1. Introduction

Project Title: AI DJ: Music Mashup Generator

Problem Statement:
Creating mashups is a complex process that requires advanced audio editing skills, professional software, and significant manual effort. Most casual music lovers, however, lack access to such tools and expertise. As a result, remixing and mashup creation remain limited to DJs and audio engineers, leaving a large audience unable to participate in creative music mixing.

Proposed Solution:
We propose building a web-based platform where users can pick any two songs, and the system automatically generates a mashup along with custom album art. Using AI, the app will align beats and tempos for seamless mixing. Additionally, GAN-based models will generate unique artwork for each mashup, giving users a complete, creative experience.

Approach:
The platform will be developed using **Next.js (frontend)**, **Flask with Librosa (backend audio processing)**, and **Firebase (database)**. Machine Learning will handle beat/tempo matching and album art generation. The portal will be simple, interactive, and engaging, with social media sharing features to boost adoption.

---

2. Project Objectives

Goals:

- Develop a centralized platform for users to create music mashups effortlessly.
- Use AI and signal processing for automatic beat/tempo alignment.
- Generate custom album art with GANs.
- Enable social media sharing for user engagement and virality.

Expected Benefits:

- Makes mashup creation accessible to non-technical users.
- Encourages creativity and music experimentation.
- Provides a fun and engaging way for people to share music.
- Opens possibilities for DJs/content creators to create remixes faster.

3. Feasibility Analysis

3.1 Technical Feasibility

- **Technology Stack:**
    - Frontend: Next.js
    - Backend: Flask (Python) with Librosa for audio analysis
    - Machine Learning:
        - Beat/tempo matching (signal processing)
        - GANs for custom album art
    - Database: Firebase (to store mashups)
    - Hosting: Vercel (frontend) + Flask API server (backend)
- **Development Skills:**
  The team has basic knowledge of frontend and backend development. Audio processing and GAN implementation will be learned during development using existing libraries and open-source resources.
- **Infrastructure:**
  Cloud hosting (Firebase, Vercel, Heroku/Render) for scalability and accessibility.

3.2 Operational Feasibility

- **Users:** Music enthusiasts, casual users, DJs, and content creators.
- **Interaction:**
    - Users pick 2 songs → System generates mashup + album art → Option to save/share.
    - The database stores user creations for playback and re-use.
- **User Adoption:** The platform will be fun, intuitive, and social, requiring no technical knowledge.

3.3 Legal & Ethical Feasibility

- **Copyright Concerns:** The platform will initially allow mashups only for personal/non-commercial use to avoid legal issues.
- **Data Privacy:** Only essential user details will be stored (username, mashup preferences).
- **Ethics:** Clearly disclaim commercial redistribution of copyrighted songs.

---

4. Risk Assessment

Potential Risks:

- Technical Challenges: Implementing audio synchronization and GANs for album art.
- Legal Risks: Copyright issues with using popular songs.
- Adoption Risk: Users may see it as a novelty rather than a serious tool.
- Performance: Audio processing may cause delays in real-time mashup generation.

Mitigation Strategies:

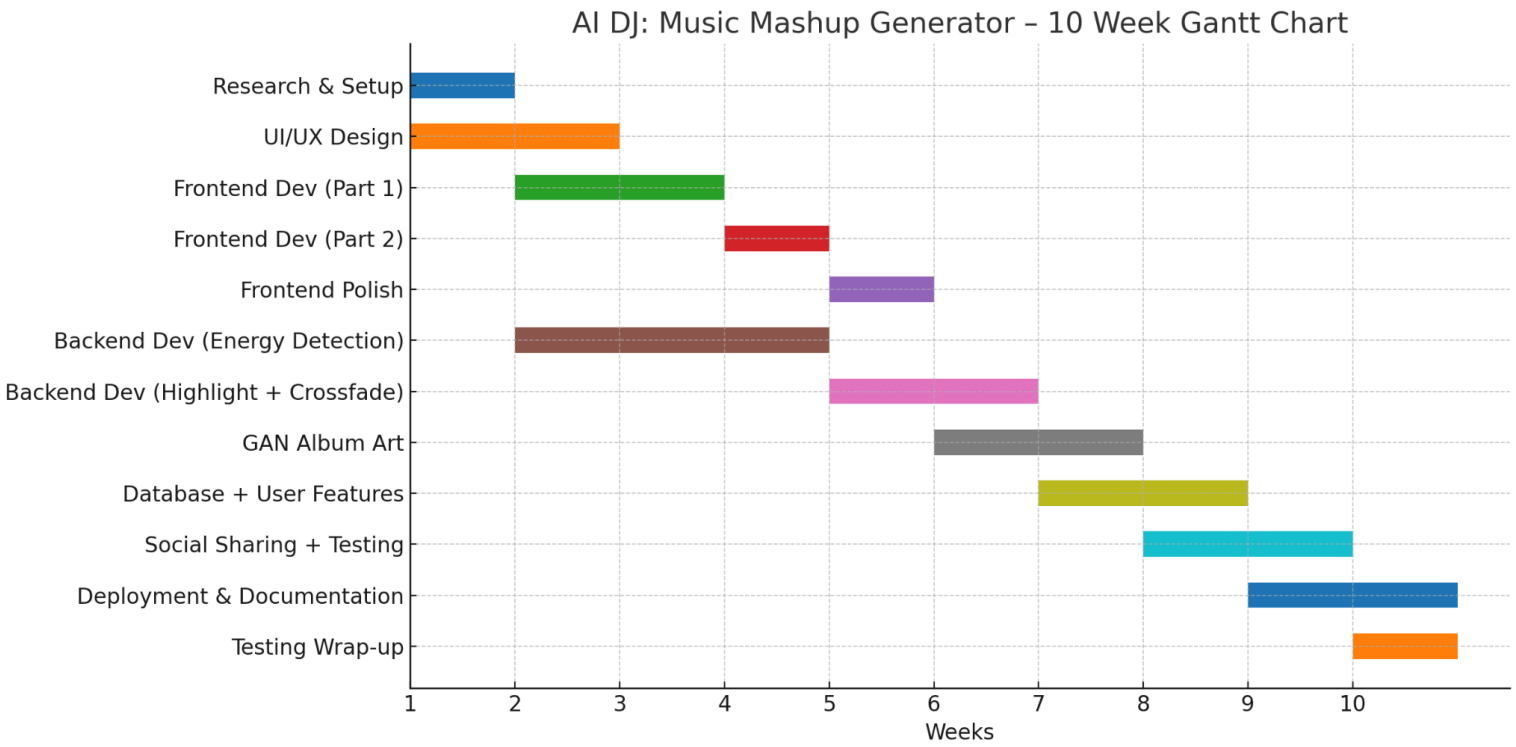
- Use established libraries (Librosa, TensorFlow/PyTorch for GANs).
- Begin with royalty-free/open-source songs to avoid copyright disputes.
- Optimize backend for efficient audio processing.
- Add gamified features (leaderboards, trending mashups) for higher engagement.

---

5. Implementation Timeline

- **Week 1-2:** Research audio processing methods, design UI/UX.
- **Week 3-4:** Build frontend with Next.js and integrate Web Audio API.
- **Week 5-6:** Develop Flask backend with Librosa for audio analysis.
- **Week 7:** Implement GANs for album art generation.
- **Week 8:** Database integration (Firebase) and user mashup storage.
- **Week 9:** Social media sharing + testing.
- **Week 10:** Deployment, documentation, and final presentation.

GANTT CHART :


AI DJ: Music Mashup Generator – 10 Week Gantt Chart
Research & Setup
UI/UX Design
Frontend Dev (Part 1)
Frontend Dev (Part 2)
Frontend Polish
Backend Dev (Energy Detection)
Backend Dev (Highlight + Crossfade)
GAN Album Art
Database + User Features
Social Sharing + Testing
Deployment & Documentation
Testing Wrap-up
1
2
3
4
5
6
7
8
9
10
Weeks