

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №2

Выполнила:

Гусейнова Марьям

Группа ФРЭНД 2.2

Проверил:

Добряков Д. И.

Санкт-Петербург

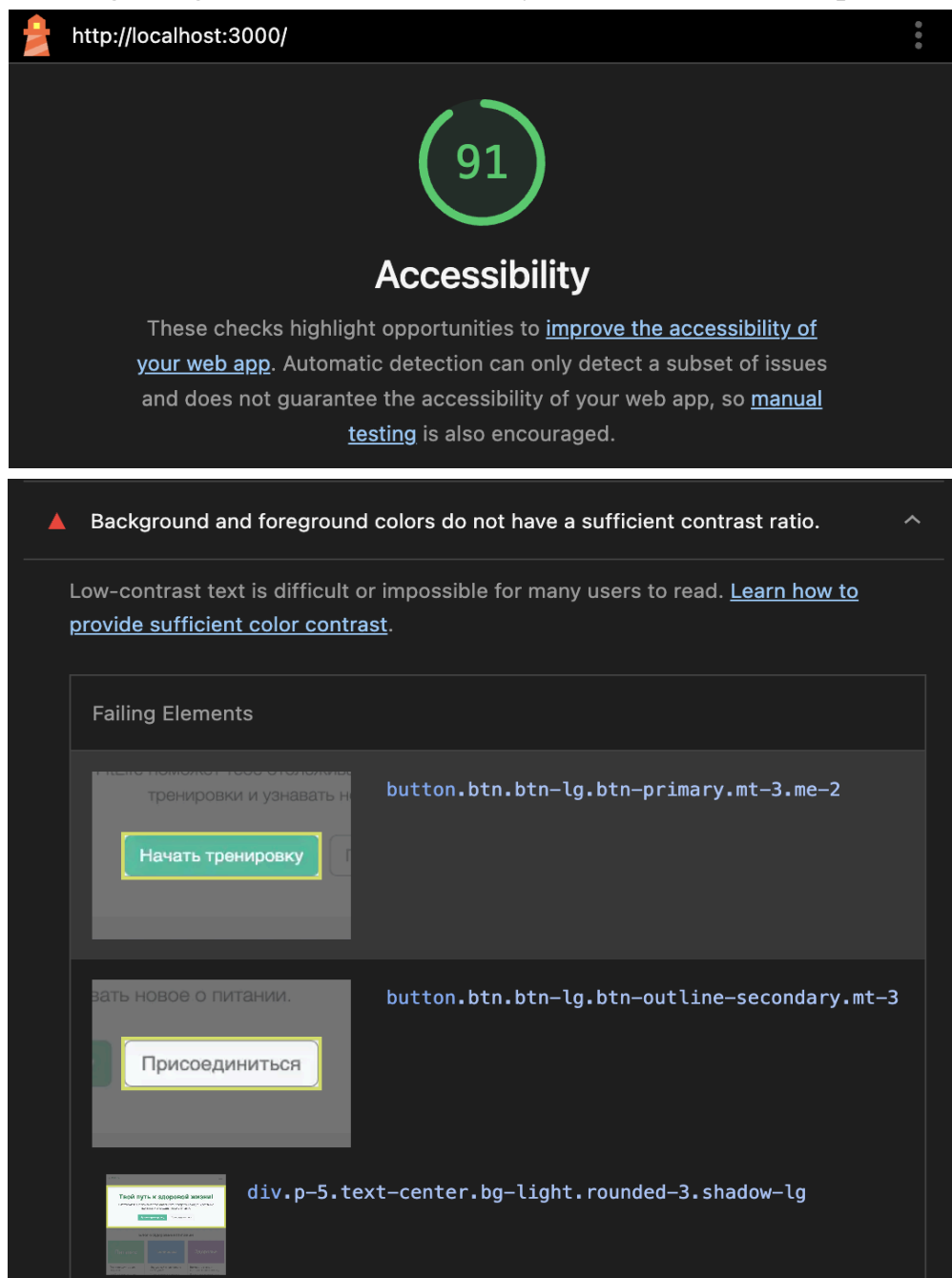
2025 г.

Задача

Задание: улучшить доступность ранее реализованного сайта. Добавить необходимые HTML-атрибуты ко всему контенту на странице и проверить это с помощью инструментов из Dev Tools браузера Firefox и сервиса Google Lighthouse.

Ход работы

Вначале Google Lighthouse показал следующее на главной странице:

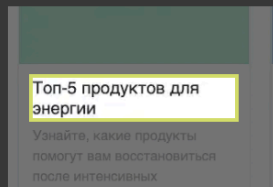


NAVIGATION

▲ Heading elements are not in a sequentially-descending order

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more about heading order.](#)

Failing Elements



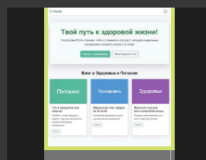
`h5.card-title`

BEST PRACTICES

▲ Document does not have a main landmark.

One main landmark helps screen reader users navigate a web page. [Learn more about landmarks.](#)

Failing Elements



`html`

These items highlight common accessibility best practices.

И следующее на подразделах (проблемы были те же, что и на главной странице):

■ 17/20

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

■ 16/19

Accessibility

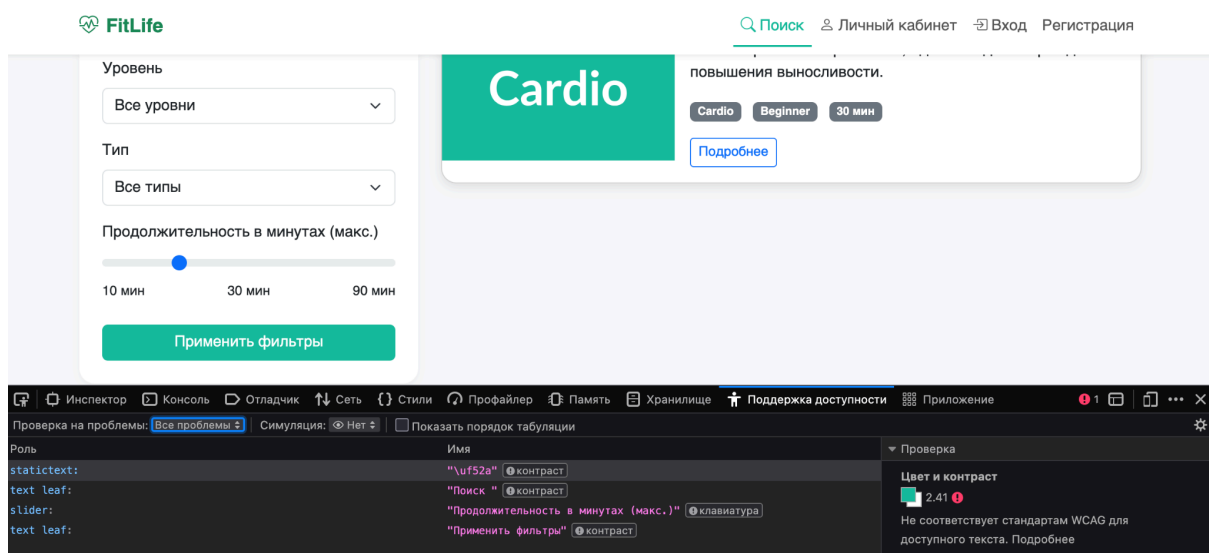
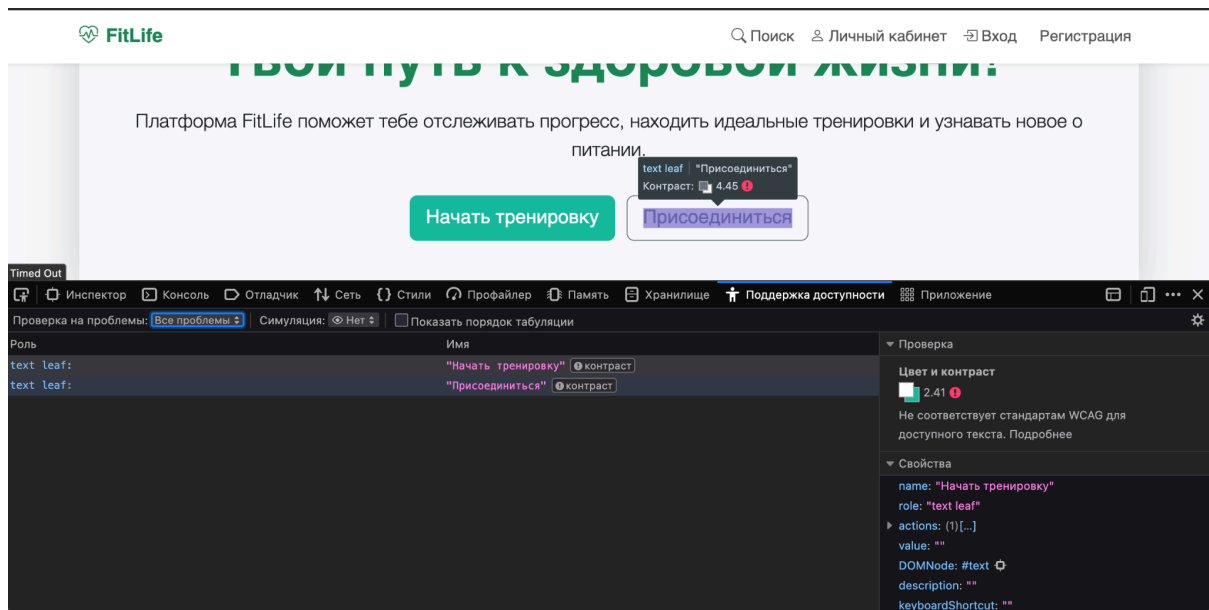
These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

■ 16/18

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

А Firefox следующее:



Были выявлены следующие проблемы:

- Отсутствие семантической структуры и меток:
 - Навигация, основные секции и интерактивные элементы не имели соответствующих ARIA-ролей (role) и меток (aria-label, aria-labelledby).
 - Декоративные иконки в интерфейсе передавались скринридерам, создавая шум.
 - Связь между заголовками секций и их содержимым не была явно обозначена.
- Нарушение иерархии заголовков: в нескольких разделах (поиск, личный кабинет) использовались заголовки <h4> сразу после <h2>.

что нарушало логическую последовательность. Это затрудняло навигацию по странице для пользователей скринридеров.

- Проблемы с интерактивными элементами:
- Ползунок для фильтрации по времени не имел полного набора ARIA-атрибутов (`aria-valuemin`, `aria-valuemax`, `aria-valuenow`), а его текущее значение не было должным образом связано с элементом интерфейса.
- Неправильная структура для динамического списка: контейнеру с результатами поиска был назначен `role="feed"`, но его дочерние элементы (карточки тренировок) не имели обязательной роли `role="article"`.
- Неоптимальная контрастность цветов: некоторые цветовые сочетания, особенно для текста в состоянии `muted` и для кнопок в состоянии `hover`, не обеспечивали достаточный контраст, что затрудняло чтение для пользователей с ослабленным зрением.

Для решения выявленных проблем был проведен комплексный рефакторинг кода.

1. Улучшение семантики и меток (HTML)

- Навигация: навигационной панели (`<nav>`) добавлен атрибут `aria-label="Основная навигация сайта"`. Кнопке «гамбургер»-меню для мобильных устройств добавлена метка `aria-label="Переключить навигацию"`.
- Секции страницы: основным секциям (`<section>`) добавлены атрибуты `aria-labelledby`, которые связывают их с уникальными идентификаторами (`id`) заголовков (`<h2>`). Это явно указывает скринридерам, какая часть контента к какому заголовку относится.
- Логические регионы: блоку фильтров добавлена роль `role="search"`, а контейнеру с результатами – `role="feed"` с меткой `aria-label`. Карточкам тренировок внутри `role="feed"` присвоена роль `role="article"`, что соответствует спецификации WAI-ARIA.
- Декоративные элементы: всем иконкам из библиотеки Bootstrap Icons, не несущим смысловой нагрузки, добавлен атрибут `aria-hidden="true"`, чтобы скринридеры их игнорировали.

- Индикация текущей страницы: в JavaScript-коде, отвечающем за переключение страниц, для активной ссылки в навигации теперь устанавливается атрибут `aria-current="page"`.

2. Исправление иерархии заголовков

- Заголовки «Фильтры», «Трекинг Прогресса» и «Твой План Тренировок» были повышены с уровня `<h4>` до уровня `<h3>`. Это восстановило корректную логическую цепочку.
- Заголовки карточек в блоге на главной странице также были подняты с `<h5>` до `<h3>` для соответствия общей структуре раздела.

3. Дооснащение интерактивных элементов

- Ползунок фильтрации:
 - Добавлены ARIA-атрибуты `aria-valuemin`, `aria-valuemax`, `aria-valuenow`.
 - Добавлен атрибут `aria-describedby="durationValue"`, который программно связывает ползунок с элементом ``, отображающим текущее числовое значение.
 - В JavaScript добавлена логика для динамического обновления атрибута `aria-valuenow` и текстового поля при перемещении ползунка.
- Динамический контент: для контейнера результатов поиска реализовано управление атрибутом `aria-busy`. Его значение меняется на `"true"` во время выполнения запроса к API и на `"false"` после получения и обработки данных, информируя вспомогательные технологии о состоянии загрузки.

4. Оптимизация визуального представления (CSS)

- Цвет текста класса `.text-muted` изменен с более светлого серого на `#555555` для повышения контрастности на светлом фоне.
- Переопределены цвета кнопок `.btn-primary` для обеспечения достаточного контраста как в обычном состоянии, так и при наведении (`:hover`).

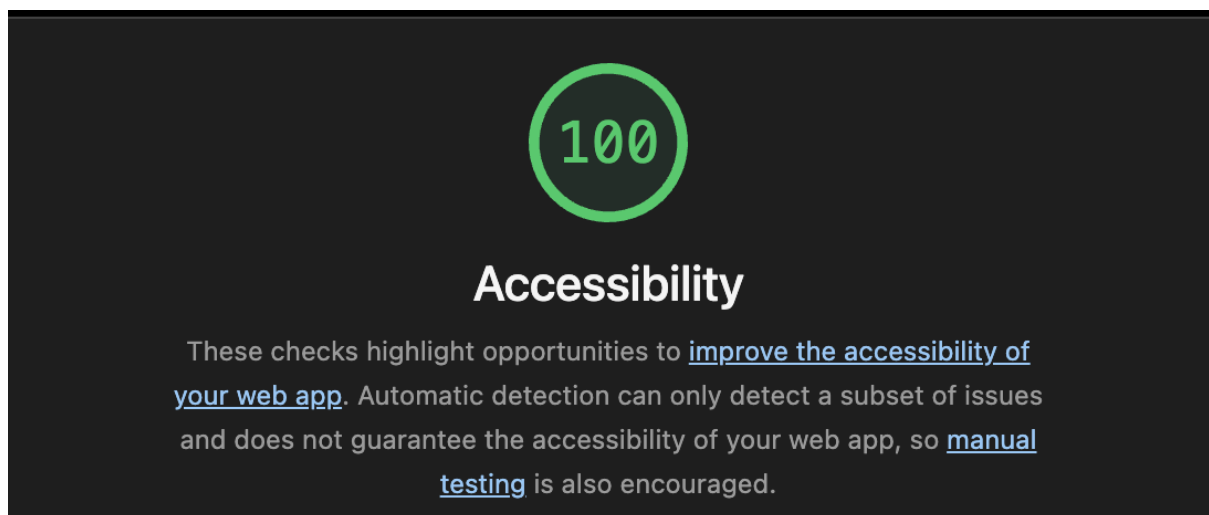
Отдельно была исправлена логика фильтрации. Изначально логика фильтрации списка тренировок выполнялась на стороне клиента (в браузере) путем перебора и скрывания DOM-элементов. Это не

соответствовало принципам RESTful API и было неэффективно при работе с большими объемами данных.

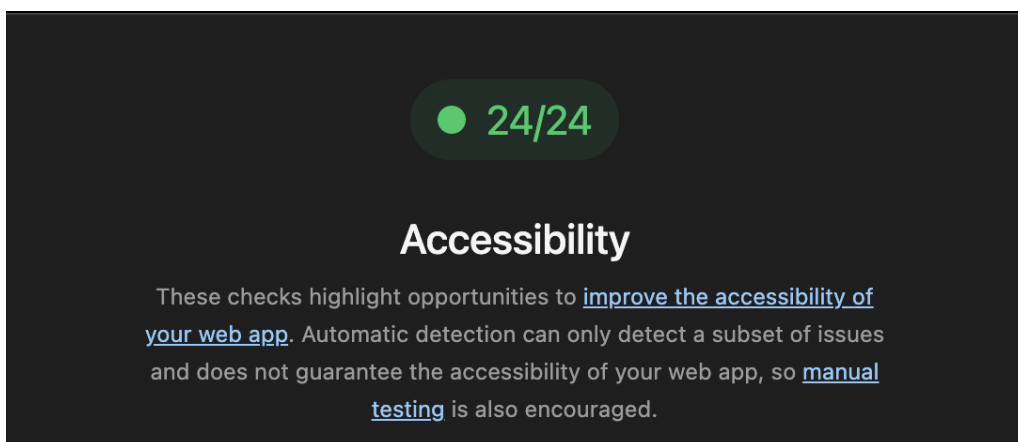
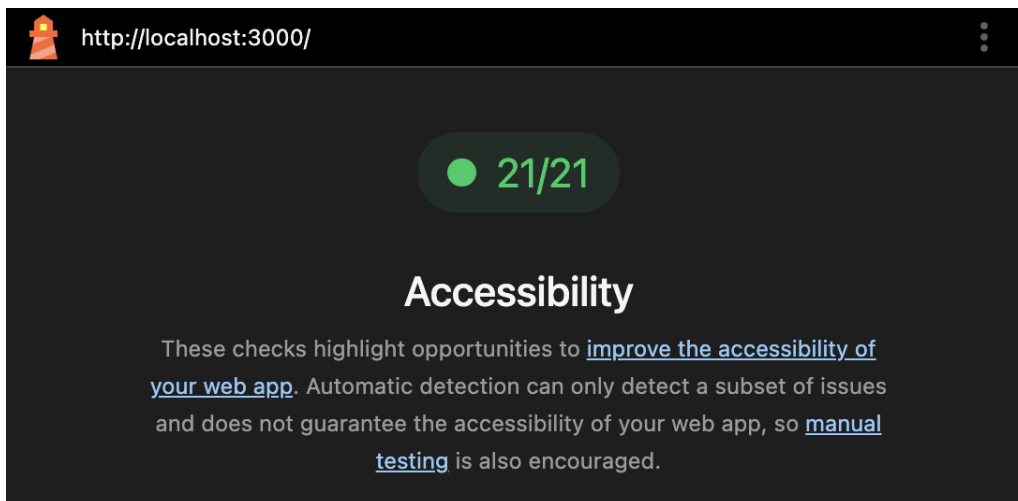
Перенос фильтрации на серверную сторону:

- Создана универсальная функция `loadWorkoutsWithFilters(filters)`, которая формирует корректный URL с query-параметрами для JSON Server (например, `?type=cardio&level=beginner&duration_lte=30`).
- Удалена клиентская логика перебора и скрывания карточек. Вместо этого сервер возвращает уже отфильтрованный набор данных.
- Функция `applyFilters()` теперь собирает параметры из формы и вызывает `loadWorkoutsWithFilters()`. Это соответствует архитектуре REST API, снижает нагрузку на клиент и повышает производительность.

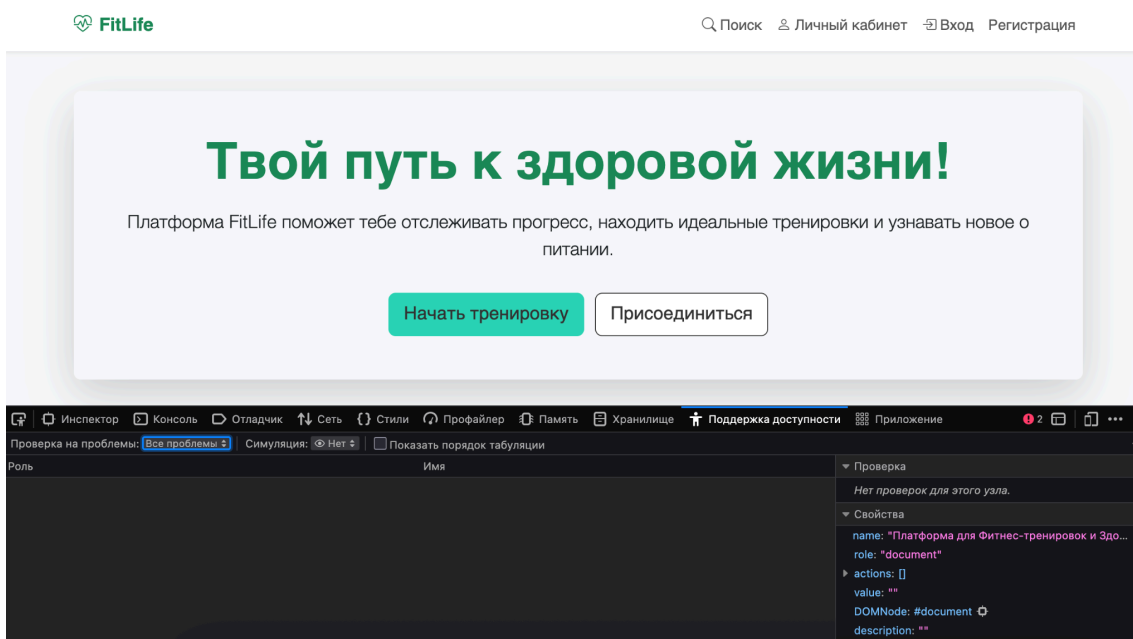
После рефакторинга Google Lighthouse показал:

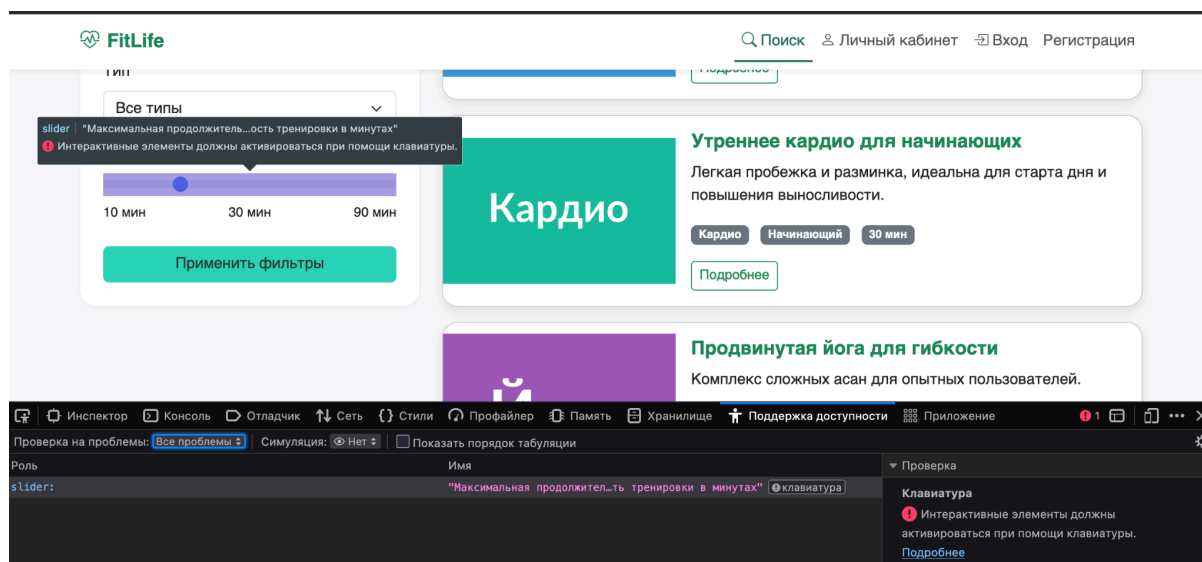


При Snapshot конкретных подразделов:



Firefox:





(проблема со слайдером осталась, хотя он теперь настраивается с клавиатуры)

Все страницы теперь полностью соответствуют лучшим принципам и доступности.

Вывод

В ходе работы были успешно идентифицированы и устранены ключевые барьеры доступности в веб-приложении «FitLife».

Основные достижения:

- Приведена в соответствие с современными стандартами семантическая структура HTML.
- Обеспечена корректная работа интерфейса с использованием только клавиатуры и со скринридерами.
- Значительно улучшены показатели в автоматизированных инструментах аудита доступности.
- Архитектура взаимодействия с бэкендом оптимизирована за счет переноса логики фильтрации на сервер, что соответствует лучшим практикам разработки.