****

**SOFTWARE REQUIREMENTS**

**SPECIFICATION (SRS)**

***for***

# Automated Video Generation from Text Using Generative AI

Version 1.0

Prepared by

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | Roll Number | Student Name | |
| 1. |  |  | |
| 2. |  |  | |
| 3. |  |  | |
|  |  |  | |
|  |  | |  |

|  |  |
| --- | --- |
| Supervisor: |  |
| Designation: |  |
| Department: |  |
| Batch ID: |  |
| Date: |  |
| Supervisor Sign. & Date |  |

**Department of Computational Intelligence**



**Title of the Project: Automated Video Generation using Generative A**

# Content

Contents I[i](#__RefHeading___Toc397335649)

Revisions I[ii](#__RefHeading___Toc397335649)

1 Introduction [1](#__RefHeading___Toc397335650)

1.1 Document Purpose [1](#__RefHeading___Toc397335651)

1.2 ProJECt Scope [1](#__RefHeading___Toc397335652)

1.3 Existing Systems 2

1.4 Problems with Existing Systems 2

1.5 Proposed Systems 3

1.6 Advantages of Proposed Systems 5

2 Overall Description 6

2.1 Feasibility Study 6

2.2 Product Functionality 7

2.3 Design and Implementation Constraints 8

2.4 Assumptions and Dependencies 9

3 Functional Requirements 10

3.1 Software Requirement Specifications 10

3.2 Hardware Requirements Specifications 10

3.3 Use Case Model 10

4 Other Non-functional Requirements 13

4.1 Performance Requirements 13

4.2 Safety and Security Requirements 13

4.3 Software Quality Attributes 14

5 Other Requirements 16

5.1 Database Requirements..................................................................................................................16

5.2 Internationalization Requirements............................................................................................16

5.3 Legal Requirements........................................................................................................................16

5.4 Reuse Objectives...............................................................................................................................17

5.5 Development Environment Requirements...................................................................................17

5.6 Documentation Requirements.......................................................................................................17

6 REFERENCES 18

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | George Orwell | Primary Revision giving an overall view of the project and document. | 24/02/25 |

# Introduction

## This section details the proposed system, "AI-Driven Video Generation Platform," an innovative content creation system designed to address the shortcomings of existing multimedia production methods. By leveraging the power of Generative AI, this system aims to provide a more efficient, scalable, and accessible approach to transforming text into high-quality video content. This section will outline the core features, functionalities, and design principles of the proposed system, providing a comprehensive overview of how this AI-driven platform will achieve its objectives.

## Document Purpose

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements for the **AI-Driven Automated Video Generator for Content Creation**. The system aims to automate the entire video production process by leveraging artificial intelligence to generate scripts, synthesize voiceovers, create captions, and compile videos seamlessly. The document serves as a comprehensive guide detailing the system’s objectives, functionalities, constraints, and implementation specifics, ensuring clarity for all stakeholders involved in the project.

## Project/Product Scope

The AI-Driven Automated Video Generator is designed to cater to content creators, educators, marketers, and businesses seeking an efficient and scalable solution for video production. By integrating AI-powered tools, the system streamlines the process from text-based input to final video output. The core functionalities include:

* **Automated Script Generation:** Uses **ChatGPT** to create structured and engaging scripts from user-specified topics or keywords.
* **Voiceover Synthesis:** Converts scripts into natural-sounding speech using **EdgeTTS** with customizable voice options.
* **Caption Generation:** Automatically generates and synchronizes captions using **Whisper**, improving accessibility.
* **Video Compilation:** Retrieves relevant stock footage from **Pexels API** or allows custom uploads, merging all components into a final video.
* **User-Friendly Interface:** Developed using **Streamlit**, ensuring an intuitive and customizable experience for non-technical users.
* **Scalability and Automation:** Enables batch processing for mass video production, optimizing efficiency for high-content demand industries.

This system eliminates the need for multiple software applications by offering an integrated, AI-powered, one-click solution for professional video creation.

## Existing System

Currently, video content creation is heavily dependent on manual intervention, with most creators using a combination of tools for scriptwriting, voiceover recording, video editing, and caption synchronization. Some existing systems include:

* **Traditional Methods:**
  + Manual scriptwriting and research.
  + Hiring voice artists for narration.
  + Using software like Adobe Premiere Pro or Final Cut Pro for video editing.
  + Manually adding captions using video editing tools.
* **AI-Assisted Solutions:**
  + Some platforms offer text-to-speech conversions but lack video integration.
  + Certain AI-driven tools generate captions but do not sync them with videos effectively.
  + Existing solutions do not provide full automation from script generation to video rendering.

## Problems with Existing System

The existing approaches to video production come with several challenges:

* **Time-Consuming Workflow:** Creating a single professional-quality video can take hours or even days due to manual effort at multiple stages.
* **Fragmented Toolsets:** Users rely on multiple applications for scriptwriting, voiceovers, video editing, and captioning, leading to inefficiencies.
* **High Costs:** Professional content creation tools and hiring voice artists add significant expenses to the production process.
* **Inconsistent Quality:** Manually created content often lacks uniformity, affecting branding and engagement consistency.
* **Scalability Issues:** Manual workflows do not allow for high-volume content production, making it difficult to scale video creation.

## Proposed System

1. To address these limitations, the AI-Driven Automated Video Generator offers a **fully integrated and automated solution**:
2. **AI-Powered Scriptwriting:** Generates topic-based scripts using **ChatGPT**, ensuring engaging and well-structured content.
3. **Realistic Voiceover Synthesis:** Uses **EdgeTTS** to produce human-like narration with multiple voice and language options.
4. **Accurate Captioning:** Implements **Whisper** to create perfectly synchronized captions for accessibility and engagement.
5. **Automated Video Retrieval:** Searches and selects suitable background footage from **Pexels API** based on AI-analyzed keywords.
6. **One-Click Video Rendering:** Combines all elements using **MoviePy and FFmpeg**, producing high-quality MP4 videos ready for sharing.
7. **Customizability:** Allows users to edit generated scripts, select different voiceovers, adjust captions, and preview the video before finalization.
8. By combining AI-based automation with a user-friendly interface, the system significantly reduces video production time while ensuring high-quality output at minimal cost.

## 1.6 Proposed System Advantages

* **Time-Saving:** Automates the entire video production workflow, reducing content creation time from hours to minutes.
* **Cost-Effective:** Eliminates the need for expensive software tools, voice actors, and manual editing.
* **Seamless Integration:** Provides a one-click solution for script generation, voice synthesis, captioning, and video assembly.
* **Scalability:** Enables high-volume content production, making it suitable for businesses and educators.
* **User-Friendly:** Designed for non-technical users with an intuitive web-based interface.

# Overall Description

## Feasibility Study

A comprehensive feasibility study was conducted to assess the viability of the **AI-Driven Video**

A feasibility study was conducted to evaluate the viability of implementing this AI-driven system across **technical, economic, and operational dimensions**:

* **Technical Feasibility:** The system is built using **proven AI technologies** such as OpenAI, EdgeTTS, and Whisper, ensuring high accuracy and reliability. Its implementation using Python and Streamlit ensures ease of development and scalability.
* **Economic Feasibility:** By automating multiple content creation steps, the system reduces overall production costs, making it **affordable for small businesses and individual creators**.
* **Operational Feasibility:** The platform’s **simple UI** and **intuitive workflow** ensure that users can operate the system with minimal training.
* **Legal Feasibility:** The system complies with **copyright laws** regarding media usage from external sources like Pexels API, ensuring legal adherence in content generation.

## Product Functionality

The AI-Driven Automated Video Generator provides the following core functionalities:

1. **User Input Processing:** Users provide a keyword or short description to initiate the content creation process.
2. **AI-Based Script Generation:** The system creates a well-structured, engaging script based on the input.
3. **Voiceover Synthesis:** Converts the script into speech using EdgeTTS, offering multiple voice options.
4. **Caption Generation:** Whisper generates **timed captions**, ensuring accessibility.
5. **Background Video Selection:** Fetches visually relevant video clips from Pexels API.
6. **Video Assembly and Rendering:** Uses MoviePy and FFmpeg to merge all assets into a final video.
7. **Preview and Download:** Users can preview the generated video and download the final output.
8. **Customization Options:** Users can manually edit the script, select different voices, adjust captions, or upload their own video assets for enhanced personalization.
9. **Scalability and Batch Processing:** Enables multiple video generation requests, making it efficient for high-content production needs.

## Design and Implementation Constraints

The design and implementation of the AI-Driven Automated Video Generator come with various constraints that need to be considered for optimal system performance and usability:

* **Hardware Limitations:** The system requires adequate computing resources, including a powerful GPU and high RAM capacity, for smooth video processing.
* **API Rate Limits:** The system relies on third-party APIs such as OpenAI, EdgeTTS, and Pexels, which have rate limits that could impact real-time processing.
* **Processing Time Constraints:** While AI-based automation reduces manual effort, certain processes such as script generation and voice synthesis must complete within predefined timeframes to maintain user experience.
* **Storage and Bandwidth Constraints:** Handling high-resolution video assets and audio files demands efficient storage management and high-speed internet connectivity.
* **Software Dependencies:** The platform depends on multiple Python libraries (e.g., MoviePy, FFmpeg, OpenAI API), requiring continuous updates and compatibility maintenance.
* **User Interface Responsiveness:** The Streamlit-based GUI should provide a seamless experience, ensuring minimal latency between user input and AI processing output.
* **Legal and Ethical Considerations:** The generated content should adhere to copyright regulations and ensure that no offensive, misleading, or inappropriate material is produced.

## Assumptions and Dependencies

For successful implementation, the system is based on several key assumptions and dependencies:

* **Availability of AI Services:** The system assumes that OpenAI’s ChatGPT, EdgeTTS, and Whisper will remain available and functional during operation.
* **Reliable Internet Connectivity:** Since the platform requires API calls and cloud-based processing, a stable internet connection is necessary for seamless functionality.
* **Pexels Video Database Accessibility:** The background video search feature depends on Pexels API, assuming that sufficient stock video assets exist for diverse keywords.
* **User Familiarity with Basic UI Navigation:** The system presumes that users can interact with simple web interfaces for entering prompts, previewing videos, and making minor edits.
* **Legal and Ethical Compliance:** Users are expected to adhere to copyright and licensing policies while using AI-generated content.
* **System Maintainability:** Regular updates to AI models and software dependencies will be required to maintain the system’s efficiency and relevance over time.

# Functional Requirements

## Software Requirement Specifications

The software components necessary for the development and deployment of this system include:

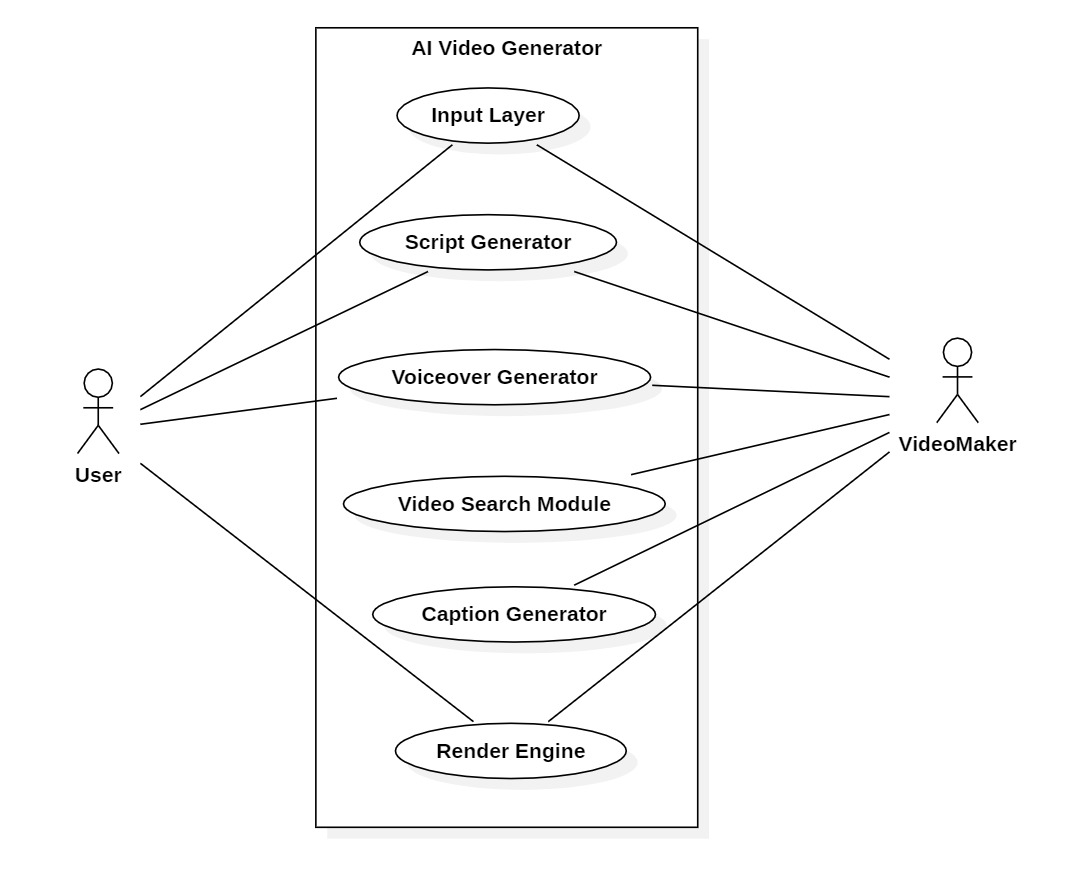
* **Operating System:** Compatible with Windows 10/11, Linux, and macOS.
* **Programming Language:** Python 3.11.
* **Web Framework:** Streamlit for UI and API interactions.
* **Libraries and Dependencies:** OpenAI API, EdgeTTS, Whisper, MoviePy, FFmpeg, NumPy, Pandas.
* **Cloud and Storage:** AWS or Google Drive for storing generated videos.
* **Version Control:** GitHub for source code management and collaboration.
* **Database:** Log storage and retrieval mechanisms for AI-generated scripts, captions, and metadata.

## Hardware Requirements Specifications

For optimal performance, the following hardware specifications are recommended:

* **Processor:** Intel Core i5 12500H or higher.
* **RAM:** Minimum 16GB (Recommended 32GB for high-performance tasks).
* **Storage:** 20GB free space (SSD preferred for faster data access).
* **GPU:** NVIDIA RTX 3050 4GB or equivalent for AI model acceleration.
* **Internet Connection:** High-speed internet for real-time API calls and media processing.

## Use Case Model



**Actors:**

* **User:** Provides input (keyword or topic) and customizes output.
* **System:** Processes AI-generated scripts, voiceovers, captions, and video rendering.

**Use Case Workflow:**

1. **User Input:** The user enters a keyword/topic to generate a video.
2. **AI Processing:** The system generates a script, synthesizes voice, and selects relevant stock videos.
3. **Customization Options:** Users can modify the script, change voice types, and adjust captions before rendering.
4. **Video Compilation:** The system merges assets into a final video.
5. **Preview and Download:** Users can preview the generated video and download it in MP4 format.

**Alternative Scenarios:**

* Users manually modify the generated script and re-run the video generation.
* The system fails to find a relevant stock video, prompting the user to upload their own.
* API limits are exceeded, causing a delay in content generation.
* Users request different voice styles or speeds for voiceover adjustments.

### **Data Flow Diagram**

## 

# Other Non-functional Requirements

## Performance Requirements

To ensure a **seamless and efficient** user experience, the following performance requirements

The system is designed to deliver high performance to meet user demands efficiently. The following performance requirements are specified:

* **Response Time:** The system should generate AI-based scripts within **3 seconds** after user input.
* **Voice Synthesis Speed:** Text-to-speech conversion should take no longer than **5 seconds** for a 1-minute script.
* **Video Rendering Time:** The complete video generation process should be completed within **2 minutes** for a standard-length video (60 seconds).
* **Concurrent Requests Handling:** The system should support **multiple concurrent users** generating videos without noticeable performance degradation.
* **Latency Management:** API requests to OpenAI, EdgeTTS, and Pexels should be optimized to minimize response delays.
* **System Uptime:** The platform should maintain a **99.9% uptime** for reliability and availability.

## Safety and Security Requirements

To ensure the protection of user data and prevent malicious activities, the following security measures are implemented:

* **User Authentication:** Secure login mechanisms should be implemented for saving user-generated projects.
* **Data Encryption:** AI-generated scripts, voiceovers, and videos should be encrypted using **AES-256 encryption** before storage.
* **API Security:** Secure API communication should be ensured using **OAuth 2.0 authentication** and **SSL/TLS encryption**.
* **Access Control:** Role-based access control (RBAC) should be implemented to restrict unauthorized modifications.
* **Content Moderation:** AI-generated scripts and captions should be checked for offensive, inappropriate, or copyright-violating content.
* **Regular Security Audits:** Periodic security assessments should be conducted to identify vulnerabilities and strengthen security.
* **Data Backup:** Automated backups should be scheduled to prevent data loss in case of system failures.

## Software Quality Attribute

The system should meet high software quality standards to ensure usability, maintainability, and efficiency:

* **Scalability:** The system should handle increasing numbers of video generation requests without performance loss.
* **Reliability:** The AI models and API services should maintain high **accuracy and consistency** in output.
* **Usability:** The platform should be user-friendly, enabling non-technical users to generate videos effortlessly.
* **Maintainability:** The system should have modular components to allow easy updates and bug fixes.
* **Portability:** The system should be cross-platform compatible, functioning on Windows, macOS, and Linux environments.
* **Extensibility:** Future enhancements, such as multilingual support, should be easily integrable.
* **Efficiency:** The system should optimize processing time and resource utilization while maintaining high-quality output.

# Other Non-functional Requirements

This section outlines additional requirements not covered in the previous sections, which are essential for the complete development and deployment of "Automated Video Generator Using Generative AI".

## ****Database Requirements (If Applicable)****

**D1. User Data Storage:** The system requires a lightweight and scalable database to store logs of AI-generated scripts, captions, and video metadata. Options include **SQLite** for local storage and **PostgreSQL/Firebase** for cloud-based solutions.

**D2. Video Metadata Logging:** The system should store details like **processing time, AI model usage, script history, video duration**, and any manual edits made by the user.

**D3. Data Security & Integrity:** All stored content should be **encrypted** to protect against unauthorized modifications or corruption.

## ****Internationalization Requirements (If Applicable)****

* **I1. Multilingual Support:** The system should allow users to generate voiceovers and captions in multiple languages using **Whisper (for captions) and EdgeTTS (for voiceovers)**.
* **I2. Unicode & Character Encoding:** The platform must support **UTF-8 encoding** to handle diverse scripts, characters, and special symbols.
* **I3. Cultural Adaptation:** AI-generated content should be **reviewed for cultural appropriateness** to avoid bias in language, voiceovers, or captions.

## ****Legal Requirements****

* **L1. Copyright Compliance:** The system should only use **royalty-free stock videos from sources like Pexels** and ensure AI-generated content does not violate copyright laws.
* **L2. Data Protection & Privacy:** If the system logs user interactions, it must comply with **GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act)** to ensure privacy.
* **L3. Content Usage Policy:** AI-generated scripts and videos must adhere to ethical AI standards and **should not be used for misinformation, political propaganda, or harmful content**.

## ****Reuse Objectives****

* **R1. Modular AI Components:** The system should allow **AI models (script generation, voice synthesis, and video processing) to be used in other projects** with minimal modifications.
* **R2. Prebuilt Video Templates:** Users should be able to select **reusable templates** to maintain consistent branding across multiple video projects.
* **R3. Cloud-Based Storage & Asset Reuse:** Frequently used assets such as **background videos, animations, and voice styles** should be stored and retrieved efficiently for future projects.

## ****Development Environment Requirements****

* **DE1. Programming Stack:** The system should be developed using:

**Python 3.11** (primary language)

**Streamlit** (web framework)

**MoviePy & FFmpeg** (video processing)

**EdgeTTS & Whisper** (voice synthesis and captions)

* **DE2. Version Control System:** The project must use **GitHub/GitLab** for version control, collaboration, and tracking changes.
* **DE3. AI Model Testing Tools:** The team should use **unit testing frameworks (pytest), integration testing, and dataset validation** to ensure AI reliability.

## ****Documentation Requirements****

* **DOC1. Code Documentation:** All AI models, APIs, and processing scripts should have **detailed inline comments and external documentation** for future developers.
* **DOC2. System Architecture Documentation:** The workflow, database structure, and API interactions should be documented using **UML diagrams and flowcharts**.
* **DOC3. User Guide & Tutorials:** The system should provide:
* **Step-by-step guides** on how to generate videos
* **FAQs and troubleshooting steps**
* **Video tutorials** explaining the platform's features
* **DOC4. Release Notes & Versioning:** Each software update must include **release notes detailing new features, bug fixes, and improvements**.

# References

* **R1. AI-Powered Content Creation** – A study on **Generative AI in multimedia production**.
* **R2. Text-to-Video AI Models** – Research on **Natural Language Processing (NLP) and Deep Learning** for automated video synthesis.
* **R3. Ethical AI Guidelines for Content Creation** – IEEE standards ensuring **AI-generated media complies with ethical AI guidelines**.

# SRS DOCUMENT REVIEW

# CERTIFICATION

This Software Requirement Specification (SRS) Document is reviewed and certified to proceed for the project development by the Departmental Review Committee (DRC).

|  |  |
| --- | --- |
| Date of SRS Submitted: |  |
| Date of Review: |  |
| Supervisor Comments: |  |
| Supervisor Sign. & Date. |  |
| Coordinator Sign. & Date |  |
| HOD Sign. & Date |  |
| Dept. Stamp |  |

**Guidelines :**

* Font Style for the Content Page is ”Times New Roman”.
* Font Style for the Remaining Pages is “Calibri(Body)”.
* Font size for Headings is 14 size

(Example – 4.3 Software Quality Attributes).

* Font size for Sub-Headings is 12 size

(Example – 4.3.1 **Usability**).

* Headings and Sub-Headings must be in bold letters.
* Font size for the content is 12 size.
* Line Spacing value is 1.15.