему урока из выпадающего списка.
3. Нажмите "Сохранить".

Замечания: убедитесь, что даты и времена указаны корректно.

### Управление темами (Topics)

Цель: Управление темами, которые будут изучаться в рамках курсов.

Действия:

1. В меню "Topics" нажмите "Добавить тему".
2. Заполните следующие поля:
  - Name: укажите название темы.
  - Description: добавьте описание темы.
  - Material Links: укажите ссылки на материалы (при необходимости).
  - Course: выберите курс, к которому относится тема.
  - Materials: загрузите материалы (если необходимо).
3. Нажмите "Сохранить".

Замечания: Описание темы должно быть подробным и соответствовать содержанию курса.

### Управление родителями (Parents)

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						119
Изм.	Лист	№ докум.	Подпись	Дата		

Цель: Управление профилями родителей.

Действия:

1. В меню "Parents" нажмите "Добавить родителя".
2. Заполните следующие поля:
  - User: выберите пользователя из выпадающего списка.
  - Phone: укажите телефонный номер.
  - Email: укажите электронную почту.
  - Relation to Child: укажите отношение к ребенку (например, мать, отец).
3. Нажмите "Сохранить".

Замечания: Контактные данные должны быть актуальными.

### Управление учениками (Students)

Цель: Управление профилями учеников.

Действия:

1. В меню "Students" нажмите "Добавить ученика".
2. Заполните следующие поля:
  - User: выберите пользователя из выпадающего списка.
  - Birth Date: укажите дату рождения.
  - Email: укажите электронную почту (при необходимости).
  - Parent: выберите родителя из выпадающего списка.
3. Нажмите "Сохранить".

Замечания: убедитесь, что все обязательные поля заполнены.

### Управление преподавателями (Teachers)

Цель: Управление профилями преподавателей.

Действия:

1. В меню "Teachers" нажмите "Добавить преподавателя".
2. Заполните следующие поля:
  - Parent: выберите родителя из выпадающего списка.
  - User: выберите пользователя из выпадающего списка.
  - Position: укажите должность.
  - Phone: укажите телефонный номер.
  - Email: укажите электронную почту.
3. Нажмите "Сохранить".

Замечания: Контактные данные должны быть актуальными.

### Управление пользователями (Users)

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						120
Изм.	Лист	№ докум.	Подпись	Дата		

Цель: Управление учетными записями всех пользователей.

Действия:

1. В меню "Users" нажмите "Добавить пользователя".

2. Заполните следующие поля:

– Username: укажите имя пользователя.

– Email: укажите электронную почту.

– Password: задайте пароль.

– User Type: выберите тип пользователя (родитель, ученик, преподаватель).

3. Нажмите "Сохранить".

Замечания: Пароль должен быть надежным и соответствовать требованиям безопасности.

					РКСИ.ДП24.09.02.07.4156.00ПЗ	Лист
						121
Изм.	Лист	№ докум.	Подпись	Дата		