Detailed Requirements Document: Adapting Music to Dramatic Short Videos.

---

1. Introduction

Project Overview: The project aims to develop a system that automatically adapts music to dramatic short videos, enhancing the emotional impact and storytelling. This involves analysing the video, understanding their dramatic elements, and matching them with suitable music tracks.

Objectives:

- Automate the adaptation of music to fit the mood and theme of the scenes\videos.
- Provide a user-friendly interface for filmmakers to customize music suggestions.
- Leverage AI and machine learning for scene analysis and music matching.

---

2. Identify Stakeholders:

The stakeholders for this project include:

1. **Filmmakers and Directors:** The primary end-users who want to enhance their movies' emotional impact through precise music adaptation.
2. **Composers and Music Producers:** Professionals who provide the music tracks used in the system.
3. **Cinema/Theatre Students:** Customers using this tool to help with their studies.
4. **Content Creators :** Enhance emotional impact, improve storytelling, and elevate their content's overall quality**.**

3. Functional Requirements

Key Features

1. **Scene Analysis**
    - Analyse short videos for visual and auditory cues such as brightness, colour grading, motion intensity, and dialogue tone.
    - Detect and classify the dramatic theme (e.g. suspense, romance, action, tragedy).
2. **Music Matching and Adaptation**
    - Match analysed scenes with suitable tracks from a pre-defined music library.
3. **Custom Music Suggestions**
    - Enable users to select a preferred track from the three recommendations provided by the system.
    - Offer alternative options upon request.

4. **Editing and Exporting**
    - Allow users to fine-tune music parameters (e.g., volume, fade-in/out effects).
    - Export the adapted music file and scene in a compatible format.

---

4. Non-Functional Requirements

1. **Performance**
    - Scene analysis and music adaptation should complete within 30 seconds per scene.
    - Support batch processing of multiple scenes simultaneously (optional).
2. **Scalability**
    - Handle increasing numbers of users and larger movie files without performance degradation.
3. **Usability**
    - Provide an intuitive and visually appealing interface for both novice and professional users.
    - Ensure accessibility across devices and platforms.

---

5. Architectural Requirements

1. **System Architecture**
    - **Frontend:** User interface for uploading short videos, viewing analysis, and customizing music.
    - **Backend:** Modules for AI/ML-based scene analysis and music adaptation.
    - **Database:** Store music tracks and scene metadata.
    - **Integration:** APIs for external music libraries and video editing tools.
2. **Performance Indicators**
    - Response time for analysis and adaptation: 5-10 seconds for small videos and 15-30 seconds for larger videos.
3. **Security**
    - Use secure protocols (e.g., HTTPS, TLS) for data transfer.

---

6. Technological Requirements

1. **Programming Languages and Frameworks**

- **Frontend:** React.js for a dynamic user interface.
- **Backend:** Node.js for server-side logic.

2. **AI/ML Tools**
   - Use PyTorch for developing machine learning models.
   - Pre-trained models for visual and audio feature extraction (e.g., OpenCV, LibROSA).
3. **Database**
   - PostgreSQL for storing metadata and user data.
4. **APIs and Libraries**
   - Music and video processing: FFMPEG, LibROSA.
   - External music libraries: Spotify API, YouTube Music API.
5. **Hosting and Deployment**
   - Cloud platforms: AWS, Azure, or Google Cloud for scalability.
   - CI/CD pipelines with GitHub Actions or Jenkins for streamlined deployment.
6. **Compliance and Standards-**
   - Ensure compliance with GDPR and CCPA for handling user data and metadata securely and ethically.
   - Adhere to standards like ISO/IEC 27001 for information security management and industry guidelines for AI transparency in machine learning models.
   - Follow licensing requirements for external APIs (e.g., Spotify API, YouTube Music API) and music libraries.
7. **Documentation Standards-**
8. **Testing and Quality Assurance**
   - Unit testing: Pytest, Jest.
   - Performance testing: JMeter, Locust.
   - Automated testing for critical workflows.

---

7. Example Use Case

Scenario: A filmmaker uploads a 2-minute dramatoc scene. The system analyzes the visual intensity and dialogue pace, identifies it as a suspenseful and tense scene, and selects a matching track from the music library. The filmmaker customizes the track by adjusting the tempo and exports the adapted file for integration.

Expected Outcome:

- The filmmaker receives a professionally adapted music track within minutes.

---

8. Conclusion

The system will bridge the gap between dramatic storytelling and music composition, enabling filmmakers to create compelling cinematic experiences effortlessly. By combining advanced AI/ML technologies with intuitive user interactions, this project will revolutionize the way music is adapted to dramatic scenes.