Пояснительная записка

к проекту

«Blackland Radio 2»

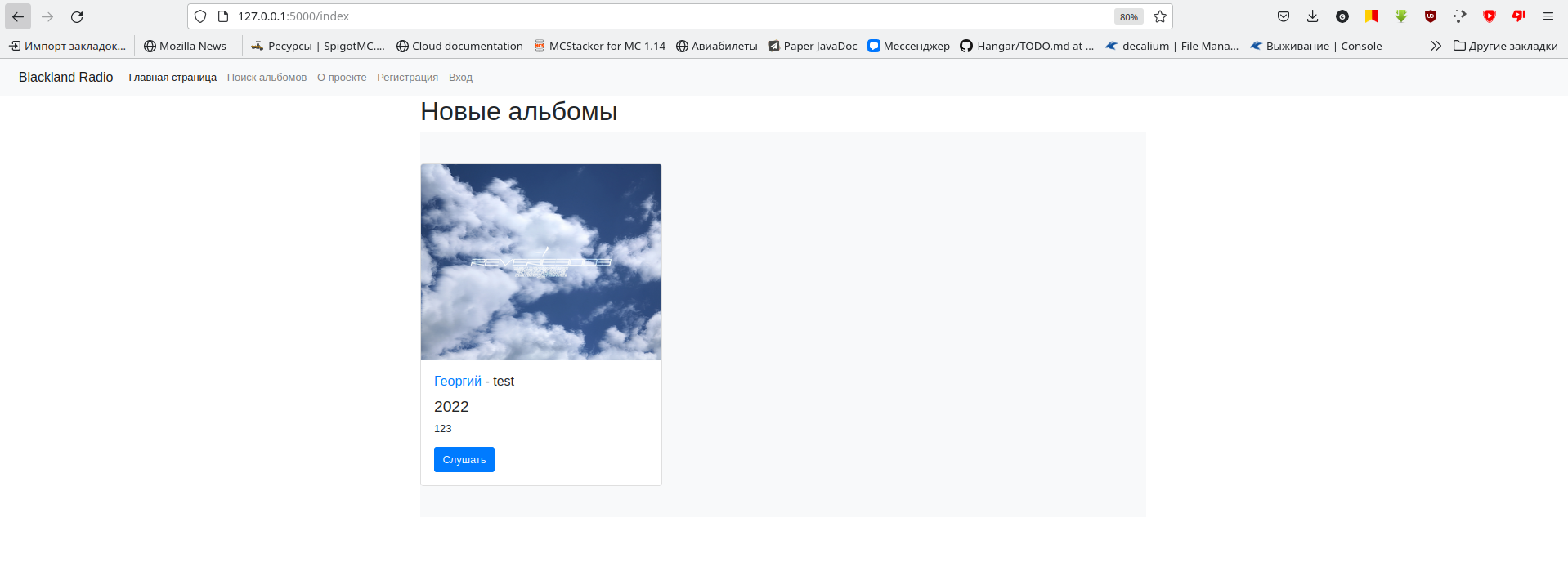
Автор: Пронюк Георгий

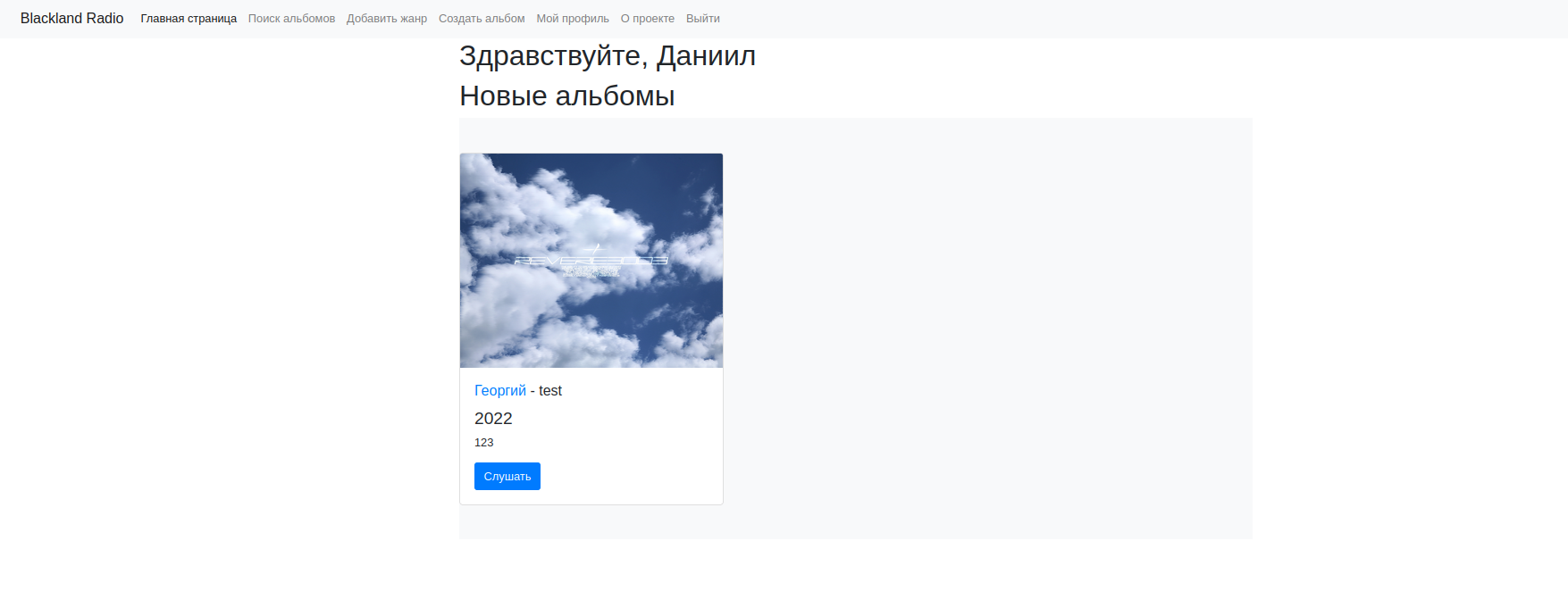
**Идея проекта**

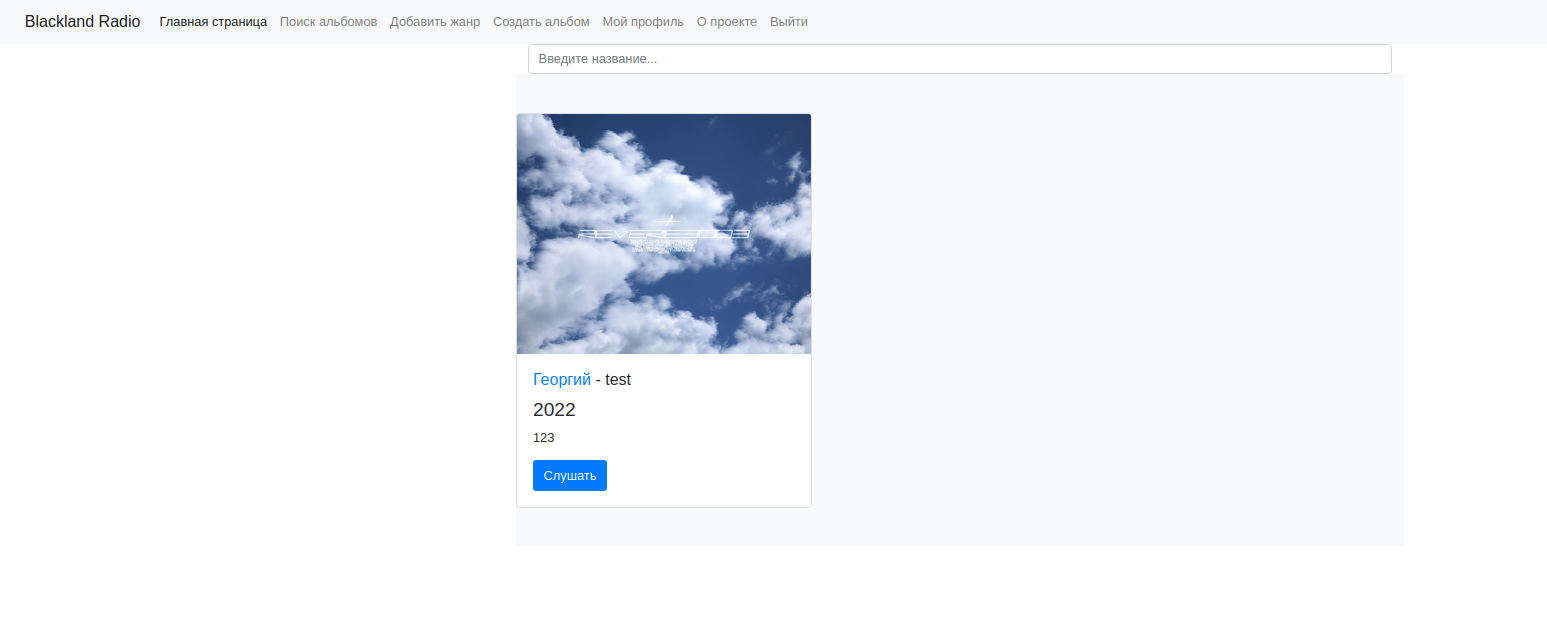
Blackland Radio 2 — музыкальный сервис, где пользователи свободно могут делиться некоммерческой музыкой.

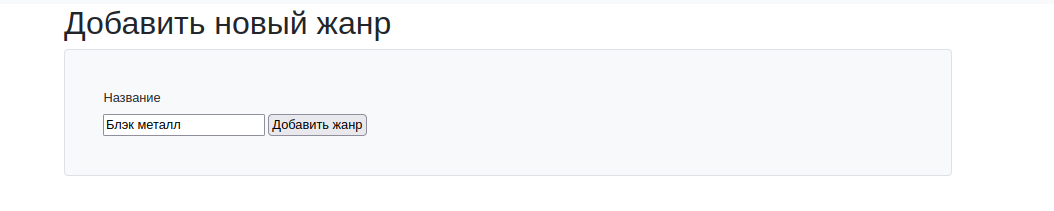
**Функционал**

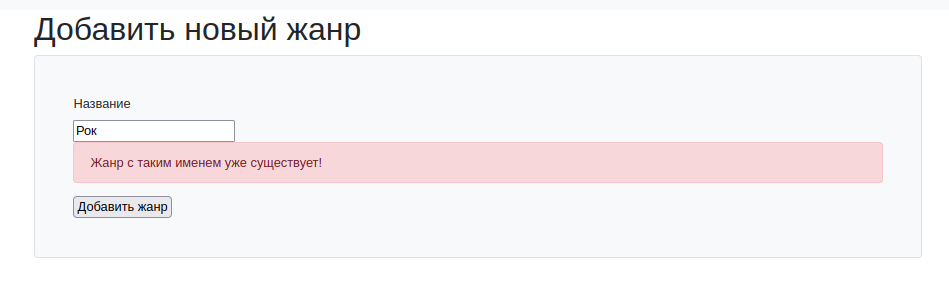
При первом входе на сайт нас встречает главная страница:

Здесь можно просмотреть новые альбомы. Если пользователь авторизан, отобразится приветствие:

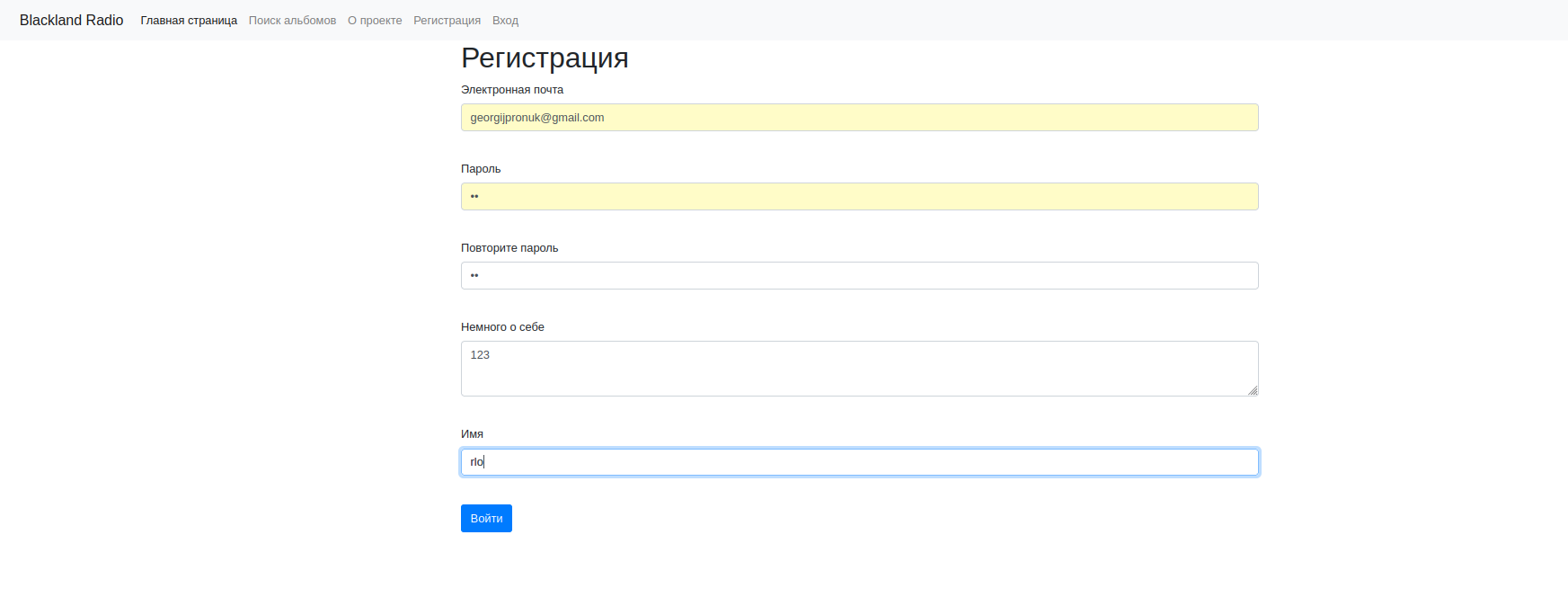
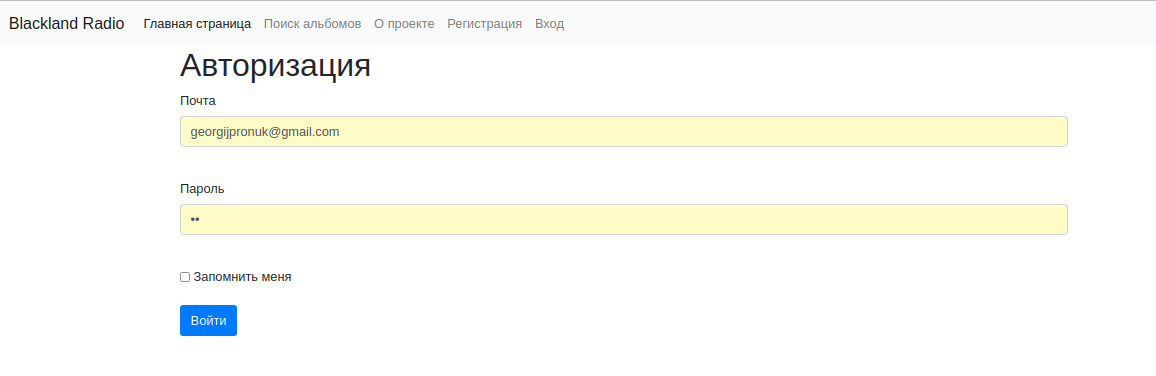
На сайте есть поиск альбомов. Можно найти альбомы по ключевым словам. Название полностью писать необязательно, поиск выведет все возможные результаты.

Есть возможность добавить свой жанр. Если он уже существует, отобразится соответсвующая ошибка.



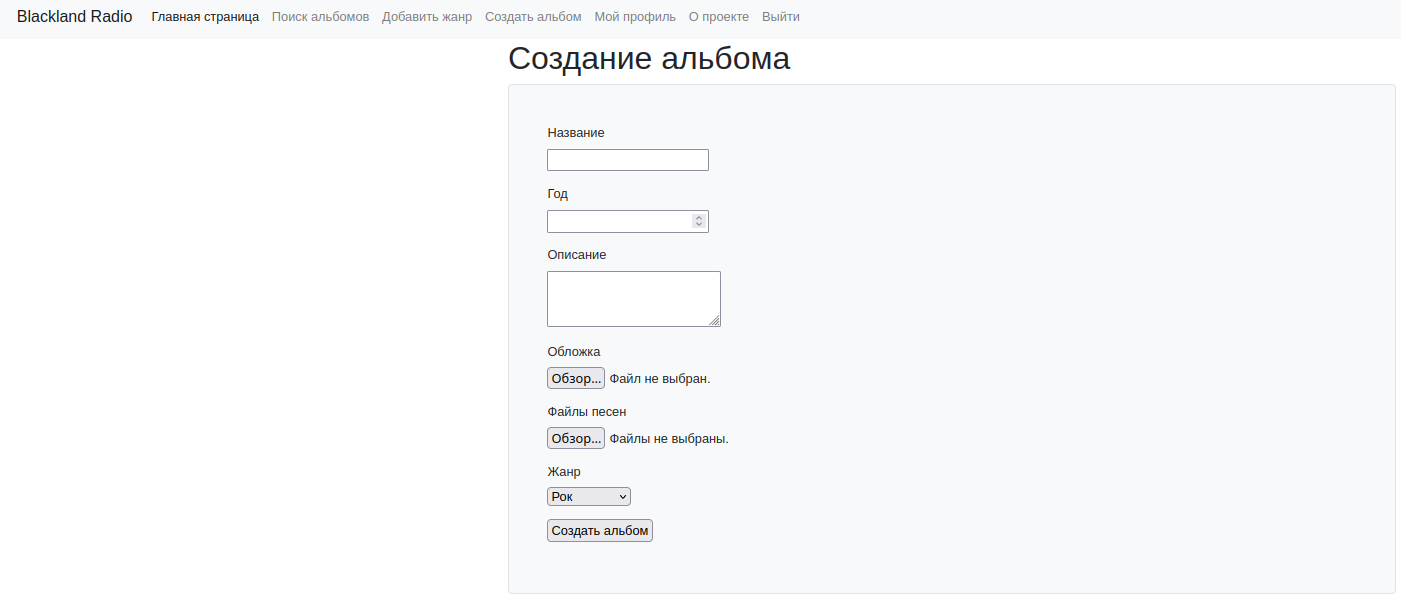


Для дальнейшего взаимодействия стоит завести аккаунт. Можно зарегистрироваться или войти в уже существующий.

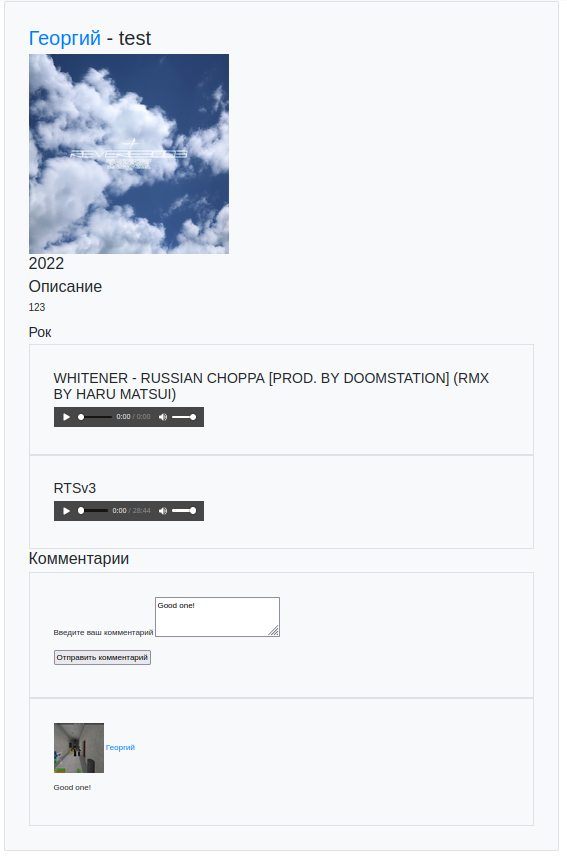
Если логин или пароль неправильный, отобразится соответсвующее сообщение.

Если аккаунт с такой почтой зарегистрирован, будет также выведено сообщение.

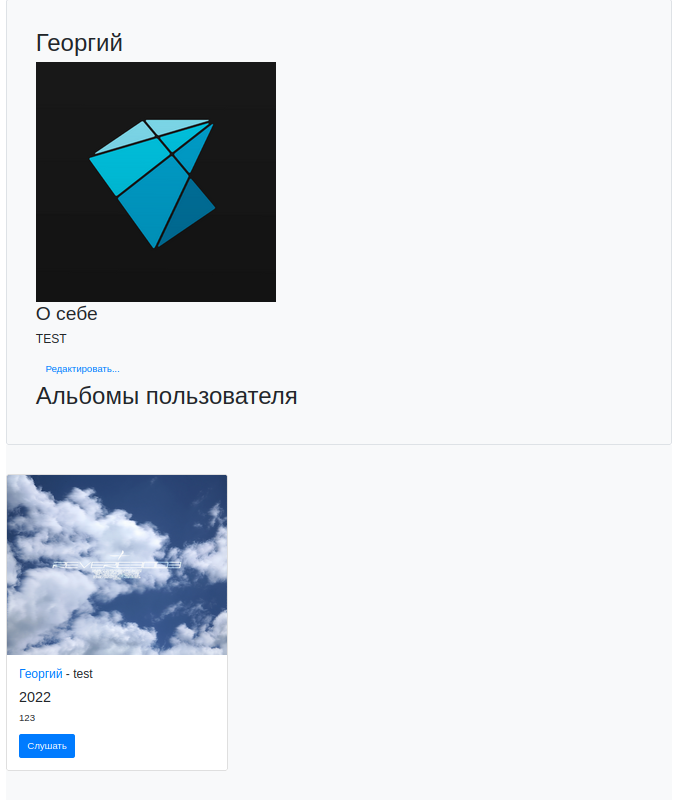
Создание альбома происходит на специальной странице.

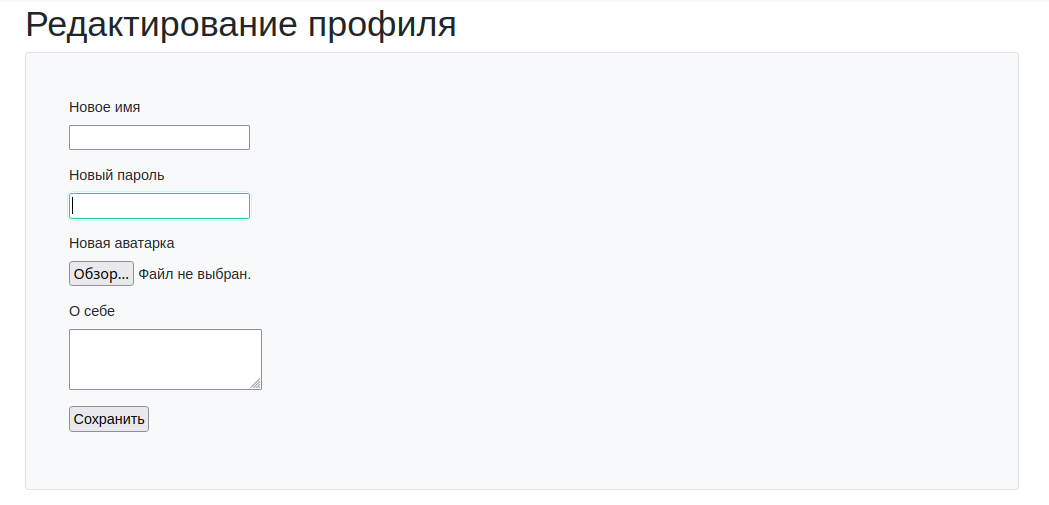
Необходимо указать название, описание, год, добавить обложку и файлы песен. Если год будет больше текущего, высветится ошибка.

После создания альбома можно зайти на его страницу, послушать песни и оставить комментарий.



Зайдя на страницу пользователя, можно просмотреть добавленные им альбому и другую информацию. Если профиль ваш, появится ссылка на редактор.





Так выглядит редактор профиля. Пустые поля останутся неизменными.

**RESTful API**

С вебсайтом можно взаимодействовать через API.

* *api/v2/albums/<int:album\_id> - получение альбома по id. Возращает информацию в json формате. Метод delete удаляет альбом с заданным id, для его использование необходимо указать логин и пароль в виде позиционных аргументов.* requests.delete("https://127.0.0.1:5000/api/v2/albums/1", params={'login': 'test@gmail.com', 'password': '1234'})
* *api/v2/albums/ - возвращает все существующие альбомы.*
* *api/v2/albums/search?value=s — поиск альбомов по названию.*
* *api/v2/users/<int:user\_id> - получение информации о пользователе по id.*
* *api/v2/users/search?value=s — поиск пользователей по имени.*

**Использованные технологии**

* **Flask - фреймворк для создания веб-приложений на языке программирования Python.**
* Flask-WTF – расширение для Flask, с помощью которого можно с легкостью создавать формы.
* Flask-Login — расширение для Flask, позволяющее легко работать с сессиями пользователей.
* SQLAlchemy — библиотекадля работы с реляционными СУБД с применением технологии ORM.
* Flask-RESTful — расширения для flask, позволяющая создавать RESTful-сервисы.

**Структура кода**

Весь код поделен на 4 модуля:

* api — RESTful api проекта.
* data — реляционные модели базы данных.
* form — формы.
* pages — страница и логика их обработки.

Api

* albums.py — апи для работы с альбомами.
* users.py — апи для работы с пользователями.

Data

* album.py — модель альбома.
* comment.py — модель комментария.
* song.py — модель песен.
* user.py — модель пользователя.
* db\_session.py — работа с сессиями sqlalchemy.

Form

* album\_creation\_form.py — форма создания альбома.
* albums\_search.py — поиск альбомов.
* auth.py — форма авторизации.
* comment\_form.py — форма комментария.
* genre\_form.py — форма добавления жанра.
* user\_form.py — редактор профиля.

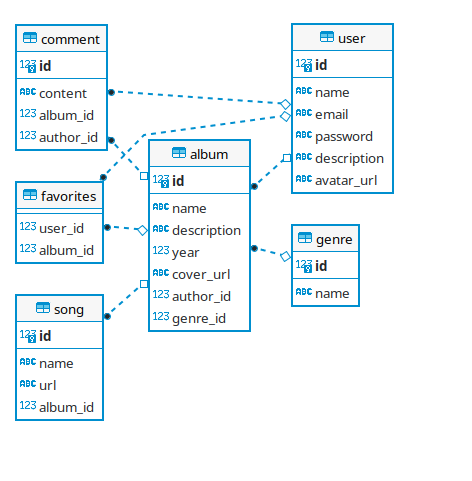
Pages

* album\_creation\_page.py — страница создания альбома.
* album\_page.py — страница альбома.
* profile\_editor\_page.py — страница редактора профиля.
* page.py — базовый класс всех страниц.

Main.py — точка входа и логика обработки базовых страниц.

В папке templates находятся шаблоны.

**Структура БД**



**Планы на будущее**

* Индивидуальные рекомендации слушателям.
* Больше способов взаимодействия для пользователей: подписки, блокировки.
* Глобальный поиск пользователей, альбомов и отдельных песен по фильтрам.