

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ И МОДЕЛИРОВАНИЯ БИЗНЕС-
ПРОЦЕССОВ

КУРСОВАЯ РАБОТА

Создание web-приложения с применением современных web-технологий по
тематике «Музыкальный портал».

Руководитель

подпись, дата

А.И. Никитин

Студент УБ23-08Б, 432319581

подпись, дата

Е.С. Чудина

Красноярск 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Анализ технологий	4
1.1 HTML.....	4
1.2 CSS.....	4
1.3 PHP.....	4
1.4 MySQL.....	5
2 Проектирование WEB-приложения.....	5
2.1 Структура базы данных	5
2.1.1 Пользователи (users):	5
2.1.2 Услуги (services):	5
2.1.3 Пользователи и услуги (users_services):	6
2.1.4 Альбомы (albums):	6
2.1.5 Артисты (artists):	6
2.1.6 Клипы (clips):.....	7
2.1.7 Новости (news):	7
2.2 Пользовательский интерфейс	8
2.2.1 Главная страница	8
2.2.2 Страница артиста	8
2.2.3 Страница популярные клипы	8
2.2.4 Страница альбомы	9
2.2.5 Профиль пользователя	9
2.2.6 Административная панель	10
3 Реализация WEB-приложения.....	11
3.1 Регистрация пользователя	11
3.2 Авторизация пользователя	13
3.3 Главная страница.....	14
3.4 Артисты	15
3.5 Популярные клипы	18
3.6 Альбомы	19
3.7 Панель администратора.....	20
3.8 Панель пользователя.....	23
ЗАКЛЮЧЕНИЕ.....	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	26

ВВЕДЕНИЕ

В современном мире web-технологии стремительно развиваются, а с ними растет и потребность в удобных и функциональных приложениях, которые могут решать различные задачи пользователей. Музыка занимает важное место в жизни людей, поэтому идея создания музыкального портала становится особенно актуальной. Такое приложение объединяет слушателей, исполнителей и музыкальные сервисы, предлагая комфортное взаимодействие с музыкальным контентом. В рамках этого проекта разрабатывается музыкальный портал, который позволит слушать музыку, смотреть клипы и следить за новинками.

Основной целью данного проекта является разработка музыкального портала – современного web-приложения, которое станет платформой для удобного доступа к музыкальному контенту. При создании приложения будут использоваться современные технологии, обеспечивающие его производительность, безопасность и адаптацию под разные устройства.

Актуальность данного проекта обусловлена тем, что музыкальные сервисы в интернете с каждым годом становятся все более востребованными. Это связано с популярностью стриминговых платформ и стремлением пользователей получать удобный доступ к музыке. Однако многие существующие приложения не предоставляют достаточной гибкости и персонализации. В разработке такого музыкального портала будут использованы такие современные технологии, как HTML5, CSS, JavaScript, PHP и MySQL. Этот проект не только актуален, но и демонстрирует возможности новых инструментов для создания функциональных и привлекательных web-приложений.

1 Анализ технологий

1.1 HTML

HTML (HyperText Markup Language) — это основной язык разметки, используемый для создания веб-страниц. За годы своего развития HTML стал основой современной веб-разработки, позволяя структурировать контент, управлять его отображением и взаимодействовать с другими технологиями, такими как CSS и JavaScript. Современные версии HTML включают в себя HTML5 и его последующие улучшения.

HTML не является языком программирования; это язык разметки, и используется, чтобы сообщать браузеру, как отображать веб-страницы, которые посещает пользователь. HTML состоит из ряда элементов, которые используются, чтобы вкладывать или оборачивать различные части контента, чтобы заставить контент отображаться или действовать определённым образом.

1.2 CSS

CSS (Cascading Style Sheets) — это каскадные таблицы стилей, которые используются для определения внешнего вида и форматирования веб-страниц. CSS отделяет содержимое документа, которое задается с помощью HTML или XML, от его представления, что позволяет улучшить гибкость и контроль над визуальными аспектами веб-страницы.

1.3 PHP

PHP (Hypertext Preprocessor) — это широко используемый язык скриптов общего назначения, который особенно подходит для веб-разработки и может быть встроен в HTML. Его основным преимуществом является возможность создавать динамические и интерактивные веб-страницы.

1.4 MySQL

MySQL — это реляционная система управления базами данных (СУБД), которая используется для хранения, управления и обработки данных. MySQL позволяет организовать данные в структурированные таблицы, поддерживая сложные запросы и манипуляции с данными.

2 Проектирование WEB-приложения

2.1 Структура базы данных

2.1.1 Пользователи (users):

1. id (INT, первичный ключ) – Уникальный идентификатор пользователя;
2. email (VARCHAR(100), уникальный, не может быть NULL) – Электронная почта пользователя;
3. password (VARCHAR(50), не может быть NULL) – Пароль пользователя;
4. role (ENUM('admin', 'user'), по умолчанию 'user') – Роль пользователя (админ или пользователь).

2.1.2 Услуги (services):

1. id (INT, первичный ключ) – Уникальный идентификатор услуги;
2. name (VARCHAR(100), уникальный, не может быть NULL) – Название услуги.

2.1.3 Пользователи и услуги (users_services):

1. id (INT, первичный ключ, автоинкремент) – Уникальный идентификатор связи;
2. users_id (INT, внешний ключ, не может быть NULL) – Идентификатор пользователя из таблицы users;
3. services_id (INT, внешний ключ, не может быть NULL) – Идентификатор услуги из таблицы services.

2.1.4 Альбомы (albums):

1. id (INT, первичный ключ, автоинкремент) – Уникальный идентификатор альбома;
2. title (VARCHAR(100), не может быть NULL) – Название альбома;
3. cover_url (VARCHAR(255), не может быть NULL) – URL обложки альбома;
4. genius_url (VARCHAR(255), не может быть NULL) – URL страницы альбома на Genius.

2.1.5 Артисты (artists):

1. id (INT, первичный ключ, автоинкремент) – Уникальный идентификатор артиста;
2. name (VARCHAR(100), не может быть NULL) – Имя артиста;
3. bio (TEXT) – Биография артиста;
4. image_url (VARCHAR(255)) – URL изображения артиста;
5. genius_url (VARCHAR(255)) – URL страницы артиста на Genius.

2.1.6 Клипы (clips):

1. id (INT, первичный ключ, автоинкремент) – Уникальный идентификатор клипа;
2. title (VARCHAR(100), не может быть NULL) – Название клипа;
3. video_url (VARCHAR(255), не может быть NULL) – URL видео клипа;
4. description (TEXT) – Описание клипа.

2.1.7 Новости (news):

1. id (INT, первичный ключ, автоинкремент) – Уникальный идентификатор новости;
2. text (TEXT, не может быть NULL) – Текст новости;
3. created_at (TIMESTAMP, по умолчанию текущее время) – Дата и время создания новости.

На основе структуры базы данных была сформирована схема, которая представлена на рисунке 2.1.

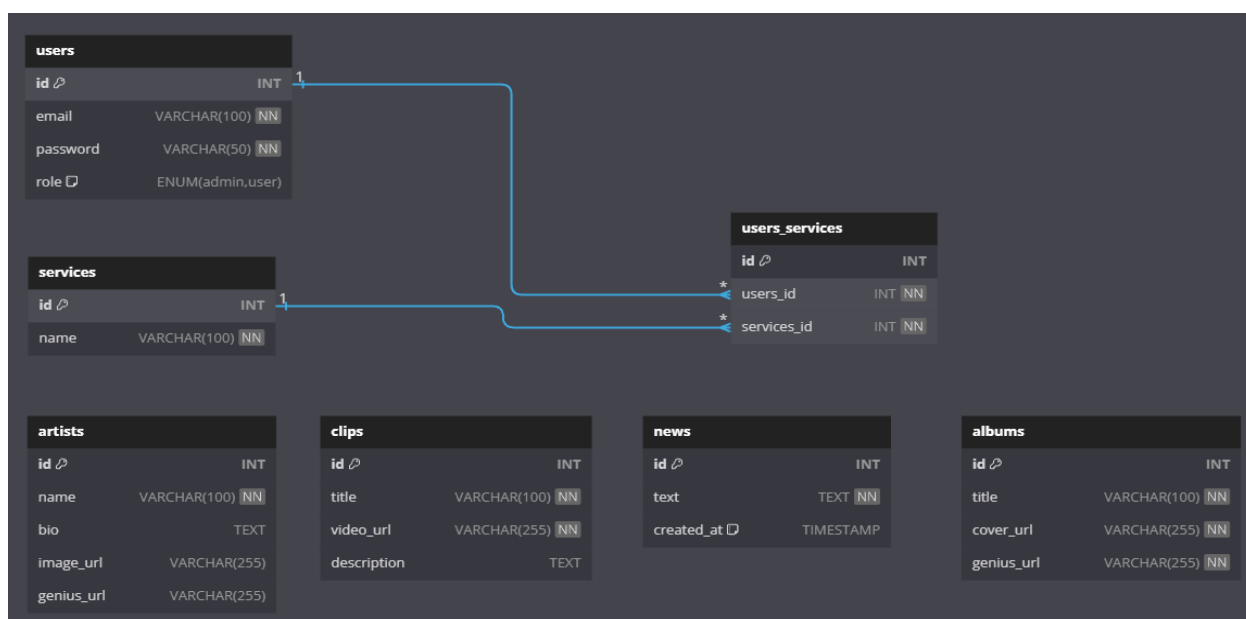


Рисунок 2.1 – Схема базы данных

2.2 Пользовательский интерфейс

2.2.1 Главная страница

Меню навигации: слева расположено вертикальное меню с ссылками на главные разделы сайта, такие как «Артисты», «Популярные клипы», «Альбомы», «Панель администратора» (если пользователь администратор), «Панель пользователя» и кнопка выхода из аккаунта.

Лента новостей: в центральной части страницы пользователи могут публиковать новости и просматривать свежие обновления. Для этого представлена форма для ввода текста новости, и кнопка «Опубликовать».

2.2.2 Страница артиста

Меню навигации: аналогично главной странице, слева расположено вертикальное меню.

Карточки артистов: в центральной части страницы отображаются карточки артистов с их именами, изображениями, краткой информацией и ссылкой на дополнительную информацию. Карточки имеют стильный дизайн и размещены в сетке для удобного просмотра.

Пагинация: внизу страницы размещены ссылки для навигации по страницам, что позволяет пользователям просматривать большое количество артистов, распределенных по страницам.

2.2.3 Страница популярные клипы

Меню навигации: слева находится вертикальное меню, аналогичное предыдущим страницам.

Карточки клипов: В центральной части страницы отображаются карточки клипов с названием, изображением и описанием. Изображения генерируются на основе ссылок на видео на YouTube.

Пагинация: внизу страницы размещены ссылки для навигации по страницам, позволяя пользователям просматривать большое количество клипов.

2.2.4 Страница альбомы

Меню навигации: слева расположено вертикальное меню, аналогичное предыдущим страницам.

Карточки альбомов: В центральной части страницы отображаются карточки альбомов с их названиями и обложками. Карточки имеют стильный дизайн и размещены в сетке для удобного просмотра.

Пагинация: внизу страницы размещены ссылки для навигации по страницам, позволяя пользователям просматривать большое количество альбомов.

2.2.5 Профиль пользователя

Меню навигации: слева расположено вертикальное меню с ссылками на главную страницу и редактирование профиля, а также кнопка выхода из аккаунта.

Информация о пользователе: в центральной части страницы отображается таблица с информацией о текущем пользователе, такой как email и роль.

Список услуг: ниже основной информации представлена таблица с услугами, связанными с пользователем.

2.2.6 Административная панель

Меню навигации: слева расположено вертикальное меню с ссылками на различные разделы управления, такие как «Добавить услугу», «Управление альбомами», «Управление артистами», «Управление клипами» и кнопка выхода из аккаунта.

Таблица пользователей: В центральной части страницы отображается таблица с информацией о пользователях, включая email, роль и действия (редактирование, удаление, управление услугами).

3 Реализация WEB-приложения

Современные веб-приложения предоставляют удобные и функциональные платформы для управления данными и взаимодействия с пользователями. Основная цель данного веб-приложения — создание музыкального портала, который позволяет пользователям регистрироваться, авторизоваться, управлять своим профилем, а администраторам предоставлять услуги и управлять контентом, включая альбомы, артистов и клипы. Проект ориентирован на создание интуитивно понятного и безопасного интерфейса с использованием популярных веб-технологий.

3.1 Регистрация пользователя

Файл `register.php` отвечает за регистрацию новых пользователей. Он включает в себя валидацию данных, проверку уникальности email и добавление нового пользователя в базу данных. В случае успешной регистрации пользователь перенаправляется на главную страницу.

Реализация работы с базой данных представлено на рисунке 3.1. Данные пользователя записываются в таблицу `user`. На рисунке 3.2 представлено окно регистрации, в котором реализована валидация данных.

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];

    if ($password !== $confirm_password) {
        $error = "Пароли не совпадают";
    } else {
        $stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");
        $stmt->execute(['email' => $email]);
        $user = $stmt->fetch();

        if ($user) {
            $error = "Пользователь с таким email уже существует";
        } else {
            $stmt = $pdo->prepare("INSERT INTO users (email, password) VALUES (:email, :password)");
            $stmt->execute(['email' => $email, 'password' => $password]);

            $stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");
            $stmt->execute(['email' => $email]);
            $user = $stmt->fetch();

            $_SESSION['user'] = [
                'email' => $email,
                'role' => $user['role']
            ];

            header("Location: index.php");
            exit();
        }
    }
}

```

Рисунок 3.1 – Реализация регистрации нового пользователя

Регистрация

Email:

Пароль:

Подтверждение пароля:

[Зарегистрироваться](#)

Уже есть аккаунт? [Войти](#)

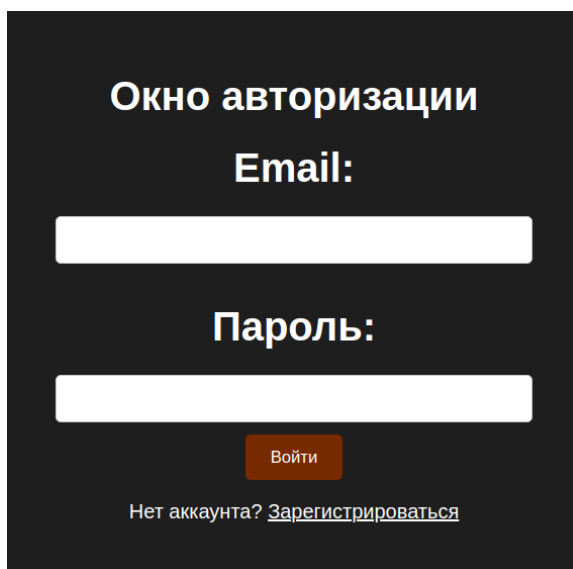
Рисунок 3.2 – Вид окна регистрации

3.2 Авторизация пользователя

Файл login.php отвечает за авторизацию пользователей. Он включает в себя валидацию данных, проверку учетных данных пользователя и установление сессии для авторизованного пользователя. В случае успешной авторизации пользователь перенаправляется на главную страницу. Реализация работы с базой данных представлено на рисунке 3.3. Вид окна авторизации представлен на рисунке 3.4.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    $email = $_POST['email'];  
    $password = $_POST['password'];  
  
    if (checkUserCredentials($email, $password)) {  
        $_SESSION['user'] = getUserByEmail($email);  
        header("Location: index.php");  
        exit();  
    } else {  
        $error = "Неверные данные для входа";  
    }  
}
```

Рисунок 3.3 – Реализация авторизации пользователя



Окно авторизации

Email:

Пароль:

Нет аккаунта? [Зарегистрироваться](#)

Рисунок 3.4 - Вид окна авторизации

3.3 Главная страница

Файл `index.php` отвечает за отображение главной страницы веб-приложения и служит точкой входа для пользователей. Он включает в себя проверку авторизации пользователя, чтобы предотвратить доступ неавторизованных лиц к содержимому сайта. Главная страница предоставляет пользователям возможность публиковать новости и отображает последние обновления в удобном формате. В дополнение к этому, файл `index.php` содержит ссылки на различные разделы сайта, такие как артисты, популярные клипы и альбомы, обеспечивая пользователям легкий доступ к различным функциям приложения. На рисунке 3.5 представлен общий вид страницы. На рисунке 3.6 представлена реализация публикаций новостей.

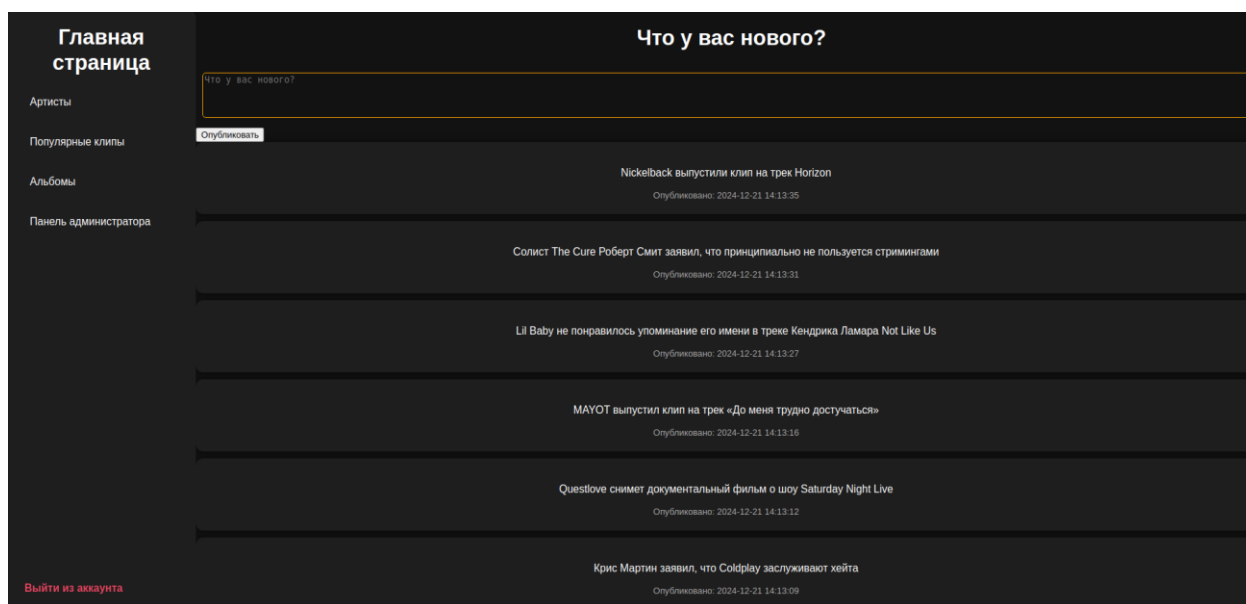


Рисунок 3.5 – Общий вид «Главной страницы»

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['news_text'])) {  
    $news_text = $_POST['news_text'];  
  
    $stmt = $pdo->prepare("INSERT INTO news (text, created_at) VALUES (:text, NOW())");  
    $stmt->execute(['text' => $news_text]);  
  
    header('Location: index.php');  
    exit();  
}
```

Рисунок 3.6 – Реализация публикации новостей

На рисунке 3.7 представлена реализация бургер-меню для перехода между страницами.

```
<div class="menu">
  <h1>Главная страница</h1>
  <a href="artist.php">Артисты</a>
  <a href="clips.php">Популярные клипы</a>
  <a href="album.php">Альбомы</a>
  <?php if (isAdmin()): ?>
    <a href="admin.php">Панель администратора</a>
  <?php else: ?>
    <a href="profile.php">Панель пользователя</a>
  <?php endif; ?>
  <div class="logout">
    <a href="logout.php">Выйти из аккаунта</a>
  </div>
</div>
```

Рисунок 3.7 – Реализация бургер-меню

3.4 Артисты

Файл `artist.php` отвечает за отображение страницы с артистами на веб-приложении и представляет пользователю информацию об артисте в удобном формате. Приложение включает проверку авторизации пользователя для предотвращения доступа неавторизованных лиц. На странице отображаются карточки с данными об артистах, включая их изображения, биографии и ссылки на дополнительные ресурсы. Основные функции файла включают в себя извлечение данных из базы данных с учетом пагинации и обеспечение удобной навигации по страницам. Этот файл является важной частью функционала сайта, предоставляя пользователям возможность легко находить и просматривать информацию об их любимых артистах. На рисунке 3.8 представлен общий вид страницы. Каждый артист — это отдельная карточка, реализация которой представлен на рисунке 3.9. Стили, навешанные на карточку представлены на рисунке 3.10.

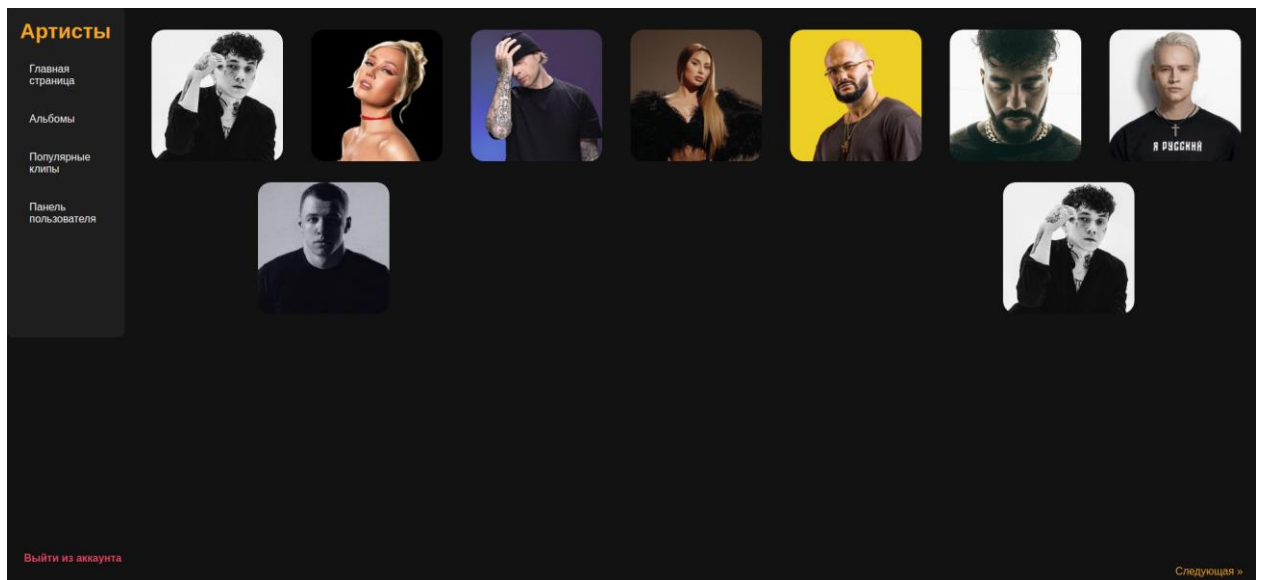


Рисунок 3.8 – Общий вид страницы «Артисты»

```
<div class="artists-block">
  <?php foreach ($artists as $artist): ?>
    <div class="card" style="background-image: url('<?php echo htmlspecialchars($artist['image_url']); ?>');">
      <a href="<?php echo htmlspecialchars($artist['genius_url']); ?>" target="_blank">
        <div class="card-body">
          <h3><?php echo htmlspecialchars($artist['name']); ?></h3>
          <p><?php echo htmlspecialchars($artist['bio']); ?></p>
        </div>
      </a>
    </div>
  <?php endforeach; ?>
</div>
```

Рисунок 3.9 – Реализация отображения карточек с артистами

```
.artists-block, .cards-block {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  padding: 20px;
}

.cards-block {
  background-color: #151515;
  border-radius: 50px;
  padding: 20px;
}

.cards-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  padding-top: 30px;
}
```

Рисунок 3.10 – Стили карточек артистов

При наведении на карточку артиста появляется имя артиста и некоторое описание, пример представлен на рисунке 3.11.

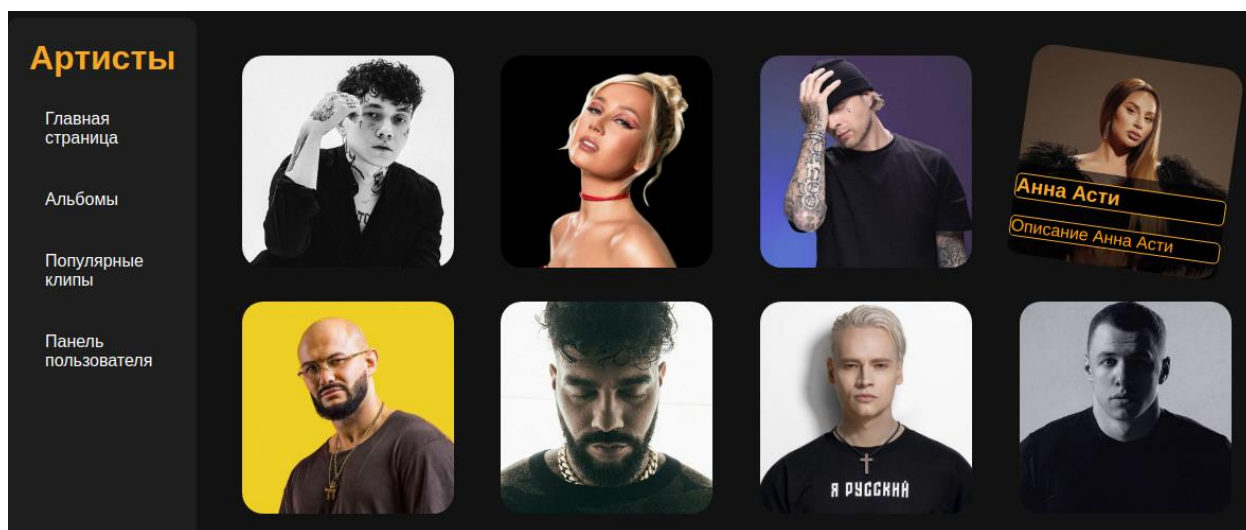


Рисунок 3.11 – Описание карточки артиста

Пагинация представляет собой процесс разделение большого количества данных на несколько страниц, что позволяет пользователю легко и удобно просматривать информация, такой функционал используется на каждой страницы «Музыкального портала», пример представлен на рисунке 3.12 в нижнем правом углу.

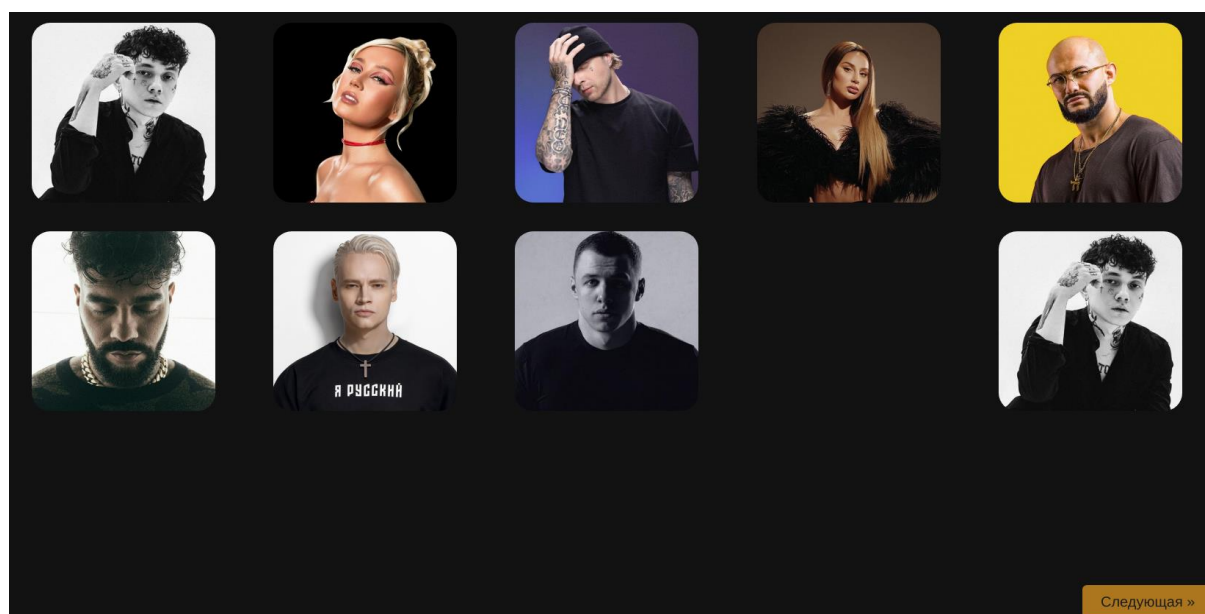


Рисунок 3.12 – Пагинация

3.5 Популярные клипы

Файл clips.php отвечает за отображение страницы с популярными клипами в вашем веб-приложении. Он включает в себя проверку авторизации пользователя для предотвращения доступа неавторизованных лиц. Основной задачей файла является извлечение данных о клипах из базы данных и отображение их на странице с использованием миниатюр YouTube. Для удобной навигации реализована пагинация, что позволяет пользователям просматривать клипы, распределенные по страницам.

Важным аспектом является создание удобного и интуитивно понятного интерфейса, который обеспечивает пользователям легкий доступ к их любимым клипам и возможность просмотра подробной информации о каждом клипе. При нажатии на карточку с клипом, запускается клип. На рисунке 3.13 представлен общий вид страницы «популярные клипы»

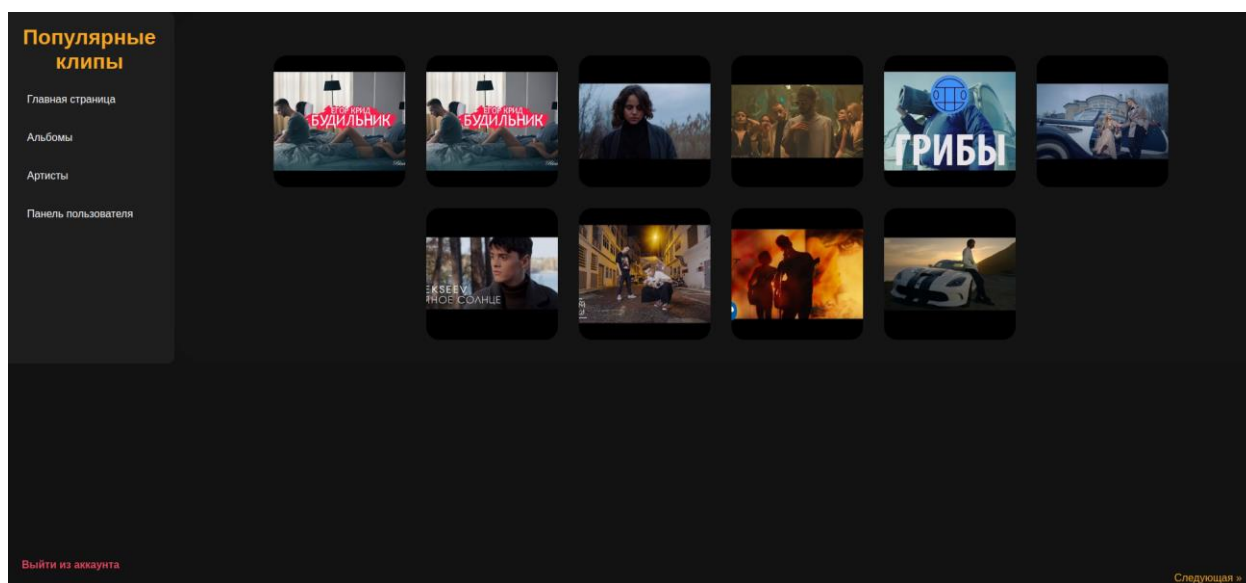


Рисунок 3.13 – Общий вид страницы «Популярные клипы»

3.6 Альбомы

Файл album.php отвечает за отображение страницы с альбомами в веб-приложении. Он включает в себя проверку авторизации пользователя для предотвращения доступа неавторизованных лиц. Основной задачей файла является извлечение данных об альбомах из базы данных и отображение их на странице в виде карточек. Для удобной навигации реализована пагинация, что позволяет пользователям просматривать альбомы, распределенные по страницам. Важным аспектом является создание удобного и интуитивно понятного интерфейса, который обеспечивает пользователям легкий доступ к информации об их любимых альбомах и возможность просмотра подробных данных о каждом из них. На рисунке 3.14 представлена страница «Альбомы». При нажатии на альбом, перенаправляет на сайт, который хранится в БД вместе с карточкой альбома. Обложка альбома подгружается локально, в БД храниться путь до файла. Перенаправления и загрузка обложки реализовано на рисунке 3.15.

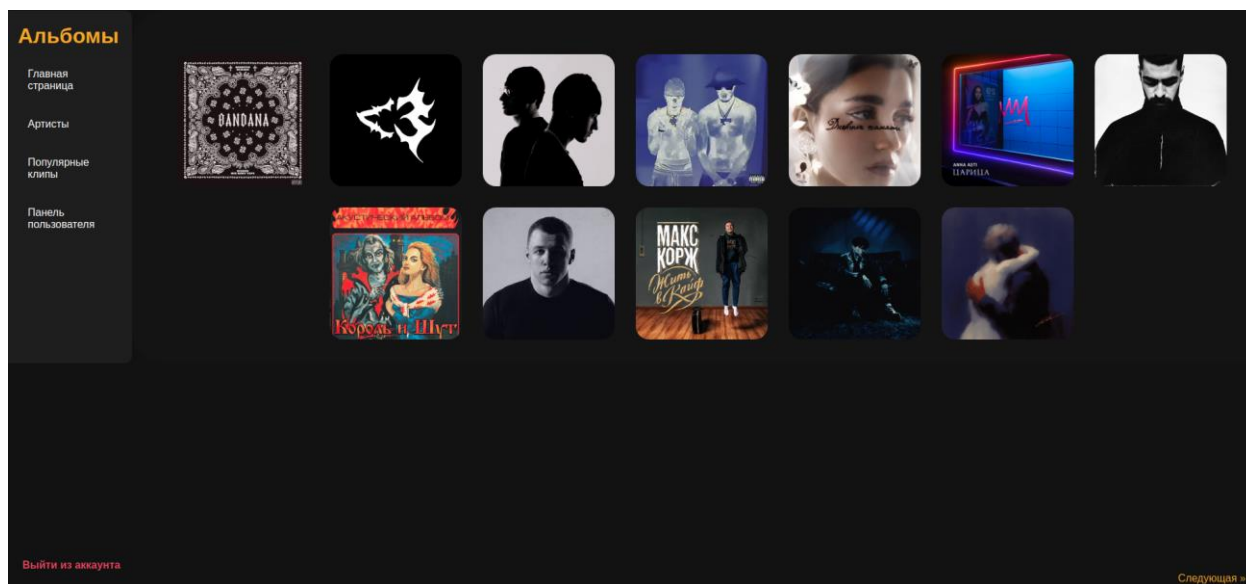


Рисунок 3.14 - Общий вид страницы «Альбомы»

```

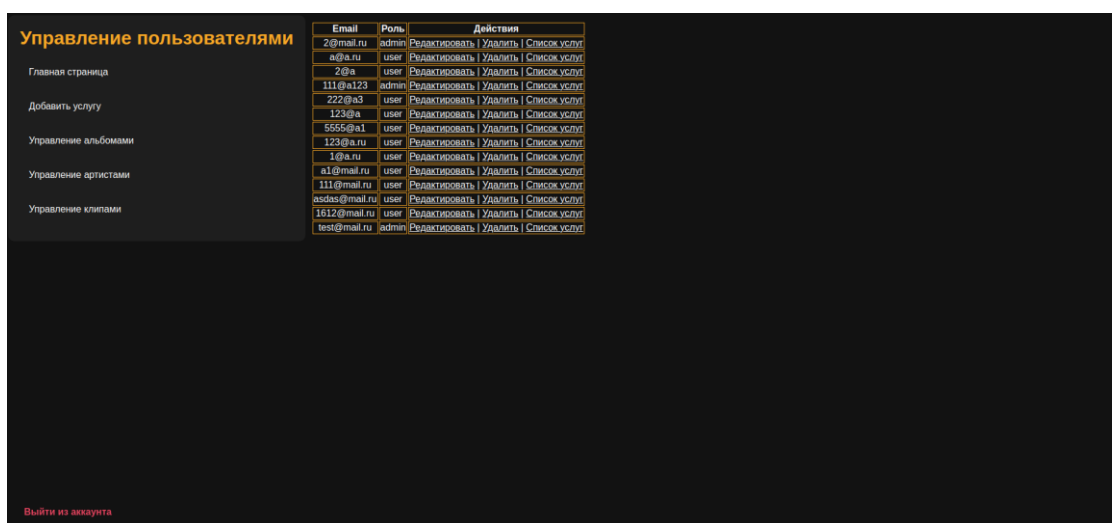
<div class="cards-block">
  <div class="cards-container">
    <?php foreach ($albums as $album): ?>
      <div class="card" style="background-image: url('<?php echo htmlspecialchars($album['cover_url']); ?>');
        onclick="location.href='<?php echo htmlspecialchars($album['genius_url']); ?>';">
        <div class="card-body"></div>
      </div>
    <?php endforeach; ?>
  </div>
</div>

```

Рисунок 3.15 – Реализация загрузки обложки и перенаправление на сайт с альбомом

3.7 Панель администратора

Файл admin.php отвечает за отображение административной панели в веб-приложении. Эта страница предоставляет администраторам доступ к управлению пользователями, а также другими сущностями, такими как услуги, альбомы, артисты и клипы. Важным аспектом является обеспечение безопасности и доступ к этой странице только для авторизованных администраторов. Основной задачей файла является извлечение данных о пользователях из базы данных и отображение их на странице в виде таблицы. Интерфейс позволяет администраторам редактировать и удалять пользователей, а также управлять связанной информацией. Это обеспечивает эффективное и безопасное управление контентом и пользователями приложения. На рисунке 3.16 общий вид панели администратора.



Рисунке 3.16 – Панель администратора

На рисунке 3.17 предоставлена реализация таблицы с пользователями, а на рисунке 3.18 стили, использованные для этой таблицы.

```
<table class="table">
  <thead>
    <tr>
      <th>Email</th>
      <th>Роль</th>
      <th>Действия</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($users as $user): ?>
      <tr>
        <td><?php echo htmlspecialchars($user['email']); ?></td>
        <td><?php echo htmlspecialchars($user['role']); ?></td>
        <td>
          <a href="edit_profile.php?id=<?php echo $user['id']; ?>">Редактировать</a> |
          <a href="delete_user.php?id=<?php echo $user['id']; ?>" onclick="return confirm('Вы уверены?')">Удалить</a>
          <a href="list_services_by_users_id.php?id=<?php echo $user['id']; ?>">Список услуг</a>
        </td>
      </tr>
    <?php endforeach; ?>
  </tbody>
</table>
```

Рисунок 3.17 – Реализация таблицы с пользователями

```
.table {
  margin: 10px;
}

.table td, th, a {
  color: white;
}

.table td, th {
  text-align: center;
  border: 1px solid #f9a826;
}
```

Рисунок 3.18 – Стили для таблицы с пользователями

При переходе на страницу «Управление альбомами» попадает на панель управления, которая позволяет нам удалять/редактировать/добавлять информацию про альбомы. Также мы можем увидеть список текущих альбомов, общий вид страницы представлен на рисунке 3.19

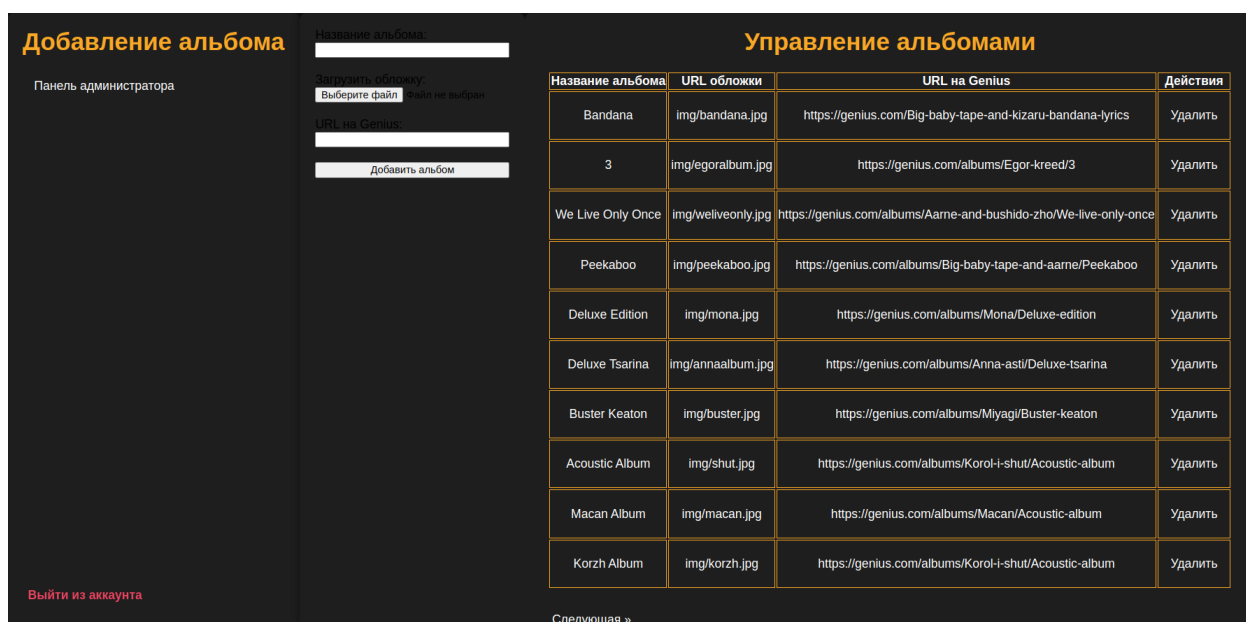


Рисунок 3.19 – Общий вид страницы «Управления альбомами»

При переходе на страницу «Управление артистами» попадает на панель управления, которая позволяет нам удалять/редактировать/добавлять информацию про артистов. Также мы можем увидеть список текущих артистов, общий вид страницы представлен на рисунке 3.20

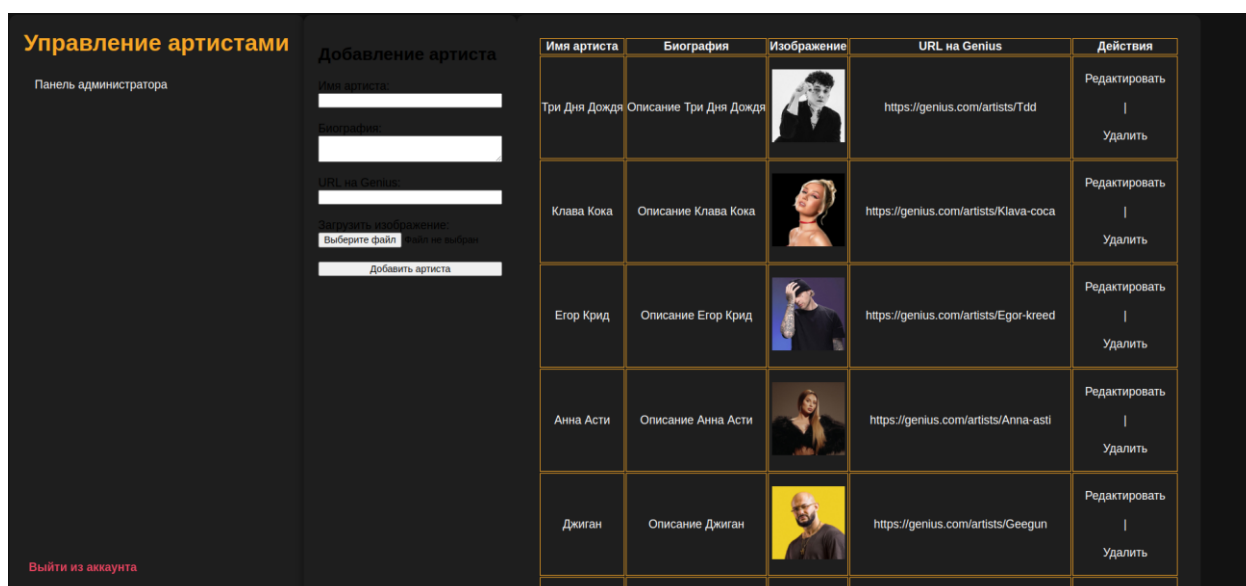


Рисунок 3.20 – Общий вид страницы «Управление артистами»

При переходе на страницу «Управление клипами» попадает на панель управления, которая позволяет нам удалять/редактировать/добавлять

информацию про клипы. Также мы можем увидеть список текущих клипов, общий вид страницы представлен на рисунке 3.21

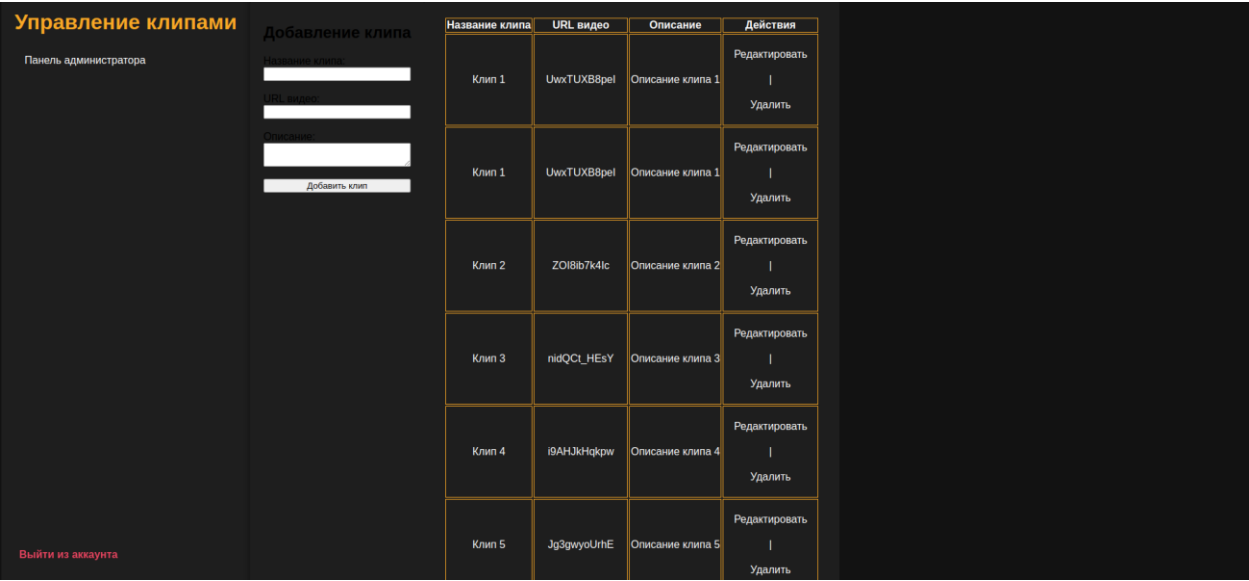


Рисунок 3.21 – Общий вид страница «Управление клипами»

3.8 Панель пользователя

Файл `profile.php` отвечает за отображение страницы профиля пользователя в веб-приложении. Он включает в себя проверку авторизации пользователя для предотвращения доступа неавторизованных лиц. Основной задачей файла является извлечение данных о текущем пользователе из базы данных и отображение этих данных на странице. Пользователи могут видеть свою учетную информацию, такую как email и роль, а также список связанных с ними услуг. Важным аспектом является создание удобного и интуитивно понятного интерфейса, который обеспечивает пользователям легкий доступ к их личной информации и возможность управления своими данными и услугами. На рисунке 3.22 представлен общий вид страницы профиля пользователя. Также, можем заметить возможность редактировать профиль, в нем есть функционал по смене «email», который представлен на рисунке 3.23.

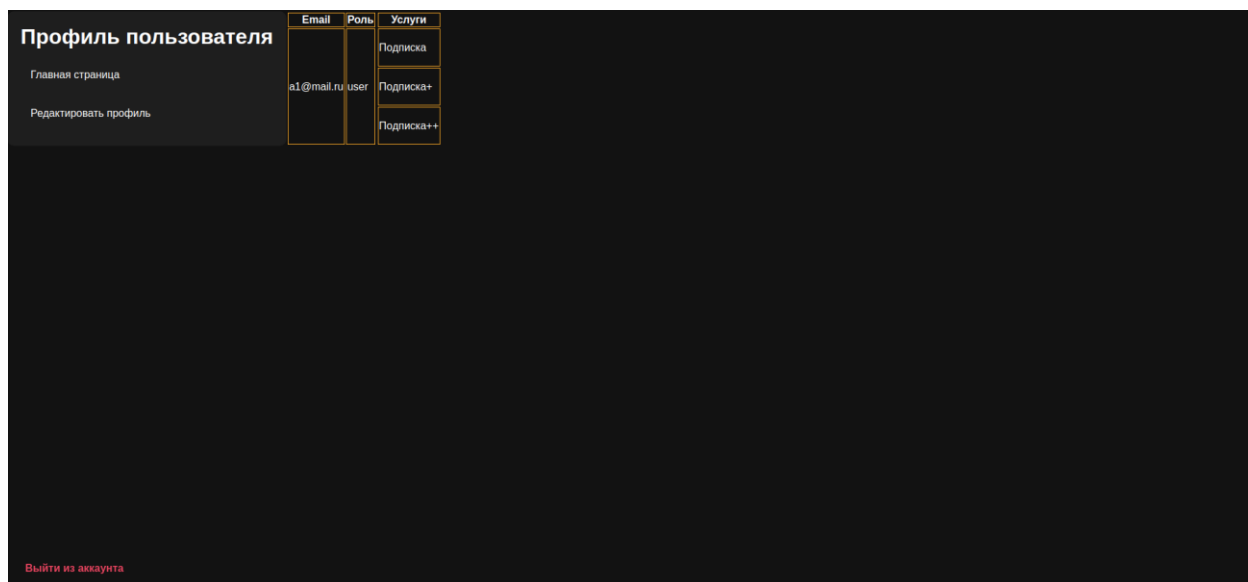


Рисунок 3.22 – Панель пользователя

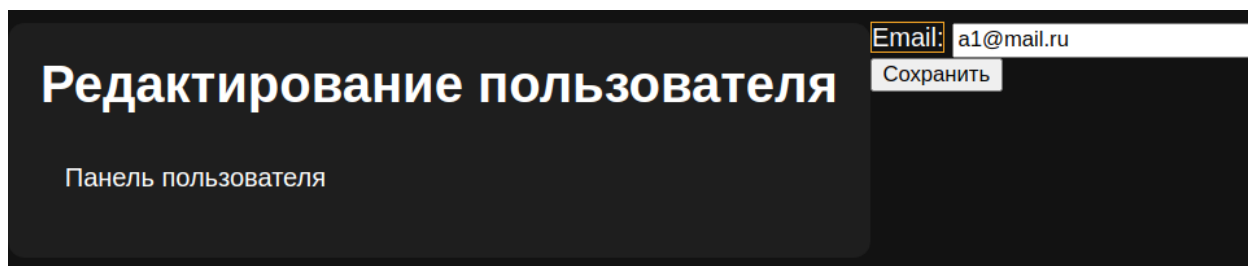


Рисунок 3.23 – Редактирование пользователя

ЗАКЛЮЧЕНИЕ

В ходе данного проекта мы реализовали веб-приложение, которое предоставляет пользователям и администраторам удобный и интуитивно понятный интерфейс для взаимодействия с музыкальным контентом. Веб-приложение включает такие ключевые разделы, как главная страница, страницы с артистами, популярными клипами и альбомами, а также административную панель для управления пользователями и контентом.

Основные этапы реализации включали проектирование структуры базы данных, разработку серверной логики на PHP, создание интерфейсов с использованием HTML, CSS, а также обеспечение безопасности данных и пользовательских сессий.

Приложение позволяет пользователям:

- Регистрироваться и авторизоваться на сайте;
- Просматривать и публиковать новости;
- Иметь доступ к персональному профилю с информацией об аккаунте и связанных услугах;
- Просматривать карточки артистов, клипов и альбомов с возможностью навигации по страницам.

Для администраторов предусмотрена возможность управления пользователями и контентом через административную панель, что включает добавление, редактирование и удаление данных.

Проект успешно достиг своих целей, предоставив пользователям и администраторам эффективные инструменты для взаимодействия с музыкальным контентом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Mozilla Developer Network (MDN). HTML: сайт. – URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата обращения: 19.12.2024).
2. Mozilla Developer Network (MDN). CSS: сайт. – URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата обращения: 19.12.2024).
3. PHP Documentation: сайт. – URL: <https://www.php.net/docs.php> (дата обращения: 19.12.2024).
4. MySQL Reference Manual: сайт. – URL: <https://dev.mysql.com/doc/> (дата обращения: 19.12.2024).
5. dbdiagram.io: сайт. – URL: <https://dbdiagram.io/d> (дата обращения: 19.12.2024).
6. Реляционные базы данных: сайт. – URL: <https://yandex.cloud/ru/docs/glossary/relational-databases> (дата обращения: 19.12.2024).
7. PHP и MySQL: сайт. – URL: <https://metanit.com/php/mysql/3.1.php> (дата обращения: 19.12.2024).
8. Работа с MySQL в PHP: сайт. – URL: <https://htmlacademy.ru/blog/php/mysql> (дата обращения: 19.12.2024).