**AI-Based Video Editing Web Application - Project Documentation**

**1. Project Overview**

**1.1 Introduction**

This project is an AI-powered web application that enables users to create high-quality videos automatically. Users can upload multiple images (in sequence or random order) and provide an audio file. Additionally, users can upload a video, and AI will automatically extract key scenes or images from the video for editing. The AI system will analyze the images and audio, arrange them in an optimal sequence, apply automatic effects, and generate an engaging video. Users can customize effects, transitions, and other parameters to enhance their videos.

**1.2 Objectives**

* Automate video creation using AI-powered image and audio processing.
* Provide a user-friendly interface for seamless video generation.
* Implement AI-based video enhancements, including transitions and effects.
* Offer customization options while keeping most processes automated.
* Extract key scenes from uploaded videos for improved storytelling.

**2. Tech Stack**

**2.1 Frontend (UI Design)**

* Framework: React.js (or Next.js for better performance)
* Styling: Tailwind CSS for a modern and responsive UI
* State Management: React Context API or Redux (if needed)
* Media Preview: HTML5 Video Player, Canvas API

**2.2 Backend**

* Framework: FastAPI (Python) or Django (if database integration is needed)
* AI Processing: OpenCV, MoviePy, FFmpeg, Whisper AI, SceneDetect
* Storage: AWS S3 or Cloudinary for media storage (or local storage for testing)
* Database: PostgreSQL or MongoDB (for user and project data management)

**2.3 AI & Media Processing Libraries**

* OpenCV + Mediapipe: Image analysis, sequencing, and feature detection.
* MoviePy: Video creation, merging images/audio, applying effects.
* FFmpeg: High-quality video rendering, format conversion.
* Whisper AI: Audio noise reduction and speech enhancement.
* SceneDetect: AI-powered scene detection to extract key frames from uploaded videos.
* Deep Learning Models (Optional): For advanced AI-based video enhancements.

**3. System Architecture**

**3.1 Workflow Overview**

1. User Uploads Images, Video & Audio: Users drag and drop or select images, videos, and audio files.
2. AI Pre-Processing:
   * If images are uploaded, AI analyzes and sequences them.
   * If a video is uploaded, AI extracts key scenes using SceneDetect and OpenCV.
   * Audio is analyzed for clarity and enhanced using Whisper AI.
3. AI-Generated Video Preview: AI applies effects, transitions, and generates a preview.
4. User Customization: Users can modify sequences, effects, and enhancements.
5. Final Video Rendering: AI renders the final video and provides a download option.

**3.2 System Components**

* Frontend UI: Handles user interactions and file uploads.
* Backend API: Processes media files and performs AI-based enhancements.
* AI Processing Module: Applies sequencing, transitions, and audio enhancements.
* Scene Extraction Module: Extracts key scenes from uploaded videos.
* Database & Storage: Manages user data and media files.

**4. Features & Functionalities**

**4.1 Core Features**

* User Upload Interface: Drag-and-drop or file selection for images, videos & audio.
* AI-Powered Sequencing: Automatically arranges images based on AI analysis.
* Scene Extraction from Videos: AI extracts key moments from uploaded videos.
* Transitions & Effects: Auto-generated and customizable.
* Audio Enhancements: Noise reduction, volume control.
* Live Video Preview: Before final rendering.
* Export & Download: Supports multiple formats.

**4.2 Advanced Features (Future Enhancements)**

* AI-Based Scene Detection with Face & Object Recognition
* Automatic Subtitle Generation
* Voiceover & Text-to-Speech Integration
* Multi-Language Support

**5. Step-by-Step Development Plan**

**5.1 Phase 1: UI Development**

1. Design wireframes and create a responsive UI.
2. Develop static UI components using React.js and Tailwind CSS.
3. Implement file upload functionality (without backend processing initially).

**5.2 Phase 2: Backend & AI Integration**

1. Set up FastAPI or Django backend.
2. Implement AI-based sequencing using OpenCV & MoviePy.
3. Develop a video scene extraction module using SceneDetect & OpenCV.
4. Develop endpoints for AI processing and video generation.

**5.3 Phase 3: API Integration & Testing**

1. Connect frontend UI to backend APIs.
2. Implement live preview and user control options.
3. Test for performance, accuracy, and responsiveness.

**5.4 Phase 4: Deployment & Optimization**

1. Optimize AI processing for faster performance.
2. Deploy backend (AWS, DigitalOcean, or Firebase backend options).
3. Deploy frontend (Vercel or Netlify for React apps).

**6. API Endpoints**

| Endpoint | Method | Description |
| --- | --- | --- |
| /upload | POST | Uploads images, videos & audio |
| /extract\_scenes | POST | Extracts key scenes from uploaded videos |
| /process | POST | AI processes images, videos & audio |
| /preview | GET | Retrieves AI-generated video preview |
| /export | POST | Final video rendering & download |

**7. Future Enhancements & Scalability**

* Advanced AI Models for Scene Understanding: Use deep learning for better scene detection.
* Cloud Storage Integration: Store videos securely on AWS S3 or Cloudinary.
* Collaboration Features: Allow multiple users to edit the same project.
* Mobile App Version: Extend functionality to mobile devices.

**8. Conclusion**

This AI-based video editing web application will provide an intuitive, high-performance solution for automated video creation. By leveraging cutting-edge AI and media processing technologies, users can effortlessly generate professional-quality videos with ease.

Next Steps: UI Development & Initial Backend Setup.