Music Collaboration Platform Documentation

Project Overview

A real-time music collaboration platform enabling musicians to create, share, and collaborate on music projects. Features include live audio recording, project sharing, and version control.

Tech Stack

- **Frontend**: JSP, HTML, CSS
- **Backend**: Java Servlet
- **Database**: My SQL
- **Authentication**: JWT

Core Features

1. **User Management**
   - Authentication using JWT
   - User profiles with musical preferences
   - Portfolio management
   - Collaboration history
2. **Project Management**
   - Create/edit music projects
   - Version control for audio files
   - Project sharing and access control
   - Real-time collaboration sessions
3. **Audio Features**
   - Live recording capabilities
   - Multi-track mixing
   - Audio effects processing
   - Export in multiple formats

Installation

Clone the Repository:
```
git clone https://github.com/lakshay-63/music-collab.git
cd music-collab
```

Database Setup:

- Ensure you have a MySQL database instance running.
- Execute the `Create_table.sql` script to create the required tables:
```
mysql -u yourUsername -p yourDatabaseName < Create_table.sql
```

Build the Project:

- The project uses Maven for dependency management.

  Build the project using the following command:
  ```
  mvn clean install
  ```

Run the Application:

- After a successful build, deploy the application on your preferred server.
- Access the platform via `http://localhost:8080/` (or the configured server address).

API Endpoints

Authentication

POST /api/auth/register
POST /api/auth/login
GET /api/auth/profile

Projects

GET /api/projects
POST /api/projects
PUT /api/projects/:id
DELETE /api/projects/:id

Collaboration

POST /api/projects/:id/collaborate
GET /api/projects/:id/sessions
POST /api/projects/:id/tracks

Database Schema

User Collection

```
{
  username: String,
  email: String,
  password: String,
  profile: {
    instruments: [String],
    genres: [String],
    portfolio: [String]
  }
}
```

Project Collection

```
{
  title: String,
  creator: ObjectId,
  collaborators: [ObjectId],
  tracks: [{
    name: String,
    audioUrl: String,
    version: Number
  }],
  settings: {
    privacy: String,
    allowedOperations: [String]
  }
}
```

Architecture

Frontend Components

- Authentication views
- Project dashboard
- Audio recording interface

- Collaboration workspace
- Mix console

Backend Services

- User service
- Project management service
- Audio processing service
- Collaboration service
- File storage service

Security Measures

- JWT authentication
- Request validation
- Rate limiting
- File upload restrictions
- CORS configuration

Error Handling

```
{
  status: number,
  message: string,
  details: object
}
```

Future Enhancements

1. Real-time chat
2. Advanced audio effects
3. Mobile application
4. AI-powered music suggestions
5. Public project marketplace