

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 2
Домашняя работа 2 и 3

Выполнил: Зайцев Кирилл
Дмитриевич

Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

Задача 1

Варианты остаются прежними. Теперь Вам нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами `fetch/axios/xhr`.

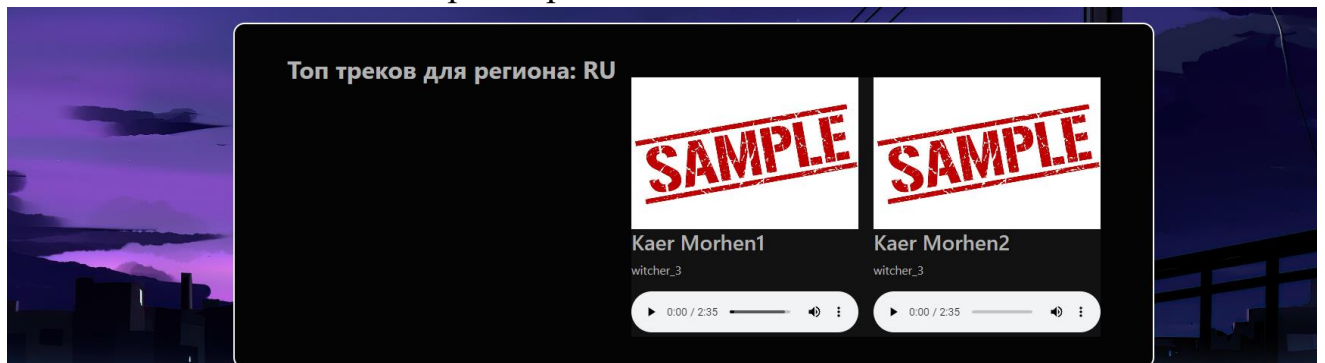
Например, для приложения для просмотра прогнозов погоды задание выглядит следующим образом:

Реализовать получение погоды (прогноз на ближайшие 7 дней) из открытого API OpenWeatherMap, в зависимости от геолокации пользователя. Реализовать вывод полученного прогноза в виде 7 карточек в три ряда (первый ряд - крупная карточка, второй ряд - три карточки в меньшем размере, третий ряд - четыре карточки в маленьком размере).

Ход работы:

В силу закрытости всех API музыкальных сайтов и плееров, был написан свой собственный API, с которым и была проведена работа.

На сайт был добавлен топ треков региона пользователя:



Код элементов на странице:

```
<div class="top-tracks-container mt-5">
  <h2 class="text-center mb-4">Топ треков для региона: <span
id="userRegion"></span></h2>
  <div class="song-list d-flex flex-wrap justify-content-center"></div>
</div>
```

Скрипт для вывода топа треков и определения региона:

```
<script>
function fetchTopTracks(region) {
  const apiUrl = `http://localhost:3000/api/top-tracks/${region}`;

  fetch(apiUrl)
    .then(response => response.json())
    .then(tracks => {
      const songList = document.querySelector('.song-list');
      songList.innerHTML = '';

      tracks.forEach(track => {
        const songItem = document.createElement('div');
        songItem.classList.add('song-item');

        const cover = document.createElement('img');
        cover.src = track.cover;
        cover.alt = track.title + ' cover';
        cover.classList.add('song-cover');

        const songInfo = document.createElement('div');
        songInfo.classList.add('song-info');

        const title = document.createElement('h3');
        title.classList.add('song-title');
        title.textContent = track.title;

        const artist = document.createElement('p');
```

```

        artist.classList.add('song-artist');
        artist.textContent = track.artist;

        const audio = document.createElement('audio');
        audio.src = track.audio;
        audio.controls = true;

        songInfo.appendChild(title);
        songInfo.appendChild(artist);

        songItem.appendChild(cover);
        songItem.appendChild(songInfo);
        songItem.appendChild(audio);

        songList.appendChild(songItem);
    });
    });
    .catch(error => {
        console.error('Error fetching top tracks:', error);
    });
}

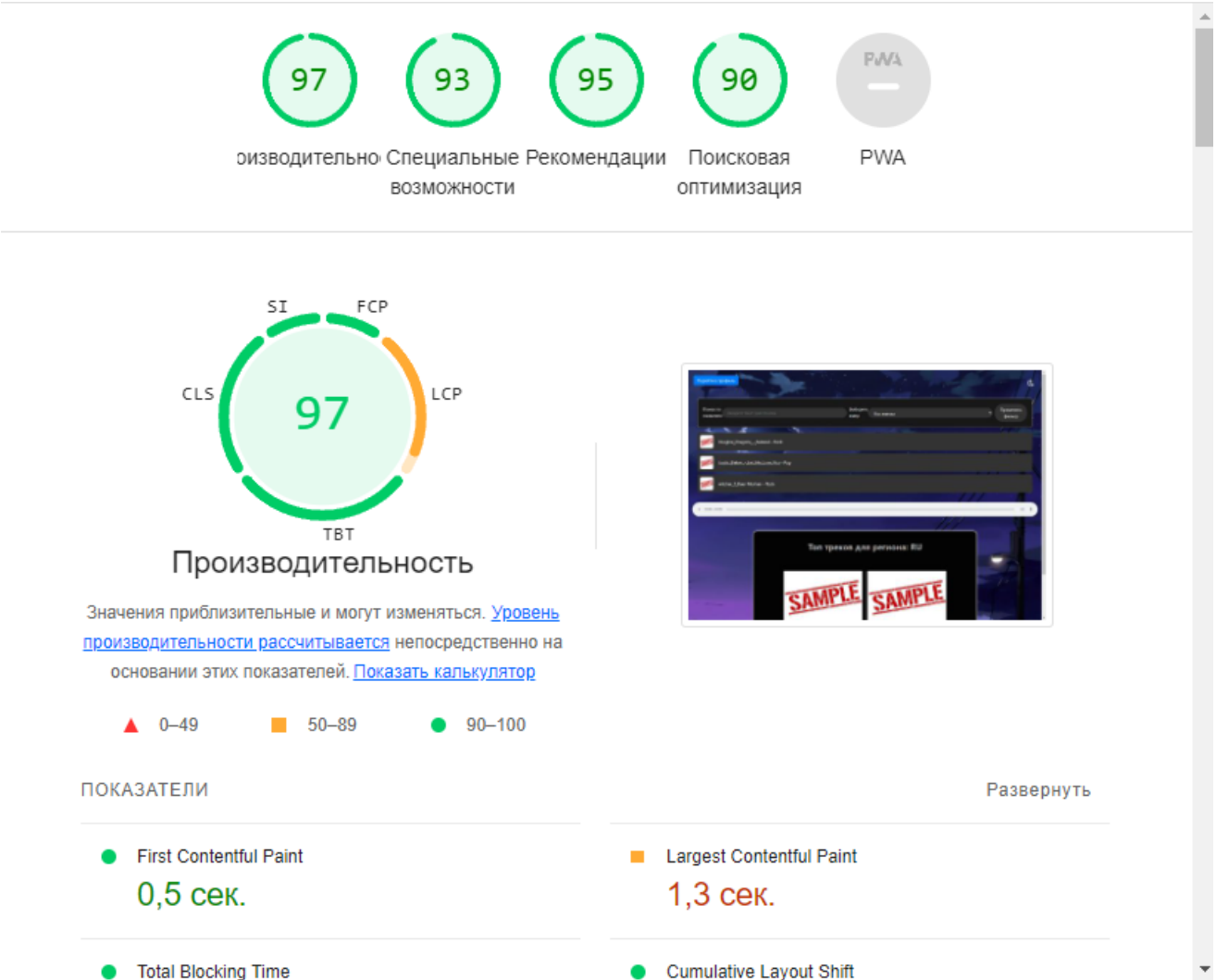
// Функция для получения региона пользователя
function getUserRegion() {
    const userLanguage = navigator.language || navigator.userLanguage;
    const countryCode = userLanguage.split('-')[1]; // Получаем код страны из
языкового тега

    // Возвращаем код страны или используем 'RU' для России по умолчанию
    return countryCode || 'RU';
}

// Определяем регион пользователя и вызываем fetchTopTracks
const userRegion = getUserRegion();
fetchTopTracks(userRegion);
document.getElementById('userRegion').textContent = userRegion;
</script>

```

В ходе работы с Lighthouse и Dev tools были изменены некоторые шрифты, фоны и цвета на сайте, для повышения доступности сайта.



SI

FCP

CLS

LCP

TBT

97

Производительность

Значения приблизительные и могут изменяться. [Уровень производительности рассчитывается](#) непосредственно на основании этих показателей. [Показать калькулятор](#)

▲ 0–49

■ 50–89

● 90–100

Развернуть

● First Contentful Paint

0,5 сек.

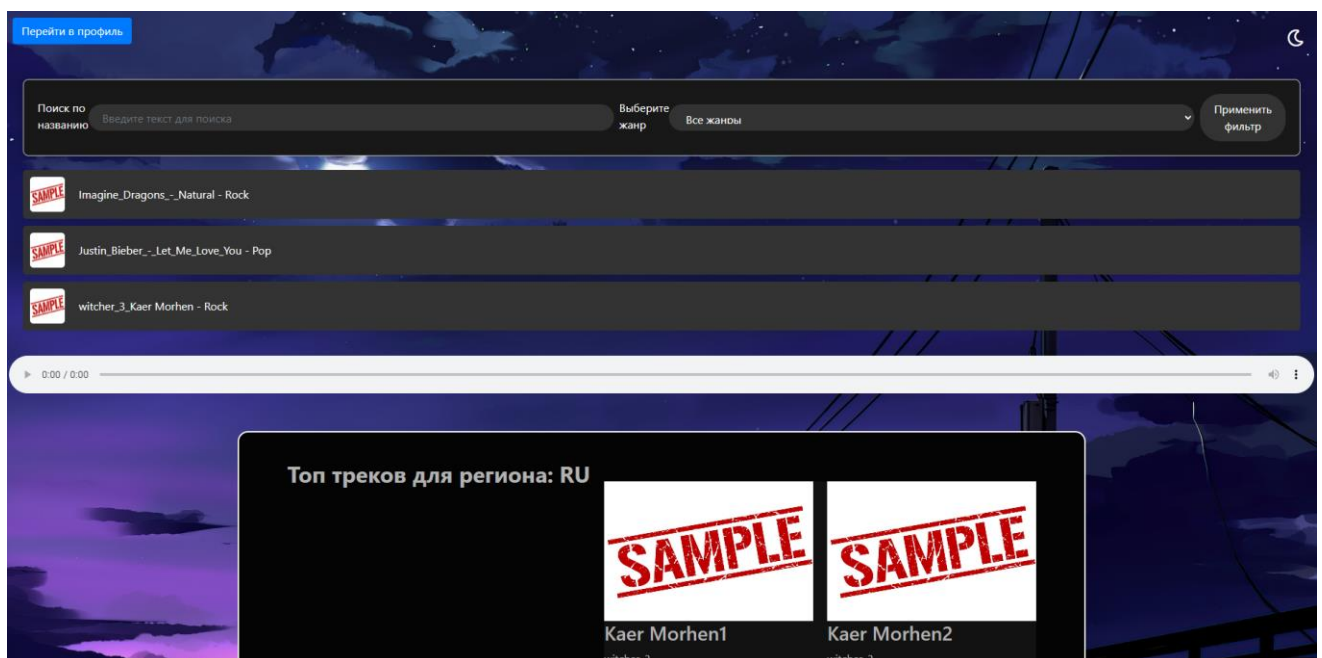
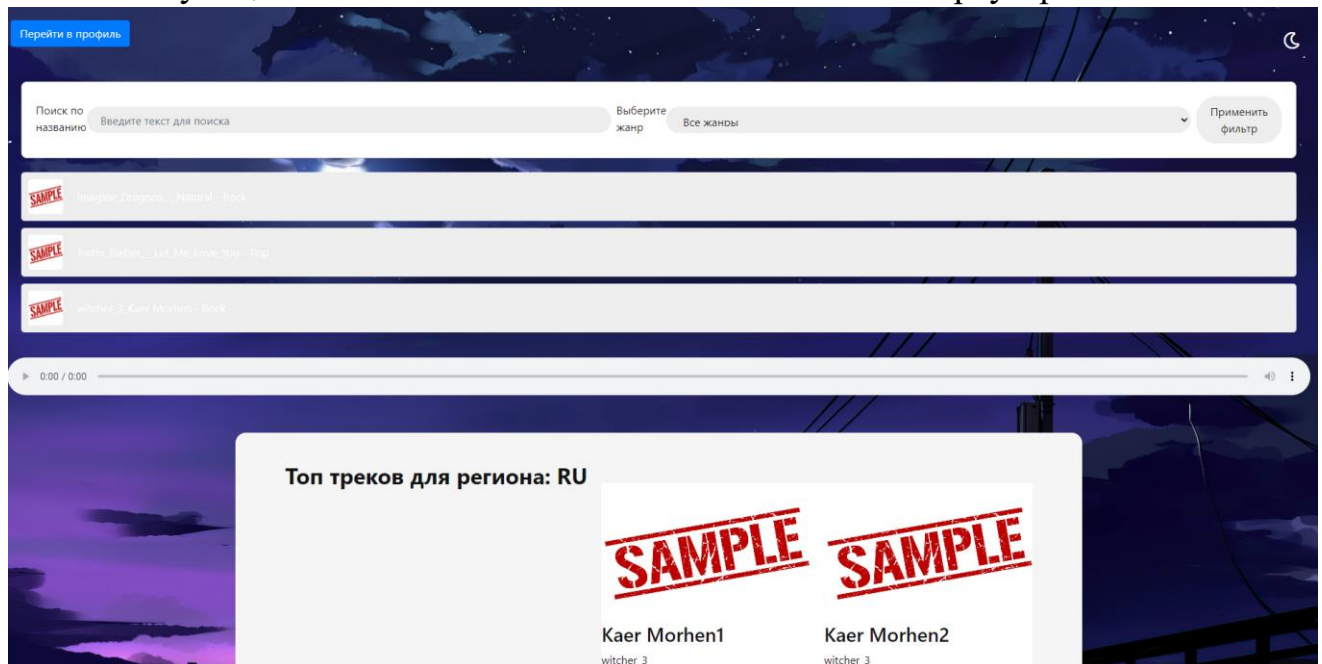
■ Largest Contentful Paint

1,3 сек.

● Total Blocking Time

● Cumulative Layout Shift

На сайте доступна возможность смены темы (светлой и темной), по нажатию соответствующей кнопки. Изначально выставляется тема браузера пользователя.



Вывод:

Лабораторная работа 2: Взаимодействие с внешним API

В соответствии с поставленным заданием, я реализовал интеграцию с внешним API для получения топа треков, с использованием кастомного API. Для этого были использованы средства работы с HTTP запросами, такие как fetch.

Процесс работы включал в себя определение геолокации пользователя, отправку запроса к APIс получением данных о топе треков в регионе и выводом двух карточек этих треков.

Реализованный функционал успешно интегрирован в приложение, и его работоспособность проверена на различных устройствах и браузерах.

Домашняя работа 2: Доступность в HTML

Для улучшения доступности ранее созданного сайта были добавлены необходимые HTML-атрибуты ко всему контенту на странице. Далее, с использованием инструментов Dev Tools браузера Firefox и сервиса Google Lighthouse, была проведена проверка доступности.

Результаты проверки подтвердили улучшение доступности, и сайт теперь соответствует стандартам доступности, рекомендуемым Google и Mozilla.

Домашняя работа 3: CSS-переменные, темизация сайта средствами CSS

В соответствии с заданием, я выполнил темизацию ранее реализованного сайта, используя CSS-переменные. Процесс включал в себя определение переменных для различных стилевых параметров и их применение на странице.

Реализованный функционал позволяет легко менять тему сайта, изменяя всего лишь несколько переменных.

Все выполненные задания удовлетворяют поставленным требованиям и стандартам, что подтверждено тестированием и проверкой.