Riff AI Documentation

1. Introduction

Riff AI is an AI-powered music companion that analyzes user text and audio inputs to recommend songs that match the user's current emotion and mood. It leverages sentiment and emotion analysis, integrates with external music databases (e.g., Last.fm), and provides a personalized listening experience.

---

2. System Architecture

```
Frontend (React Native + Expo)
       |
Backend 1 (Django REST Framework) → Auth, User data, Song history
       |
Backend 2 (FastAPI) → Audio transcription, Sentiment + Emotion Analysis
       |
External APIs: Last.fm, GeoIP
```

---

3. Features

3.1 Audio & Text Input

- Users can speak or type messages.
- Input is transcribed if audio.
- All inputs are analyzed for sentiment and emotion.

3.2 Emotion & Sentiment Analysis

- Uses fine-tuned transformer models (e.g., GoEmotions, Sentiment140).
- Maps emotions (joy, sadness, anger, etc.) and polarity (positive/negative/neutral).

3.3 Music Recommendation

- Extracts mood/emotion/keywords → queries Last.fm.
- Retrieves and ranks song recommendations.
- Plays songs via in-app player.

3.4 Personalized Radio Jockey

- Auto-generated voice introduces songs.
- Transitions between songs using context-aware comments.

3.5 User Listening History

- Tracks emotion over time.
- Builds a personalized emotional music graph.

---

4. Tech Stack

Layer	Tech Stack
Frontend	React Native + Expo
Backend 1	Django REST Framework
Backend 2	FastAPI, Transformers, Torch
Models	`cardiffnlp/twitter-roberta-base-sentiment`, GoEmotions
Audio	Whisper / SpeechRecognition

Layer	Tech Stack
API	Last.fm, GeoIP
Deployment	GCP VM, Gunicorn + Nginx + HTTPS

---

5. Key API Endpoints

FastAPI Endpoints

- `POST /user_input/analyze_text`
  → Analyzes user text input, returns emotion, mood, and keywords.
- `POST /user_input/analyze_audio`
  → Accepts audio, returns transcribed text and analysis.
- `POST /get_song_recommendation`
  → Takes mood, emotion, keywords → returns top 5 songs.

Django Endpoints

- `POST /auth/login/`
- `GET /user/history/`
- `POST /user/save_recommendation/`

---

6. Data Flow Example

1. User sends audio → `/analyze_audio`
2. Text is transcribed and analyzed
3. Analysis output → `/get_song_recommendation`
4. Top 5 songs are fetched from Last.fm
5. Playlist + transitions by Riff AI DJ is returned

---

7. Future Enhancements

- Replace Last.fm with a hybrid vector + keyword retriever.
- Local LLM integration for full offline RAG.
- Adaptive DJ personality using voice cloning.
- Emotion tracking dashboard for each user.

---

8. Contributors

- **Siddharth** – UI/UX,Mobile Dev, NLP & Audio Integration
- **Sheron** – UI/UX,Mobile Dev, NLP & Audio Integration
- **Vishnu** – Frontend Components
- **Wilson** – Frontend Components