ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ТЮМЕНСКОЙ ОБЛАСТИ

Государственное автономное профессиональное образовательное учреждение

Тюменской области

«КОЛЛЕДЖ ЦИФРОВЫХ И ПЕДАГОГИЧЕСКИХ ТЕХНОЛОГИЙ»

Курсовая работа

по специальности 09.02.07 Информационные системы и программирование

МДК 02.02 Инструментальные средства разработки программного обеспечения

РАЗРАБОТКА WEB-СЕРВИСА ДЛЯ ФИТНЕС-ТРЕНИНГА

|  |  |
| --- | --- |
|  | Выполнил:  студент Сытдыкова Сабина Рамильева  (подпись) |
| Работа защищена  « » 20 г.  с оценкой « » | Руководитель Приходько Дарья Владимировна  (подпись) |

Тюмень, 2025

**Содержание**

[Введение 3](#_Toc198233584)

[Глава 1 Теоретические основы web-разработки в фитнес-тренинге 5](#_Toc198233585)

[1.1 Основные аспекты цифровизации в сфере фитнеса 5](#_Toc198233586)

[1.2 Аналитический обзор существующих решений для фитнес-тренинга 8](#_Toc198233587)

[1.3 Средства и технологии разработки 11](#_Toc198233588)

[Глава 2 Проектирование «Web-сервиса для фитнес-тренинга» 17](#_Toc198233589)

[2.1 Реализация хранения данных в «web-сервисе для фитнес-тренинга» 17](#_Toc198233590)

[2.2 Проектирование функциональной части web-сервиса 20](#_Toc198233591)

[2.3 Проведение отладки и тестирования готово решения 24](#_Toc198233592)

[Заключение 29](#_Toc198233593)

[Список использованных источников 32](#_Toc198233594)

[Приложение А 34](#_Toc198233595)

[Проектирование функциональной части web-сервиса 34](#_Toc198233596)

[Приложение Б 37](#_Toc198233597)

[Интерфейс web-сервиса 37](#_Toc198233598)

## Введение

Современное общество сталкивается с рядом проблем, таких как малоподвижный образ жизни, стресс, увеличение числа заболеваний, связанных с образом жизни. В условиях цифровизации и автоматизации все больше людей ведут сидячий образ жизни, что становится причиной таких заболеваний, как ожирение, гипертония, сердечно-сосудистые и опорнодвигательные патологии. В этих условиях физическая активность становится важным инструментом для профилактики и улучшения здоровья. Технологии затронули почти каждую сферу нашей жизни, не осталась в стороне и физическая культура. Одна из самых актуальных тенденций – диджитализация. С появлением умных устройств и мобильных приложений стало проще отслеживать физическую активность и корректировать программы тренировок.

С помощью цифровых технологий мы можем контролировать физическую активность, достигать лучших результатов в физической подготовке и тренировках, а также оптимизировать процесс планирования и проведения физкультурных занятий. Помимо этого, цифровые технологии, такие как дистанционные тренировки и онлайн занятия, делают физическую активность более доступной, особенно для тех, кто не имеет возможности посещать спортивные залы и клубы.

Web-сервис для фитнес-тренинга может помочь отслеживать тренировочный процесс. Предоставляет готовые программы тренировок и возможность расчета суточной нормы потребляемых калорий, учитывая цели пользователя. Используя web-сервис для фитнес-тренинга, пользователь не привязан ко времени или месту проведения тренировок.

Целью данной курсовой работы является разработка web-сервиса для фитнес-тренинга.

Для достижения цели, мы поставили следующие задачи:

1. Выявить основные аспекты цифровизации в сфере фитнеса.
2. На основе проведенного анализа существующих решений выделить функциональные требования web-сервиса для фитнес-тренинга.
3. Определить средства и технологии разработки.
4. Реализовать хранение данных в «web-сервисе для фитнес-тренинга».
5. Спроектировать функциональную часть web-сервиса.
6. Провести тестирование и отладку готово решения.

# 

## Глава 1 Теоретические основы web-разработки в фитнес-тренинге

## 1.**1 Основные аспекты цифровизации** **в сфере фитнеса**

Сегодня информационные технологии (ИТ), прочно вошли в нашу жизнь и закрепились в ней. Современный мир просто невозможно представить без ИТ, свое применение они нашли во всех сферах человеческой жизни. ИТ — это всегда процедура поиска, сбора, хранения, обработки, предоставления и распространения информации [7]. Это способствует повышению эффективности развития в той или иной сфере деятельности человечества. ИТ способствуют оптимизации процессов обработки информации, снижению трудоемкости использования ресурсов, применению новых форм информационной поддержки любых видов деятельности, при этом экономятся затраты труда, времени, энергии, вещественных средств. Они представляют собой важнейшую часть процесса использования информационных ресурсов общества. Основной целью ИТ является снижение трудоемкости использования информационных ресурсов.

В условиях ускоряющегося темпа жизни, роста информационных технологий и урбанизации возрастает необходимость ведения активного образа жизни и укрепления здоровья через физическую активность. В последние годы физическая культура претерпевает значительные изменения, обусловленные глобальными тенденциями и локальными вызовами. Современное общество сталкивается с рядом проблем, таких как малоподвижный образ жизни, стресс, ухудшение экологической ситуации и увеличение числа заболеваний, связанных с образом жизни. В условиях цифровизации и автоматизации все больше людей ведут сидячий образ жизни, что становится причиной таких заболеваний, как ожирение, гипертония, сердечно-сосудистые и опорно-двигательные патологии. В этих условиях физическая культура становится важным инструментом для профилактики и улучшения здоровья. Физическая активность способна улучшить не только физическое, но и психоэмоциональное состояние человека. Сегодня физическая культура рассматривается как важный фактор формирования устойчивости к стрессу, поддержания высокого уровня энергичности и общего качества жизни [4].

Одна из самых актуальных тенденций – диджитализация. С появлением умных устройств и информационных сервисов стало проще отслеживать физическую активность и корректировать программы тренировок. Инновационные технологии в системе оздоровительной физической культуры появляются и разрабатываются в большом количестве. Значительная часть оздоровительных технологий связана с развитием именно фитнес-индустрии. Благодаря данному направлению оздоровления организма человека в физической культуре появился такой популярный термин, как «фитнес-технологии». Фитнес-технологии отличаются тем, что направленны, прежде всего на результативность при занятии фитнесом с использованием инновационных средств, методов и современного оборудования [9].

Не все потенциальные клиенты сферы фитнеса могут позволить себе выделить время или средства для посещения фитнес-клубов, многие предпочитают формат домашних тренировок, либо тренировок на улице. Именно поэтому, современные гаджеты, методики и технологии дистанционного фитнеса и домашних онлайн тренировок – это одно из направлений, обладающих огромным потенциалом развития. Онлайн тренировки удовлетворяют такие человеческие потребности как комфорт, обособленность и индивидуальность подходов к построению тренировочного процесса. Помимо этого, в последнее время рост спроса на фитнес-услуги перемещается из столичных городов в регионы. В связи с отсутствием достаточного количества фитнес-предприятий в некоторых отдаленных регионах, занятия по системе онлайн-ведения повышают свою популярность [2].

Цифровые технологии в сфере фитнес тренинга помогают решать следующие задачи:

* предоставление информации о правильной технике выполнения упражнений, питании, восстановлении и других аспектах фитнеса;
* создание и подбор персонализированных тренировочных программ;
* организация индивидуального расписания для тренинга;
* отслеживание прогресса на тренировках, измерениям и общему состоянию здоровья;
* отсутствие привязки к спортивному инвентарю, времени, месту и формату проведения тренировок.

Всё вышеперечисленное позволяет улучшить качество тренировочного процесса пользователей, что способствует повышению уровня заботы о здоровье и общему благополучию сообщества.

Таким образом, рассмотрев вопрос о основных аспектах цифровизации в фитнес-тренинге можно сделать вывод о том, что ИТ оказывают значительное влияние на различные аспекты жизни общества. С развитием Интернета веб-технологии распространяются с огромной скоростью, и сегодня они стали неотъемлемой частью нашей повседневной жизни. Постоянное развитие цифровых технологий открыло новые возможности для совершенствования и модернизации огромного количества сфер, в том числе и сферы фитнеса. Основным фактором в разработке веб-сервисов является обратная связь от целевой аудитории, удовлетворение запросов которой должно быть в центре разработки. Автоматизация тренировочного процесса значительно упрощает процесс планирования тренировочных нагрузок и понижает вероятность получения травмы или патологических изменений.

# 1.2 Аналитический обзор существующих решений для фитнес-тренинга

На фоне возникшей пандемии в мире спорта произошёл активный толчок к развитию и применению цифровых технологий. На время ограничений посетители спортивных клубов нашли альтернативные способы поддержания формы. Внедрение технологий спровоцировало активный рост сервисов и приложений, которые позволяют тренироваться дома. Интеграция информационных технологий и физической культуры приносит как предпринимателям, так и потребителям набор выгод. Для предпринимателей это открывает широкие перспективы, позволяя формировать новые рынки и предлагать инновационные продукты и услуги. Персонализированные решения повышают эффективность тренировок, а разнообразие бизнес-моделей предоставляет свободу выбора оптимального пути. Для потребителей ИТ технологии предоставляют доступ к информации о физической активности и питании. Персонализированные программы тренировок учитывают индивидуальные потребности, а соревнования и социальное взаимодействие могут способствовать повышению мотивации. В настоящее время онлайн-фитнес растёт в среднем на 33,1% в год, и прогнозируется, что к 2027 году его объём достигнет 59 млрд долларов [9]. Главное основание для разработки web-сервиса – возможность предоставления тренировочных программ, которые учитывают индивидуальные особенности и цели пользователей, повышая эффективность тренировок. Примеры web-сервисов для фитнес-тренинга приведены в Таблице 1.1.

Таблица 1.1 – Анализ готовых решений для фитнес-тренинга

| Критерий  Решение | Alo Moves | WowFit | FitStars |
| --- | --- | --- | --- |
| Поддержка русского языка | – | + | + |
| Узконаправленный выбор программ | + | – | – |
| Возможность подбора тренера | + | + | – |
| Оффлайн-доступ | + | – | + |
| Поддержка русского языка | – | + | – |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий  Решение | Alo Moves | WowFit | FitStars |
| Интеграция с гаджетами | + | + | – |
| Отслеживание прогресса | + | + | + |
| Сообщество/форум | + | – | + |
| Цена подписки (в месяц) | $20 | $15 | $10 |
| Рейтинг (из 5) | 4.8 | 4.5 | 4.2 |

На основе проведенного анализа были выделены следующие функциональные требования для «Web-сервиса для фитнес-тренинга»:

1. Авторизация и аунтификация:
   1. Приложение должно иметь возможность регистрации новых пользователей.
   2. Должна быть предусмотрена процедура проверки подлинности, путем сравнения введенных данных с данными, сохраненными в БД.
2. Профиль пользователя:
   1. Пользователь должен иметь возможность настроить свой профиль, включая параметры, такие как возраст, пол, вес и фитнес-цели.
3. Профиль администратора:
   1. Функционал для ведения отчетности web-сервиса (просмотр активности и профилей пользователей).
   2. Удаление и добавление пользователей.
4. Тренировки:
   1. Web-сервис должен предлагать персонализированные тренировки, основанные на фитнес-целях пользователя и предпочтениях. Возможны тренировки в зале, дома, на открытом воздухе или с использованием специального оборудования.
   2. Предоставление текстовых и видео инструкций.
5. Рекомендации по питанию:
   1. Возможность расчета количества потребляемых калорий для достижения целей пользователя, учитывая его физические данные и образ жизни.
6. Отслеживание прогресса:
   1. Web-сервис должен предоставлять возможность отслеживать прогресс пользователя в достижении его фитнес-целей.

К нефункциональным требования относятся:

1. Производительность:
   1. Web-сервис должен обеспечивать быструю загрузку страниц и отзывчивость интерфейса.
   2. Минимизация задержек при выполнении операций и запросов к БД.
2. Надежность:
   1. Web-сервис должен быть стабильным и не подвержен частым сбоям.
   2. Защищен от утечки данных.
3. Удобство использования:
   1. Интуитивно понятный интерфейс для пользователей.
   2. Удобная навигация и взаимодействие с web-сервисом.
4. Совместимость и масштабируемость:
   1. Возможность масштабирования функционала web-сервиса для потенциального расширения.
5. Адаптивный дизайн:
   1. Поддержка различных размеров экранов и ориентаций устройств для обеспечения удобного использования на разных устройствах.

Таким образом, проведенный обзор цифровых решений для фитнес-тренинга показал, что облачные сервисы создают виртуальную среду, единое цифровое пространство, вокруг каждого пользователя с помощью которого возможен обмен, мониторинг, интеллектуальный и статистический анализ данных. Существующие решения лишь частично удовлетворяют запросы пользователей. Разработка web-сервиса для фитнес-тренинга требует тщательного анализа потребностей целевой аудитории, а также учета трендов и конкурентного окружения. Ключевым аспектом разработки является обеспечение безопасности использования web-сервиса. Политика безопасности должна быть нацелена на защиту персональной информации пользователя на протяжении всего периода использования данных.

# 1.3 Средства и технологии разработки

С каждым годом наблюдается рост использования информационных технологий, что сопровождается увеличением спроса на них и расширением спектра предлагаемых возможностей. В настоящий момент различные виды сервисов, ориентированные на использование широким кругом пользователей, предоставляют возможность удаленной работы с ними. Дистанционный доступ можно организовать различными способами. Классическим вариантом организации дистанционного доступа является клиент-серверная архитектура системы. Web-сервис, разработанный для бизнеса или других целей, должен обладать высокой функциональностью и обеспечивать максимальную совместимость с различными платформами и устройствами [3]. Ключевую роль привлечения посетителей на web-сервис выполняет графическое оформление. Дизайн отвечает за легкость использовании и навигации, а также за то, чтобы весь контент был доступен и понятен для пользователей.

Инструментом для разработки дизайна web-сервиса был выбран онлайн-сервис – Figma. Главное преимущество Figma – кроссплатформенность, не перегруженность инструментария и инновационная особенность, заключающаяся в возможности осуществления параллельной работы нескольких пользователей в одном проекте. Figma объединяет в себе все нужные функции для реализации дизайна: работа с графикой, проектирование UX, создание макетов прототипирование, работа с компонентами и многое другое [10]. Данное программное обеспечение является комбинацией инструментов, платформ и приложений, позволяющей создавать логичный и простой дизайн с точки зрения UX/UI.

Frontend-разработка представляет собой процесс создания пользовательского интерфейса и интерактивных компонентов веб-приложения, обеспечивающих непосредственное взаимодействие с конечными пользователями [8]. Для реализации клиентской части был выбран современный JavaScript-фреймворк Vue.js. Vue.js – это прогрессивная структура, которая может быть использована для создания пользовательских интерфейсов. Его основная библиотека ориентирована только на определенный слой представления, и ее легко сочетать и интегрировать с другими библиотеками или уже существующими проектами. Vue использует такую концепцию как Virtual DOM, поддерживается серверный рендеринг, возможность использовать JSX и т.д. Основными концепциями Vue являются:

* конструктор;
* компоненты;
* директивы;
* переходы.

Vue.js подходит для разработки большинства пользовательских интерфейсов и является отличным выбором для разработки IT проекта любой сложности, благодаря хорошей гибкости, читаемости и простоте API, а инструменты, которые предоставляет данный фреймворк, позволят эффективнее разрабатывать компоненты с наиболее положительным опытом для разработчика [5].

В качестве CSS использована технология Tailwind CSS, который предоставляет более удобный и модульный подход к конструированию интерфейсов, что позволяет легко адаптировать стиль под конкретные требования проекта. Идея использования tailwind заключается в том, чтобы прописывать стили напрямую в директиву class, а не под селекторами в CSS-файле. Tailwind поддерживает более редкие и продвинутые свойства: анимации, радиус закругления рамок, свойства transform, letter-spacing и им подобные. Помимо самого фреймворка, разработчики создали коллекцию уже готовых элементов на основе Tailwind, которые можно добавить на свой сайт и использовать в качестве шаблона для дальнейшей разработки сайта или приложения. Основное преимущество Tailwind CSS заключается в том, что он предоставляет большой набор готовых классов. Это значительно сокращает время на создание пользовательского интерфейса, так как не нужно писать каждый раз новые стили для каждого элемента. Использование готовых классов позволяет легко и быстро изменять внешний вид элементов, не затрагивая их структуру.

Для разработки backend (серверной части), хранения и обработки данных web-сервиса выбрана облачная платформа Firebase. Firebase, инновационная платформа от Google, которая обеспечивает набор инструментов для разработки мобильных и веб-приложений, а также для проведения аналитики. База данных (БД) Firebase Realtime (RTDB) позволяет разработчикам хранить данные в облачной базе данных, к которой могут получать доступ и обновлять ее в режиме реального времени несколько пользователей. Эта функция особенно полезна для разработки приложений для совместной работы, когда нескольким пользователям необходимо получать доступ к одним и тем же данным и изменять их. Firebase Cloud Messaging (FCM) — мощный инструмент для отправки сообщений и уведомлений пользователям. FCM позволяет разработчикам отправлять пользователям персонализированные сообщения и уведомления, которые можно использовать для улучшения пользовательского опыта и повышения вовлеченности. FCM также предоставляет надежную и масштабируемую инфраструктуру для отправки сообщений, гарантируя быструю и эффективную доставку сообщений. В зарубежной интерпретации Firebase часто используется как пример успешной платформы для создания мобильных приложений. Она упоминается в девятикратном использовании облачных сервисов, таких как AWS и Azure, для создания и развертывания приложений [1].

В качестве среды разработки выбран текстовый редактор Visual Studio Code (VS Code). VS Code был разработан компанией Microsoft. Это облегчённый редактор кода, который содержит такие функции, как подсветка синтаксиса, сопоставление скобок, автоматические отступы, выделение блоков, фрагментов кода и т. д. Главная особенность, которая выделяет VS Code, — это функция отладки кода. Она помогает пользователю находить ошибки в коде, устанавливая соответствующие расширения. В ней также есть поддержка Git, которая позволяет пользователю работать с системой контроля версий, не выходя из редактора. VS Code обладает интуитивно понятным интерфейсом, что упрощает навигацию и настройку для начинающих и опытных разработчиков. Независимо от типа проекта, VS Code предоставляет инструменты для оптимизации рабочего процесса. Различные функции, такие как отладчик, IntelliSense, интеграция Git, поддержка множества языков программирования и его экосистема, имеющая множество расширений, дают VS Code преимущество перед другими редакторами.

При разработке «web-сервиса для фитнес-тренинга» были использованы такие библиотеки как:

Tailwind – утилитарный CSS-фреймворк для быстрой разработки интерфейсов.

Flowbite – библиотека UI-компонентов, построенная на основе Tailwind CSS.

Vue Router – маршрутизация в SPA-приложении.

Pinia – state-менеджер для управления глобальным состоянием приложения.

Firebase Authentication – аутентификация.

Firestore – NoSQL база данных.

date-fns – работа с датами (format, локализация ru).

Xlsx – работа с Excel файлами.

Отладка и тестирования готово решения проводилось с использованием Postman. Postman – это популярный инструмент для тестирования API, который позволяет создавать запросы API, отправлять их на сервер и анализировать ответы. Postman также позволяет автоматизировать тестирование API с помощью коллекций запросов, скриптов и тестовых сценариев [6].

Таким образом, рассмотрев вопрос о средствах и технологиях разработки, был сформирован технологический стек, оптимально соответствующий требованиям создания веб-сервиса для фитнес-тренинга. Основой клиентской части выбран фреймворк Vue.js, обладающий рядом ключевых преимуществ: открытая лицензия MIT, подробная документация, реактивная система данных и компонентная архитектура. Вес фреймворка всего 20 Кб, что обеспечивает высокую производительность приложения при минимальной нагрузке на клиентские устройства. Для реализации серверной функциональности применена облачная платформа Firebase, предоставляющая комплекс готовых решений: аутентификацию пользователей через различные провайдеры, масштабируемую NoSQL-базу данных Cloud Firestore с синхронизацией в реальном времени, а также облачные функции для выполнения серверной логики. Такой подход позволил существенно сократить время разработки backend-составляющей, избежав необходимости развертывания и администрирования собственного сервера. Все выбранные технологии позволяют вести качественную разработку для web-сервисов.

## Глава 2 Проектирование «Web-сервиса для фитнес-тренинга»

## 2.1 Реализация хранения данных в «web-сервисе для фитнес-тренинга»

Работа с базами данных является неотъемлемой частью процесса разработки современных веб-приложений. Независимо от того, создается ли небольшой блог или сложная корпоративная система, правильное управление играет ключевую роль в обеспечении надежности, производительности и безопасности проекта. Для хранения данных используется облачная платформа Firebase. В отличие от традиционных локальных СУБД, Firebase представляет собой полноценный облачный Backend-as-a-Service (BaaS), который предоставляет готовую серверную инфраструктуру для веб-приложений. Главное преимущество Firebase для веб-разработки — это отсутствие необходимости развертывать и поддерживать собственный сервер. Все данные хранятся в облаке Google и доступны через удобные клиентские SDK.

Обобщая особенности Firebase, можно выделить:

1. Готовая инфраструктура;
2. Автоматическое масштабирование;
3. Интеграция с другими сервисами Google (Analytics, Auth, etc.);
4. Гибкие правила безопасности (Firebase Security Rules).

Облачные функции Firebase (FCF) позволяют разработчикам запускать серверный код в ответ на события, предоставляя масштабируемый и экономичный способ решения сложных задач. FCF позволяет разработчикам писать код, который взаимодействует с базой данных Firebase Realtime, службой аутентификации и другими службами Firebase, упрощая создание масштабируемых и динамических приложений. Модель БД «web-сервиса для фитнес-тренинга» представлена на рисунке 2.1.

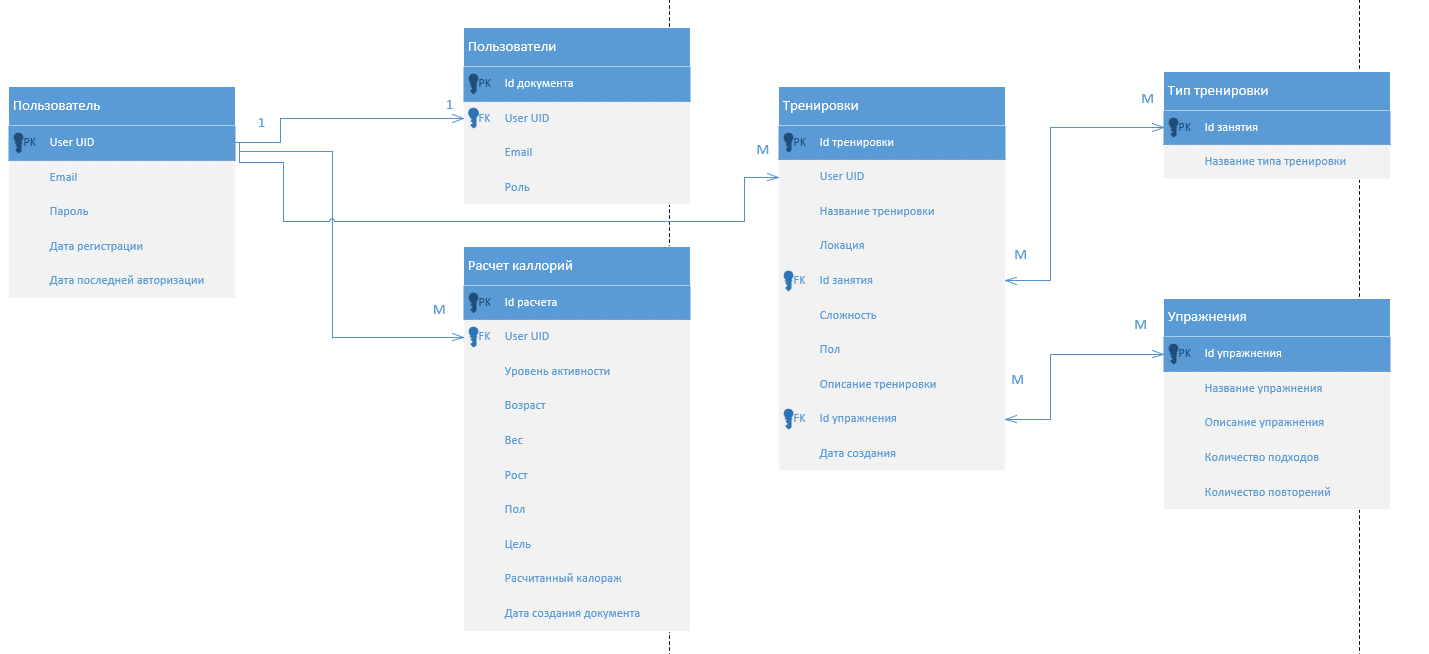


Рис.2.1 – Модель базы данных

Перед началом работы с Firebase, необходимо перейти в Firebase Console и создать проект. При создании базы данных важно настроить правила доступа, разрешив операции с данными только авторизованным пользователям (рис. 2.2).



Рис.2.2 – Правила безопасности (Firestore Rules)

Следующий этап, установка Firebase SDK в проект, с помощью команды: npm install firebase и создание файла конфигурации Firebase (рис.2.3).



Рис.2.3 – Подключение Firebase к проекту

Хранение данных построена на современных облачных технологиях. Архитектура решения включает трехуровневую структуру, состоящую из клиентского приложения на Vue.js, серверного уровня на базе сервисов Firebase и уровня хранения данных с использованием Cloud Firestore и Firebase Storage. Основу системы составляют три ключевые модели данных. Модель пользователей содержит идентификационные данные, информацию о роли (пользователь, администратор), а также персональные данные профиля. Модель тренировок включает идентификаторы, параметры тренировки и описательные характеристики. Модель для учета персональных метрик пользователей и расчета калорий. Для реализации хранения используется комплекс сервисов Firebase. Firebase Authentication отвечает за безопасное хранение учетных данных и управление пользовательскими сессиями. В приложении реализована регистрация и авторизация новых пользователей с использованием метода встроенного метода Firebase Authentication – createUserWithEmailAndPassword, который включает в себя:

* валидацию вводимых данных на клиентской стороне;
* проверку совпадения паролей;
* обработку возможных ошибок с преобразованием технических кодов ошибок Firebase в понятные пользователю сообщения;
* автоматическое создание профиля пользователя в базе данных после успешной регистрации.

Cloud Firestore выступает в качестве основного хранилища структурированных данных, организованных в коллекции users, trainings и caloriesRecords. При регистрации нового пользователя в коллекции users автоматически создается документ с уникальным идентификатором (uid), полученным из Firebase Auth, который содержит данные пользователя (email, роль и дату создания записи).

Таким образом, в разработанном веб-сервисе для фитнес-тренинга реализовано хранение данных с использованием облачной NoSQL-технологии Firebase, что обеспечивает высокую производительность, надежность и масштабируемость решения.

# 2.2 Проектирование функциональной части web-сервиса

Основная задача графического интерфейса веб-приложения — обеспечить пользователю интуитивно понятный доступ ко всем функциям приложения, будь то просмотр тренировок, расчет калорий или управление личным аккаунтом (рис. Б.1 – Б.10). Архитектура интерфейса web-сервиса для фитнес-тренинга основана на принципах модульности и реактивности, что позволяет динамически подстраивать содержимое страницы под различные сценарии использования. На главной странице расположено панель для навигации по web-сервису. Навигационный компонент Header, подключаемый в верхней части layout'а, прописан в компоненте – App.vue, который служит корневым контейнером, задающим общую разметку приложения и отвечающим за базовую организацию контента. Навигационная панель построена по принципу прогрессивного раскрытия функционала (рис.2.4).

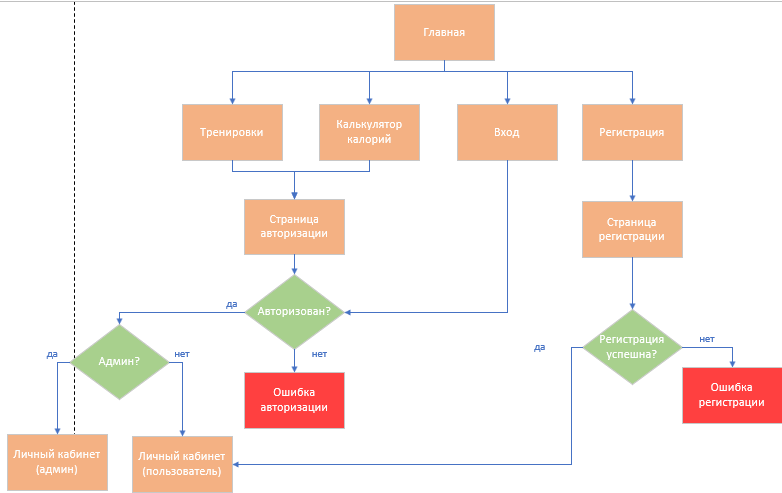
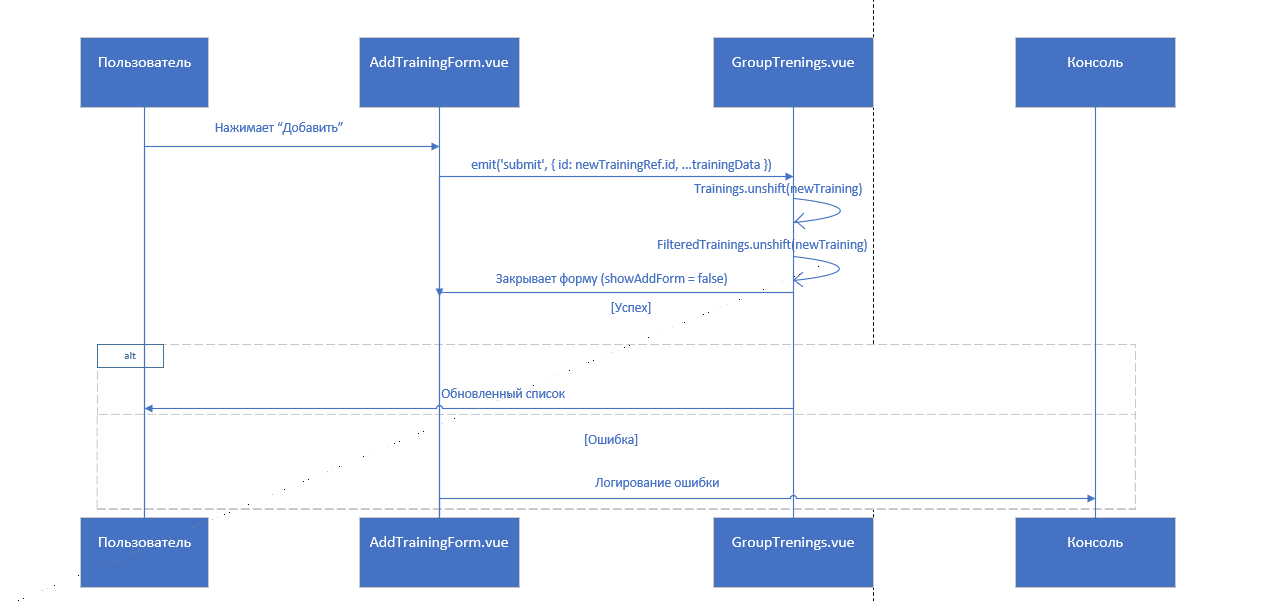
****

Рис.2.4 – Схема функционала навигационной панели

В рамках web-сервиса реализована система ролевого доступа, предусматривающая два уровня авторизации: для обычных пользователей и администраторов. При попытке перехода на защищённые страницы (тренировки, калькулятор калорий) без предварительной аутентификации система автоматически перенаправляет пользователя на страницу авторизации (рис.А.1). В случае успешного входа через учетную запись администратора система осуществляет переадресацию в личный кабинет администратора, где доступны функции просмотра статистики, формирования отчетов, а также управления пользователями, включая их добавление и удаление. Для формирования статистических данных в компоненте Account.vue реализован метод loadUserReports(), который обеспечивает загрузку из базы данных Firestore полного списка зарегистрированных пользователей с последующим подсчетом их общего количества и определением числа новых регистраций за заданный временной период (рис.А.2). Формирование отчетов реализовано с помощью метода exportToExcel(), который предоставляет возможность экспорта пользовательской статистики в стандартный формат электронных таблиц XLSX. Данный метод формирует структурированный отчет, содержащий идентификаторы пользователей, их электронные адреса, даты регистрации и уровень доступа (рис.А.3). Вторая страница личного кабинета администратора реализует функционал для управления пользовательскими аккаунтами. Процесс добавления новых пользователей представляет собой двухэтапную процедуру: сначала учетная запись создается в системе аутентификации Firebase с указанием электронной почты и пароля, затем в базе данных Firestore формируется соответствующая запись с указанием роли (пользователь или администратор) и даты создания. Функционал удаления пользователей включает защитные механизмы, предотвращающие случайное удаление учетных записей. Перед выполнением операции система запрашивает подтверждение действия, а также блокирует возможность удаления текущего активного пользователя (рис.А.4). При авторизации через обычного пользователя, он попадает в персональный кабинет, где отображается история его расчетов калорийности с возможностью удаления отдельных записей (рис.А.5, рис.А.6).

Страница тренировок отображает все доступные программы тренировок, а также обладает функционалом добавления и управления программами тренировок, обеспечивает разграничение прав доступа между обычными пользователями и администраторами. В случае выявления некачественных тренировочных программ, администратору доступна возможность удаления тренировок. Отображение тренировочных программ реализует метод loadTrainings(), осуществляя запрос к коллекции базы данных Firestore – «trainings". При выполнении запроса применяется сортировка документов по полю createdAt (дата создания) в порядке убывания. Асинхронный метод getDocs() получает набор документов, которые затем преобразуются в массив объектов, содержащих как уникальный идентификатор записи (id), так и все остальные данные тренировочной программы. Результат сохраняется в реактивную переменную this.trainings (рис.А.7). UML-диаграмма последовательностей, которая наглядно отображает взаимодействие компонентов при добавлении новой тренировки, представлена на рисунке 2.5.

  
Рис.2.5 – UML-диаграмма последовательностей

Страница калькулятора калорий позволяет пользователям рассчитать и сохранить индивидуальную суточную норму калорий для достижения выбранных целей (похудение, поддержание веса или набор массы). При загрузке компонент автоматически подключается к системе аутентификации Firebase, определяя статус текущего пользователя и сохраняя его данные в реактивной переменной user. Ключевая функция calculateCalories реализует алгоритм расчета суточной калорийности на основе формулы Миффлина-Сан Жеора, которая учитывает базовые антропометрические параметры (возраст, вес, рост, пол), уровень физической активности и поставленную цель (рис.А.8). Функция addNewRecord обеспечивает сохранение результатов расчета базе данных. Создается структурированная коллекция для каждого пользователя, включая временные метки сервера и отформатированные даты. Это позволяет получать актуальные расчеты, вести историю изменений и анализировать динамику.

Таким образом, графический интерфейс web-сервиса выполняет ключевую функцию — обеспечивает пользователю интуитивно понятный и эффективный доступ ко всем возможностям системы, включая управление тренировками, расчет калорий и работу с персональным аккаунтом Основываясь на принципах компонентной архитектуры Vue.js, интерфейс интегрирует три ключевых модуля: управление тренировочными программами, расчет суточной потребности в калориях и персонализированный аккаунт пользователя, образуя единую экосистему фитнес-трекинга. Центральным элементом архитектуры выступает навигационная панель Header, встроенная в корневой компонент App.vue, которая выполняет системообразующую функцию, обеспечивая согласованную навигацию между разделами web-сервиса.

# 2.3 Проведение отладки и тестирования готово решения

Тестирование программных продуктов является неотъемлемым этапом их создания и разработки. Наиболее простым способом тестирования является мануальное (ручное), однако оно способно быть неэффективным, поскольку напрямую зависит от классификации, внимания, профессионализма, логики и опыта тестировщика. Кроме того, ручное тестирование даже простых функций приложения занимает достаточно большое количество времени, поскольку тестировщик проверяет и анализирует каждый тестов кейс, разрабатывает их, строит схемы зависимостей. В контексте разработки веб-сервиса для фитнес-тренинга был выбран подход автоматизированного тестирования API, позволяющий существенно повысить эффективность процесса проверки работоспособности системы. Тестирование API – это процесс проверки функциональности и производительности программного интерфейса приложения (API), который используется для взаимодействия с другими приложениями и системами. API – это набор протоколов, инструментов и структур данных, который позволяет разным программным приложениям обмениваться информацией и взаимодействовать друг с другом. Postman обладает возможностью создания коллекций запросов к API, что значительно ускоряет разработку и оптимизирует время и ресурсы, зачастую тратимые на поиск запроса в памяти или ручное заполнение параметров.

Для удобства тестирование было организовано в Postman Collections, где каждая группа связанных запросов объединены в отдельную коллекцию. Первоначально был выполнен базовый тест доступности сервера через GET-запрос по адресу localhost. На рисунке 2.6 представлен результат проверки доступности сервера. GET-запрос возвращает статус 200 OK, что подтверждает работоспособность серверной части веб-сервиса.

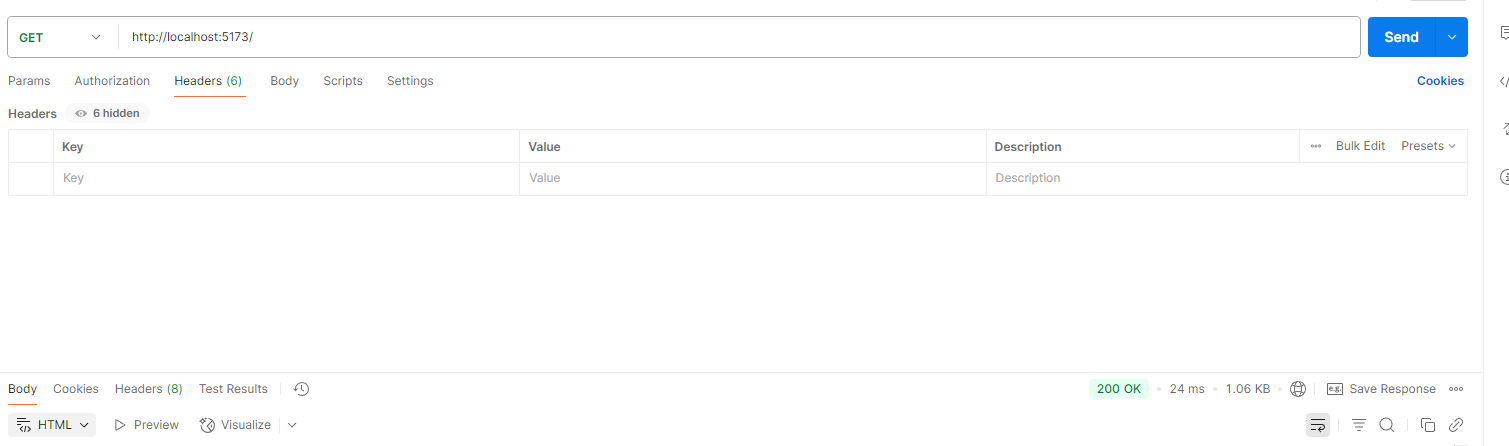


Рис.2.6 – Проверка доступности сервера

Коллекция Auth включает в себя тестирование авторизации и регистрации. Для тестирования процесса регистрации создаётся POST-запрос, отправляющий данные нового пользователя на сервер Firebase. Запрос включает JSON-объект с полями email, password и returnSecureToken, а его адрес соответствует Firebase REST API. Для автоматизации проверок в Postman добавлены тесты, валидирующие статус ответа (рис.2.7).

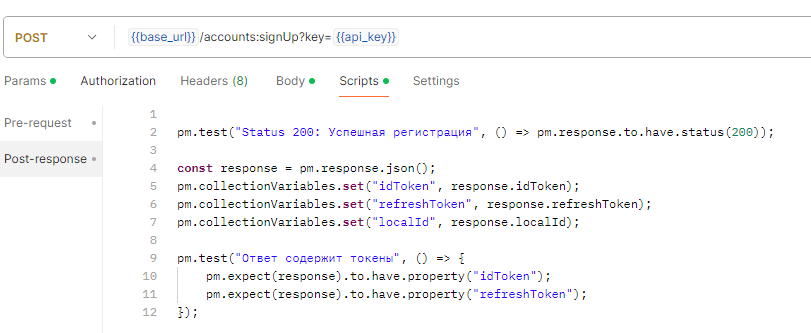


Рис.2.7 – Автоматизированные тесты регистрации

После отправки POST-запроса на регистрацию сервер возвращает ответ (рис.2.8). Тесты проверяют:

* Статус ответа;
* Наличие обязательных полей.



Рис.2.8 – Ответ сервера при успешной регистрации

Запрос Sign In предназначен для тестирования процесса авторизации пользователя в системе. Он отправляет на сервер учетные данные (email и пароль) для проверки подлинности и возвращает токены доступа в случае успешной аутентификации (рис.2.9).



Рис.2.9 – Ответ сервера при успешной авторизации

В коллекции workouts создан запрос Get Trainings, который предназначен для проверки корректности получения списка тренировок из базы данных (рис.2.10). GET-запрос отправляется по адресу {{base\_url}}/trainings.json и выполняет следующие функции:

* Проверяет доступность сервера и работоспособность API;
* получает полный список тренировочных программ;
* оценивает производительность системы.

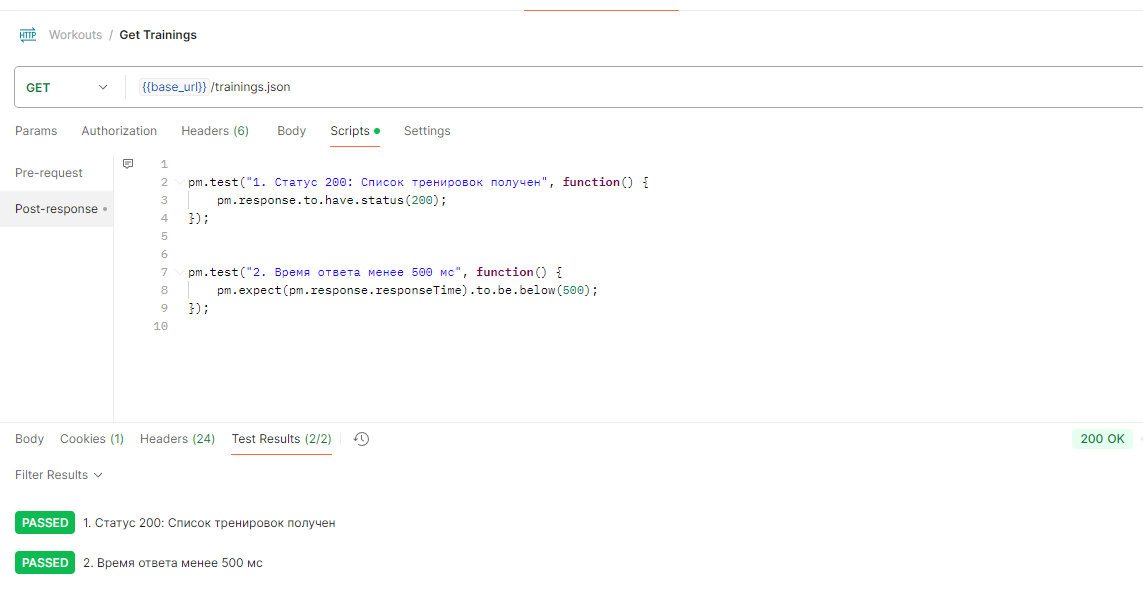


Рис.2.10 – Запрос Get Trainings

Запрос Get Records из коллекции Calculate Callories проверяет корректность функционирования эндпоинта получения списка записей о расчетах калорий (рис.2.11).

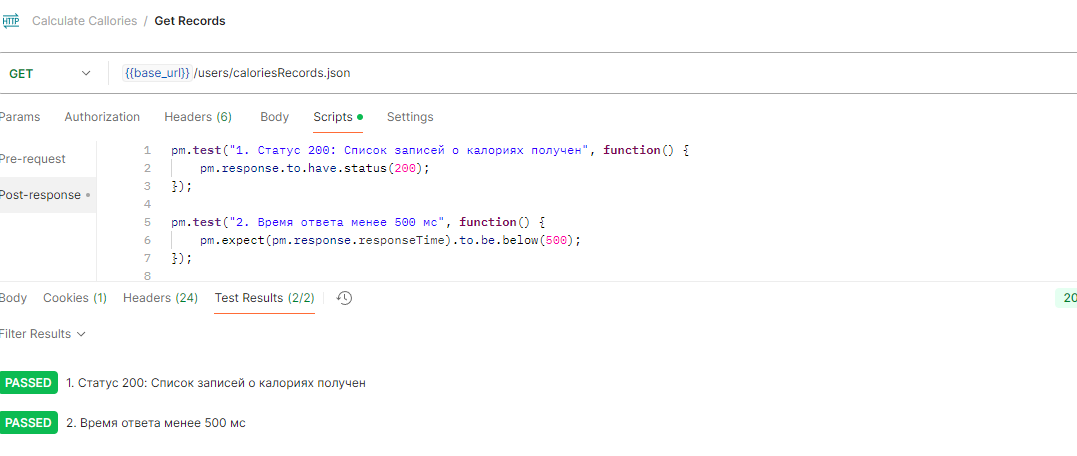


Рис.2.11 – Запрос Get Records

Таким образом, проведенное тестирование REST API веб-сервиса для фитнес-тренинга с использованием Postman позволило комплексно оценить работоспособность и надежность ключевых функциональных модулей системы.

## Заключение

Современный мир просто невозможно представить без ИТ, свое применение они нашли во всех сферах человеческой жизни. В условиях ускоряющегося темпа жизни, роста информационных технологий и урбанизации возрастает необходимость ведения активного образа жизни и укрепления здоровья через физическую активность. Современное общество сталкивается с рядом проблем, таких как малоподвижный образ жизни, стресс, ухудшение экологической ситуации и увеличение числа заболеваний, связанных с образом жизни. Сегодня физическая культура рассматривается как важный фактор формирования устойчивости к стрессу, поддержания высокого уровня энергичности и общего качества жизни. С появлением умных устройств и мобильных приложений стало проще отслеживать физическую активность и корректировать программы тренировок. Онлайн тренировки удовлетворяют такие человеческие потребности как комфорт, обособленность и индивидуальность подходов к построению тренировочного процесса.

В настоящий момент различные виды сервисов, ориентированные на использование широким кругом пользователей, предоставляют возможность удаленной работы с ними. Классическим вариантом организации дистанционного доступа является клиент-серверная архитектура системы. Ключевую роль привлечения посетителей на web-сервис выполняет графическое оформление. Дизайн отвечает за легкость использовании и навигации, а также за то, чтобы весь контент был доступен и понятен для пользователей. Figma объединяет в себе все нужные функции для реализации дизайна. Данное программное обеспечение является комбинацией инструментов, платформ и приложений, позволяющей создавать логичный и простой дизайн с точки зрения UX/UI. JavaScript является одним из наиболее популярных и широко применяемых языков программирования в области веб-разработки. Благодаря встроенной поддержке во всех современных браузерах, он служит основным инструментом для создания клиентской части веб-приложений. Кроме того, с появлением платформы Node.js JavaScript стал использоваться и для разработки серверной логики, что позволило реализовывать полноценные full-stack решения в рамках единой технологической экосистемы. Для упрощения и ускорения процесса разработки в JavaScript существует множество специализированных фреймворков и библиотек, таких как React, Angular и Vue.js. Vue.js представляет собой прогрессивный фреймворк для создания пользовательских интерфейсов. Его ключевыми особенностями являются гибкость, высокая производительность и интуитивно понятная архитектура, что делает его особенно популярным среди разработчиков. Благодаря компонентному подходу и реактивности, Vue.js позволяет эффективно разрабатывать как небольшие интерактивные элементы, так и сложные приложения.

Хранение и обработка данных в системе реализованы с использованием облачной платформы Firebase – инновационного решения от Google, предоставляющего комплекс инструментов для разработки мобильных и веб-приложений. Платформа включает в себя различные сервисы, такие как база данных реального времени, облачное хранилище, аутентификация пользователей и аналитика. Архитектура данных системы построена на трех взаимосвязанных моделях. Модель пользователей содержит идентификационные данные, информацию о ролевой принадлежности и персональные данные профиля. Модель тренировок включает уникальные идентификаторы, параметры тренировочного процесса и описательные характеристики. Отдельная модель предназначена для учета персональных метрик пользователей и автоматического расчета расхода калорий, что обеспечивает персонализацию тренировочного процесса.

Важной частью разработки является тестирование готово решения. Postman является популярным инструментом для тестирования, который позволяет автоматизировать тестирование API с помощью коллекций запросов, скриптов и тестовых сценариев. В рамках проекта тестирование было организовано с использованием Postman Collections, где логически связанные группы запросов были сгруппированы в отдельные коллекции с общей конфигурацией и скриптами. Такой подход позволил систематизировать процесс тестирования, обеспечить повторное использование тестовых сценариев и упростить командную работу над проектом.

Таким образом, все поставленные задачи и цель исследования достигнуты.

## Список использованных источников

1. Блинков А. С. Архитектура веб-приложений при использовании фреймворка vue.js и облачного решения firebase / А. С. Блинков, Н. А. Спирин, И. А. Гурин // Теплотехника и информатика в образовании, науке и производстве : сборник докладов X Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (TИМ’2022) с международным участием (Екатеринбург, 19–20 мая 2022 г.). — Екатеринбург: УрФУ, 2022. — С. 191-195;
2. Давидян И. Г., Шокола Я. В. Цифровая трансформация сервисных процессов предприятий сферы фитнес-услуг / И. Г. Давидян, Я. В. Шокола // Технико-технологические проблемы сервиса. — 2023. — № 1(63). — С. 80 – 84;
3. Итинсон К. С. Web 1.0, web 2.0, web 3.0: этапы развития веб-технологий и их влияние на образование / К. С. Итинсон // Карельский научный журнал. — 2020. — № 1(30). — С. 19 – 21;
4. Мартыщенко С.О. Актуальность онлайн фитнес-приложений в России / С.О. Мартыщенко // International Journal of Humanities and Natural Sciences. — 2023. — № 79. — С. 219 – 224;
5. Матвеев В. С. Современные фреймворки для разработки информационных систем на примере платформы vue / В. С. Матвеев // Форум молодых ученых. — 2020. — № 5(45). — С. 301 – 306;
6. Медаев М. К. Тестирование api / М. М. Казбекович // E-Scio. — 2023. — № номер 8 (83). — С. 1 – 6;
7. Орехов Г. С. Ит в современном мире: возможности, задачи, проблемы, перспективы / Орехов Г. С. // Современные материалы, техника и технологии. — 2021. — № №1 (1). — С. 175 – 177;
8. Урусов Т. Т. Создание веб-приложения интернет-магазина с использованием современных инструментов разработки / Т. Т. Урусов // Инновации и инвестиции. — 2023. — № 6. — С. 179 – 185;
9. Фарахутдинов Ш.Ф. Цифровой фитнес: грозит ли традиционным клубам вымирание? / Ш.Ф. Фарахутдинов // Теория и практика физической культуры. — 2021. — № 6. — С. 110;
10. Хван Д. О., Воронкин Е. Ю. Разработка frontend части киноафиши для компании «И20» по медиа франшизе «Звездные войны» / Д. О. Хван, Е. Ю. Воронкин // Интерэкспо Гео-Сибирь. — 2022. — № 6. — С. 256 – 260;

## Приложение А

# Проектирование функциональной части web-сервиса

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Рисунок А.1 – Система маршрутизации с ролевой моделью доступа   |  |  | | --- | --- | |  |  | | Рисунок А.2 – Метод получения статистики по пользователям | Рисунок А.3 – Метод для экспорта статистики в Excel |     Рисунок А.4 – Метод удаления пользователей   |  |  | | --- | --- | |  |  | | Рисунок А.5 – Загрузка записей из счетчика калорий | Рисунок А.6 – Удаление записей из счетчика калорий | |
|  |
| Рисунок А.7 – Отображение тренировочных программ    Рисунок А.8 – Расчет суточной калорийности |

## Приложение Б

# Интерфейс web-сервиса

|  |
| --- |
|  |
|  |
|  |

Рисунок Б.1 – Главная страница

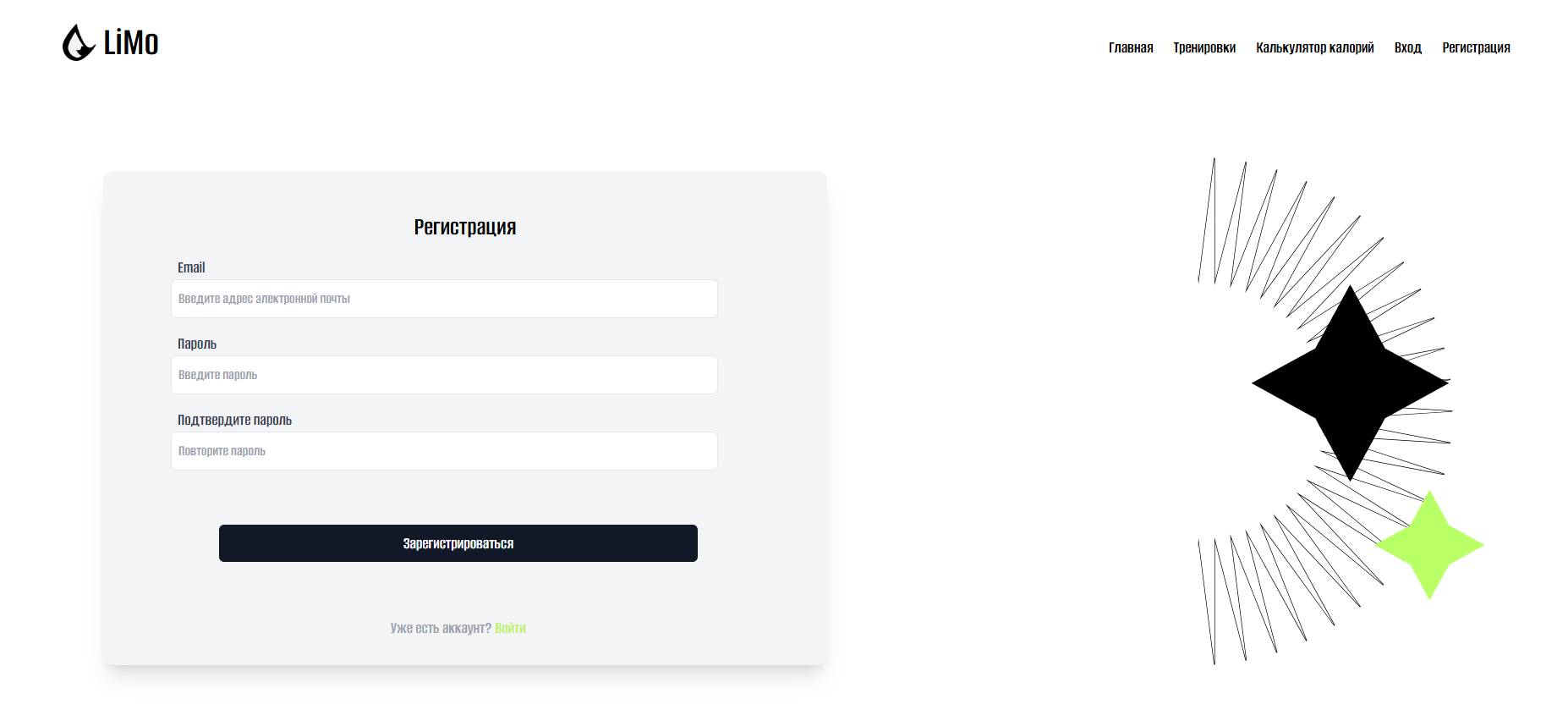


Рисунок Б.2 – Страница регистрации

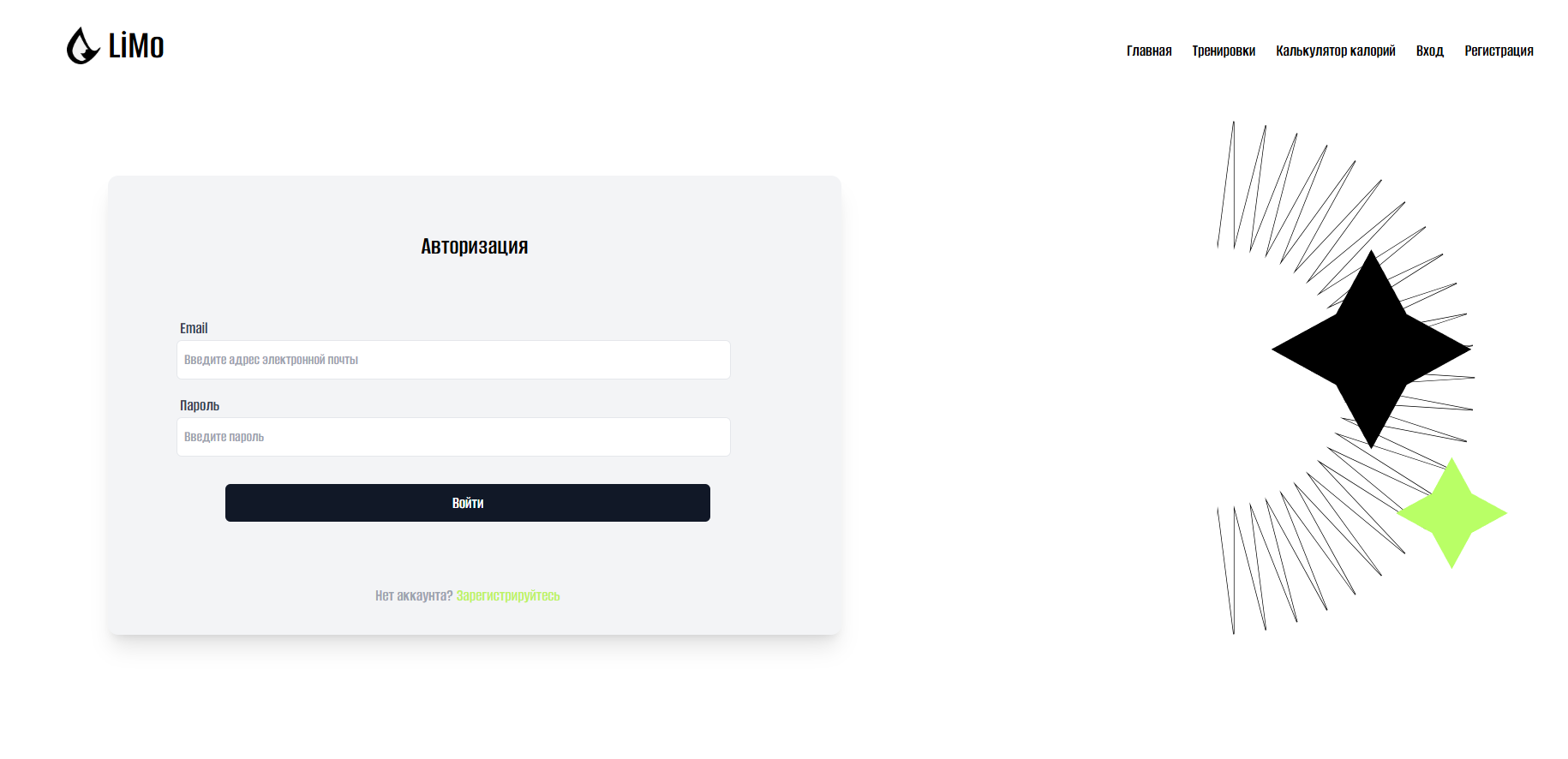


Рисунок Б.2 – Страница авторизации

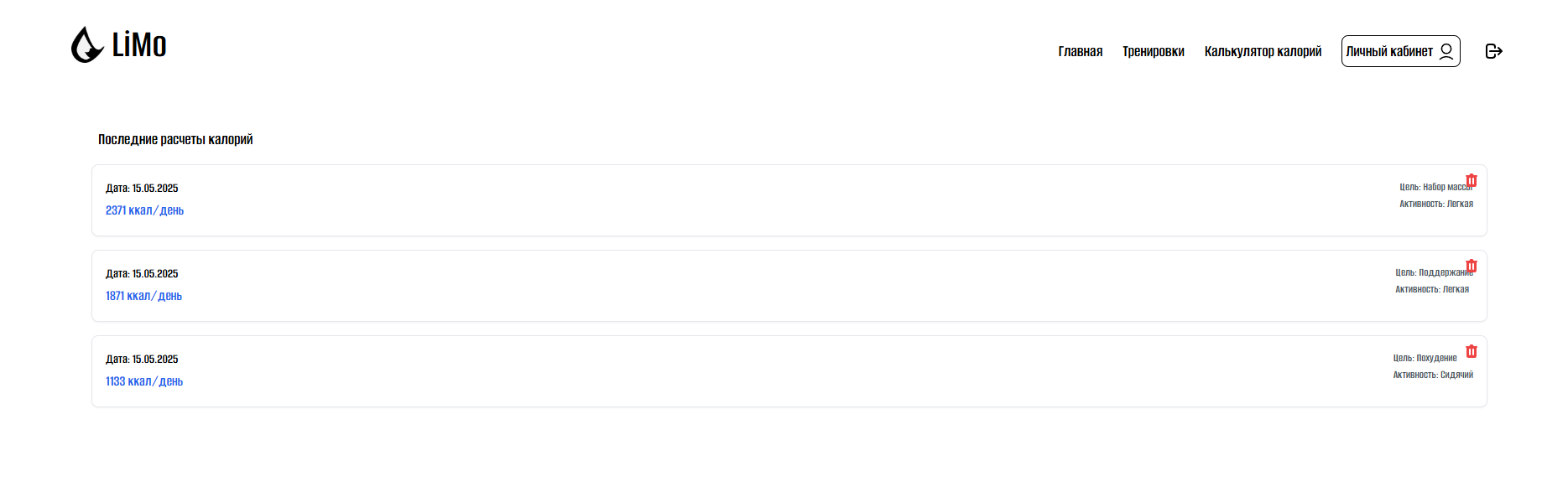


Рисунок Б.3 – Личный кабинет пользователя

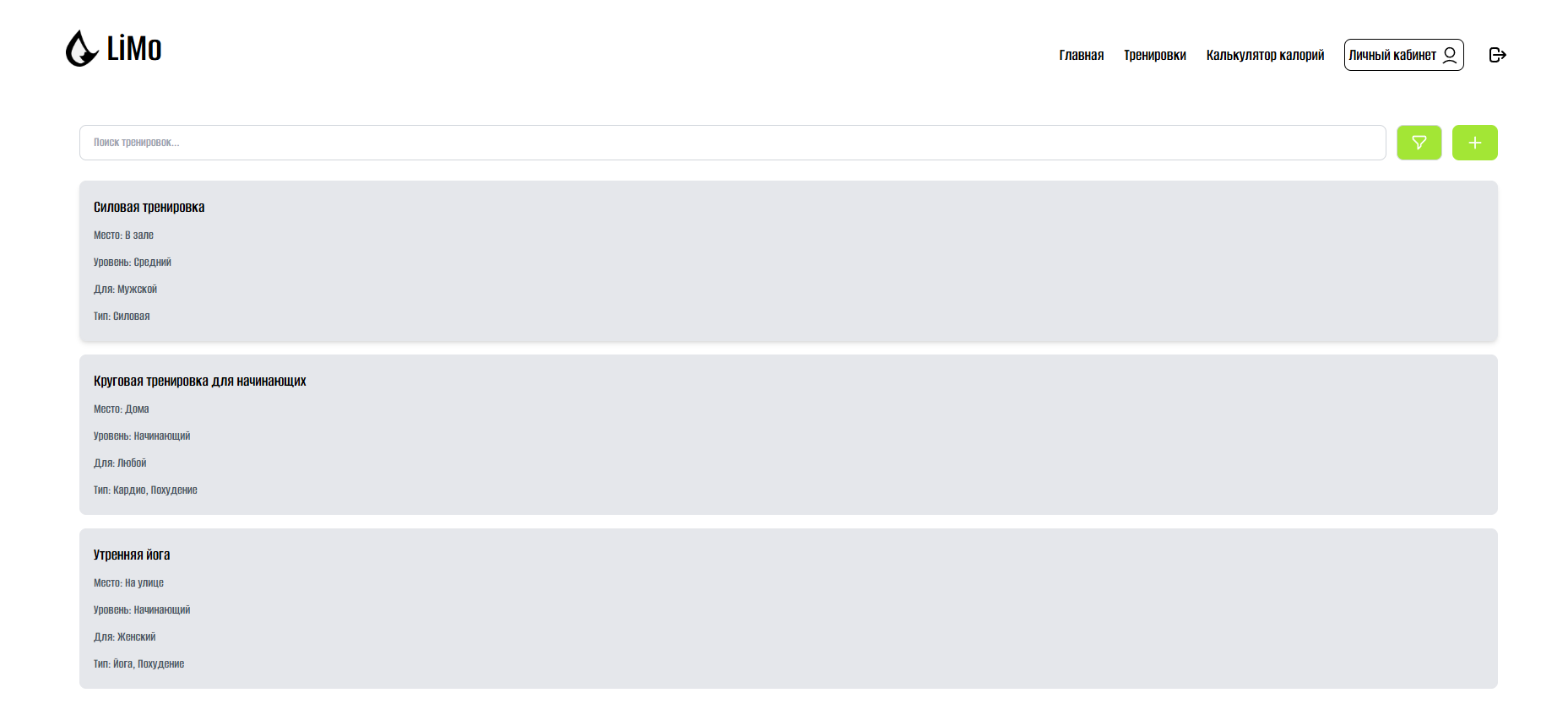


Рисунок Б.4 – Страница тренировок

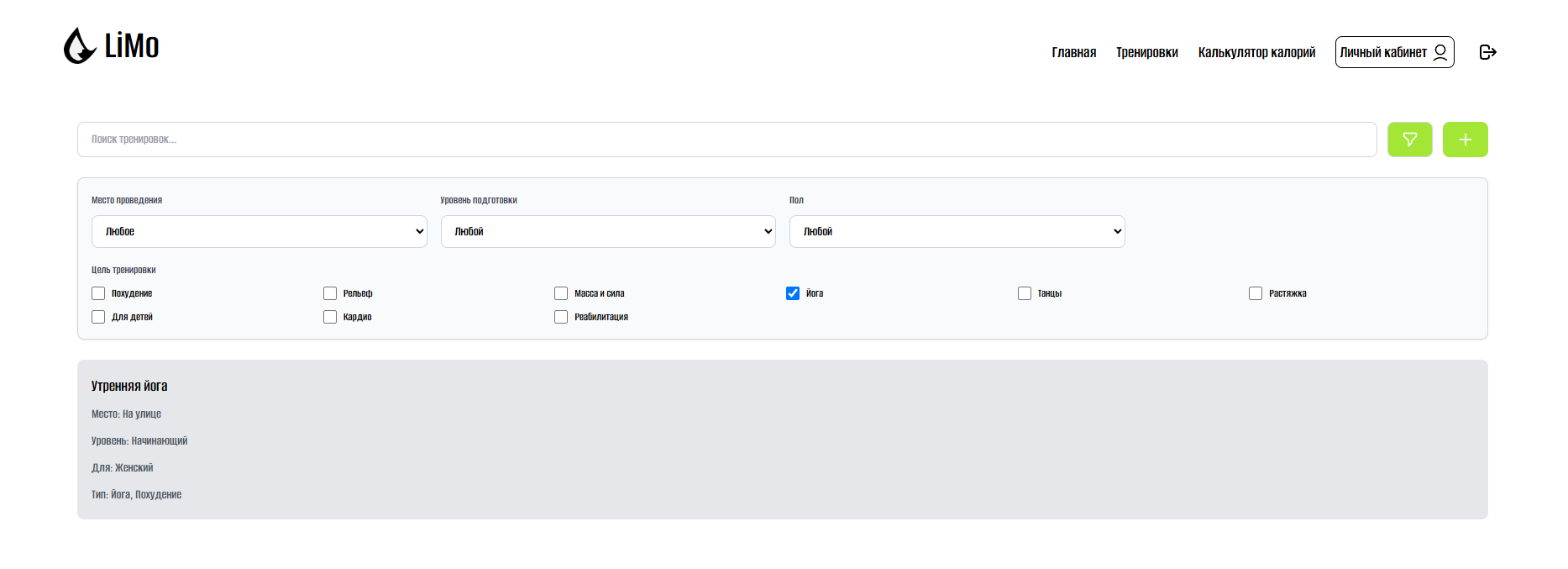


Рисунок Б.5 – Фильтры по параметрам

|  |
| --- |
|  |
|  |

Рисунок Б.6 – Просмотр содержимого тренировки

|  |  |
| --- | --- |
|  |  |

Рисунок Б.7 – Окно создания новой тренировки

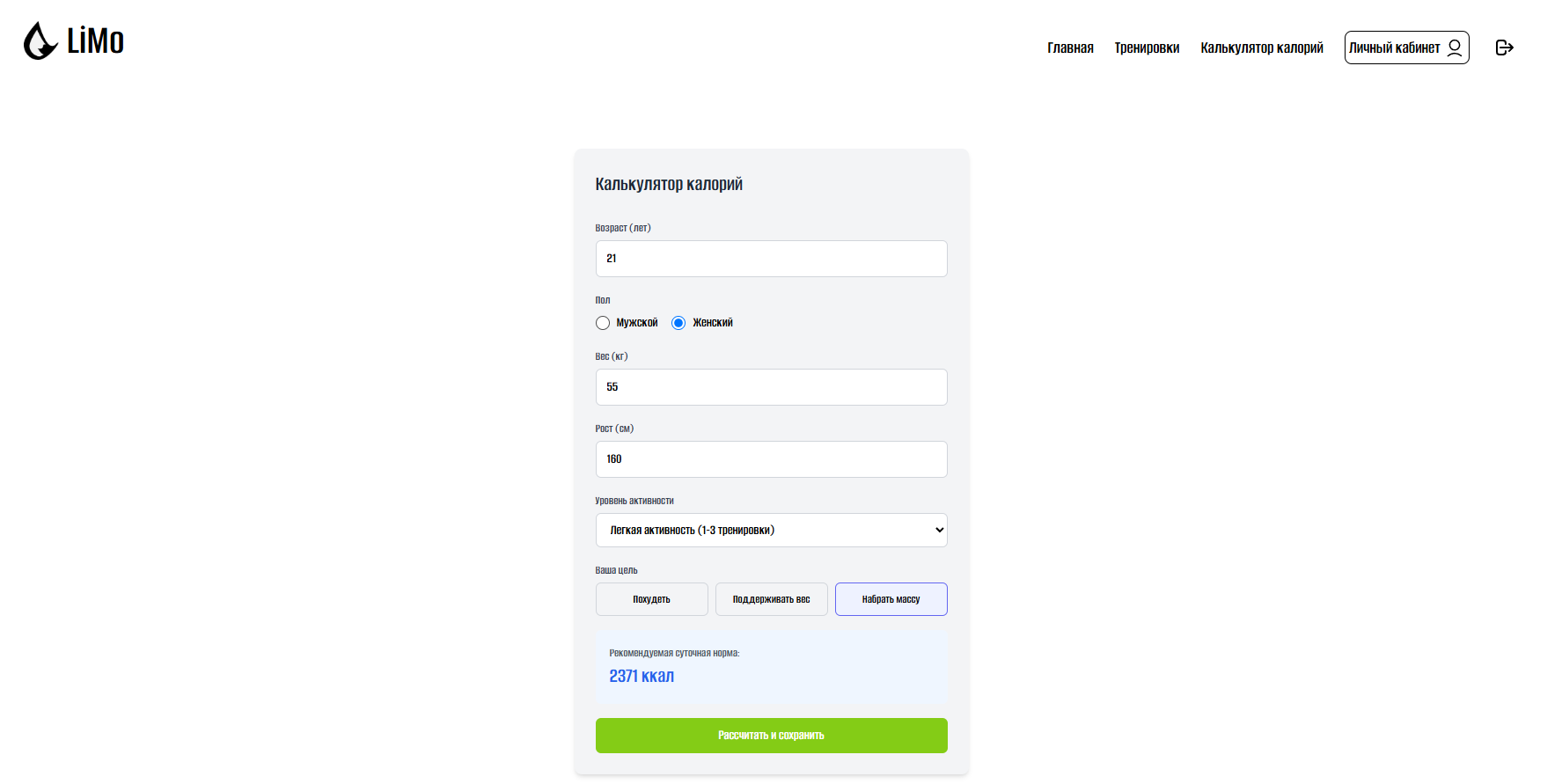


Рисунок Б.8 – Страница калькулятора калорий

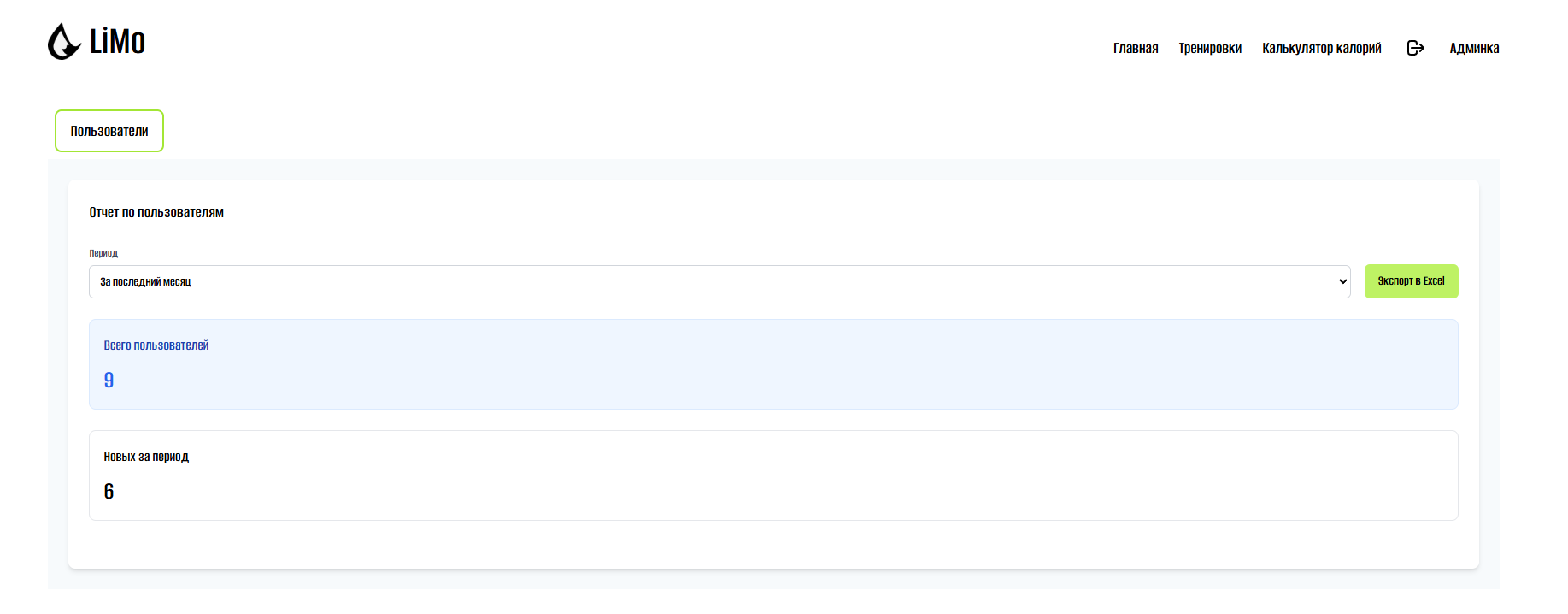


Рисунок Б.8 – Страница формирования отчетности

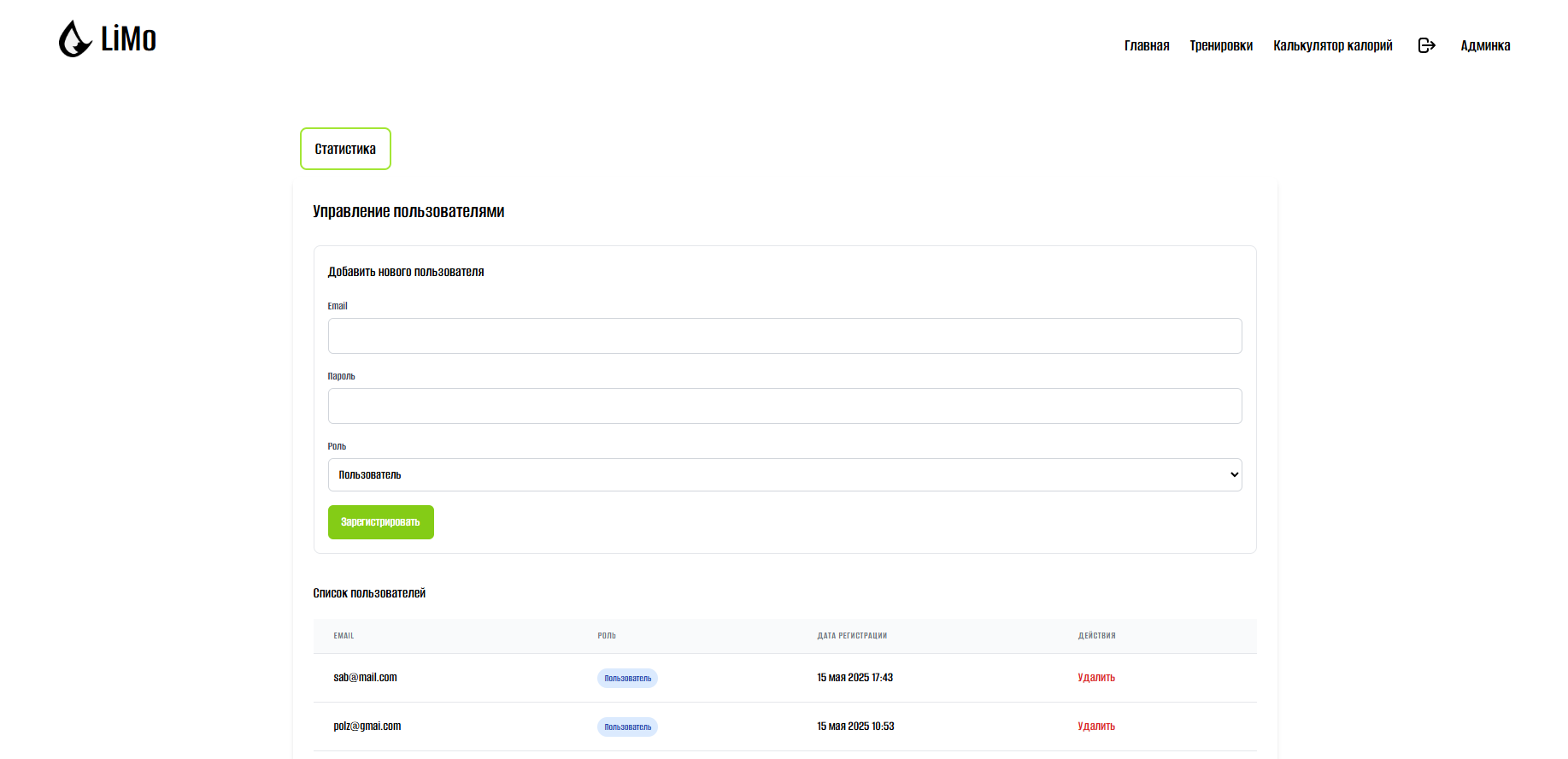


Рисунок Б.9 – Страница управления пользователями

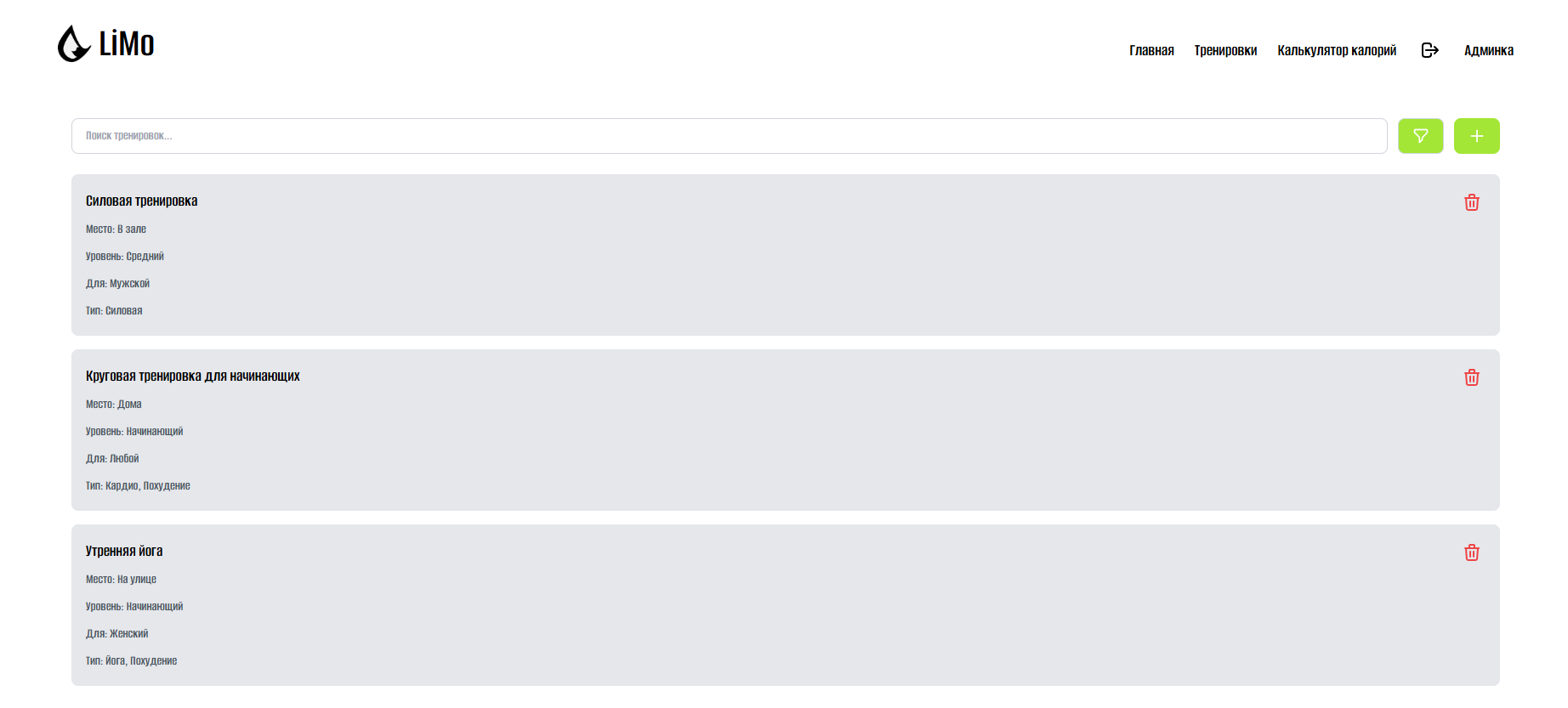


Рисунок Б.10 – Страница тренировок от лица администратора