****

**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 outlines the requirements for "AI Driven Video Generation Platform," a system designed to automate the conversion of text into professional-quality multimedia content. This document specifically details the software requirements for version 1.0 of the project.

## The scope of this SRS encompasses text processing, multimedia generation, AI-powered script creation, and video assembly. This document focuses on the software aspects of the project, including the implementation of AI models for summarization, image retrieval, voice synthesis, and video assembly using tools such as MoviePy, FFmpeg, and ImageMagick. It does not cover hardware specifications, marketing plans, or detailed art asset creation, which are considered outside the scope of this document. The aim is to provide a clear and comprehensive understanding of the software functionalities required to achieve the project's objectives.

## Project/Product Scope

The AI-Driven Video Generation Platform is designed to provide an efficient and scalable solution for creating multimedia content from text-based sources. This platform automates the entire process of video production, from text extraction and summarization to video synthesis and voiceover generation. The system utilizes advanced AI models and machine learning techniques to streamline content creation, making it accessible to a wider audience.

The core objectives of the system include:

* Enhancing efficiency in multimedia production by automating scriptwriting, image retrieval, and voice synthesis.
* Enabling multilingual content generation to increase global reach and accessibility.
* Reducing the time and manual effort required for video production while ensuring high-quality output.

## Existing System

Currently, multimedia content creation relies heavily on **manual workflows** that are time-intensive and require significant human effort. The primary methods used today include:

* **Traditional Methods**
* **Manual Scriptwriting:** Content creators manually draft scripts from raw text.
* **Image and Video Editing:** Visual elements are sourced and edited using traditional software.
* **Voiceover Production:** Voiceovers are recorded manually and synchronized with video content.
* **Digital Tools**
* **Basic Automation Tools:** Some AI-assisted tools exist but often lack full end-to-end automation.
* **Limited Integration:** Existing platforms may offer individual automation solutions but do not provide a **seamless workflow** from text to video.

## Problems with Existing System

Currently, multimedia content creation involves manual effort at multiple stages, making it time-consuming, expensive, and inefficient.

The existing methods for transforming text into video content lack seamless automation, resulting in several key challenges:

**Time-Consuming and Labor-Intensive:**

* Traditional methods require extensive manual work, making large-scale content production inefficient.

**Limited Automation and Scalability:**

* Most current systems lack the ability to automate the entire multimedia production pipeline.

**Lack of Multilingual Support:**

* Manually producing content in multiple languages requires significant additional effort.

**High Costs and Resource Requirements:**

* Professional video production is costly, making it inaccessible to smaller creators and businesses.

## Proposed System

The **AI-Driven Video Generation Platform** automates the process of transforming text into **high-quality, engaging video content** using advanced AI-powered tools and machine learning models. This end-to-end system streamlines video production, reducing the time and effort required while ensuring professional-grade results. Below is a detailed breakdown of the system’s key steps:

**1. Document Selection**

Users begin by **uploading or selecting** a text-based document from a repository. The system supports various input formats, including:

* **PDFs, Word documents, and plain text files**
* **Web articles and blogs** (via web scraping or API integration)

**2. Text Summarization**

Since raw documents can be lengthy and contain **extraneous information**, the system employs **AI-based text summarization models** (such as **BART, BERT, or T5**) to extract the **key points** from the document. This step ensures:

* Removal of **redundant information**
* Extraction of **core themes and essential insights**
* **Condensation** of long-form content into a concise format

**3. Script Generation**

Once the content is summarized, the system utilizes **Large Language Models (LLMs)** (e.g., **GPT-4, T5, or LLaMA**) to generate a **structured narration script**. The script is:

* **Contextually relevant**, ensuring a natural storytelling flow
* **Formatted for speech**, making it easy to convert into a voiceover
* **Adaptable for different tones**, such as **informative, casual, or promotional**

**4. Image and Audio Asset Sourcing**

To enhance the **visual and auditory appeal** of the generated video, the system automatically sources **relevant images and audio assets**:

* **4.1 Image Selection**
* **4.2 Voiceover Generation**

The system employs AI-powered **text-to-speech (TTS) models** (e.g., **Google WaveNet, Amazon Polly, or VITS**) to generate **natural-sounding voiceovers**. Features include:

* **Multiple voice options** (male, female, different accents)
* **Multilingual support** for accessibility

## Advantages of Proposed Systems

The AI-Driven Video Generation Platform offers several distinct advantages over existing multimedia production methods:

* Time and Cost Efficiency:

Automates scriptwriting, image selection, and voice synthesis, reducing manual effort.

* End-to-End Automation:

Provides a seamless workflow from text input to a finished video.

* Enhanced Scalability:

Supports large-scale content production for businesses, educators, and content creators.

* Multilingual Capabilities:

AI-generated voiceovers and subtitles allow content to be produced in multiple languages effortlessly.

* Professional-Quality Output:

Uses AI-enhanced tools like WaveNet for natural-sounding voiceovers and GANs for video frame synthesis.

* Contextualized Learning and Engagement:

Ideal for e-learning and training content, improving information retention.

The proposed system revolutionizes multimedia production by leveraging AI to enhance efficiency, accessibility, and content quality.

This automated workflow allows creators to produce professional videos with minimal effort, bridging the gap between **text-based information and engaging visual content**.

# Overall Description

## Feasibility Study

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

This evaluation covered four key areas: **technical, economic, operational, and scheduling feasibility**.

* **Technical Feasibility**

**Technology Availability**: The project relies on established AI frameworks, including **Hugging Face Transformers, OpenAI CLIP, Google WaveNet, and MoviePy**, all of which provide robust support for text processing, image retrieval, and video generation.

**Software Integration**: The use of widely adopted tools such as **FFmpeg, ImageMagick, and TensorFlow/PyTorch** ensures smooth integration and compatibility.

**Scalability**: The platform's **modular architecture** allows for easy updates and the addition of **new AI models** as technology evolves.

**Hardware Requirements**: While AI-powered video processing can be computationally intensive, cloud-based solutions (e.g., **AWS, Google Cloud**) can be leveraged for scalability, allowing users to generate videos efficiently even on lower-end systems.

* **Economic Feasibility**

**Development Costs**: The system leverages **open-source AI models and existing APIs**, reducing the cost of development.

**Potential Revenue Streams**: Monetization opportunities include **subscription-based services, licensing for enterprises, and integration with content creation platforms**.

**Return on Investment (ROI)**: By significantly **reducing manual effort in video production**, the platform offers a **high ROI** for businesses, educators, and content creators.

* **Operational Feasibility**

**User Adoption**: Automating video creation simplifies content production for non-technical users, increasing adoption rates among **educators, businesses, and marketing professionals**.

**Maintenance & Support**: AI models and processing pipelines are designed to be **modular and scalable**, ensuring maintainability.

**Integration with Existing Systems**: The platform can integrate with **social media APIs, e-learning systems, and content management platforms**, making it a seamless addition to existing workflows.

* **Scheduling Feasibility**

**Project Timeline**: A structured development timeline has been established, including **phases for model training, testing, optimization, and deployment**.

**Resource Availability**: The project is designed to be completed with **available AI tools, cloud services, and a skilled development team**.

**Risk Management**: Potential risks, such as **AI model inaccuracies, performance bottlenecks, and evolving technology trends**, have been identified, with contingency plans in place.

The feasibility study confirms that the **AI-Driven Video Generation Platform** is **technically, economically, and operationally feasible**.

The project has the potential to **revolutionize multimedia content creation** by automating video production, improving scalability, and increasing accessibility.

## Product Functionality

The **AI-Driven Video Generation Platform** offers a set of powerful functionalities to streamline the process of **automated video creation**:

* **Automated Text Processing**: Extracts, summarizes, and structures content using **AI models (BART, BERT, GPT-4)**.
* **Script Generation**: Transforms summarized text into **narration scripts optimized for voice synthesis**.
* **Multimedia Asset Sourcing**: Retrieves **relevant images, animations, and background visuals** using **CLIP and OpenCV**.
* **AI-Generated Voiceovers**: Converts text into **lifelike speech** using **WaveNet, gTTS, and multilingual TTS models**.
* **Video Assembly**: Uses **MoviePy, FFmpeg, and ImageMagick** to create dynamic, professional-quality videos.
* **Multilingual Support**: Enables **automatic translation** and **voice synthesis** in multiple languages for global reach.
* **Automated Publishing**: Integrates with platforms like **YouTube, Instagram, TikTok, and LMS systems** for seamless content distribution.

This **end-to-end automation** eliminates manual effort, making video production **efficient, scalable, and accessible**.

## Design and Implementation Constraints

The development of the **AI-Driven Video Generation Platform** must adhere to several key constraints:

**1. Hardware & Performance Limitations**

* The system must support **high-performance video processing** while remaining accessible to **users with lower-end hardware**.
* **Cloud-based processing** will be leveraged for **scalability** and to reduce the computational burden on local devices.

**2. Software Platform**

* The project will be developed using **Python**, integrating **AI libraries (Transformers, TensorFlow, PyTorch)** with **video processing tools (MoviePy, FFmpeg)**.
* The development team must adhere to **best practices in AI model deployment and optimization**.

**3. Modular Design**

* The system must follow a **modular architecture**, ensuring easy integration of **new AI models and features**.
* Individual components (text summarization, voice synthesis, video assembly) should be **interchangeable** for flexibility and scalability.

**4. User Interface (UI) and User Experience (UX)**

* The UI must be **intuitive and accessible**, catering to users **without technical expertise** in AI or video editing.
* Features like **drag-and-drop text input, preview functionality, and customization options** must be integrated for **ease of use**.

**5. Data Management**

* **Efficient storage and retrieval** mechanisms must be in place for **processed video assets**.
* The system must provide **data visualization and reports** for insights into user engagement.

**6. Security Considerations**

* **User data privacy** must be ensured when processing and storing content.
* The system should prevent **unauthorized access and content tampering**.

**7. Compliance with Industry Standards**

* The platform must adhere to **video format and compression standards** (MP4, H.264).
* AI-generated content must comply with **ethical AI guidelines and copyright laws**.

## Assumptions and Dependencies

The following assumptions and dependencies are critical to the successful development and implementation of "Automated Video Generation using Generative AI":

**Assumptions**

* User Familiarity with Basic Content Creation Tools: Users are expected to have a basic understanding of text input, video preview, and content customization.
* Availability of AI Models: The system assumes continued support and updates from AI libraries such as Hugging Face, OpenAI, and Google TTS.
* Cloud Processing Capabilities: It is assumed that users will have access to cloud-based processing for high-performance video generation.
* User Interest in AI-Generated Content: The platform assumes a demand for automated multimedia generation for marketing, education, and content creation.
* Sufficient Development Resources: The project is expected to be completed within the allocated timeframe with available AI expertise and computing resources.

**Dependencies**

* AI Framework Stability: The platform relies on pre-trained AI models, meaning any major changes or deprecations in these frameworks could impact performance.
* Third-Party APIs and Tools: The system depends on external APIs (Google TTS, CLIP, FFmpeg, OpenCV, AWS S3) for processing and storage.
* Content Licensing and Copyright: Ensuring compliance with intellectual property rights for AI-generated visuals and voiceovers is crucial.

**Conclusion**

The AI-Driven Video Generation Platform is designed with a scalable, modular, and user-friendly architecture that leverages AI for automated multimedia content creation. By addressing the constraints and dependencies identified, the system ensures high-quality, efficient, and accessible video production.

# Functional Requirements

## Software Requirement Specifications

The AI-Driven Video Generation Platform requires the following software components:

* **Operating System**: Windows 10/11, macOS, or Linux (Ubuntu preferred)
* **Programming Language**: Python 3.x
* **Web Framework**: Flask/Django (for UI and API integrations)
* **Cloud & Storage**: AWS S3, Google Drive, Local Storage
* **Version Control System**: Git/GitHub
* **Deployment**: Docker, Kubernetes

## Hardware Requirements Specifications

The platform's AI models and video generation processes require:

* **Processor**: Intel Core i7/i9 or AMD Ryzen 7/9 (or higher)
* **RAM**: Minimum **16GB** (32GB recommended for AI model training)
* **Storage**: 500GB SSD (1TB+ preferred for large media files)
* **GPU**: NVIDIA RTX 3060 or higher (for AI processing and video rendering)
* **Internet**: High-speed internet required for cloud-based processing and data retrieval

## Use Case Model

### **3.3.1 Use Case #1 (Game Play – U1)**

**Author** – Team AI Video Generation  
**Purpose** – This Use Case Diagram provides a high-level overview of how a user interacts with the platform to generate a video from text input.

**Requirements Traceability:**

* **R1**: User input processing
* **R5**: AI-based script generation
* **R10**: Automated video assembly
* **R14**: Multilingual voiceover integration
* **R20**: Social media publishing

**Priority** – High

**Preconditions** – User has uploaded text content (document, script, or raw text).

**Postconditions** – A **fully generated video** is ready for preview, download, or publishing.

**Actors** – User (Content Creator, Educator, Business Owner)

**Extends** – N/A

**Flow of Events**

1. **Basic Flow**
   * **User uploads a text document or enters raw text.**
   * **AI processes the text** using BART/BERT for summarization.
   * **Script is generated** using LLM-based models.
   * **Relevant images are sourced** using OpenAI CLIP.
   * **AI-generated voiceover** is created using Google WaveNet.
   * **Video is assembled** with MoviePy, FFmpeg, and ImageMagick.
   * **Multilingual versions** are generated (if selected).
   * **Final video preview** is displayed, and user can download or publish.
2. **Alternative Flow**
   * **User modifies generated script** before finalizing video.
   * **User manually selects images/audio** if AI recommendations are not satisfactory.
3. **Exceptions**
   * Text input **fails to process** due to unreadable format.
   * **Network failure** interrupts video assembly.
4. **Includes**
   * AI-generated **text-to-speech synthesis**
   * Automated **video editing and transitions**
5. **Notes/Issues**
   * Additional **content customization features** can be integrated in future versions.

### **Data Flow Diagram**

## Block diagram depicting data flow in the adopted video analytics... | Download Scientific Diagram

# Other Non-functional Requirements

## Performance Requirements

To ensure a **seamless and efficient** user experience, the following performance requirements must be met:

* **P1. Processing Speed**: The system should process **text-to-video conversion within 30 seconds** for standard-length content (up to 500 words).
* **P2. Video Generation Time**: The platform must generate a fully assembled video within **2 minutes** for a 1-minute video output.
* **P3. Text Summarization Speed**: AI models (BART/BERT) should extract key points from documents in **under 5 seconds**.
* **P4. Voiceover Rendering Time**: Google WaveNet-based voice synthesis must generate **multilingual speech output within 3 seconds**.
* **P5. Image Retrieval Speed**: CLIP-based image sourcing should complete within **4 seconds** after text input.
* **P6. Video Assembly Optimization**: MoviePy and FFmpeg must optimize asset processing to ensure **smooth rendering with minimal resource usage**.
* **P7. Cloud Processing Scalability**: The system must scale efficiently in **cloud environments (AWS, Google Cloud)** to handle multiple concurrent video generation tasks.
* **P8. User Interaction Responsiveness**: The platform’s UI must respond to user inputs **within 0.5 seconds**, ensuring a smooth experience.
* **P9. Storage Efficiency**: The system should compress video files using **H.264/H.265 encoding** to minimize storage size while maintaining quality.
* **P10. Background Processing**: AI-based **video rendering should run asynchronously**, allowing users to continue working while the video is being processed.

## Safety and Security Requirements

Since the **AI-Driven Video Generation Platform** processes user-generated content, the following security measures must be implemented:

**S1. Data Integrity**: User content (uploaded text, AI-generated scripts, video files) must be **stored securely** and protected from corruption.

**S2. Protection Against Malicious Code**:

* + Regularly scan for vulnerabilities in **third-party AI models and libraries**.
  + Implement **input validation** to prevent **injection attacks**.
  + Apply **secure coding practices** in Python (Flask/Django).

**S3. Secure File Handling**: All user-uploaded files (text, images) should be **sanitized** before processing to prevent malicious code execution.

**S4. Secure API Usage**: The system should use **OAuth 2.0 authentication** for external API calls (Google TTS, AWS S3).

**S5. Encrypted Data Transmission**: Ensure **SSL/TLS encryption** for all data transfers.

**S6. Access Control**: Implement **role-based access control (RBAC)** for managing user permissions.

**S7. User Content Privacy**:

* + Ensure AI-generated content complies with **copyright laws**.
  + Provide options for **data anonymization** in text-based input.

**S8. No Collection of Personal Data**: The platform **does not store** personally identifiable information (PII). Any future user data collection must adhere to **GDPR/CCPA** compliance.

**S9. Software Updates & Security Patching**:

* + Regularly update **AI models and system libraries** to prevent security vulnerabilities.
  + Deliver software updates through **secure channels** to prevent tampering.

**S10. Error Handling & Logging**: Implement robust **error reporting** mechanisms to track failures in **AI model processing, video assembly, and API requests**.

## Software Quality Attributes

The following attributes are critical for ensuring the platform’s usability, maintainability, and scalability.

**4.3.1 . Usability**

**Requirement**: The user interface (UI) must be **intuitive and easy to navigate** for both **technical and non-technical users**.

**Implementation**:

* Provide **drag-and-drop** file uploads.
* Display **real-time progress bars** for AI processing.
* Offer **preview options** before finalizing video generation.
* Conduct **usability testing** to ensure accessibility across different user demographics.

**Verification**: User surveys, feedback analysis, and **A/B testing** will validate usability.

**4.3.2 Maintainability**

**Requirement**: The system architecture should be designed to support **easy updates, bug fixes, and AI model enhancements**.

**Implementation**:

* Use **modular coding practices** (e.g., separate text processing, video generation, and UI modules).
* Implement **version control (GitHub)** to track updates and prevent regressions.
* Document **API endpoints, AI workflows, and deployment procedures**.
* Perform **unit tests and integration tests** to maintain system reliability.

**Verification**: Maintainability will be assessed through **code reviews and automated testing reports**.

**4.3.3 Adaptability (Design for Change)**

* **Requirement**: The platform must be **flexible and scalable** to incorporate **new AI models, video editing features, and additional languages**.
* **Implementation**:
  + Use **plug-and-play AI model integration** to swap text summarization or voice synthesis models.
  + Support **custom user-defined templates** for video styling.
  + Design **API-based integrations** for future compatibility with **third-party content platforms**.
  + Implement **serverless architecture (AWS Lambda, Google Cloud Functions)** for cloud scalability.
* **Verification**: Successfully adding **new AI models and features** within a reasonable timeframe will verify adaptability.

**4.3.4 Reliability**

* **Requirement**: The system must be **highly reliable and stable**, minimizing video processing failures.
* **Implementation**:
  + Implement **retry mechanisms** for AI API calls.
  + Use **distributed processing** for handling high workloads efficiently.
  + Ensure **fail-safe video assembly** so that partially processed content is recoverable.
  + Conduct **extensive testing (unit, integration, system-wide testing)** to minimize bugs.

# 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)****

* **R1. User Data Storage:** If user-generated content needs to be saved, the system should use a lightweight and scalable database (e.g., SQLite, PostgreSQL, or Firebase).
* **R2. Video Metadata Logging**: The platform may log video creation statistics (e.g., processing time, AI model usage, video duration) for analytics and performance tracking.
* **R3. Data Security & Integrity:** Stored content should be encrypted and protected against corruption or unauthorized modifications.

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

* **R4. Multilingual Video Generation:** The system must support **automatic translation** and **multilingual voice synthesis** for **global accessibility**.
* **R5. Cultural Sensitivity:** AI-generated content should be reviewed for **cultural appropriateness** to avoid unintended bias.
* **R6. Unicode & Character Support:** The platform must support **UTF-8 encoding** to handle **diverse languages, scripts, and special characters.**

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

* **R7. Copyright Compliance:** All assets (e.g., **images, audio, AI-generated content**) must comply with **copyright and licensing laws**.
* **R8. Privacy Policy & Data Protection:** If the platform stores **user-generated content or metadata**, a **clear privacy policy** must be in place, adhering to **GDPR and CCPA regulations**.
* **R9. Accessibility Standards:** The platform should adhere to **accessibility guidelines (e.g., WCAG, ADA)** to support **users with disabilities**.

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

* **R10. Modular AI Components:** The platform should be designed to allow **AI modules (text processing, script generation, voice synthesis, video editing) to be reused across multiple projects**.
* **R11. Prebuilt Video Templates:** Users should be able to **select reusable templates** to maintain **consistent branding** across their video content.
* **R12. Cloud-Based Storage & Asset Reuse:** Frequently used assets (**backgrounds, animations, voice styles**) should be **stored and retrieved efficiently**.

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

* **R13. Programming Stack: The system will be developed using Python, Flask/Django (backend), and TensorFlow/PyTorch (AI models).**
* **R14. Version Control System: The development team must use GitHub/GitLab for source control and collaboration.**
* **R15. AI Model Testing Tools: The team must use unit testing (pytest), integration testing, and dataset validation for model reliability.**
* **R16. Cloud Deployment & Scaling:** The platform should be deployable on cloud services (AWS, GCP, or Azure) for large-scale video processing.

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

* **R17. Code Documentation:** All AI models, APIs, and processing scripts must include **detailed documentation and inline comments**.
* **R18. System Architecture Documentation:** The system’s **workflow, database structure, and API interactions** must be documented.
* **R19. User Guide & Tutorials:** The platform should provide **step-by-step guides, FAQs, and tutorials** to assist users.
* **R20. Release Notes & Versioning:** Each software update must include release notes detailing new features, bug fixes, and improvements.

# References

* **AI-Powered Content Creation** – A Study on Generative AI in Multimedia Production.
* **Text-to-Video AI Models** – Research on NLP and Deep Learning for Automated Video Synthesis.
* **Handbook of Video Processing and Computer Vision** – MIT Press.
* **Automated Multimedia Generation** – ACM Digital Library.
* **Ethical AI Guidelines for Content Creation** – IEEE AI Standards.

# 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.