

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Лабораторная работа 1
«Вёрстка сайта средствами HTML, CSS и Bootstrap»

Выполнил:
Студент гр. К33402
Тарабанов Алексей Вячеславович
Проверил:
Добряков Д. И.

Санкт-Петербург

2024 год

Задача лабораторной работы 1:

В рамках данной лабораторной работы было предложено выбрать один из нескольких вариантов, выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать моменты, в которых необходим JS (например, открытие модальных окон).

Выбранный вариант: музыкальный плеер.

Музыкальный плеер: создание веб-приложения для воспроизведения музыки, включающего функции поиска и добавления треков, создания плейлистов и контроля громкости.

Ход работы:

Были написаны следующие HTML страницы:

- Файл main.html:

```
● <!DOCTYPE html>
● <html lang="en">
● <head>
●   <meta charset="UTF-8">
●   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
●   <title>Музыкальный плеер</title>
●   <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css" rel="stylesheet">
●   <link
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/a
ll.min.css" rel="stylesheet">
●   <style>
●     body {
●       margin: 0;
●       font-family: Arial, sans-serif;
●       background-image: url('fon.jpg');
●       background-size: cover;
●       background-position: center;
●       color: #fff;
```

```
        position: relative;
    }

    .container-fluid {
        padding-top: 60px;
        padding-right: 20px;
        padding-left: 20px;
    }

    .form-container {
        border-radius: 10px;
        background-color: #181818;
        color: #fff;
        padding: 20px;
        box-shadow: 0 4px 16px rgba(0, 0, 0, 0.2);
        position: relative;
        margin-right: 20px;
        margin-left: 20px;
    }

    .search-bar input[type="text"],
    .search-bar select {
        border-radius: 30px;
        padding: 10px 20px;
        border: none;
        color: #fff;
    }

    .search-bar label {
        display: block;
        margin-bottom: 5px;
    }

    .search-bar button {
        border-radius: 30px;
        padding: 10px 20px;
        border: none;
        color: #fff;
        background-color: transparent;
    }

    #audioPlayer {
        width: 100%;
        margin-top: 20px;
    }

    .music-list {
        padding: 0;
        margin-right: 20px;
    }
```

```
        margin-left: 20px;
    }

    .music-item {
        display: flex;
        align-items: center;
        margin-bottom: 10px;
        cursor: pointer;
        transition: background-color 0.3s ease;
        border-radius: 5px;
        padding: 10px;
        background-color: #333;
    }

    .music-item img {
        width: 50px;
        height: 50px;
        margin-right: 20px;
        border-radius: 5px;
    }

    .music-item span {
        color: #fff;
    }

    .music-item:hover {
        background-color: transparent;
    }

    .music-item:hover span {
        color: #fff;
    }

    .music-item:hover img {
        border: 2px solid #1db954;
    }

    .profile-link {
        top: 20px;
        left: 20px;
        z-index: 10;
    }

    .top-tracks-container h2 {
        font-size: 32px;
        font-weight: bold;
    }
```

```

    .song-item img {
        width: 100%;
        height: auto;
        max-width: 100%;
        max-height: 200px;
        object-fit: cover;
    }

    .light-mode .theme-toggle::before {
        content: "\f186";
        font-family: 'Font Awesome 5 Free', serif;
    }

    .dark-mode .theme-toggle::before {
        content: "\f186";
        font-family: 'Font Awesome 5 Free', serif;
    }

</style>
<link href="css/style.css" rel="stylesheet">
</head>
<body>
<svg xmlns="http://www.w3.org/2000/svg" style="display: none;">
    <symbol id="icon-search" viewBox="0 0 27 27">
        <svg width="35" height="35" viewBox="00 0 30 30">
            <path d="M19.39 17.97l-4.42-4.42c.74-1.14 1.18-2.48
1.18-3.95C16.15 5.59 12.56 2 8.07 2S0 5.59 0 10.08s4.58 8.08 9.07
8.08c1.47 0 2.81-.44 3.95-1.18l4.42 4.42c.49.49 1.28.49 1.77 0
.49-.49.49-1.28 0-1.77z"m2 10.08C2 6.69 5.36 3.33 8.75 3.33S15.5 6.69
15.5 10.08s-3.36 6.75-6.75 6.75S2 13.47 2 10.08z" fill="red"/>
        </svg>
    </symbol>

    <symbol id="headphones-icon" viewBox="0 0 24 24">
        <path d="M3 17V12C3 7.02944 7.02944 3 12 3C16.9706 3 21 7.02944
21 12V17M8.28571 21C7.02335 21 6 19.9767 6 18.7143V15.2857C6 14.0233
7.02335 13 8.28571 13C9.23249 13 10 13.7675 10 14.7143V19.2857C10
20.2325 9.23249 21 8.28571 21Z"m15.7143 21C16.9767 21 18 19.9767 18
18.7143V15.2857C18 14.0233 16.9767 13 15.7143 13C14.7675 13 14 13.7675
14 14.7143V19.2857C14 20.2325 14.7675 21 15.7143 21Z" stroke="#000000"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>
    </symbol>
</svg>
<div class="container-fluid">
    <a href="profile.html" class="btn btn-primary mb-3
profile-link">Перейти в профиль</a>
    <style>
    .form-container {

```

```

    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}

.headphones-icon {
    display: flex;
    align-items: center;
    margin-bottom: 20px;
}

.search-bar {
    display: flex;
    align-items: center;
    justify-content: space-between;
    width: 100%;
    max-width: 800px;
    margin-top: 20px;
}

label, #searchInput, #genreFilter {
    flex: 1;
    margin-right: 10px;
}

.icon-search {
    width: 30px;
    height: 30px;
    fill: red;
}

button {
    border: none;
    background: none;
    cursor: pointer;
}
</style>
<div class="form-container mt-5">
    <div class="headphones-icon">
        <svg class="icon" width="37" height="37">
            <use xlink:href="#headphones-icon"/>
        </svg>
        <span>Музыка - это хорошо</span>
    </div>
    <div class="d-flex align-items-center search-bar">
        <style>
            label {
                font-size: 16px;

```

```

        margin-bottom: 60px;
        display: block;
    }

    #searchInput {
        width: 550px;
        height: 40px;
        font-size: 16px;
    }
</style>

<label for="searchInput">Поиск</label>
<input type="text" id="searchInput" placeholder="Введите
текст для поиска" class="form-control mr-2">

<style>
    #genreFilter {
        width: 350px;
        height: 50px;
        font-size: 14px;
    }
</style>

<label for="genreFilter">Поиск по жанру</label>
<select id="genreFilter" class="form-control mr-2">
    <option value="all">Жанры</option>
    <option value="rock">Рок</option>
    <option value="pop">Поп</option>
    <option value="hiphop">Хип-хоп</option>
</select>

<style>
    .icon-search {
        width: 70px;
        height: 18px;
        fill: red;
    }
</style>
<button onclick="filterMusic()" class="btn btn-primary">
    <svg class="icon-search" viewBox="0 0 27 27">
        <use xlink:href="#icon-search"></use>
    </svg>
</button>
</div>
</div>
<ul class="music-list">
    <li class="music-item"
onclick="playMusic('music/apillsinjuice-freetato-mp3.mp3')">

```

```

    
    <span>Fretato - Rock</span>
  </li>
  <li class="music-item"
onclick="playMusic('music/icyk-cyper-feat.-igor-cyba-dymok.mp3')">
    
    <span>Дымок - Pop</span>
  </li>
  <li class="music-item"
onclick="playMusic('music/ЧёрноеОлово-Митоз_(muzmo.su).mp3')">
    
    <span>Митоз! - Rock</span>
  </li>
</ul>
<div id="playlistModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    <h3>Выберите плейлист</h3>
    <label for="modalPlaylistSelect"></label><select
id="modalPlaylistSelect"></select>
    <button onclick="addToSelectedPlaylist()">Добавить в
плейлист</button>
  </div>
</div>
<audio id="audioPlayer" controls></audio>
</div>

<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.m
in.js"></script>
<link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap
.min.css" rel="stylesheet">
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.m
in.js"></script>
<script src="js/main.js"></script>
<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);
    });
}

function getUserRegion() {
    const userLanguage = navigator.language ||
navigator.userLanguage;
    const countryCode = userLanguage.split('-')[1];
}

```

```

    return countryCode || 'RU';
  }

  const userRegion = getUserRegion();
  fetchTopTracks(userRegion);
  document.getElementById('userRegion').textContent = userRegion;
}

</script>
</body>
</html>

```

- Файл registration.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Регистрация</title>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .form-container {
      max-width: 400px;
      padding: 20px;
      background-color: #fff;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      margin: 50px auto 0;
    }

    input[type="text"],
    input[type="password"],
    input[type="submit"] {
      width: 300px;
      padding: 10px;
      margin: 10px 0;
      border-radius: 5px;
      border: 1px solid #ccc;
      font-size: 16px;
    }

```

```

    }
    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        cursor: pointer;
    }
    input[type="text"],
    input[type="password"],
    input[type="submit"] {
        transition: all 0.3s ease;
    }

    input[type="text"]:focus,
    input[type="password"]:focus {
        border-color: #4CAF50;
        box-shadow: 0 0 5px rgba(76, 175, 80, 0.5);
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }

    .error-message {
        color: #f00;
        margin-top: 5px;
        display: none;
    }
}
</style>
<link href="css/style.css" rel="stylesheet">
</head>
<body>
<div class="form-container">
    <h2>Регистрация</h2>
    <form id="registrationForm">
        <label for="username"></label><input type="text" id="username"
placeholder="Имя пользователя">
        <br>
        <label for="email"></label><input type="text" id="email"
placeholder="Email">
        <br>
        <label for="password"></label><input type="password"
id="password" placeholder="Пароль">
        <br>
        <label for="confirmPassword"></label><input type="password"
id="confirmPassword" placeholder="Повторите пароль">
        <br>
        <input type="submit" value="Создать аккаунт"
style="background-color: #222eaf; color: white;">
    </form>

```

```

•   <p>Есть аккаунт? <a href="login.html">Войти</a></p>
•   <div class="error-message" id="registrationError">Введите
корректные данные</div>
• </div>
• <script src="js/registration.js"></script>
•
• <script>
•   const registrationForm =
document.getElementById('registrationForm');
•   const usernameInput = document.getElementById('username');
•   const emailInput = document.getElementById('email');
•   const passwordInput = document.getElementById('password');
•   const confirmPasswordInput =
document.getElementById('confirmPassword');
•   const registrationError =
document.getElementById('registrationError');
•
•   registrationForm.addEventListener('submit', function(event) {
•     event.preventDefault();
•     if (!validateRegistration(usernameInput.value,
emailInput.value, passwordInput.value, confirmPasswordInput.value)) {
•       registrationError.style.display = 'block';
•     } else {
•       // Simulate successful registration
•       alert('Успешная регистрация!');
•       window.location.href = 'login.html';
•     }
•   });
•
•   function validateRegistration(username, email, password,
confirmPassword) {
•     const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
•     return username.trim() !== '' &&
•       emailRegex.test(email) &&
•       password.length > 4 &&
•       password === confirmPassword;
•   }
• </script>
• </body>
• </html>

```

- Файл login.html:

```

• <!DOCTYPE html>
• <html lang="en">
• <head>
•   <meta charset="UTF-8">
•   <meta name="viewport" content="width=device-width,
initial-scale=1.0">

```

```
<title>Вход</title>
<style>
  body {
    margin: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f0f0f0;
    font-family: Arial, sans-serif;
  }
  .form-container {
    text-align: center;
    padding: 20px;
    border-radius: 10px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  input[type="text"],
  input[type="password"],
  input[type="submit"] {
    width: 300px;
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
    border: 1px solid #ccc;
    font-size: 16px;
  }
  input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    cursor: pointer;
  }
  input[type="text"],
  input[type="password"],
  input[type="submit"] {
    transition: all 0.3s ease;
  }
  input[type="text"]:focus,
  input[type="password"]:focus {
    border-color: #4CAF50;
    box-shadow: 0 0 5px rgba(76, 175, 80, 0.5);
  }
  input[type="submit"]:hover {
    background-color: #45a049;
  }
}
```

```

    .error-message {
        color: #f00;
        margin-top: 5px;
        display: none;
    }
</style>
</head>
<body>
<div class="form-container">
    <h2>Вход</h2>
    <form id="loginForm">
        <label for="username"></label><input type="text" id="username"
placeholder="Логин">
        <br>
        <label for="password"></label><input type="password"
id="password" placeholder="Пароль">
        <br>
        <input type="submit" value="Войти" style="background-color:
#222eaf; color: white;">
    </form>
    <p>Нет аккаунта? <a href="registration.html">Создать
аккаунт</a></p>
    <div class="error-message" id="loginError">Введите корректные
данные</div>
</div>
<script src="js/login.js"></script>
<script>
    const loginForm = document.getElementById('loginForm');
    const usernameInput = document.getElementById('username');
    const passwordInput = document.getElementById('password');
    const loginError = document.getElementById('loginError');

    loginForm.addEventListener('submit', function(event) {
        event.preventDefault();
        if (!validateLogin(usernameInput.value, passwordInput.value)) {
            loginError.style.display = 'block';
        } else {
            alert('Успешный вход!');
            window.location.href = 'main.html';
        }
    });

    function validateLogin(username, password) {
        // Просто проверим, что поля не пустые
        return username.trim() !== '' && password.length > 4;
    }
</script>
<script></script>
</body>

```

```
• </html>
```

- Файл profile.html:

```
• <!DOCTYPE html>
• <html lang="en">
• <head>
•   <meta charset="UTF-8">
•   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
•   <title>Личный кабинет</title>
•   <style>
•     body {
•       margin: 0;
•       font-family: Arial, sans-serif;
•       height: 100vh;
•       display: flex;
•       justify-content: center;
•       align-items: center;
•     }
•
•     .form-container {
•       text-align: center;
•       padding: 20px;
•       border-radius: 10px;
•       background-color: #fff;
•       box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
•       width: 80%;
•       max-width: 600px;
•       color: #222;
•     }
•     input[type="text"],
•     input[type="password"],
•     input[type="submit"] {
•       width: 300px;
•       padding: 10px;
•       margin: 10px 0;
•       border-radius: 5px;
•       border: 1px solid #ccc;
•       font-size: 16px;
•     }
•     input[type="submit"] {
•       background-color: #4CAF50;
•       color: white;
•       cursor: pointer;
•     }
•     input[type="text"],
•     input[type="password"],
```

```

    input[type="submit"] {
        transition: all 0.3s ease;
    }

    input[type="text"]:focus,
    input[type="password"]:focus {
        border-color: #4CAF50;
        box-shadow: 0 0 5px rgba(76, 175, 80, 0.5);
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }

    .profile-info {
        text-align: center;
        margin-bottom: 20px;
    }

    .playlist-container {
        text-align: center;
        margin-bottom: 20px;
    }

</style>
<link href="css/style.css" rel="stylesheet">
</head>
<body>
<div class="form-container">
    <h2>Личный кабинет</h2>
    <a href="main.html">Вернуться на главную</a>
    <div class="profile-info">
        <p>Имя: Алексей</p>
        <p>Email: tarabanov@example.com</p>
        <h2>Плейлисты пользователя</h2>
        <div class="playlist-container">
            <h3>Плейлисты</h3>
            <button onclick="createPlaylist()">Создать
плейлист</button>
            <ul id="playlistContainer">
            </ul>
        </div>
        <button onclick="logout()">Выйти из аккаунта</button>
    </div>
</div>
<script src="js/profile.js"></script>
<script>
    const currentUser = 'user123';

```



```

    const userPlaylists =
JSON.parse(localStorage.getItem(currentUser)) || {};
    const playlistContainer =
document.getElementById('playlistContainer');

    for (let playlist in userPlaylists) {
        const li = document.createElement('li');
        li.textContent = playlist;
        li.classList.add('playlist-item');
        li.onclick = function () {
            alert('Открыт плейлист: ' + playlist);
        };
        playlistContainer.appendChild(li);
    }
</script>
<script src="js/profile.js"></script>
<script>
    const currentUser = 'user123'; // На будущее
    const userPlaylists =
JSON.parse(localStorage.getItem(currentUser)) || {};
    const playlistContainer =
document.getElementById('playlistContainer');

    for (let playlist in userPlaylists) {
        const li = document.createElement('li');
        li.textContent = playlist;
        playlistContainer.appendChild(li);
    }

</script>

<script>
    function createPlaylist() {
        // Добавление нового плейлиста (для примера, пока только
имитация)
        const newPlaylistName = prompt('Введите название плейлиста:');
        if (newPlaylistName) {
            const playlistContainer =
document.querySelector('.playlist-container');
            const newPlaylist = document.createElement('div');
            newPlaylist.classList.add('playlist-item');
            newPlaylist.textContent = newPlaylistName;
            newPlaylist.addEventListener('click', function() {
                alert('Открыт плейлист: ' + newPlaylistName);
            });
            playlistContainer.appendChild(newPlaylist);
        }
    }

```

```

•     function logout() {
•         // Перенаправление на страницу входа (имитация выхода из
аккаунта)
•         window.location.href = 'login.html';
•     }
• </script>
• </body>
• </html>

```

Были написаны в том числе и JS файлы для обработки функционала:

- Файл main.js:

```

•     function playMusic(song) {
•         const audioPlayer = document.getElementById('audioPlayer');
•         audioPlayer.src = song;
•         audioPlayer.play();
•     }
•
•     // Функция для фильтрации музыки
•     function filterMusic() {
•         const searchInput =
document.getElementById('searchInput').value.toLowerCase();
•         const genreFilter =
document.getElementById('genreFilter').value.toLowerCase();
•         const musicItems = document.querySelectorAll('.music-item');
•
•         musicItems.forEach(function(item) {
•             const songInfo = item.textContent.toLowerCase();
•             const songGenre = songInfo.includes(genreFilter) || genreFilter
=== 'all';
•             const songMatch = songInfo.includes(searchInput) || searchInput
=== '';
•             if (songGenre && songMatch) {
•                 item.style.display = 'flex';
•             } else {
•                 item.style.display = 'none';
•             }
•         });
•     }
•
•     // Функция для добавления песни в плейлист профиля
•     function addToPlaylist(song) {
•         const currentUser = 'user123';
•
•         let userPlaylist = JSON.parse(localStorage.getItem(currentUser)) ||
[];

```

```

•   userPlaylist.push(song);
•   localStorage.setItem(currentUser, JSON.stringify(userPlaylist));
•
•   alert('Песня добавлена в ваш плейлист!');
• }
•
• function closeModal() {
•   const myModal = new
bootstrap.Modal(document.getElementById('playlistModal'));
•   myModal.hide();
• }
•
•
• // Функция для добавления трека в выбранный плейлист
• function addToSelectedPlaylist() {
•   const selectedPlaylist =
document.getElementById('modalPlaylistSelect').value;
•   // Логика добавления трека в плейлист
•   alert('Трек добавлен в плейлист');
•   closeModal();
• }

```

- Файл registration.js:

```

•   document.getElementById('registrationForm').addEventListener('submit',
function(event) {
•   event.preventDefault();
•   // Здесь будет код для отправки данных на сервер и создания нового
аккаунта
•   });

```

- Файл login.js:

```

•   document.getElementById('loginForm').addEventListener('submit',
function(event) {
•   event.preventDefault();
•   // Здесь можно добавить код для отправки данных на сервер или
проверки логина/пароля
•   });

```

- Файл profile.js:

```

•   function createPlaylist() {
•   const newPlaylistName = prompt('Введите название плейлиста:');
•   if (newPlaylistName) {
•   // Логика создания нового плейлиста
•   }

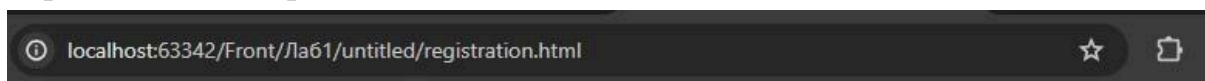
```

```
• }  
•  
•  
• // Функция для выхода из аккаунта  
• function logout() {  
•     // Логика выхода из аккаунта  
• }
```

Итог выполненной работы:

Скриншоты страниц в браузере:

- Страница “Регистрация”:



Регистрация

Создать аккаунт

Есть аккаунт? [Войти](#)

- Страница “Вход в профиль”:

localhost:63342/Front/Лаб1/untitled/login.html

Вход

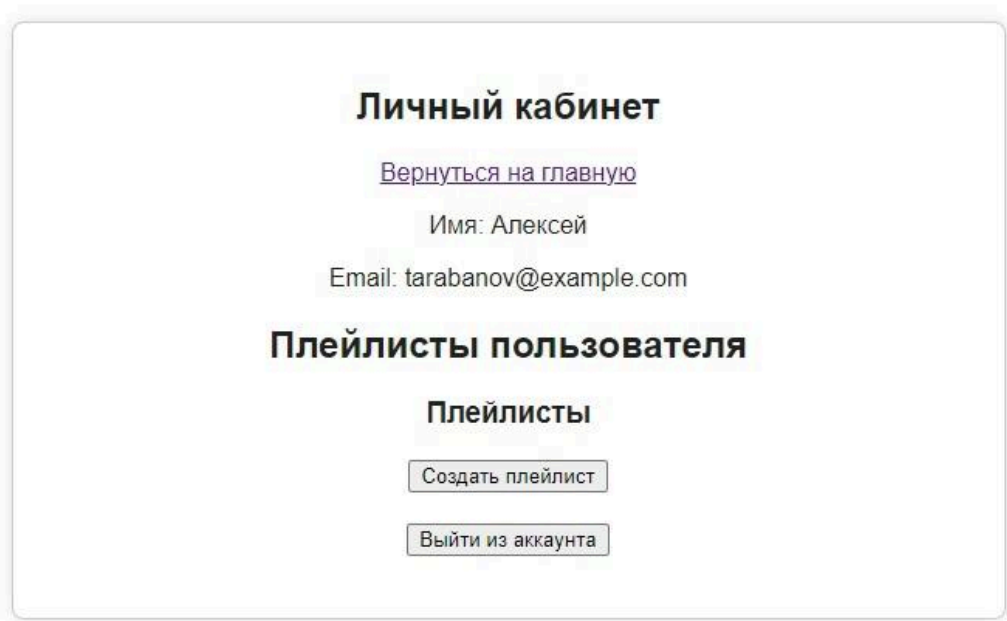
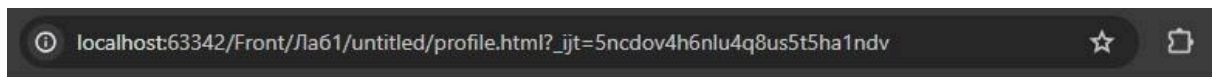
Логин

Пароль

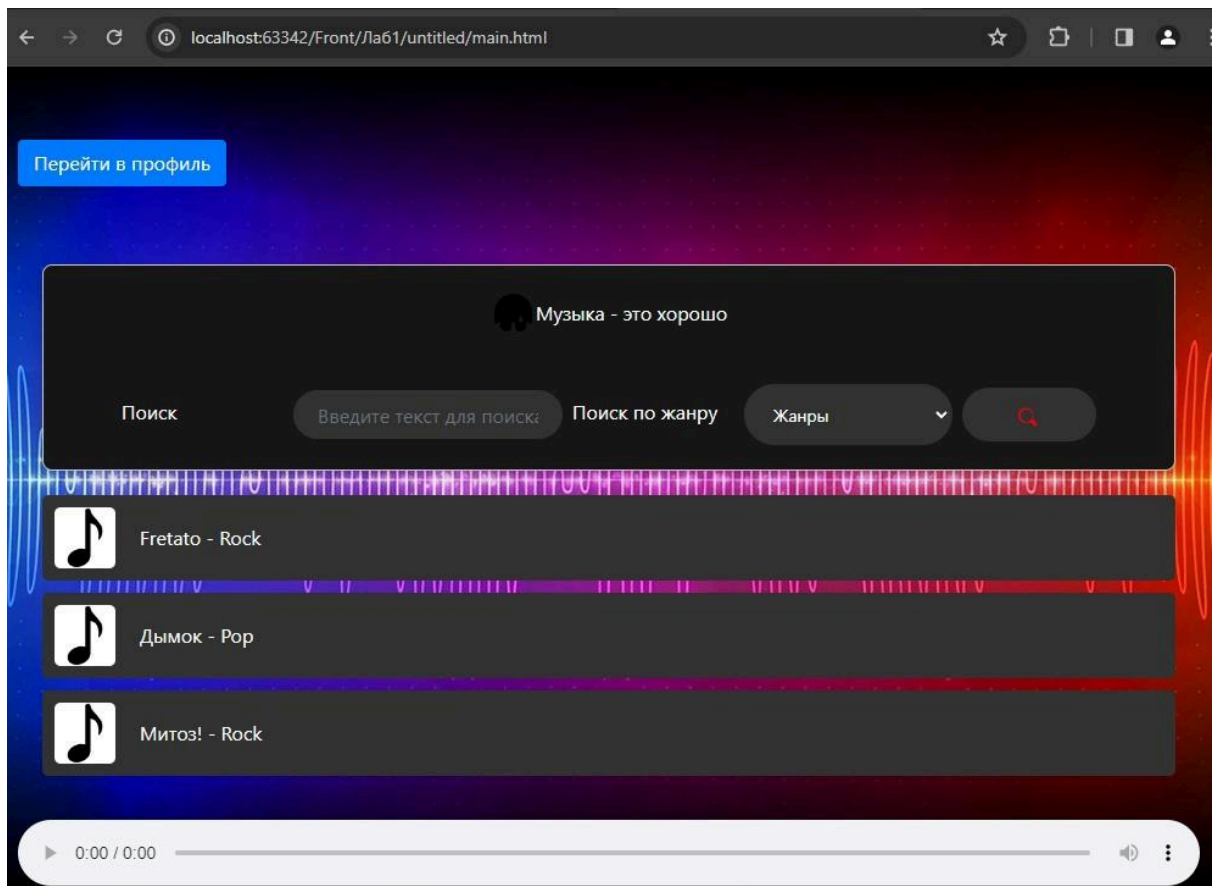
Войти

Нет аккаунта? [Создать аккаунт](#)

- Страница “Профиль”:



- Главная страница плеера:



Все кнопки на каждой странице кликабельны и выполняют свой функционал - на сколько возможно в рамках написания только клиентской части.

Вывод:

Работа по созданию музыкального плеера веб-приложения включала ряд ключевых этапов, к ним относятся: верстка сайта, а именно создание пользовательского интерфейса с использованием HTML, CSS и Bootstrap для обеспечения удобства и эстетики; интерактивные элементы с JS - реализация функционала с помощью JS для управления основными возможностями приложения: поиск и фильтрация, что дает возможность поиска треков по названию и, непосредственно, сама фильтрация по жанру музыки, создание плейлистов и создание новых плейлистов, управление громкостью воспроизведения музыки, открытие модальных окон для удобного выбора плейлиста при добавлении трека, создание обратной

связи и уведомлений для пользователя, таких как сообщения об успешном добавлении трека или удалении плейлиста.

Так, подводя итог, можно сказать, что в результате проделанной работы получилось веб-приложение, которое обеспечивает возможность прослушивания музыки, управление плейлистами, добавление и удаление треков и позволяет пользователям сохранять свои настройки музыкального контента между сессиями.