

О нас

Цель:

Создание веб-платформы-агрегатора для совместного управления музыкальными плейлистами с возможностью кросс-платформенной синхронизации (Yandex Music, Spotify, VK Music).

## 2. Функциональные требования (MVP)

### 2.1. Пользователь и Авторизация

- Регистрация (Email/Password) и вход в систему.
- Управление профилем: привязка аккаунтов стриминговых сервисов  
*Поддерживаемые сервисы (пока не определились до конца не определились):*  
Yandex Music, Spotify, VK Music.

### 2.2. Группы и Плейлисты

- Создание "Комнаты" (Группы) владельцем.
- Генерация уникальной ссылки-приглашения для вступления в группу.
- Создание общего плейлиста внутри группы (CRUD операции).

### 2.3. Работа с музыкой

- **Универсальный поиск:** Поиск трека по базе агрегатора (мета-поиск сразу по API подключенных сервисов).
- **Добавление трека:** Пользователь добавляет трек в общий плейлист (в БД сохраняется мета-информация: Исполнитель, Название, Длительность).
- **Экспорт:** Возможность выгрузить собранный плейлист в свой личный аккаунт подключенного сервиса (с автоматическим поиском соответствий треков).

## 3. Технологический стек

### Backend

- **Язык:** Python 3.11+
- **Фреймворк:** FastAPI (Асинхронный).
- **База данных:** PostgreSQL (Драйвер `asyncpg`).
- **ORM:** SQLAlchemy 2.0 (Async) + Alembic (Миграции).
- **Auth:** `fastapi-users` (JWT strategy).
- **Integrations:**
  - `yandex_music` (Unofficial API wrapper).
  - `spotipy` (Official Spotify Web API).
  - `vkpymusic` (Unofficial API wrapper).

- **Data Validation:** Pydantic v2.
- **Background Tasks (Optional for MVP):** Redis + Arq/Celery (для длительных операций экспорта).

## Frontend

### Основной стек:

- **React** для построения компонентного пользовательского интерфейса.
- **TypeScript** для строгая типизация кода.
- **Vite** для сборки проекта и сервер для локальной разработки.

### Инструменты и библиотеки:

- **TanStack Query (React Query)** управление серверным состоянием, кэширование и синхронизация с API.
- **Mantine UI** библиотека готовых компонентов для быстрой верстки интерфейса.
- **React Router v6** маршрутизация для навигации между страницами без перезагрузки.
- **React Hook Form** управление формами и валидация данных ввода пользователя.

## DevOps / Infrastructure

- Docker & Docker Compose (для локальной разработки и деплоя).
- Nginx (Reverse Proxy).