

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«___»_____ 2023 г.

Разработка веб-сайта для организации работы репетитора

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 02.03.02.2023.308-496.ВКР

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.

_____ М.В. Сухов

Автор работы,
студент группы КЭ-402

_____ Т.М. Галимханов

Ученый секретарь
(нормоконтролер)

_____ И.Д. Володченко

«___»_____ 2023 г.

Челябинск, 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

_____ Л.Б. Соколинский

06.02.2023 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студенту группы КЭ-402

Галимханову Тимуру Максимовичу,

обучающемуся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

1. Тема работы (утверждена приказом ректора от 25.04.2023 г. № 753-13/12)

Разработка веб-сайта для организации работы репетитора.

2. Срок сдачи студентом законченной работы: 05.06.2023 г.

3. Исходные данные к работе

3.1. Артеменко Ю.Н. MySQL: Справочник по языку. – Спб.: Диалектика, 2009. – 429 с.

3.2. Гаевский А.Ю. Создание Web-страниц и Web-сайтов. // «Технолоджи – 3000», 2008. – 464 с

4. Перечень подлежащих разработке вопросов

4.1. Провести обзор литературы.

4.2. Выполнить проектирование веб-приложения.

4.3. Выполнить реализацию веб-приложения.

4.4. Провести тестирование веб-приложения.

5. Дата выдачи задания: 06.02.2023 г.

Научный руководитель,

доцент кафедры СП, к.т.н.

М.В. Сухов

Задание принял к исполнению

Т.М. Галимханов

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. РЕПЕТИТОРСКИЕ УСЛУГИ В РОССИИ.....	7
1.1. Репетиторство как образовательный процесс.....	7
1.2. Описание предметной области.....	11
1.3. Анализ существующих решений.....	11
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ.....	16
2.1. Функциональные и нефункциональные требования к проектируемой системе.....	16
2.2. Диаграмма вариантов использования.....	17
2.3. База данных веб приложения.....	19
3. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ.....	24
3.1. Реализация авторизации на сайте.....	24
3.2. Реализация списка учеников у репетитора.....	27
3.3. Реализация создания занятия.....	29
3.4. Реализация домашнего задания.....	32
3.5. Реализация финансов.....	35
4. ТЕСТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ.....	38
4.1. Функциональное тестирование.....	38
4.2. Usability-тестирование.....	39
ЗАКЛЮЧЕНИЕ.....	40
ЛИТЕРАТУРА.....	41
ПРИЛОЖЕНИЕ. Отображение веб-сайта со стороны ученика.....	43

ВВЕДЕНИЕ

Актуальность

В настоящее время возрастает потребность в квалифицированных рабочих местах, которые требуют хорошей подготовки, а также в большей мобильности людей с высшим образованием в сравнении с теми, у кого образование начальное или среднее. Кроме того, социальная функция высшего образования приобретает все большее значение. Оно является минимальным требованием для профессионального и карьерного роста. Важным элементом качественного обучения является репетиторство.

Данная работа имеет значимость, поскольку не все ученики школ и студенты университетов способны полностью усвоить материал, который представлен на уроках, а также на лекциях и практических занятиях. Это может происходить по причинам большой нагрузки на ученика. Поэтому важность репетиторства в нынешнее время нельзя не преувеличить. Работа репетитора может оказать положительное воздействие на процесс обучения учеников и студентов, привести к повышению успеваемости и формированию устойчивых знаний.

Важно ответственно относиться к основополагающим школьным дисциплинам, поскольку именно они закладывают фундамент для дальнейшего карьерного роста. Отсутствие пробелов и недостатков в этих дисциплинах является особенно важным, поскольку они влияют на формирование углубленных знаний и навыков, необходимых для продвижения по карьерной лестнице.

В настоящее время наблюдается заметный рост рынка образовательных услуг, и одной из его важных составляющих является внедрение инноваций, направленных на улучшение качества образования.

В современном мире все большую популярность набирает онлайн-образование, так как каждый человек может осваивать навыки, не выходя из дома. Это удобно и экономит время. Именно поэтому все больше людей

предпочитают онлайн-обучение и прохождение курсов от различных компаний. Дистанционная работа с репетитором имеет множество преимуществ и превосходит традиционный очный формат обучения. Существует множество программ, которые помогают объяснять учебный материал.

У репетиторов доступно множество приложений, которые упрощают сам процесс обучения, а также помогают работать с дополнительными учебными материалами и проверкой домашних заданий учеников. Дополнительные приложения для репетиторов также облегчают ведение учета прогресса учеников, обеспечивают возможность проведения онлайн-тестирования и оценки знаний, а также обеспечивают коммуникацию с родителями или опекунами учеников для обмена информацией о прогрессе и достижениях.

Для эффективной работы репетитором необходимо использовать множество различных приложений, каждое из которых выполняет свою уникальную задачу. Например, мессенджер обеспечивает коммуникацию с учениками, позволяя обмениваться информацией и задавать вопросы. Приложение для составления расписания помогает планировать занятия и управлять временем. Таблицы об оплатах занятий помогают отслеживать информацию о платежах и финансовом состоянии. Кроме того, приложения для статистики по доходам позволяют анализировать и отслеживать финансовые результаты и улучшать финансовое планирование.

Для удобства и упрощения работы можно объединить функционалы некоторых этих приложений для оптимизации деятельности.

Постановка задачи

Целью выпускной квалификационной работы является разработка веб-сайта для организации работы репетитора. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) произвести обзор существующих решений;
- 2) определить требования веб-приложения;

- 3) спроектировать веб-приложение;
- 4) реализовать веб-приложение;
- 5) произвести тестирование веб-приложения.

Структура и содержание работы

Работа состоит из введения, четырёх глав, заключения, списка литературы и приложения. Объем работы составляет 45 страниц, объем списка литературы – 15 источников.

В первом разделе описывается постановка задачи, а также предметная область. Также произведен анализ предметной области и выбор инструментов и технологий для реализации.

Во втором разделе были описаны функциональные и нефункциональные требования к проекту, а также составлена диаграмма вариантов использования.

В третьем разделе описывается реализация таких страниц как, расписание занятий, раздел с учениками, заданиями, а также раздел с финансами.

В четвертом разделе было произведено тестирование как функциональное, так и usability тестирование.

В заключении приведены результаты проделанной работы.

1. РЕПЕТИТОРСКИЕ УСЛУГИ В РОССИИ

1.1. Репетиторство как образовательный процесс

Из-за различных объективных причин, школьные преподаватели не всегда могут адекватно адаптировать образовательный процесс под индивидуальные потребности каждого ученика, учитывая их уровень способностей, мотивацию и интерес к определенным предметам. Это приводит к возникновению так называемых «образовательных пробелов» у учащихся, которые чувствуют неудовлетворенность своими образовательными потребностями. В таких ситуациях многие родители ищут возможности для «закрытия» этих пробелов и обращаются за помощью к репетиторам. Репетиторы предоставляют индивидуальное обучение, которое позволяет ученикам заполнить пробелы в знаниях, повысить успеваемость и достичь лучших результатов в учебе.

В то же время, по мнению Т. В. Базжиной, повышению спроса репетиторских услуг способствует снижение коммуникативной культуры в школе в системе «ученик – учитель» [1].

С ростом рынка репетиторских услуг возрастает интерес научного сообщества к этой сфере и требует проведения подробного исследования. Анализ современных исследований в области частного дополнительного образования и рынка репетиторских услуг позволяет сделать вывод, что для большинства исследователей репетитор рассматривается в качестве важной экономической составляющей этой сферы взаимодействий. Однако, помимо экономического аспекта, такие исследования также должны учитывать педагогические, социальные и психологические аспекты роли репетиторов в образовательном процессе.

Ряд исследований ставит перед собой задачу рассмотрения экономических эффектов репетиторства, некоторые исследования изучают социальные эффекты и возникающие при этом социальные неравенства, другие – отсутствие официального законодательно закреплённого статуса ре-

петитора и возникающий при этом «нормативный вакуум». Однако мало таких исследований, которые рассматривали бы методическую сторону репетиторства: поведенческие тактики и педагогические технологии, используемые им в ходе проведения занятий, качества репетитора как педагога, которые отличают его от школьного преподавателя и позволяют ему достигать поставленных перед ним задач [2].

Такое исследование имеет актуальность, поскольку среди репетиторов существует разделение на специалистов, обладающих глубоким предметным знанием, но не всегда обладающих дидактическими навыками, такими как умение ясно объяснять и подавать материал. И наоборот, некоторые репетиторы обладают отличными навыками обучения и умеют эффективно взаимодействовать с учащимся, но могут быть недостаточно компетентны в преподаваемом предмете. Поэтому исследование таких аспектов важно для понимания и оптимизации работы репетиторов и обеспечения качественного образования.

В этом контексте требуется осуществить анализ деятельности репетитора с целью выявления ключевых качеств и тактик, необходимых для успешного выполнения его задач и достижения планируемых результатов. Также следует обратить внимание на особенности применения педагогических технологий и различных форм проведения урока. Это позволит определить оптимальные подходы и стратегии, которые способствуют эффективному взаимодействию репетитора и ученика, а также повышению образовательных результатов.

Анализом такого субъекта образовательного процесса, как «репетитор», занимается Н. Г. Крылова в статье «Репетиторы выходят из тени» [3], в которой она приводит свой взгляд на данный вид педагогической деятельности и рассматривает его следующим образом:

1) вид преподавательской деятельности, смысл которой в том, чтобы обучать определенному предмету индивидуально или в малых группах, с целью успешного поступления в учебное заведение;

2) теневой социальный институт общества по использованию социальных ресурсов, позволяющий удовлетворить определенные потребности;

3) особый вид образовательной услуги [4].

А.А. Наумова предлагает свое определение: «Под репетиторством мы будем понимать вид платных образовательных услуг, представляющий собой дополнительные учебные занятия по предметам школьной программы с целью закрепления или углубления знаний» [5]. Однако вышеприведенные определения термина «репетитор» не являются исчерпывающими в области целеполагания и задач, которые ставятся перед современным репетитором. На сегодняшний день формы самой репетиторской деятельности различны. Зачастую они целиком и полностью определены спросом и стоящими перед репетитором задачами, так как «репетиторство» – это образовательная услуга, работа «по заказу».

Наиболее распространенным видом поиска репетиторов для учеников по-прежнему считается так называемое сарафанное радио. Все, на что могут полагаться родители при выборе преподавателя, – отзывы нескольких человек и результаты ЕГЭ, полученные детьми друзей.

Не менее разнообразны задачи, которые ставятся перед преподавателем-репетитором заказчиками образовательных услуг (родителями ученика, реже – самим учеником). Исследованием востребованности задач и направлений деятельности репетиторов в 2018 году занималась Е.Н. Шипкова. В нем были опрошены репетиторы и родители учеников с 5 по 11 класс с целью выявить характер и степень востребованности задач, стоящих перед репетитором. По результатам исследования можно говорить о том, что в области задач можно выявить три основных направления (по степени убывания востребованности со стороны учеников и их родителей):

- 1) подготовка к решению конкретных образовательных задач (около 87% опрошенных);
- 2) восполнение пробелов в знаниях по уже пройденным общеобразовательным программам и помощь ученику в освоении нового материала (около 39% опрошенных);
- 3) удовлетворение личных образовательных запросов ученика и повышение его интереса к изучаемой дисциплине (около 24% опрошенных) [6].

Степень достижения задач, стоящих перед репетитором, оценивается в первую очередь по достижению учеником тех или иных целей и получению им образовательных результатов. Однако исследования в сфере эффективности работы репетиторов дают достаточно противоречивые результаты. В Германии и Японии наблюдается положительная взаимосвязь между индивидуальными дополнительными занятиями и академическими успехами учеников, тогда как исследования в Корее и Сингапуре такой связи не обнаружили [7–9].

Также мы можем отметить позитивные и негативные качества частных репетиторов.

Первостепенные аспекты, отмеченные учениками в качестве «положительных» характеристик репетитора, включали в себя способность создавать комфортную и дружелюбную атмосферу на занятиях, интересное и понятное изложение материала. Родителям же особенно важен был индивидуальный подход к особенностям и потребностям ученика.

К негативным качествам, родители указали непунктуальность, большую теоретическую нагрузку, и иногда конфликтное поведение, а также недостаточно индивидуальное обучение с учениками.

1.2. Описание предметной области

Предметная область веб-сайта по организации работы репетитора состоит в том, чтобы установить коммуникацию между репетитором и учеником. Веб-сайт будет служить для отображения домашнего задания, а также расписания для учеников. Ученики смогут узнавать информацию о следующем уроке через данный веб-сайт. Это может решить проблему, что ученик забыл, когда будет следующее занятие и может просто посмотреть это из своего аккаунта.

Веб-сайт включает в себя такие функции как: отображение расписания занятий, отображение оплаты занятий, действия с домашним заданием.

1.3. Анализ существующих решений

В этом подразделе будет проведен анализ имеющихся решений в данной области и представлены результаты в виде таблицы. Будут рассмотрены различные аспекты и особенности существующих решений, а также их преимущества и недостатки. Таблица будет содержать сведения о каждом решении, его основных характеристиках и результаты сравнительного анализа. Это позволит получить полное представление о текущем состоянии рынка и выбрать наиболее подходящее решение для задачи, руководствуясь конкретными потребностями и требованиями.

GeekBrains [9]

При анализе сайта «GeekBrains» можно отметить следующее: на главной странице акцентируется внимание на преимуществах платформы и особенностях специальностей, по которым предоставляются курсы. Также представлены различные направления обучения и популярные программы. На сайте присутствует блок с популярными статьями, охватывающими различные тематики. Посетители могут ознакомиться с последними тестированиями, пройденными пользователями. Дополнительно, име-

ется блок с комментариями и отзывами, а также раздел с историями успеха студентов и вакансиями, предлагаемыми на платформе.

Логотип «GeekBrains» выполнен в минималистичном стиле, что соответствует современным дизайнерским трендам. Он символизирует мозг, что связывается с образованием и знаниями. Основной цветовой схемой сайта является белый, что создает чистый и простой общий вид.

Дизайн главной страницы представлен на рисунке 1.

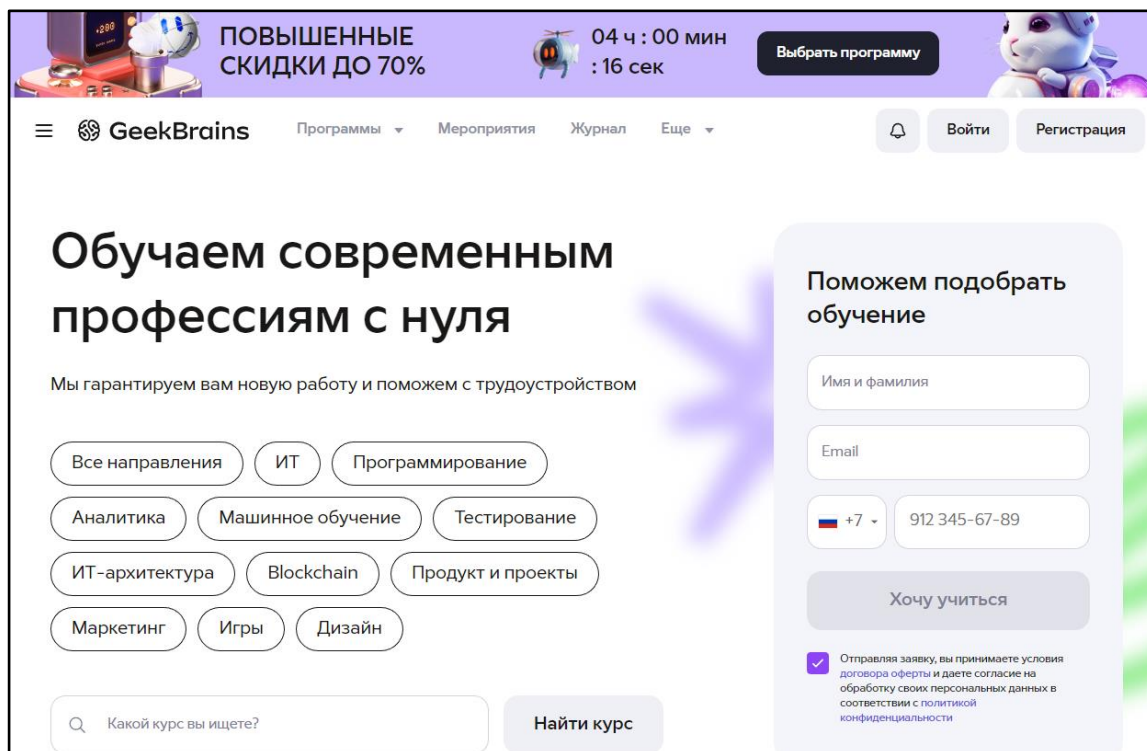


Рисунок 1 – Главная страница сайта GeekBrains

Ulearn [10]

На главной странице сайта «Ulearn.me» немедленно привлекают внимание представленные курсы и их краткие описания. Интерфейс сайта обладает минималистичным дизайном, где помимо этого выделяются только название сайта и кнопки для авторизации и регистрации.

У данного сайта отсутствует традиционный логотип. Вместо этого, в качестве логотипа использовано название сайта «Ulearn.me».

Увидеть это можно на рисунке 2, на котором представлена главная страница сайта «Ulearn».

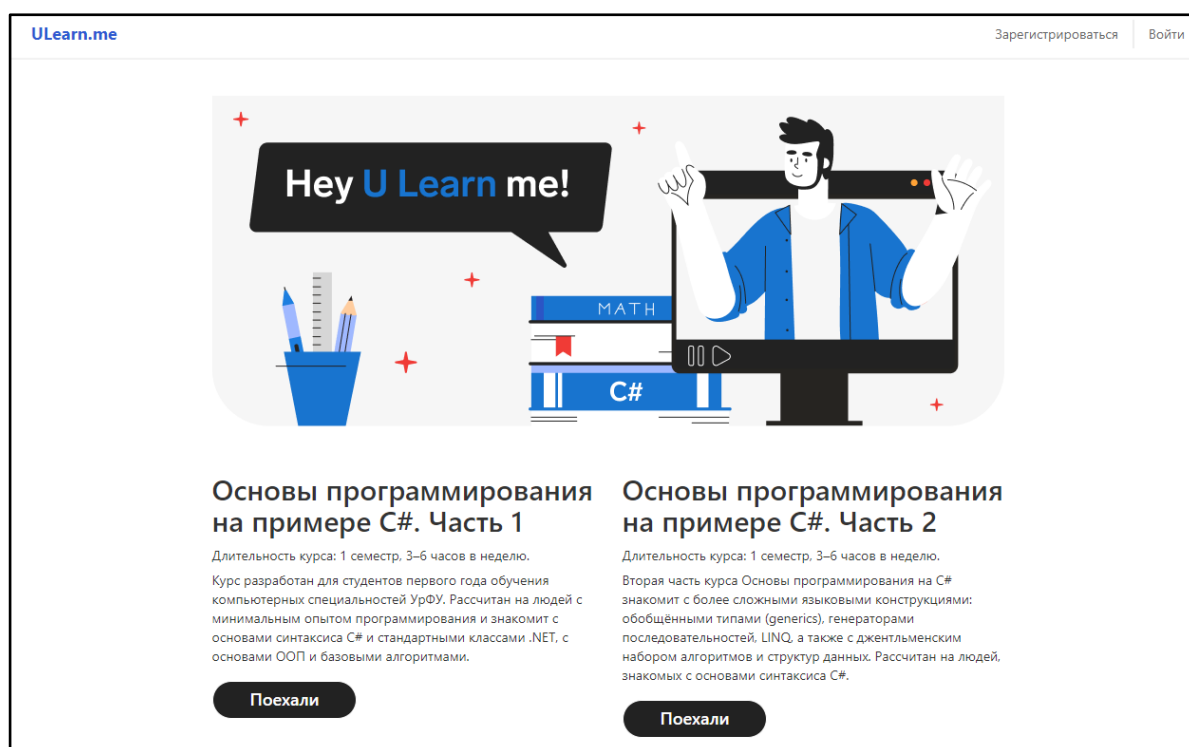


Рисунок 2 – Главная страница Ulearn

Skillbox [11]

При проведении анализа сайта «Skillbox» выявлены следующие особенности: на главной странице пользователей встречается функция поиска курсов по выбранной профессии, что позволяет найти необходимые программы обучения. Кроме того, предоставляется описание преимуществ платформы. На сайте также отображаются актуальные и востребованные профессии на текущий момент, чтобы помочь пользователям выбрать наиболее перспективные области. Также присутствуют отзывы пользователей и информация о партнерах данного сервиса.

В качестве логотипа на сайте «Skillbox» присутствует название платформы «Skillbox».

Главная страница этого сайта представлена на рисунке 3.

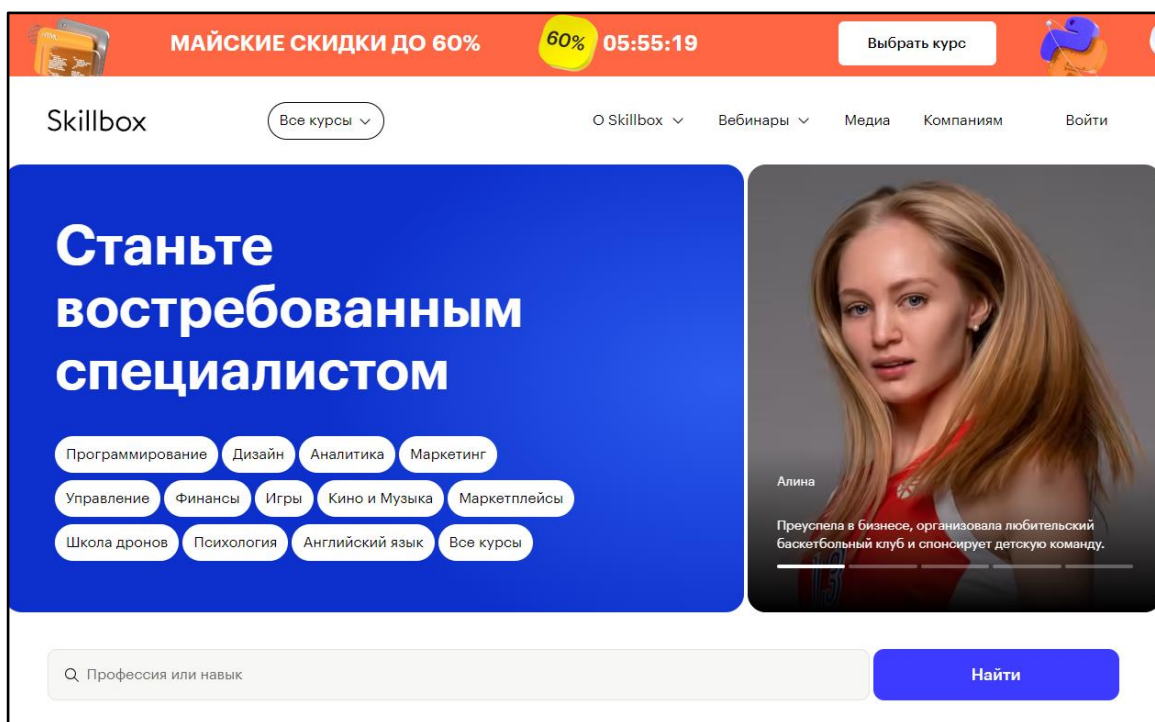


Рисунок 3 – Главная страница сайта Skillbox

Краткие сведения о аналогичных решениях можно увидеть в таблице 1.

Таблица 1 – Анализ существующих проектов

Сайт	«GeekBrains»	«Ulearn»	«Skillbox»
Регистрация на сайте	+	+	+
Система отзывов клиента	+	—	+
Современный интерфейс	+	+	+
Поддержка	+	+	+

После изучения имеющихся вариантов было решено применить фреймворк SvelteKit для разработки пользовательского интерфейса веб-сайта онлайн курсов. Кроме того, клиентское приложение будет развернуто с использованием платформы Vercel.

Это решение позволит создать эффективную и отзывчивую систему интерфейса, основанную на SvelteKit, который обладает высокой производительностью и простотой разработки. Клиентское приложение, развернутое на платформе Vercel.

Vercel – это платформа развертывания и хостинга для клиентских приложений и статических сайтов. Она предоставляет простой и удобный способ развертывания вашего клиентского приложения и обеспечивает высокую производительность, масштабируемость и безопасность [12].

Некоторые из возможностей, предоставляемых Vercel при создании клиентского приложения, включают.

1. Автоматическая масштабируемость. Платформа Vercel автоматически масштабирует приложение в зависимости от его нагрузки, что позволяет обрабатывать большое количество пользователей и сохранять высокую производительность.

2. Мгновенные обновления. Когда вы вносите изменения в ваше приложение и публикуете их, Vercel автоматически обновляет ваше развернутое приложение, что позволяет вам быстро внедрять изменения без простоев.

Также для разработки использовались Tailwind CSS и Flowbite-Svelte. Это две различные инструменты, связанные с разработкой пользовательского интерфейса (UI) [13].

Flowbite-Svelte построен на базе Tailwind CSS и предоставляет готовые компоненты, такие как кнопки, формы, карточки, навигационные панели и т. д. Flowbite-Svelte позволяет быстро создавать интерфейсы.

Вывод по первой главе

В данной главе был проведен анализ существующих площадок для веб-сайта по организации работы репетитора. Также был осуществлен выбор инструментов и технологий, необходимый для реализации сайта.

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

2.1. Функциональные и нефункциональные требования к проектируемой системе

Анализ предметной области позволил выделить следующие функциональные требования [14]:

- 1) авторизация репетитора и ученика на сайте;
- 2) добавление ученика и создание для него аккаунта;
- 3) перенос ученика в архивный список;
- 4) возможность репетитора создать домашнее задание, просмотр предыдущих заданий и возможность оценивать их;
- 5) возможность репетитора отметить занятие, как оплаченное;
- 6) возможность ученика посмотреть полностью своего расписания за все время;
- 7) возможность ученика смотреть задолженности по оплате занятий;
- 8) возможность ученика просматривать весь список своего домашнего задания и выполнять его;
- 9) возможность ученика посмотреть свою успеваемость.

А также нефункциональные требования:

- 1) приложение должно быть интуитивно понятным и простым в использовании для пользователей разного уровня опыта;
- 2) реализация клиентского приложения должна быть развернута на Versel;
- 3) приложение должно быть стабильным и надежным, обеспечивая минимальное время простоя и отказоустойчивость;
- 4) Приложение должно поддерживать разные платформы и браузеры. Также должно быть доступно и работоспособно в разных браузерах, например в Chrome, Safari, Firefox.

2.2. Диаграмма вариантов использования

Диаграмма вариантов использования (use case diagram) представляет собой схематическое отображение различных сценариев использования и взаимодействия между ролями (которые выступают в роли специальных классов). Эти диаграммы используются для описания статической структуры функциональных возможностей системы. На основе функциональных требований была создана диаграмма вариантов использования системы на языке UML, изображение которой представлено на рисунке 4.

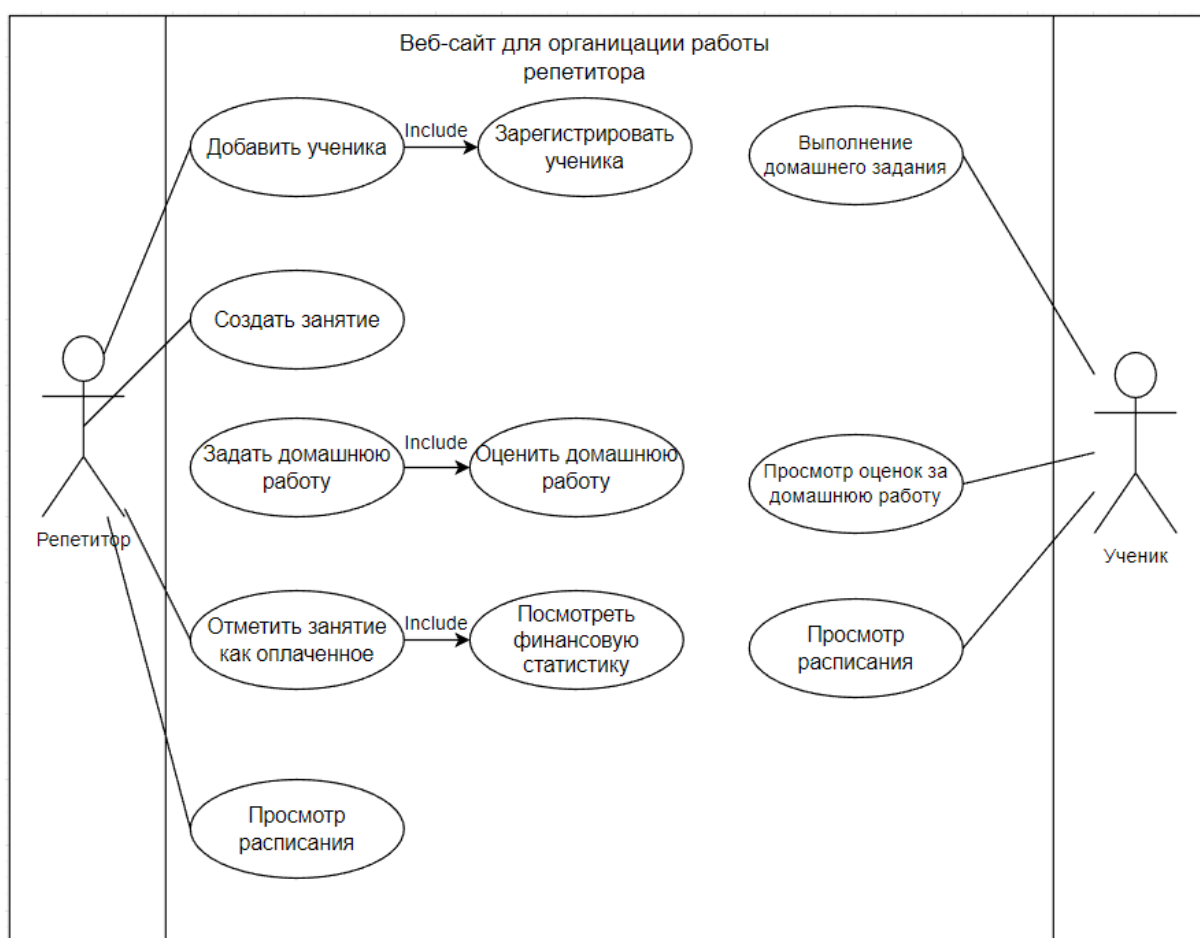


Рисунок 4 – Диаграмма вариантов использования

В проекте системы представлены 2 актера.

1. Репетитор – человек, который обладает правами по управлению всеми разделами в веб-приложении.

2. Ученик – человек, который заходит в веб-приложение для обучения.

Ниже перечислены варианты использования системы.

1. Авторизоваться на сайте – репетитор и ученик могут авторизоваться в веб-приложении.

2. Добавить ученика – репетитор может добавить нового ученика к себе в список учеников.

3. Создать занятие – репетитор может создать занятие, указав при этом время, дату, а также ученика, с которым будет проходить занятие.

4. Задать домашнюю работу – репетитор имеет возможность задать ученику домашнюю работу.

5. Оценить домашнюю работу – после того как репетитор задал домашнюю работу, он может ее оценить после выполнения.

6. Отметить занятие как оплаченное – репетитор после получения оплаты за занятие отмечает это напротив каждого занятия.

7. Посмотреть финансовую статистику – после того как репетитор отмечает, что занятие оплачено, у него изменяется статистика, которая показывает его доходы.

8. Просмотр расписания – репетитор имеет возможность просмотреть все занятия, которые были поставлены как в прошлом, так и будущие занятия.

9. Просмотр расписания – ученик может просматривать только свое расписание, которое отображает все поставленные занятия.

10. Выполнение домашнего задания – ученик выполняет домашнее задание, которое ему задает репетитор.

11. Просмотр оценок за домашнюю работу – ученик может увидеть все свои оценки, которые он получил за домашние задания.

2.3. База данных веб приложения

Как и для любого сайта, так и для данного веб-приложения требуется база данных. Для разработки веб-приложения используется PocketBase.

PocketBase – это открытая серверная часть с открытым исходным кодом, которая включает в себя встроенную базу данных SQLite с проверкой данных, подписками в реальном времени, а так же панель администратора.

PocketBase полностью переносим и требует минимальной настройки. Кроме того, он интегрируется с популярными технологиями, такими как Vue, Svelte, React, Angular и Flutter.

ER-диаграмма представляет собой графическую модель базы данных, которая иллюстрирует взаимосвязи между элементами. Она состоит из двух основных компонентов: сущностей и связей.

ER-диаграмма, представленная на рисунке 5, демонстрирует взаимосвязи между разделами базы данных, такими как:

- 1) ученики (`users`);
- 2) уроки (`lessons`);
- 3) задания (`tasks`);
- 4) финансы (`finance`).

База данных `users` связана с базой данных `lessons` через связь между атрибутами `usermane-students`.

С использованием связи между `name` и `user_id` устанавливается связь между базами данных `users` и `tasks`.

Также база данных `finance` и база данных `lessons` связаны через связь атрибутов `student-students`.

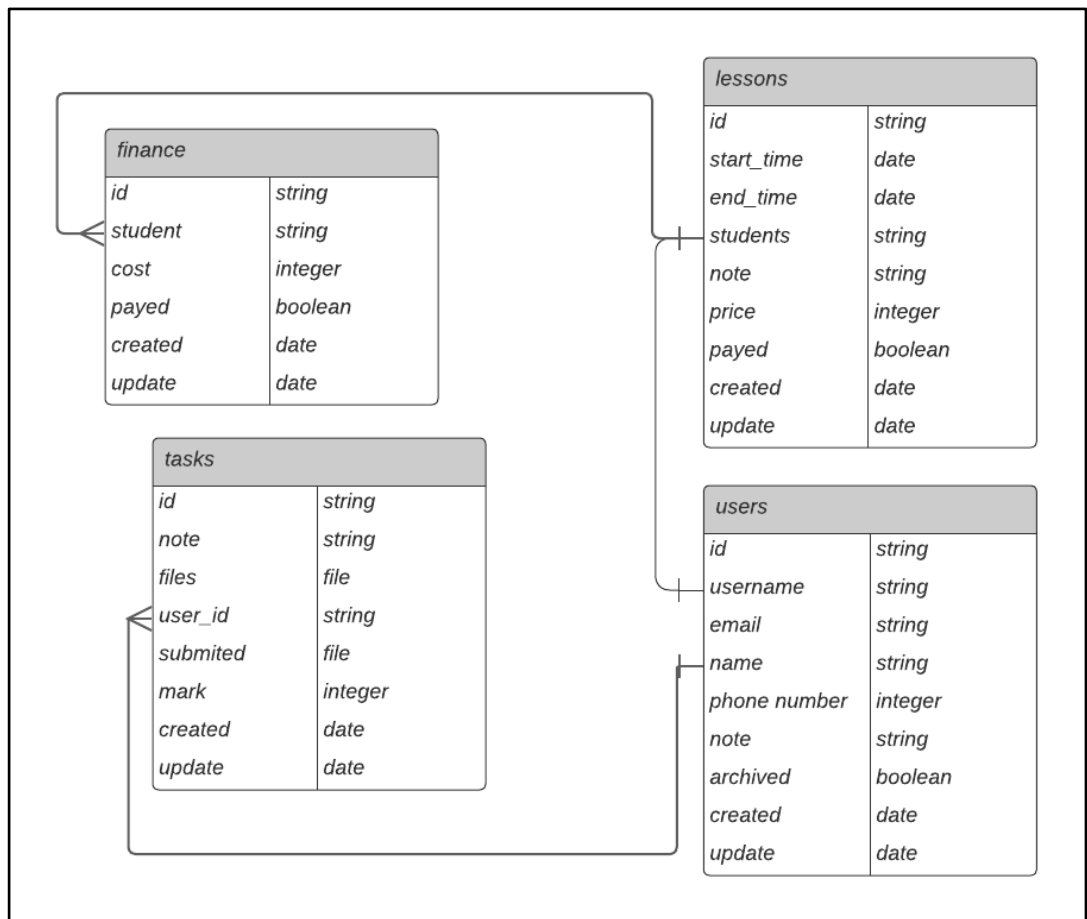


Рисунок 5 – Диаграмма базы данных

База данных также предлагает функциональность аутентификации (листинг 1) и файловое хранилище [13].

Листинг 1 – Реализация аутентификации пользователя

```

import { derived, writable } from 'svelte/store';
import PocketBase, { Admin } from 'pocketbase';
import { PUBLIC_POCKETBASE_URL } from '$env/static/public';

export const pb = new PocketBase(PUBLIC_POCKETBASE_URL);
export const currentUser = writable(pb.authStore.model);
export const isAdmin = derived(currentUser, ($currentUser) => {
  return $currentUser instanceof Admin;
});

pb.authStore.onChange(() => {
  currentUser.set(pb.authStore.model);
});
  
```

Данный код является примером использования библиотеки Svelte и PocketBase для создания веб-приложения с функциональностью аутентификации и управления пользователями. Он позволяет управлять аутенти-

фикацией и получать информацию о текущем пользователе, а также определить, является ли текущий пользователь администратором.

В процессе аутентификации веб-приложение просит пользователя предоставить учетные данные, такие как имя пользователя и пароль. Затем веб-приложение проверяет предоставленные учетные данные с сохраненными значениями в базе данных.

В личном кабинете репетитор может просматривать список своих учеников. Реализация представлена в листинге 2.

Листинг 2 – Реализация списка учеников в профиле репетитора

```
import { writable } from 'svelte/store';
import { pb } from './pocketbase';
import type { User } from './types';

export const students = writable({ values: [] as User[], isLoading: false },
  () => {
    fetchStudents();
  });

export const fetchStudents = async () => {
  if (!pb.authStore.model) {
    return;
  }
  students.update((s) => ({ ...s, isLoading: true }));
  const records = await pb.collection('users').getFullList({
    sort: '-created'
  });
  students.set({ values: records as unknown as User[], isLoading:
false });
};
```

Далее можно увидеть код представляет собой модуль, где определены хранилища для списка учеников и функция для получения этого списка. Этот код представлен в листинге 3.

Листинг 3 – Список учеников и функция для его отображения

```
export type PocketBaseCollection = {
  id: string;
  created: string;
  updated: string;
  expand: Record<string, unknown>;
};

export type User = PocketBaseCollection & {
  name: string;
  username: string;
  email: string;
  phone_number: string;
```

```

        note: string;
        archived: boolean;
        emailVisibility: boolean;
        verified: boolean;
    };

    export type Task = any;

    export type Lesson = PocketBaseCollection & {
        start_time: Date;
        end_time: Date;
        students: string[];
        note: string;
        price: number;
        payed: boolean;
    };

    export type Finances = PocketBaseCollection & {
        revenue: number;
        month: string;
    };

```

Тип `User` представляет информацию о пользователе. Он наследуется от типа `PocketBaseCollection` и расширяет его, добавляя дополнительные поля, такие как `name`, `username`, `email`, `phone_number`, `note`, `archived`, `emailVisibility` и `verified`. Эти поля представляют различные атрибуты пользователя, такие как имя, имя пользователя, электронная почта, номер телефона и т.д.

Тип `Task` представляет задачу. В данном случае, он определен как `any`, что означает, что тип задачи может быть любым.

Тип `Lesson` представляет информацию о занятии. Он также наследуется от типа `PocketBaseCollection` и добавляет дополнительные поля, такие как `start_time`, `end_time`, `students`, `note`, `price` и `payed`. Эти поля содержат информацию о времени начала и окончания занятия, списке студентов, заметке, цене и галочке, указывающей на оплату.

Тип `Finances` представляет информацию о финансах. Он также наследуется от типа `PocketBaseCollection` и добавляет дополнительные поля `revenue` и `month`, которые содержат информацию о доходе и месяце соответственно.

Также в базе данных присутствуют SQL-запросы, которые обеспечивают функциональность финансовой вкладки репетитора. Эти запросы позволяют получать и обрабатывать финансовые данные, связанные с деятельностью репетитора. Данный запрос представлен в листинге 4.

Листинг 4 – SQL запрос для раздела финансов

```
SELECT
    (ROW_NUMBER() OVER()) as id,
    strftime('%Y-%m', start_time) AS month,
    SUM(price) AS revenue
FROM lessons
WHERE paid = true
GROUP BY month
ORDER BY month;
```

А также есть запрос к ведомости об оценках учениках, который представлен в листинге 5.

Листинг 5 – SQL запрос для раздела оценок

```
SELECT
    (ROW_NUMBER() OVER()) as id,
    user_id,
    GROUP_CONCAT(mark) AS marks
FROM
    tasks
GROUP BY
    user_id
```

Вывод по второй главе

Во второй главе, исходя из требований к поставленной задаче, были определены функциональные и нефункциональные требования к разрабатываемому веб-приложению. Была описана база данных веб-приложения и запросы к ней, диаграмма вариантов использования.

3. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ

3.1. Реализация авторизации на сайте

В разрабатываемом веб-приложении существует авторизация. Репетитор является администратором в базе данных PocketBase, у поэтому у репетитора изначально есть доступ ко всем разделам на сайте.

Для учеников, репетитор создает карточки ученика и создается личный кабинет для каждого ученика внутри этой самой карточки. Фрагмент кода представлен в листинге 6.

Листинг 6 – Реализация авторизации

```
<script lang="ts">
  import { goto } from '$app/navigation';
  import { currentUser, pb } from '$lib/pocketbase';
  import { Label, Input, Checkbox, A, Button } from 'flowbite-svelte';

  let username: string;
  let password: string;
  let isAdmin = false;
  let authForm: HTMLFormElement;
  let isAuth = false;

  async function login() {
    isAuth = true;
    try {
      if (isAdmin) {
        await pb.admins.authWithPassword(username, password);
      } else {
        await pb.collection('users').authWithPassword(username,
password);
      }
    } catch (e: any) {
      authForm.reset();
      isAuth = false;
      alert(e.message);
      return;
    }
    goto('/');
  }
  function signOut() {
    pb.authStore.clear();
  }
</script>
```

Общая цель этого фрагмента кода, состоит в реализации функциональности входа и выхода из системы для пользователей, используя pocketbase в качестве аутентификационной системы.

После этого можно разобрать создание отображение формы на листинге 7 для входа в систему, если пользователь не авторизован.

Листинг 7 – Отображение формы для авторизации

```
{#if !$currentUser}
  <form on:submit|preventDefault={login} bind:this={authForm}>
    <div class="mb-6">
      <Label for="username" class="mb-2">Имя пользователя</Label>
      <Input type="text" id="username" placeholder="username" required bind:value={username} />
    </div>
    <div class="mb-6">
      <Label for="password" class="mb-2">Password</Label>
      <Input type="password" id="password" placeholder="....." required bind:value={password} />
    </div>
    <Checkbox class="mb-6 space-x-1" bind:checked={isAdmin}>Учитель</Checkbox>
    <Button type="submit" disabled={isAuth}>Submit</Button>
  </form>
{:else}
  <p>
    Signed in as {$currentUser.username ?? 'Репетитор'}
    <button on:click={signOut}>Sign Out</button>
  </p>
{/if}
```

Внутри блока {#if !\$currentUser} находится форма для входа в систему, которая будет отображаться, если пользователь не аутентифицирован (\$currentUser является ложным значением).

В начале указано событие on:submit|preventDefault={login}, которое привязано к функции login() и предотвращает отправку формы по умолчанию.

Затем форма содержит несколько разделов с элементами ввода и кнопкой отправки:

<Label> – компонент, отображающий метку для соответствующего поля ввода.

<Input> – компонент ввода текста, связанный с переменными username и password.

<Checkbox> – компонент галочки, связанный с переменной isAdmin.

`<Button>` – кнопка отправки формы с текстом «Submit». Она будет отключена, если пользователь уже аутентифицирован (`isAuth` является истинным значением).

В блоке `{:else}` находится раздел, который будет отображаться, если пользователь уже аутентифицирован (`$currentUser` является истинным значением). В этом разделе выводится информация о текущем пользователе и кнопка выхода из системы.

`<p>` – абзац, содержащий текст «Signed in as» и имя пользователя (если оно доступно) или "Репетитор" в случае отсутствия имени.

`<button>` – кнопка, при нажатии на которую будет вызвана функция `signOut()` для выхода из системы.

Общая цель этого кода состоит в отображении формы для входа в систему, если пользователь не аутентифицирован, и отображении информации о пользователе и кнопки выхода, если пользователь аутентифицирован.

В результате реализована страница с двумя полями для ввода логина и пароля. А также указание того, что авторизуется репетитор и кнопка для авторизации.

Реализация страницы авторизации представлена на рисунке 6.

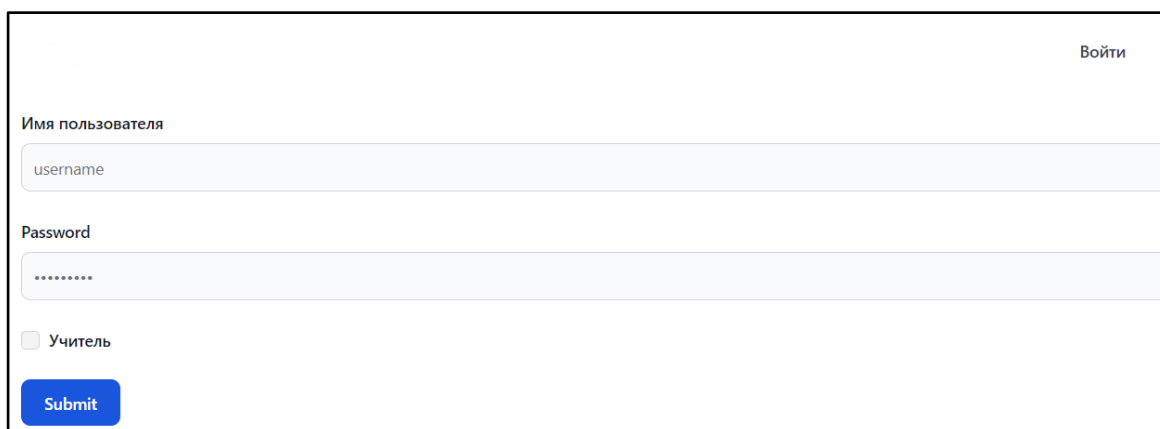


Рисунок 6 – Реализация авторизации на сайте

3.2. Реализация списка учеников у репетитора

Со стороны репетитора имеется такая возможность, как создать карточку ученика. Это нужно для того, чтобы при создании занятия отображался конкретный ученик в каждом занятии. Фрагмент кода реализации можно увидеть на листинге 8.

Листинг 8 – Реализация раздела списка учеников

```
</div>
    <section>
        <div class="flex flex-col gap-3">
            {#each $students.values.filter((student) => stu-
dent.archived === false) as student}
                <StudentCard id={student.id} name={student.name} />
            {:else}
                {#if !$students.isLoading}
                    Пока нет учеников
                {/if}
            {/each}
        </div>
        <ButtonNew on:click={() => goto('/students/new')} />
    </section>
</TabItem>
<TabItem>
    <div slot="title" class="flex items-center gap-2">
        <svg
            aria-hidden="true"
            class="w-5 h-5"
            fill="currentColor"
            viewBox="0 0 20 20"
            xmlns="http://www.w3.org/2000/svg"
        ><path
            d="M5 3a2 2 0 00-2 2v2a2 2 0 002 2h2a2 2 0 002-2v2a2 2 0 00-2 2H5z"
            M11 3a2 2 0 00-2 2v2a2 2 0 002 2h2a2 2 0 002-2v2a2 2 0 00-2 2H11z"/>
        </svg>
```

Этот код является частью шаблона веб-страницы и отображает список студентов.

Сначала создается блок `<div>` с классом «flex flex-col gap-3». Затем используется директива `#each` для прохода по массиву студентов, отфильтрованному по `archived = false`. Для каждого студента создается компонент `StudentCard` с передачей ему `id` и `name` студента.

Если массив студентов пустой и данные еще не загружены, то отображается текст «Пока нет учеников» с помощью директивы `#if`.

В конце блока `<div>` находится компонент `ButtonNew`, который при клике перенаправляет пользователя на страницу создания нового студента.

Код в блоке `<TabItem>` отвечает за отображение вкладки со списком студентов. В блоке `<div>` с классом `«flex items-center gap-2»` находится иконка и текст, который будет отображаться во вкладке.

Реализацию отображения страницы со списком учеников можно увидеть на рисунке 7.



Рисунок 7 – Отображение списка учеников

Также в разделе со списком учеников, репетитор может переносить учеников в архив, что означает, что ученик либо отказался от занятий навсегда, либо на определенный период. Как это реализовано, можно увидеть в листинге 9.

Листинг 9 – Реализация архива учеников

```
</div>
  <section>
    <div class="flex flex-col gap-3">
      {#each $students.values.filter((student) => student.archived === true) as student}
        <StudentCard id={student.id} name={student.name} />
      {:else}
        {#if !$students.isLoading}
          Нет архивных учеников
        {/if}
      {/each}
    </div>
```

Этот код создает список архивированных студентов с использованием фреймворка Svelte.

Первая строка создает блок `<div>` с классом «flex flex-col gap-3». Этот блок будет содержать все компоненты `StudentCard`, которые будут созданы.

Директива `#each` используется для перебора массива учеников, отфильтрованных по `archived = true`. Отфильтрованный массив получается с помощью массива `$students.values` и метода `filter()`. Для каждого ученика в отфильтрованном массиве создается компонент `StudentCard` с передачей `id` и `name` ученика в качестве `props`.

Если в отфильтрованном массиве нет архивированных учеников и данные уже загружены, то выполняется блок `{:else}`. В этом блоке используется директива `#if` для отображения текста «Нет архивных учеников», если переменная `$students.isLoading` равна `false`.

Результат реализации списка архивированных учеников можно увидеть на рисунке 8.

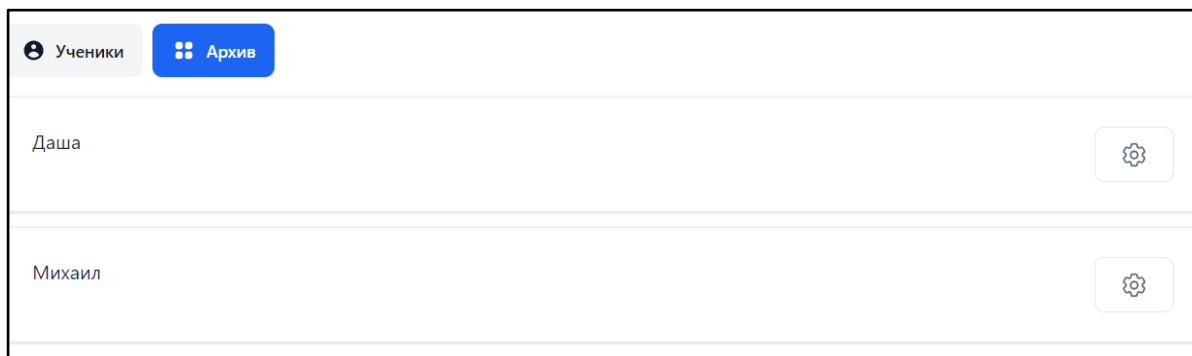


Рисунок 8 – Реализация списка архивированных учеников

3.3. Реализация создания занятия

Со стороны репетитора, одной из основных возможностей является создание занятий. Для создания занятия необходимо указать время начала занятия и его окончания, а также указать, с каким именно учеником проходит занятие. Реализацию этой части для создания занятия можно увидеть в листинге 10.

Листинг 10 – Часть реализации создания занятия с указанием времени и выбором ученика

```
async function createLesson() {
  if (!selected) {
    alert('Выберите ученика');
    return;
  }
  isCreatingLesson = true;
  const start = new Date(date);
  const end = new Date(date);

  const [startHours, startMinutes] = startTime.split(':');
  const [endHours, endMinutes] = endTime.split(':');

  start.setHours(+startHours);
  start.setMinutes(+startMinutes);

  end.setHours(+endHours);
  end.setMinutes(+endMinutes);
}
```

Также для занятия есть ограничение, что начало занятия не может быть позже его окончания. Реализацию этого ограничения можно увидеть на листинге 11.

Листинг 11 – Реализация создания занятия

```
if (generate) {
  const start = new Date(generateDateStart);
  const end = new Date(generateDateEnd);

  if (start > end) {
    alert('Дата начала не может быть позже даты окончания');
    return;
  }
  while (start <= end) {
    pb.collection('lessons').create(
      {
        start_time: start,
        end_time: new Date(start.getTime() +
(end.getTime() - start.getTime())),
        note,
        price,
        students: [selected.value]
      },
      { $autoCancel: false }
    );
    start.setDate(start.getDate() + 7);
  }
  goto('/');
  return;
}
```

Данный код создает ограничения при создании занятия. При создании дата начала не может быть позже даты окончания занятия. Также этот код выполняет создание нескольких уроков в базе данных, если переменная `generate` имеет значение `true`. Он создает уроки в интервале от начальной даты `generateDateStart` до конечной даты `generateDateEnd`, увеличивая дату на 7 дней после каждого созданного урока. Это нужно для генерации занятий, если у репетитора сформировано повторяющееся расписание с учеником, чтобы не проставлять каждое занятие вручную.

Увидеть реализацию можно увидеть на рисунках 9–11.

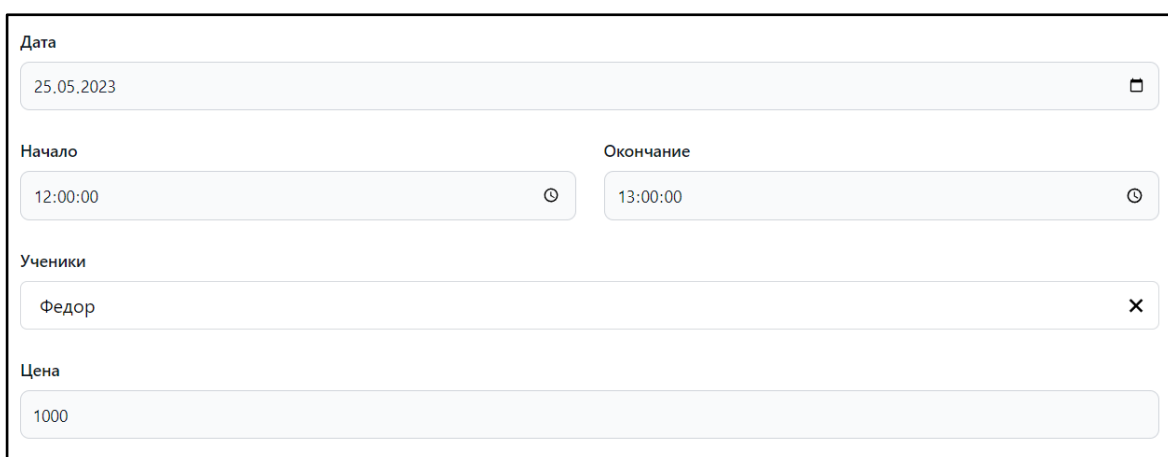


Рисунок 9 – Создание занятия

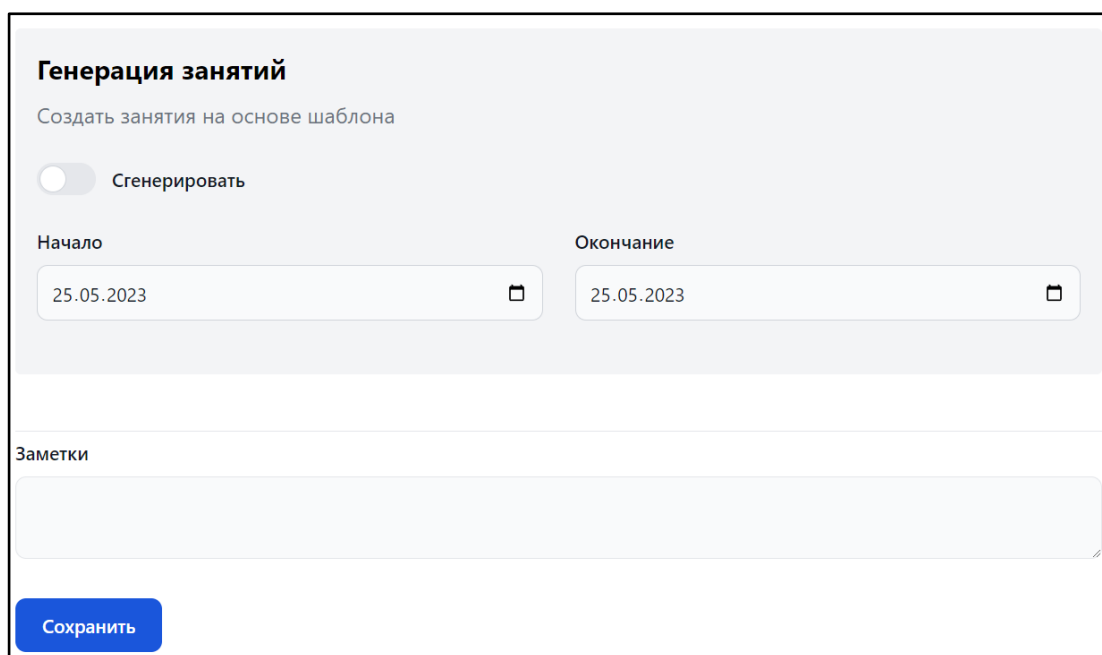


Рисунок 10 – Создание генерации занятий

Расписание

воскресенье, 23 апреля

🕒 07:00 - 08:00

✓

Михаил

среда, 26 апреля

🕒 12:00 - 12:00

✓

Тимур

четверг, 27 апреля

🕒 21:00 - 04:00

✓

Михаил

+

Рисунок 11 – Список созданных занятий

3.4. Реализация домашнего задания

Данный раздел в веб-приложении реализован для того, чтобы репетитор задавал домашнее задание ученикам, для оценки их прогресса, через домашнее задание.

Реализацию можно увидеть в листинге 12.

Листинг 12 – Реализация раздела домашнего задания

```
let isCreating = false;
let note = '';
let selected: { value: string; label: string };
async function createTask() {
    if (!selected) {
        alert('Выберите ученика');
        return;
    }
    if (!formData) {
        formData = new FormData();
    }
    isCreating = true;
    formData.append('note', note);
    formData.append('user_id', selected.value);
    await pb.collection('tasks').create(formData);
    isCreating = false;
    goto('/tasks');
}
```

Данный код отвечает за создание задачи и содержит переменные и функцию для этой цели. В функции `createTask()`, после проверки вы-

бранного ученика, данные задачи добавляются в объект `formData`. Затем вызывается метод `create()` для создания задачи с использованием объекта `formData`. После завершения операции создания, пользователь перенаправляется на страницу с задачами.

Отображение страницы в веб-приложении показано на рисунке 12.



Студент	Дата
Тимур	13 апреля 2023 г.
Тимур	16 апреля 2023 г.
Михаил	21 апреля 2023 г.
Михаил	23 апреля 2023 г.

Рисунок 12 – Отображение раздела с заданиями

В листинге 13 код отображает форму для создания задачи. В форме присутствуют поля для выбора ученика, загрузки файлов и ввода заметок. При отправке формы вызывается функция `createTask()`, которая обрабатывает данные формы и создает задачу. Во время создания задачи отображается спиннер, а кнопка сохранения становится неактивной.

Листинг 13 – Создание задания

```
<form on:submit|preventDefault={createTask}>
  <div class="mb-6">
    <Label for="students" class="mb-2">Ученик</Label>
    {#await $students}
      <Select placeholder="Загрузка..." />
    {:then students}
      {@const items = students.values.map((student) => ({
        value: student.id,
        label: student.name
      })))}
      <Select id="students" {items} bind:value={selected} placeholder="Выберите ученика..." />
    {/await}
  </div>
  <div class="mb-6">
    <Label class="pb-2" for="multiple_files">Файлы задания</Label>
    <form bind:this={fileForm}>
```

```

        <Fileupload id="multiple_files" multiple
on:change={onFileuploadChange} />
    </form>
</div>

<div class="mb-6">
    <Label for="notes" class="mb-2">Заметки</Label>
    <Textarea id="notes" placeholder="" bind:value={note} />
</div>

</form>

```

На рисунке 13 можно увидеть, как реализовано создание домашнего задания.

Ученик

Даша

Файлы задания

Выбрать файлы

Файл не выбран

Заметки

Сохранить

Рисунок 13 – Отображение создание задания

На рисунке 14 представлен архив заданий. В него попадают те задания, которые оценил репетитор.

Задания

Архив

Оценки

Архив

Михаил	12 апреля 2023 г.
Михаил	21 апреля 2023 г.
Михаил	24 апреля 2023 г.

Рисунок 14 – Отображение архива заданий

На рисунке 15 можно увидеть сформированную ведомость по оценкам, которые ученики получили за свои домашние задания. У репетитора отображается полный список со всеми учениками. У учеников в свою очередь отображение только своих отметок.

Оценки					
Тимур	0	0			
Михаил	5	4	0	0	1

Рисунок 15 – Отображение оценок учеников

3.5. Реализация финансов

Раздел финансов сделан для того, чтобы репетитору показывать результаты его работы на протяжении каждого месяца. Исходя из этих данных, можно увидеть свои лучшие месяцы по заработной плате.

Реализацию списка оплат занятий можно увидеть на листинге 14.

Листинг 14 – Реализация списка оплат

```
{#each filtered as group}
  <h2 class="text-xl font-semibold place-self-center my-4">{formatDay(group.date)}</h2>
  <div class="flex flex-col gap-2">
    {#each group.values as lesson (lesson.id)}
      <Card class="flex justify-between">
        <div class="flex gap-5">
          <div class="flex gap-2">
            <Icon class="w-6 h-6" src={Clock} />
            {formatTime(lesson.start_time, lesson.end_time)}
          </div>
          <div class="flex gap-2">
            <span class="text-ellipsis overflow-hidden">
              {lesson.students.join(', ')}
            </span>
          </div>
        </div>
      </Card>
    </div>
  </div>
```

Данный код отображает список уроков, разделенных по группам и отсортированных по дням. Для каждой группы уроков выводится заголо-

вок с датой. Внутри каждой группы отображается карточка (Card), содержащая информацию о времени урока и списка учеников.

Отображение раздела показано на рисунке 16. На нем отображается расписание, но только с возможностью отметки об оплате.

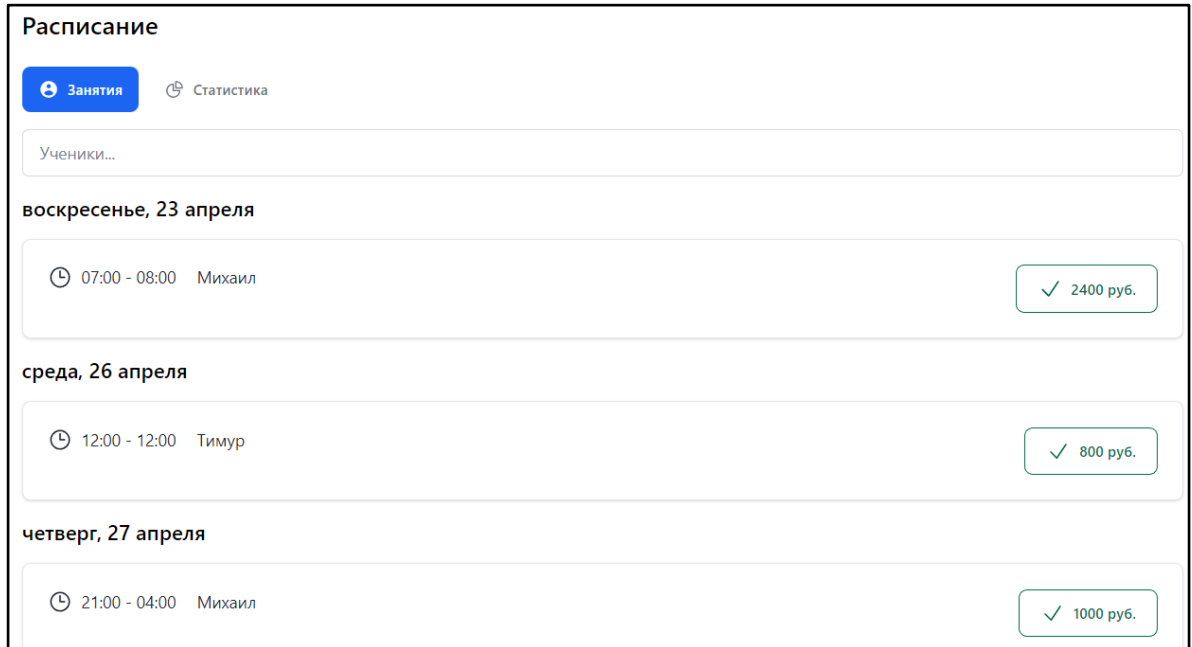


Рисунок 16 – Расписание с возможностью отметки оплаты

В разделе финансов реализован подраздел со статистикой. Где отображается доход репетитора за каждый месяц.

Реализацию можно увидеть в листинге 15.

Листинг 15 – Реализация статистики дохода

```
</div>
<section>
  <!-- bar chart -->
  {#await pb.collection('finances').getList(1, 3, { sort: 'month' })}
  <div class="flex justify-center my-6">
    <Icon class="w-6 h-6 animate-spin" src={Clock} />
  </div>
  {:then finances}
  <div class="w-full md:w-1/2 mx-auto">
    <BarChart
      labels={finances.items.map((i) => i.month)}
      values={finances.items.map((i) => i.revenue)}
    />
  </div>
  {/await}
</section>
```

Данный код отображает вкладку «Статистика» с графиком столбцов (BarChart). Внутри вкладки происходит запрос к базе данных (`pb.collection('finances').getList(1, 3, { sort: 'month' })`) для получения данных о финансовых показателях. Если данные еще загружаются, отображается анимация загрузки (`Icon class="w-6 h-6 animate-spin" src={Clock}`). После загрузки данных, график столбцов отображается с помощью компонента BarChart, где на оси X отображаются месяцы, а на оси Y отображается выручка (revenue) для каждого месяца.

Как выглядит раздел со статистикой можно увидеть на рисунке 17.

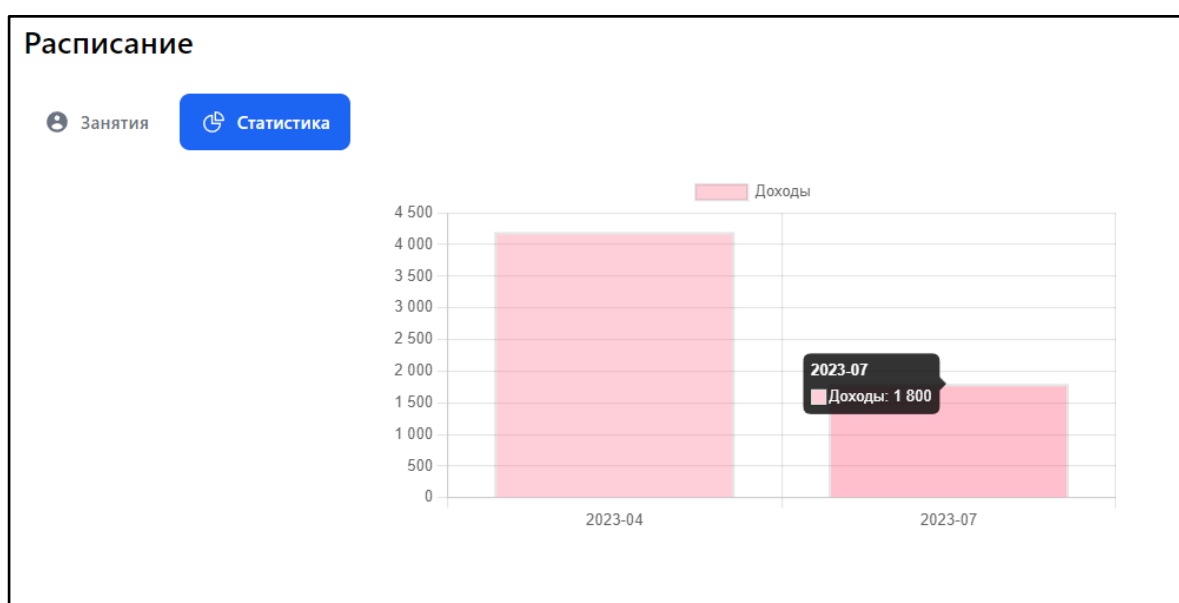


Рисунок 17 – Отображение статистики

Вывод по третьей главе

В рамках данной работы был разработан веб-сайт для репетитора, соответствующий поставленной цели. Для создания адаптивного и отзывчивого пользовательского интерфейса веб-сайта были использованы JavaScript и TypeScript. Также были применены SQL запросы для взаимодействия с базой данных.

Результаты разработки веб-сайта представлены в виде скриншотов страниц веб-сайта и примеров исходного кода.

4. ТЕСТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

Одним из важных шагов после завершения верстки страниц и реализации функциональности веб-сайта является проведение тестирования разработанного веб-сайта.

Тестирование проводилось в два этапа:

- 1) функциональное тестирование;
- 2) usability-тестирование.

4.1. Функциональное тестирование

При выполнении функционального тестирования проводится проверка всех функциональных задач, описанных в требованиях. В таблице 2 приведены результаты функционального тестирования. Обнаруженные результаты тестирования совпадают с ожидаемыми результатами, что свидетельствует о правильной реализации функциональных требований.

Таблица 2 – Функциональное тестирование

№	Название теста	Ожидаемый результат	Полученный результат	Тест пройден?
1.	Авторизоваться	Пользователь успешно авторизован	Пользователь успешно авторизован	Да
2.	Пропустить поле логина	Ошибка авторизации	Ошибка авторизации	Да
3.	Ввести неправильный пароль	Ошибка авторизации	Ошибка авторизации	Да
4.	Создать занятие	Занятие создано	Занятие создано	Да
5.	Генерация занятий	Занятие продублировано на каждый выбранный день недели в выбранном интервале	Занятие продублировано на каждый выбранный день недели в выбранном интервале	Да
6.	Создание ученика	Ученик добавлен в список	Ученик добавлен в список	Да
7.	Создание задания	Задание создано и отображается у ученика	Задание создано и отображается у ученика	Да
8.	Оценить задание	Задание перешло в архив выполненных и оценка появилась в ведомости	Задание перешло в архив выполненных и оценка появилась в ведомости	Да

№	Название теста	Ожидаемый результат	Полученный результат	Тест пройден?
9.	Отметить занятие как оплаченное	Появилась галочка напротив занятия. В статистике по финансам сумма заработной платы увеличилась на сумму занятия	Появилась галочка напротив занятия. В статистике по финансам сумма заработной платы увеличилась на сумму занятия	Да
10.	Выйти из профиля	Пользователь вышел из профиля и попал на страницу авторизации	Пользователь вышел из профиля и попал на страницу авторизации	Да

4.2. Usability-тестирование

В ходе usability-тестирования была оценена удобность и эффективность использования системы. В тестировании участвовали два человека, которым были предложены определенные задачи. Задачи включали авторизацию в системе, создание занятий, генерацию занятий, добавление ученика в список, задание логина и пароля для ученика, оценку задания и отметку занятия как оплаченного.

Тестировщики успешно выполнили все предложенные задачи без затруднений и неудобств. Это свидетельствует о том, что система прошла usability-тестирование успешно, и ее использование является удобным и эффективным.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы было разработано веб-приложение для репетиторов, которое объединяет различные функции и обеспечивает удобство использования как для репетиторов, так и для учеников. Основные цели проекта были достигнуты.

1. Был проведен анализ предметной области, что позволило определить требования и функциональность приложения.
2. Создан дизайн веб-сайта, который учитывает эстетические и пользовательские аспекты.
3. Выполнена верстка страниц, обеспечивающая правильное отображение и взаимодействие элементов интерфейса.
4. Разработана база данных для хранения и управления информацией, связанной с репетиторской деятельностью.
5. Реализованы функциональные возможности веб-приложения, такие как авторизация, создание занятий, работа с учениками и финансовым учетом.
6. Проведено тестирование веб-сайта, чтобы убедиться в его правильной работе и соответствии требованиям.

В будущем планируется продолжить разработку и внедрение улучшений в функциональность разработанного веб-приложения.

ЛИТЕРАТУРА

1. Дефицит учителей в стране. [Электронный ресурс] URL: https://ria.ru/sn_edu/20171010/1506540004. (дата обращения: 08.02.2023 г.).
2. Учеба в современной школе. [Электронный ресурс] URL: https://mel.fm/vazhny_razgovor/5217086-useless. (дата обращения: 08.02.2023 г.).
3. Крылова Н. Г. Репетиторы выходят из тени. // Народное образование. 2011. – 184 с.
4. Определение образования. [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Образование>. (дата обращения: 08.02.2023 г.).
5. Буркина, Т.А. Анализ внешней среды организации: Методические рекомендации к выполнению контрольной работы. // Самар. гос. техн. ун-т, филиал в г. Сызрани; Сызрань: СфСамГТУ, 2007. – 21 с.
6. Зуб, А.Т. Стратегический менеджмент. Теория и практика. Учебное пособие для вузов. – М.: Аспект Пресс, 2006. – 415 с.
7. Журавлева Л.В. Образовательный продукт: понятие и ценность. // Вестн. Том. гос. ун-та, 2009. – № 321.
8. Сударкина Х.В. Маркетинг образовательных услуг вуза: особенности, возможности, перспективы / Х.В. Сударкина // Известия ЮФУ. Технические науки, 2012. – № 8. – С. 138–142.
9. GeekBrains. [Электронный ресурс] URL: gb.ru (дата обращения: 05.05.2023 г.).
10. Ulearn. [Электронный ресурс] URL: ulearn.me (дата обращения: 05.05.2023 г.).
11. Skillbox. [Электронный ресурс] URL: skillbox.ru (дата обращения: 05.05.2023 г.).
12. Vercel [Электронный ресурс] URL: <https://vercel.com/> (дата обращения: 05.05.2023 г.).

13. PocketBase. [Электронный ресурс] URL: <https://pocketbase.io/> (дата обращения: 05.05.2023 г.).
14. Головатый А. Django. Подробное руководство // под ред. Адриан Головатый, Джейкоб Каплан. // Мос. СПб.: Питер, 2010 г. – 97 с.
15. Современный учебник JavaScript. [Электронный ресурс] URL: <https://learn.javascript.ru> (дата обращения 16.04.2023 г.).

ПРИЛОЖЕНИЕ. Отображение веб-сайта со стороны ученика

При авторизации ученика в свой личный кабинет у него сразу же отображается расписание занятий, которые были в прошлом, так и занятия, которые будут в будущем с указанием даты. Также в профиле у ученика отображается, отметил ли репетитор, что занятие оплачено или нет. Этим могут воспользоваться родители, для контроля. Отображение расписания из профиля ученика представлено на рисунке 1.

The screenshot shows a web interface titled "Расписание" (Schedule). It lists lessons for four dates: Sunday, April 23; Thursday, April 27; Wednesday, July 5; and Wednesday, July 19. Each lesson entry includes a clock icon, a time range, a status icon (green checkmark or red X), and the name "Михаил".

Дата	Время	Статус	Репетитор
воскресенье, 23 апреля	07:00 - 08:00	✓	Михаил
четверг, 27 апреля	21:00 - 04:00	✓	Михаил
среда, 5 июля	05:00 - 05:00	✗	Михаил
среда, 19 июля	05:00 - 05:00	✓	Михаил

Рисунок 1 – Отображение расписания ученика

Также ученику доступен раздел с домашним заданием, которое ему задал репетитор (рисунок 2).

Ученик имеет возможность просматривать свои предыдущие домашние работы (рисунок 3) и посмотреть примечания репетитора к ним (рисунок 4).

А также он может контролировать свою успеваемость, для этого ему необходимо посетить раздел «Оценки» (рисунок 5).



Рисунок 2 – Отображение предстоящих домашних работ



Рисунок 3 – Отображение работ, которые уже оценили

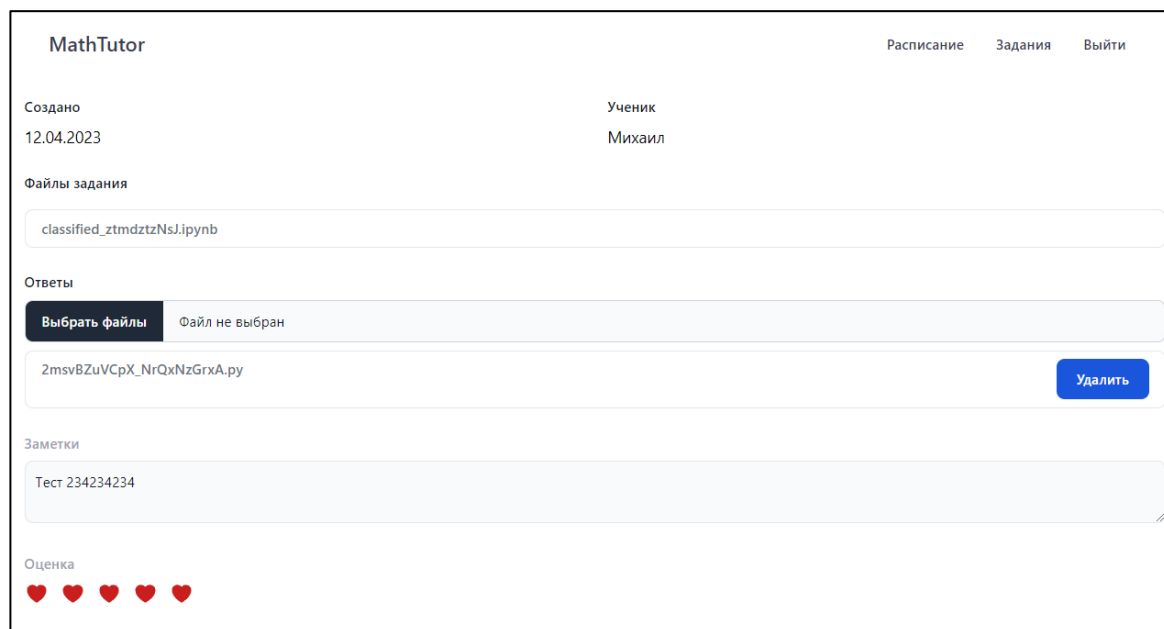


Рисунок 4 – Отображение домашнего задания

Задания

Архив

Оценки

Оценки

Михаил	5	4	0	0	1
--------	---	---	---	---	---

Рисунок 5 – Отображение оценок в профиле ученика