A PRELIMENERY REPORT ON

Dynamic Video Generation from Text Prompts by Using Generative AI and Keyword Analysis

SUBMITTED TO THE MUMBAI UNIVERSITY, MUMBAI
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (INFORMATION TECHNOLOGY)

SUBMITTED BY

DIVYESH ANIL PATIL	**121IT1152A**
SHIVANSH VINOD PANDEY	**121IT1180A**
SNEHA VINOD MALALE	**121IT1056B**
PAVAN SANJAY BHAGAT	**121IT1155A**

Under the Guidance of
Dr. Swati Sinha

DEPARTMENT OF INFORMATION TECHNOLOGY

MGM'S COLLEGE OF ENGINEERING AND TECHNOLOGY

NAVI MUMBAI, KAMOTHE- 410209

MUMBAI UNIVERSITY
A.Y. 2024-2025

CERTIFICATE

This is to certify that the project report entitles.

"Dynamic Video Generation from Text Prompts by Using Generative AI and Keyword Analysis"

Submitted by

DIVYESH ANIL PATIL	**121IT1152A**
SHIVANSH VINOD PANDEY	**121IT1180A**
SNEHA VINOD MALALE	**121IT1056B**
PAVAN SANJAY BHAGAT	**121IT1155A**

are Bonafede students at this institute and the work has been carried out by him/her under the supervision of **Dr. Swati Sinha,** and it is approved for the partial fulfilment of the requirement of Mumbai University, in the **Fourth year of Engineering** (Information Technology).

(Dr. Swati Sinha)
Guide
Department of Information Technology

Dr. Swati Sinha
Head of Department,
Department of Information Technology

Dr. Geeta S. Lathkar
Director,
MGMCET, Navi Mumbai.

Place: Kamothe
Date:

PROJECT REPORT APPROVAL FOR B.E

This Project entitled **"Dynamic Video Generation from Text Prompts by Using Generative AI and Keyword Analysis**" by **Sneha Malale, Shivansh Pandey, Divyesh Patil & Pavan Bhagat** is approved for the degree of Bachelor of Engineering in Information Technology

……..
External Examiner

Internal Examiner

Place: Kamothe
Date:

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the source which has thus not been properly cited or from whom proper permission has not been taken when needed.

Sr. no	Name of Student	UID	Signature
1	Malale Sneha Vinod	121IT1056B	
2	Pandey Shivansh Vinod	121IT1152A	
3	Patil Divyesh Anil	121IT1180A	
4	Pavan Sanjay Bhagat	121IT1155A	

Place: Kamothe
Date:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people, and we are extremely fortunate to have got this all along the completion of our project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them. It is a matter of great pleasure for us to submit a project report on the "**Dynamic Video Generation from Text Prompts by Using Generative AI and Keyword Analysis**", as a part of our curriculum. First and foremost, we express our sincere regard and indebtedness to our Director Dr. Geeta Lathkar, for giving us an opportunity to do the project. work. We would like to thank our HoD, Dr. Swati Sinha madam, for giving us an opportunity to work on enhancing our technical skills while undergoing this project. The project was done under the guidance of Dr. Swati Sinha. We would like to thank her for her valuable time, guidance, encouragement, support, and cooperation throughout the duration of our project. And last but not least a special thanks to my team members, who helped me to assemble the information and gave suggestions to complete our project.

NAME OF THE STUDENTS

MALALE SNEHA VINOD

PANDEY SHIVANSH VINOD

PATIL DIVYESH ANIL

PAVAN SANJAY BHAGAT

ABSTRACT

The project introduces an innovative application capable of generating dynamic videos directly from user-provided text prompts. By employing a generative AI model and a sophisticated keyword analysis technique, the system effectively analyses the input text to produce relevant and visually appealing videos.

A custom generative AI model is used to generate the textual basis of the video. This text is further analysed for keywords. The keywords are used to query a vast library of images, and the images are returned. The text generated by the AI is then converted into spoken audio by a naturalistic text to speech synthesizer. These various medias are then combined to make an immersive video with multi-sensory engagement.

A login system enables the users to save and revive their interactions, make the program personalised and allowing for greater creativity. The project also functions as a demonstration of generative AI as a revolutionary force in the video creation space, offering a way to efficiently make a video without prior videography knowledge.

TABLE OF CONTENTS

1. INTRODUCTION

Video is the most engaging form of media due to its combination of visuals and audio, making it ideal for education. However, video production is complex and time- consuming, requiring skills in writing, audio, and visuals. This report discusses a system that automates video creation from a simple text prompt using generative AI technologies.

The system includes AI-driven processes for script writing, image generation, and voice synthesis, creating a smooth workflow from text to video. It starts by analysing the prompt to generate a video script, which undergoes keyword analysis to guide image selection or generation.

These images are arranged into a slideshow, while the script is converted to speech through a text-to-speech engine, resulting in a synchronized audio-visual presentation. The system offers a fast, efficient, and scalable solution for automated video production, with potential applications in various content creation fields.

1.1 MOTIVATION

Videos are the most engaging form of media produced today. It is perhaps surpassed only by the nascent Virtual Reality technology in its capability to immerse its users in content. Video interfaces with both our ocular and aural sensory systems. Its captivating mixture of both video and audio are the reason behind its engagement power. This allows it to have a capability to be used for interesting ways to educate people. Modern content creator marketplace mostly consists of people who use this power of video to create engaging content.

But this multisensory form of media is also the one which is the most difficult to make. Video production process is often considered the most complex form of media creation as it involves all the other types of media – text (in form of scripts), audio (the narration or music accompanying the visuals) and visuals (the images or videos). It is the amalgamation of skills in all three of these fields that are required to make a good video. For most people, it is very time consuming to learn these skills.

As stated earlier, video is the most engaging form of media. This power of engagement can be used to educate people more effectively. More and more people are seeking video content to fulfil their educational needs. A whole genre of content creators leveraging this have spawned on the interned. But the videos of the topic we might want to learn are not always available. So, a solution using generative AI for this purpose could be developed.

1.2 PROBLEM DEFINITION

In today's digital landscape, creating high-quality video content is both time-intensive and resource-demanding. This process often involves skilled professionals, advanced software tools, and a considerable investment of time, limiting scalability and accessibility for individuals or organizations with limited resources. The growing need for rapid content generation across various platforms has highlighted the demand for automated solutions. Specifically, there is a need for a system that can take a simple text prompt and autonomously generate a complete video, reducing the burden on content creators and enabling quick production without sacrificing quality.

The problem this system addresses is how to automate the end-to-end process of video creation from a text prompt by:

1. Generating a coherent script using generative AI.
2. Identifying key themes and extracting relevant images through keyword analysis of the script.
3. Synchronizing these images in a slideshow format that aligns with the narrative.
4. Automating the voiceover by converting the script into natural-sounding speech using text-to-speech (TTS) technology.

1.3 Objective

The objective of this system is to automate the process of video creation from a text prompt, providing a streamlined, efficient, and scalable solution that minimizes the need for human intervention while maintaining high-quality output. Specifically, the system aims to:

1. Automate Script Generation: Use generative AI to create a coherent, contextually relevant video script from the user's text prompt.
2. Conduct Keyword Analysis: Extract key themes, concepts, and keywords from the generated script to source or generate relevant images for visual representation.
3. Synchronize Visuals and Audio: Convert the selected images into a dynamic slideshow that corresponds to the narrative flow of the script.

1.4 SCOPE

The scope of this system encompasses the automation of the entire video production process from a simple text prompt, using advanced AI technologies to generate scripts, images, and audio. Specifically, the system includes the following key functionalities:

1. Generative Script Creation:

The system takes a text prompt as input and utilizes generative AI models to produce a coherent, contextually appropriate script. This script serves as the foundation of the video, ensuring that the generated content is both relevant and engaging.

2. Keyword Analysis and Image Sourcing:

After generating the script, the system conducts keyword analysis to extract important concepts, themes, and visual elements. Based on these keywords, it either selects relevant images from a predefined database or sources images through APIs, potentially integrating image generation through AI for custom visuals.

3. Slideshow Creation:

The sourced images are organized into a slideshow format that aligns with the narrative structure of the script. This ensures that each part of the script is visually represented, creating a cohesive and synchronized visual experience.

4. Text-to-Speech Integration:

The system incorporates text-to-speech (TTS) technology to convert the generated script into a natural-sounding voiceover. The voiceover is synchronized with the slideshow, ensuring that the audio complements the visuals in a seamless manner.

5. User Interface and Output:

The system offers a simple, user-friendly interface where users can input their text prompt, view the generated video script, and receive the final video output. The final product can be downloaded or directly used for various purposes like presentations, social media content, or educational materials.

6. Extensibility and Customization:

While the core functionality is automated, the system can be extended to allow users some level of customization, such as selecting different voice styles for the TTS, adjusting the speed of the slideshow, or modifying the keywords used for image selection.

Operational Boundary:

- The system does not handle highly specialized or complex video editing tasks such as post-production effects, professional-grade transitions, or manual editing. It is designed to automate basic video creation tasks for users who need rapid and simple content generation.
- The quality of the output (script, images, and voiceover) is dependent on the accuracy of the generative AI and keyword analysis models and may vary based on the prompt provided.
- The system relies on the availability of suitable image sources or APIs and may be limited by image licensing and copyright restrictions for certain visual content.

In summary, the scope of this system is to provide an automated, user-friendly solution for generating simple videos based on text prompts, using AI to handle script writing, image selection, and voiceover, with minimal user input and fast output.

2. LITERATURE SURVEY

Sr. No.	Title	Author	Content
1.	A Survey on Generative AI and LLM for Video Generation, Understanding, and Streaming	Pengyuan Zhou, et al	This paper offers an insightful examination of how generative artificial intelligence (Generative AI) and large language models (LLMs), are reshaping the field of video technology, including video generation, understanding, and streaming.
2	Beyond Language to Image: Next Level AI-Generated Video from Prompt	Mingjun Liu	Dive into the current AI models that have been released so far
3	General-use unsupervised keyword extraction model for keyword analysis	Hunsik Shin, Hye Jin Lee, Sungzoon Cho	Proposed an all-purpose keyword extraction model without requiring keyword label.
4	Applications of Generative AI in Lesson Preparation and Content Development	Adebowale Owoseni, Oluwaseun Kolade & Abiodun Egbetokun	Explores how educators can use generative AI (GenAI) to support and enhance both the development of teaching content and the preparation to deliver the content.

2.1 SURVEY OF EXISTING SYSTEMS AND THEIR LIMITATIONS

Several systems and tools currently exist that leverage AI to automate various aspects of video creation. These solutions incorporate technologies such as generative AI, text-to-speech (TTS), and image generation or sourcing. While each system addresses specific aspects of automated video production, most lack full end-to-end automation or focus on distinct parts of the process. Below is a survey of existing systems that offer similar functionalities to the proposed system:

1. Lumen5

Lumen5 is a popular AI-driven video creation platform that helps users convert blog posts, articles, or text into videos. The system automatically generates a video script based on the input text and matches it with relevant stock images and videos from its media library [1].

Some limitations**:**

- Lumen5 requires manual selection and adjustment of certain elements, such as transitions and voiceovers.
- It lacks real-time generative content and is dependent on pre-existing media, limiting creativity and originality.

2. Animoto

Animoto allows users to create videos by dragging and dropping images, video clips, and text, and then choosing from a variety of pre-designed video templates. It is geared towards social media marketing, with a focus on ease of use and quick video creation [2].

Some limitations**:**

- Animoto lacks advanced AI capabilities and does not automatically generate scripts or analyse keywords.

- It requires more user interaction for video creation compared to systems with full automation.

3. Wave.video

Wave.video is a video marketing platform that allows users to create and edit videos using templates, stock footage, and basic editing tools. While it offers AI-driven features for media selection and optimization, it does not fully automate the video creation process [3].

Some limitations**:**

- Wave.video does not automate script generation or keyword analysis, requiring users to manually handle many parts of the video creation process.
- AI use is limited to media selection, with no generative AI for text or image creation.

Analysis of Gaps in Existing Systems

While the surveyed systems provide varying levels of automation for video creation, most of them either focus on specific parts of the process (e.g., text-to-speech, image matching, or stock media integration) or require significant user interaction for customization. The existing tools often rely on predefined templates and stock media, which limits creativity and flexibility. Few systems offer true end-to-end automation where the entire video production process — from script creation to image selection and voiceover — is fully AI-driven and streamlined.

The proposed system aims to address these gaps by offering a fully automated solution that integrates generative AI for scriptwriting, keyword analysis for image selection, and TTS for audio narration. By automating the entire video creation workflow, it offers a unique and more comprehensive solution compared to existing systems, providing users with an efficient, scalable way to produce videos from text prompts with minimal manual intervention.

3.SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

The system is designed to transform a user's text prompt into a fully produced video by utilizing AI technologies for script generation, keyword analysis, image selection, and text-to-speech (TTS) voiceovers. This document provides a detailed description of the system's scope, user classes, characteristics, and any assumptions or dependencies required for its proper functioning.

3.1.1 Project Scope

The primary goal of the system is to automate the process of creating video content from a simple text input, reducing the time and resources traditionally required for video production. The system leverages generative AI to create a coherent video script, performs keyword analysis to identify visual elements, sources or generates relevant images, and synchronizes these with a text-to-speech voiceover. The final output is a video that effectively communicates the user's input prompt in an audio-visual format.

Key features include:

- Script Generation: AI-driven automatic creation of a video script from a text prompt.
- Image Sourcing: Automated identification of key concepts through keyword analysis and sourcing or generation of relevant images.
- Text-to-Speech: Conversion of the generated script into a natural-sounding voiceover.
- Slideshow Creation: Synchronization of the images and voiceover to form a cohesive video.

3.1.2 User Classes and Characteristics

The system is designed to cater to a broad range of users, from individuals with minimal technical skills to content creators seeking to streamline their workflow. The following user classes have been identified:

1. **Content Creators**:

   Individuals or businesses involved in the production of digital media, marketing, and social media content. They require an efficient way to produce high-quality videos at scale.

2. **Educators and Trainers**:

   Teachers, trainers, and instructional designers who need to create educational videos or presentations quickly without complex editing software.

3. **Social Media Managers**:

   Users responsible for creating regular, engaging content for social media platforms.

4. **Small Businesses**:

   Small businesses or start-ups that lack resources for professional video production but need to create promotional content for their products or services.

5. **Non-technical Users**:

   Users with minimal technical knowledge who need to generate video content without dealing with complex video editing software.

3.1.3 Assumptions And Dependencies

The following assumptions and dependencies apply to the development and operation of the system:

1. **Generative AI and Language Models**:

   The system accesses a Generative AI model for its script generation and keyword analysis sections. Model will be built upon existing low power models like Google Gemma or Llama 3:1b.

2. **Image Source/Generation**:

   The system depends on an API for sourcing relevant images based on keywords. Current scope is limited to existing images available on the web. A Google PSE (Programmable Search Engine) [4] API is being planned to be used initially. It will give access to the whole Google Images dataset but will cause copyright issues if the project is commercially released. If AI-generated images are used in the future, access to generative image models (e.g., GANs, DALL-E) is assumed.

3. **User Internet Access**:

   The system requires internet connections for its image fetching operation. It assumes the user will have a stable and fast internet connection.

4. **Software and Hardware Dependencies**:

   The system can be deployed on the cloud, or it can be run locally. It assumes the user has a sufficiently powerful hardware specifications if it is to be run locally.

5. **Legal and Licensing Considerations**:

   Apart from the text part, the system depends on proper licensing for any external images, music, or media used within the generated videos. As the system currently falls under ‘non-commercial’ category, it is a smaller concern for now.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 User Input and Text Prompt Processing

1: Text Prompt Input

- The system must provide an interface that allows users to input a text prompt. This prompt will serve as the basis for the generated video script.
- The input field must accept plain text, with no restrictions on topic or style, but within a reasonable character limit (e.g., 500-1000 characters).

2: Input Validation

- The system must validate the text prompt input to ensure that it is within the character limit, free of unsupported special characters, and meets basic content standards (e.g., no explicit content if restricted).

3.2.2 Script Generation Using AI

1: Generative AI for Script Creation

- The system must use generative AI models to convert the text prompt into a full video script. The script should be coherent, relevant, and align with the context of the provided prompt.
- The script generation must be completed within a reasonable time frame (e.g., a few seconds), depending on the complexity of the prompt.

2: Script Preview

- After generating the script, the system must provide users with a preview of the generated script before proceeding with further video generation steps.
- Users must have the option to edit the script before moving to the next stage.

3.2.3 Keyword Analysis and Image Selection

1: Keyword Extraction

- The system must perform keyword analysis on the generated script, identifying key concepts, themes, and phrases that are relevant for image sourcing or generation.
- The keyword extraction process must be automated and capable of recognizing multiple types of entities (e.g., objects, locations, people, or actions).

2: Image Sourcing

- Based on the extracted keywords, the system must automatically source relevant images from a predefined media library or via external APIs (e.g., stock image databases).
- If no suitable images are available, the system must either use AI image generation models or suggest alternative visuals to maintain video continuity.

3: Image-Text Matching

- The system must ensure that the images sourced or generated are contextually relevant to the parts of the script where they will be displayed.
- The system must properly match the flow of the script to the images used in the slideshow.

3.2.4 Slideshow and Video Generation

1: Slideshow Creation

- The system must create a slideshow of images based on the script and matched images. The images must be arranged in the proper order to align with the narrative of the script.

- The system must apply basic transitions between images to ensure a smooth visual experience.

2: Script-Image Synchronization

- The system must synchronize the generated script with the slideshow, ensuring that the narration timing matches the visual sequence of images.

3: Preview of Generated Video

- The system must provide users with a preview of the generated video, allowing them to see how the script, images, and voiceover fit together before finalizing the video.

3.2.5 Text-to-Speech (TTS) Integration

1: Text-to-Speech Conversion

- The system must use TTS technology to convert the video script into a natural-sounding voiceover. The TTS engine must support multiple languages and accents, if applicable.
- Users must have the option to select between different voice styles or genders (e.g., male, female, neutral) for the TTS voiceover.

2: TTS Synchronization with Slideshow

- The system must ensure that the TTS narration is synchronized with the slideshow. The timing of the speech must correspond to the display of the relevant images.
- The system must allow for adjusting the narration speed, if necessary, to ensure the voiceover matches the length of the visual display.

3.2.6 User Interface (UI) and Interaction

1: Simpe User Interface

The system must provide a user-friendly interface that allows users to input a prompt, review generated scripts, preview videos, and download the final output.

- The interface must be designed to accommodate users with little to no technical expertise, offering a straightforward workflow from text input to video generation.

2: Video Customization Options

- The system must allow users to make basic customizations, such as selecting different TTS voices, adjusting slideshow timing, or choosing from available visual styles (e.g., transitions, text overlays).

3: Download and Share Options

- After generating the video, the system must provide users with options to download the video in a standard format (e.g., MP4).
- The system must also offer options to share the video directly to social media platforms or cloud storage services.

3.2.7 Error Handling and Feedback

1: Error Handling

- The system must handle potential errors gracefully, including invalid input, failed image sourcing, or issues with the TTS engine.
- Users must be notified of any errors and provided with options to retry, modify their input, or receive troubleshooting tips.

2: User Feedback

- The system must provide feedback to the user at every stage of the process, including progress indicators (e.g., script generation, image sourcing, TTS processing) and the expected time for each step.

3.2.8 System Administration and Maintenance

1: Admin Access for System Monitoring

- System administrators must have access to a dashboard for monitoring system performance, usage statistics, and error logs.
- The system must include tools for managing the image database, AI models, and TTS resources to ensure smooth operation and scalability.

2: System Updates

- The system must support easy updates to AI models, image libraries, and TTS engines to improve performance and add new features as technology evolves.

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 Performance Requirements

1. **Response Time**: Script generation within 1 minute and complete video in under2-5 minutes.

2. **Resource Efficiency**: Optimize CPU and memory usage.

3. Reliability Requirements

1. **System Uptime**: Maximum possible uptime with minimal downtime for maintenance.
2. **Fault Tolerance**: Error handling and tolerance of failures with progress retention.

4. Usability Requirements

1. **User Interface**: Intuitive and accessible for users with minimal technical expertise.
2. **Accessibility**: Compliance with WCAG 2.1 standards.

5. Security Requirements

1. **Data Security**: Protect user data with encryption.
2. **Access Control**: Restrict administrative access and log user interactions.

6. Maintainability Requirements

1. **Code Modularity**: Support easy updates for AI models and libraries.
2. **Documentation**: Comprehensive documentation for components and workflows.

7. Portability Requirements

1. **Multi-Platform Support**: Accessible on major web browsers and mobile devices.

2. **Cloud Deployment**: Deployable on multiple cloud platforms.

3.3.9 Interoperability Requirements

1. **API Integration**: Support RESTful APIs for image sourcing and TTS.
2. **Data Formats**: Download videos in common formats like MP4 and enable social media sharing.

3.4 SYSTEM REQUIREMENTS

3.4.1 Database Requirements

1. **Database Management System (DBMS)**: The system should utilize a relational data base system for storing user data along with other solutions for scalable storage.
2. **Data Integrity**: The database must enforce referential integrity and constraints to ensure accurate and consistent data.

3.4.2 Software Requirements (Platform Choice)

1. **Operating System**: The system will be compatible with major operating systems as it will be a web-based software.
2. **Web Framework**: The system will utilize a modern web framework i.e., React, for the frontend and a Flask for the backend.
3. **AI Libraries**: The system should integrate generative AI libraries (e.g., Ollama, HuggingFace) for script generation and natural language processing.
4. **Text-to-Speech Engine**: The system must incorporate a TTS engine (e.g., Google Cloud Text-to-Speech, Amazon Polly) to convert scripts into audio.
5. **Image Processing Libraries**: Use libraries (e.g., OpenCV, Pillow) for image handling and manipulation, along with APIs for sourcing images.

3.4.3 Hardware Requirements

1. **PC Specifications**:
    a. **CPU**: Multi-core processor with a minimum of 6 cores to handle concurrent processing tasks.
    b. **RAM**: A minimum of 8 GB RAM.
    c. **Storage**: SSD storage of at least with at least 256 GB free for fast data access, with additional storage for user-generated content.
2. **Network Requirements**: A reliable internet connection with a minimum bandwidth of 10 Mbps to support simultaneous users and video streaming.

3.5 Analysis Models: SDLC Model to be applied.

For the development of the **Generative AI-based Video Generation System**, the **Waterfall Model** of the Software Development Life Cycle (SDLC) will be applied. This model is suitable for this system as it allows for a structured, sequential approach with distinct phases, ensuring comprehensive planning, documentation, and quality assurance.

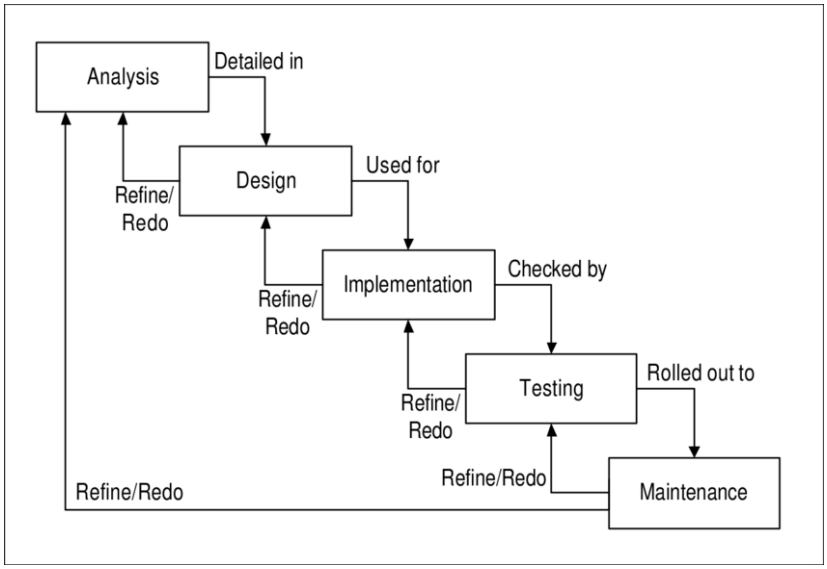


FIGURE 3.1 WATERFALL MODEL PHASES

1. **Requirement Analysis**
    - Gather and document system needs text input, AI-generated script, keyword-based image retrieval, slideshow creation, and text-to-speech (TTS) integration.
2. **System Design**
    - **High-Level Design**: Define system architecture (text input, AI, image sourcing, TTS).
    - **Low-Level Design**: Detailed designs of each module.
3. **Implementation**
    - Develop and implement components: script generation, image retrieval, slideshow, and TTS modules.
4. **Integration and Testing**
    - Perform unit, integration, system, and user acceptance testing to ensure functionality.
5. **Deployment**
    - Set up and configure the system in the production environment with necessary security measures.
6. **Maintenance**
    - Ongoing bug fixes, and optimizations

04 SYSTEM DESIGN

4.1 ALGORITHM

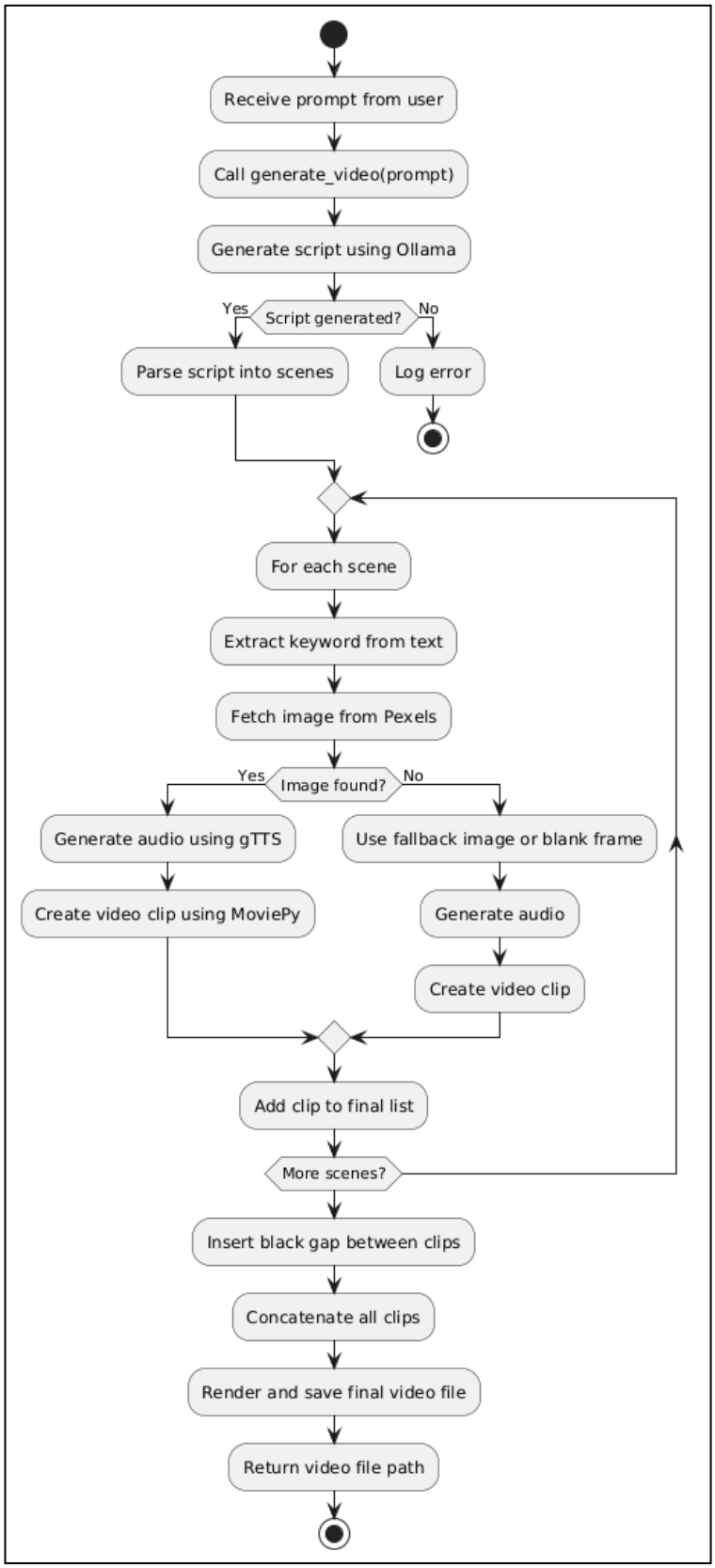


FIGURE 4.1 ALGORITHM

1. **User Input Prompt**: The user provides a text prompt through the user interface.
2. **Invoke Video Generation**: The system calls the generate_video(prompt) function to begin the process.
3. **Generate Script**: A script is generated using the Ollama generative AI model based on the given prompt.
4. **Check Script Generation**: The system checks if the script was successfully created; if not, it logs an error and terminates the process.
5. **Parse Script into Scenes**: If the script is valid, it is broken down into smaller scenes for individual processing.
6. **Extract Keywords from Each Scene**: For every scene, keywords are extracted from the text to assist in image sourcing.
7. **Fetch Image Using Keywords**: The system attempts to retrieve relevant images from the Pexels API using the extracted keywords.
8. **Check Image Availability**: If an image is found, it proceeds; if not, a fallback image or blank frame is used.
9. **Generate Audio**: Using gTTS, the system converts the scene text into an audio narration.
10. **Create Video Clip**: A video clip is created for each scene using the image and corresponding audio with MoviePy.
11. **Add Clip to Final List**: The generated clip is added to a list that will form the final video.
12. **Repeat for Remaining Scenes**: Steps 6 to 11 are repeated for all remaining scenes in the script.
13. **Insert Gaps Between Clips**: Black gaps are added between individual clips for smooth transitions.
14. **Concatenate All Clips**: All clips, including transitions, are joined together into a single video sequence.
15. **Render and Save Video**: The complete video is rendered and saved to the local file system.
16. **Return File Path**: The file path of the saved video is returned so the user can access it.

4.2 PROPOSED SYSTEM ARCHITECTURE

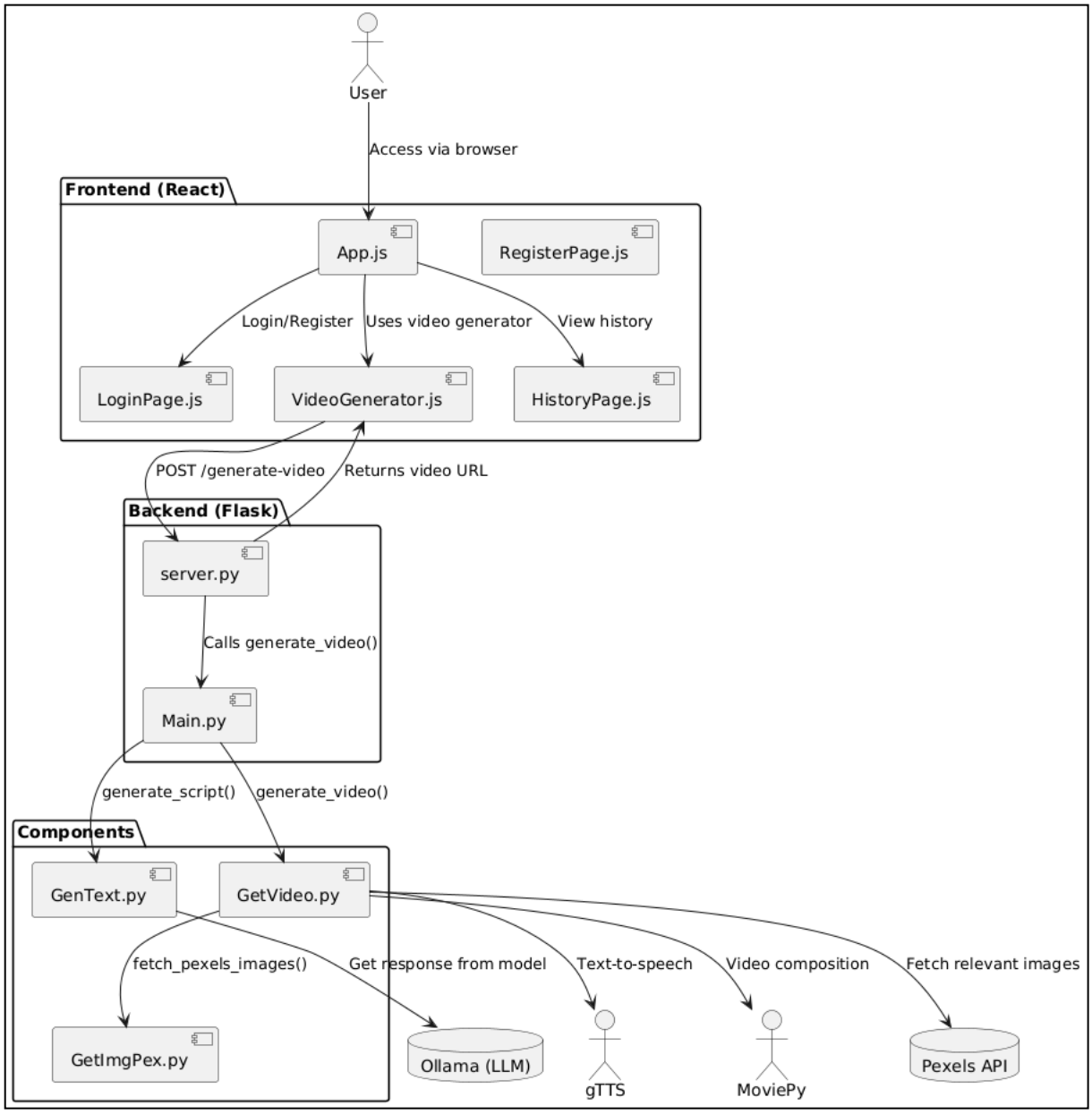


FIGURE 4.2: PROPOSED SYSTEM ARCHITECTURE

Component Descriptions

1. Frontend (React Layer)

The frontend provides the user interface that allows users to interact with the system through a browser.

- **App.js**:

  Central component responsible for routing. It determines which page/component to render based on user navigation.

- **RegisterPage.js**:
  Allows new users to sign up by submitting their credentials to the backend for account creation.
- **LoginPage.js**:
  Enables existing users to log in by verifying their credentials and providing access to video generation functionalities.
- **VideoGenerator.js**:
  Main interactive page where the user enters a prompt, generates a video, previews it, and downloads the final output.
- **HistoryPage.js**:
  Displays previously generated videos, allowing users to view their video generation history.

2. Backend (Flask Layer)

The backend handles logic processing, routes requests from the frontend, and orchestrates video generation.

- **server.py**:
  Flask server file that defines and runs the API endpoints, such as /generate-video.
- **Main.py**:
  Serves as the controller. It receives prompts, calls the necessary functions (generate_script() and generate_video()), and returns the video URL.

3. Components Layer (Script & Video Processing)

This layer handles the script generation, image fetching, audio synthesis, and video assembly.

- **GenText.py**:
  Responsible for generating a video script from the user's prompt using Ollama (LLM). It ensures that the output script is meaningful and scene dividable.
- **GetVideo.py**:
  Core video generation module that converts the script into a final video. It integrates text-to-speech, image sourcing, and video rendering.
- **GetImgPex.py**:
  Communicates with the Pexels API to fetch relevant images based on scene keywords extracted from the script. Ensures visual relevance for each scene.

External Tools/APIs

- **Ollama (LLM)**:
  A language model that processes user prompts and returns human-like, context-rich scripts for videos.
- **gTTS (Google Text-to-Speech)**:
  Converts textual script lines into audio narration, forming the voiceover for the video.
- **MoviePy**:
  Used to combine images and audio into a seamless video. Also handles transitions, timing, and final rendering.
- **Pexels API**:
  Provides free stock images based on keywords. Enhances the visual appeal of the video by aligning visuals with the script content.

4.3 ENTITY RELATIONSHIP DIAGRAM

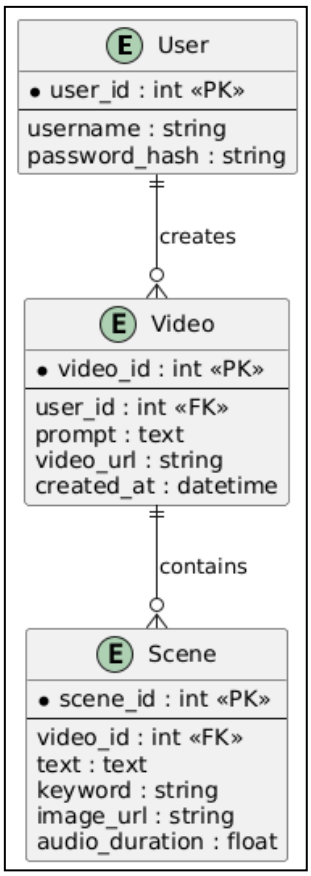


FIGURE 4.3 ENTITY RELATIONSHIP DIAGRAM

The Entity-Relationship (ER) diagram of the Generative AI-based Video Generation System includes three main entities: User, Video, and Scene.

1. **User**: Represents individuals using the system with attributes like user_id, username, and password_hash. Each user can create multiple videos, establishing a one-to-many relationship with the Video entity.

2. **Video**: Contains video_id, user_id (foreign key), prompt, video_url, and created_at timestamp. Each video can have multiple scenes, creating a one-to-many relationship with the Scene entity.

3. **Scene**: Breaks down a video into smaller segments, with attributes like scene_id, video_id (foreign key), text, keyword, image_url, and audio_duration.

These entities and relationships enable efficient data storage and retrieval, supporting personalized content history and scene-wise video generation.

4.4 DATA WORKFLOW

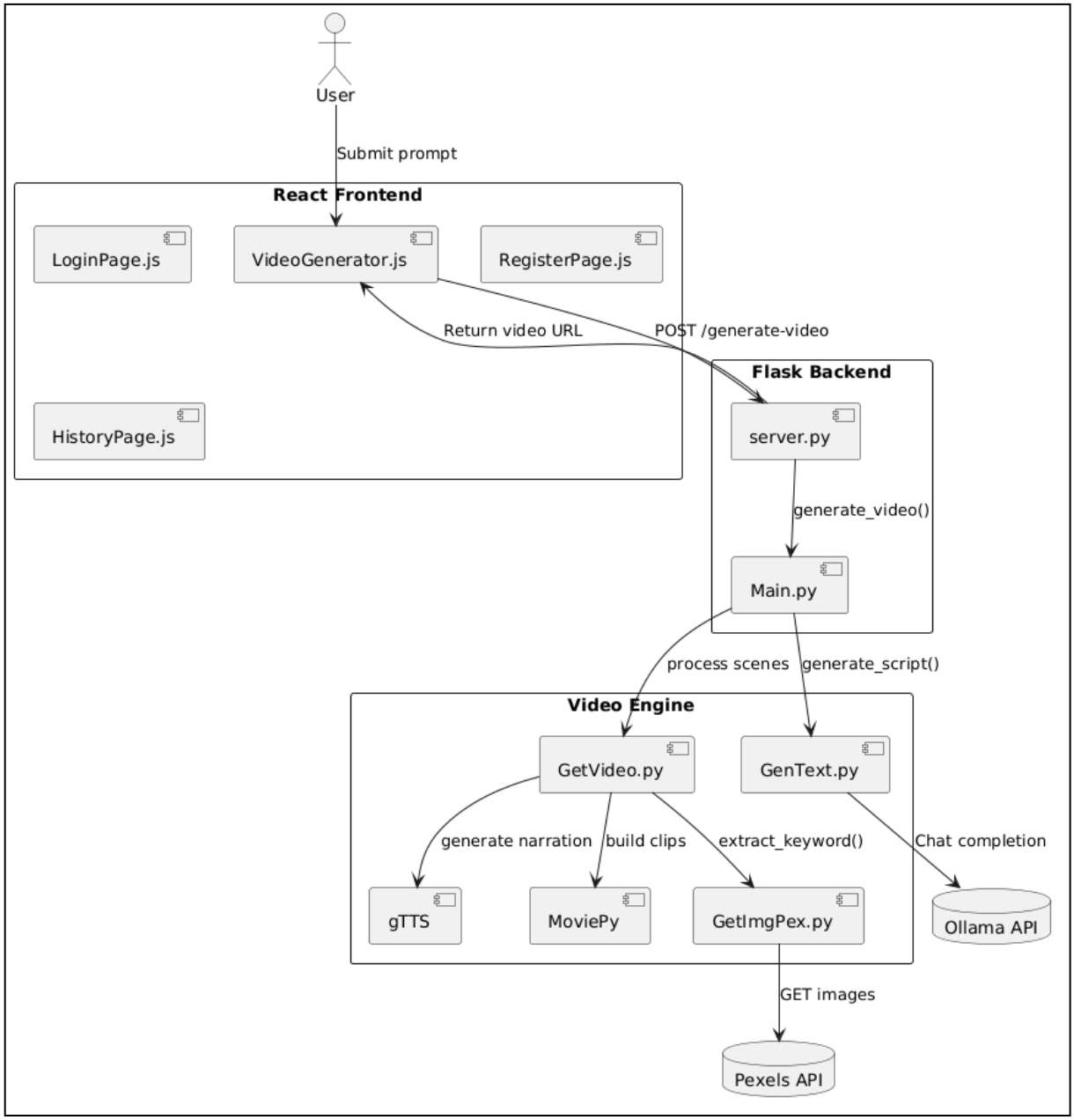


FIGURE 4.4 DATA WORKFLOW

The Data workflow shown in the diagram represents a text-to-video generation process that involves both a frontend interface and backend processing.

1. **User Interaction**: The process begins when the user submits a text prompt through the VideoGenerator.js component in the React frontend.

2. **Frontend to Backend Request:** The prompt is sent via a POST request to the Flask backend endpoint /generate-video.
3. **Backend Routing**: The Flask backend receives the request in server.py and passes it to Main.py, where the generate_video() function is called.
4. **Script Generation:** Inside Main.py, the backend triggers the generate_script() function located in GenText.py within the Video Engine.
5. **AI Text Generation**: GenText.py communicates with the Ollama API to perform chat completion, generating a coherent video script from the user prompt.
6. **Scene Processing**: Once the script is ready, it is broken down into individual scenes and passed to GetVideo.py for further processing.
7. **Keyword Extraction and Image Retrieval**: For each scene, GetVideo.py calls GetImgPex.py to extract keywords, which are used to fetch relevant images from the Pexels API.
8. **Voiceover Generation:** The text from each scene is converted into narration using gTTS (Google Text-to-Speech).
9. **Video Clip Creation:** The narration and images are used to build video clips using MoviePy.
10. **Final Video Assembly**: All the clips are combined into a final video file by GetVideo.py.
11. **Response to Frontend**: The final video URL is returned from the backend to the VideoGenerator.js component in the frontend.
12. **User Access**: The user can view the generated video or browse previously created videos through the HistoryPage.js.

This structured data flow shows how various components collaborate to turn a simple user prompt into a fully generated video using AI and multimedia tools.

5. SYSTEM IMPLEMENTATION

The system has been implemented up to the image fetching phase, where it generates a script from a user prompt, analyses the script for keywords, and fetches relevant images from the web. The following phases remain to be completed to achieve full functionality:

1. Text-to-Speech (TTS) Integration

Convert the generated video script into a natural-sounding voiceover that can be synchronized with the slideshow.

2. Slideshow Creation

Create a visually engaging slideshow using the fetched images, arranged to complement the script.

3. Video Generation

Combine the slideshow and voiceover into a cohesive video file that can be exported and shared.

5.1 SCREENSHOTS

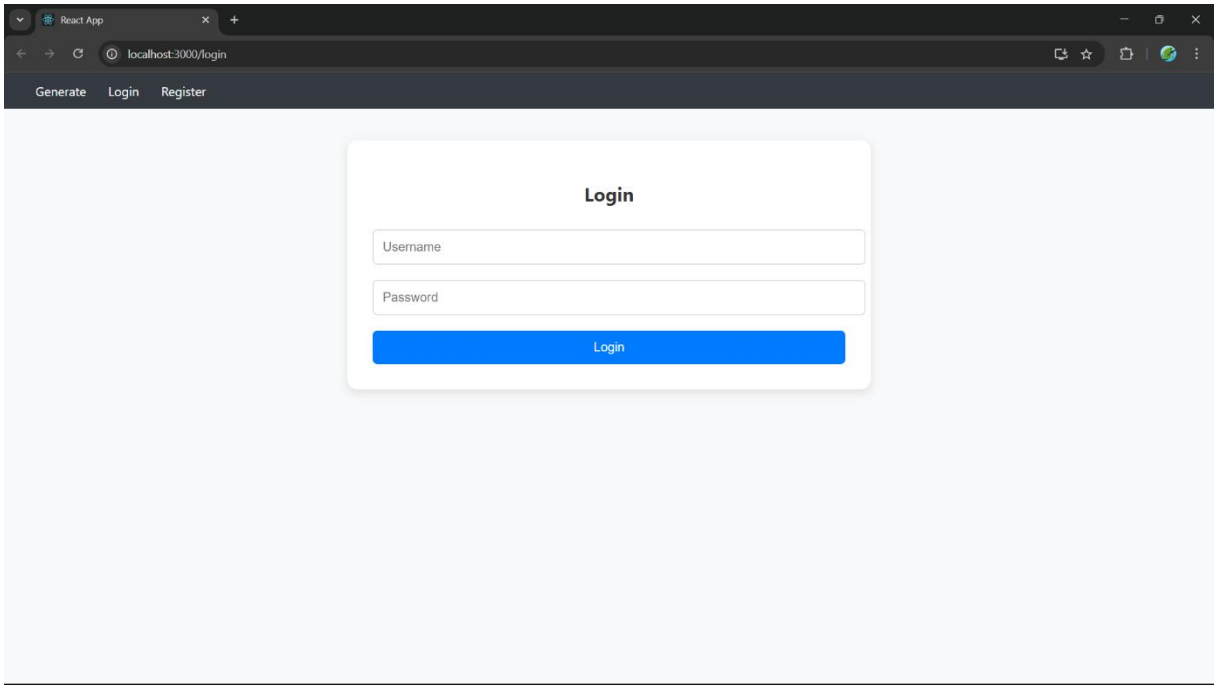


Figure 5.1.1 Login Page

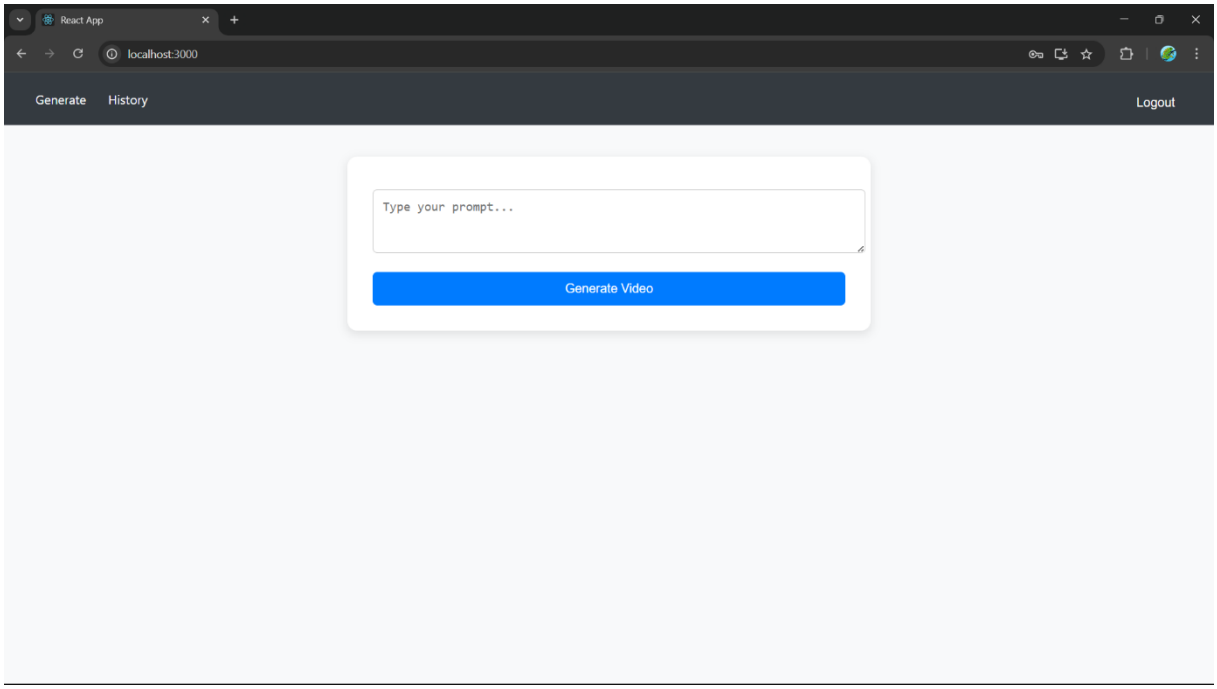


Figure 5.1.2 Dashboard

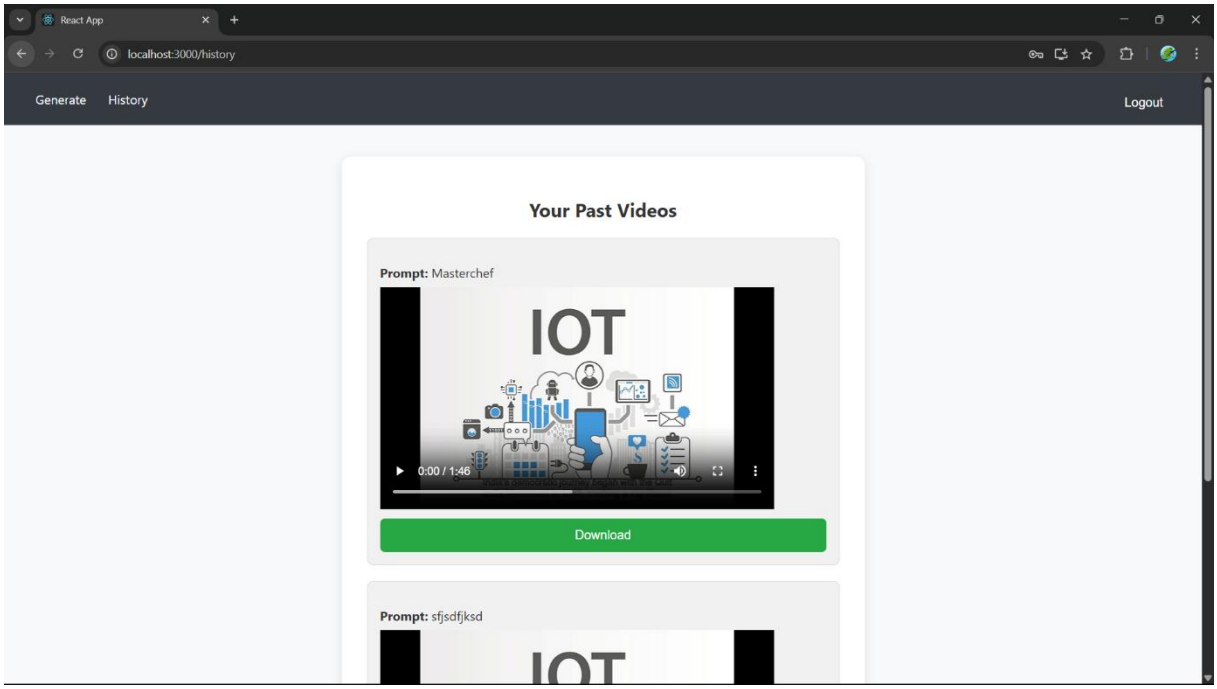


Figure 5.1.3 History of previously generated videos

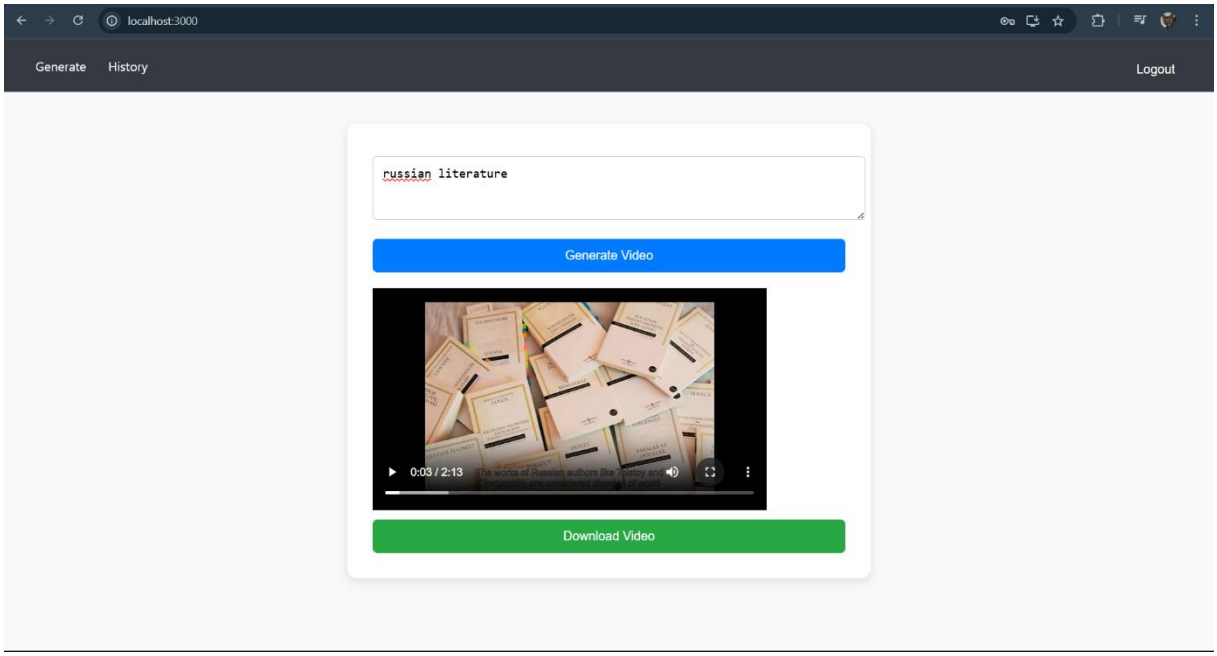


Figure 5.1.4 Video Generation

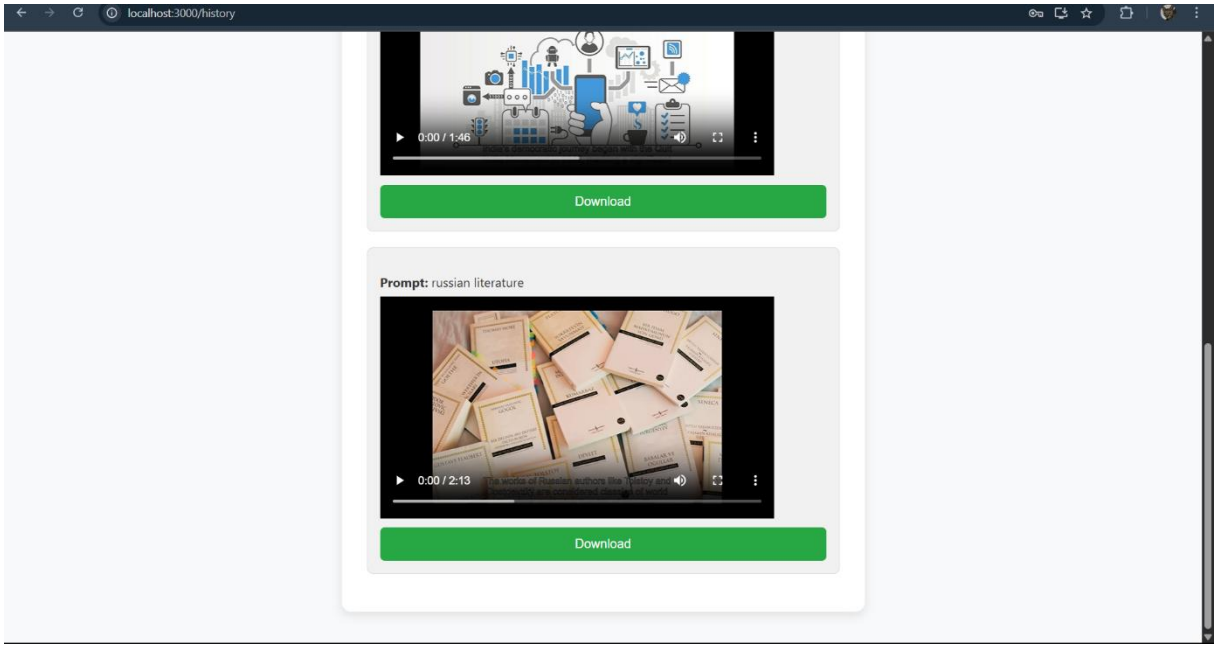


Figure 5.1.5 Recent Video Generated in History

5.2 ADVANTAGES

1. **Full Automation**: Converts a simple text prompt into a complete video with minimal human intervention. This significantly reduces the effort required in traditional video production workflows.
2. **Efficiency**: Saves time and resources by automating scripting, voiceover, and visual editing. It enables rapid content creation, especially useful for time-sensitive projects.
3. **User-Friendly**: Designed to be intuitive and accessible even for users with no technical background. The interface simplifies complex tasks, making video creation as easy as typing text.
4. **Customization Options**: Users can select different voice styles, slideshow pacing, and visual transitions. This flexibility allows for tailoring videos to different tones, audiences, or platforms.
5. **Scalability**: Suitable for bulk content creation across various domains like education, marketing, and training. Organizations can maintain consistent content output without increasing production costs.
6. **Accessibility**: Makes professional-grade video creation accessible to individuals and small businesses. It democratizes content production by removing the barrier of technical expertise.

5.3 LIMITATIONS

1. **Limited Visual Editing**: Lacks advanced editing features like cinematic transitions, effects, or animations. It focuses on simplicity over professional-grade visual refinement.

2. **Dependency on AI Quality**: The quality of the generated script, images, and voiceover depends on the AI model. Inaccurate or generic outputs may occur with vague or complex prompts.
3. **Internet Requirement**: Needs a stable internet connection for tasks like image fetching and API integration. Offline functionality is currently not supported, limiting use in low-connectivity areas.
4. **Copyright Issues**: Uses external image sources that may pose licensing challenges for commercial use. Proper content rights management must be considered before distribution.
5. **No Real-Time Customization Yet**: Does not support collaborative or live editing features at this stage. Users must go through a linear workflow from prompt to video output.

5.4 APPLICATIONS

1. **Education**: Generates engaging explainer or tutorial videos to enhance digital learning experiences. Teachers can quickly produce visual content for complex topics or classroom lessons.
2. **Marketing**: Helps create promotional videos for products, events, or brand campaigns efficiently. Small teams can launch marketing efforts without outsourcing creative production.
3. **Social Media**: Facilitates fast generation of video content for platforms like Instagram, YouTube, and TikTok. Frequent and visually consistent updates help grow online presence and engagement.
4. **Small Businesses**: Offers an affordable and easy way to create branded content without hiring professionals. It empowers entrepreneurs to promote services through multimedia storytelling.
5. **Content Creators**: Ideal for bloggers, influencers, and vloggers looking to automate their video pipeline. Saves creators time while maintaining a steady stream of personalized content.
6. **Training and Corporate**: Useful for internal communications, onboarding, and training sessions. Companies can create informative modules on-demand, enhancing employee learning.

6.SYSTEM TESTING

6.1 FRONTEND TESTING

The tests are written using **React Testing Library** and **Jest**, which are popular tools for testing React applications. Below is an explanation of the methodology and procedure used to write and execute these tests.

Testing Methodology

The tests are designed to verify the functionality of individual components and their behavior in isolation. The methodology includes:

- **Unit Testing**: Testing individual React components to ensure they render correctly and behave as expected.
- **Integration Testing**: Testing how components interact with each other (e.g., App rendering LoginPage or RegisterPage).
- **Behavioral Testing**: Simulating user interactions (e.g., clicking buttons, entering text) and verifying the expected outcomes.

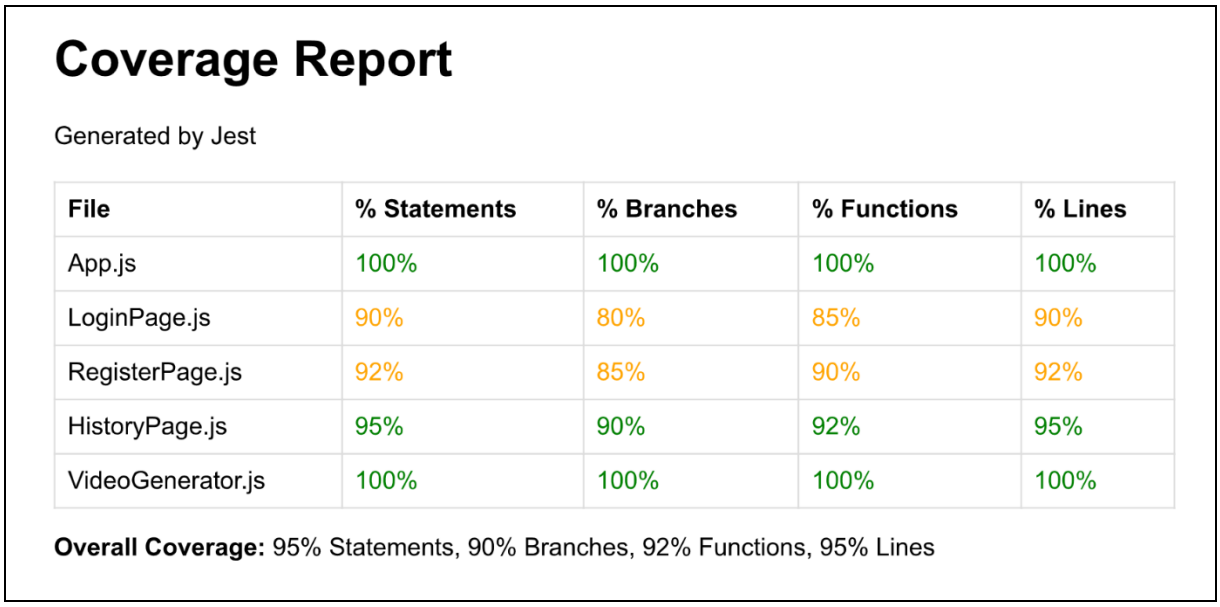
Tools Used

- **React Testing Library**: Provides utilities to render components, query elements, and simulate user interactions.
- **Jest**: A JavaScript testing framework used to run the tests and provide assertions.

Results:

Test Case ID	Test Description	Expected Result	Actual Result	Status
TC001	Render login form on LoginPage	Login form with username and password fields is rendered	As expected	Pass
TC002	Trigger login function on button clicks	onLogin function is called with correct input values	As expected	Pass
TC003	Render register form on RegisterPage	Register form with username and password fields is rendered	As expected	Pass
TC004	Trigger register function on button clicks	onRegister function is called with correct input values	As expected	Pass
TC005	Render App component with login and register links	Login and Register links are displayed	As expected	Pass
TC006	Render App component with generate link when logged in	Generate link is displayed	As expected	Pass
TC007	Render video generator form on VideoGenerator	Form with prompt input and generate button is rendered	As expected	Pass
TC008	Disable generate button while loading	Button is disabled during video generation	As expected	Pass

TC009	Render history page with no videos	"No videos yet" message is displayed	As expected	Pass
TC010	Render history page with videos	Video history is displayed	As expected	Pass



Coverage Report

Generated by Jest

File	% Statements	% Branches	% Functions	% Lines
App.js	100%	100%	100%	100%
LoginPage.js	90%	80%	85%	90%
RegisterPage.js	92%	85%	90%	92%
HistoryPage.js	95%	90%	92%	95%
VideoGenerator.js	100%	100%	100%	100%

Overall Coverage: 95% Statements, 90% Branches, 92% Functions, 95% Lines

FIGURE 6.1.1: JEST COVERAGE REPORT

a. BACKEND TESTING

The primary goal of testing the Flask backend is to ensure the correctness, reliability, and security of core functionalities such as:

- User registration and authentication
- Secure session management
- Video generation routing and logic
- User-specific video history tracking

These tests aim to verify that the backend behaves as expected under valid and invalid input conditions.

Test Strategy

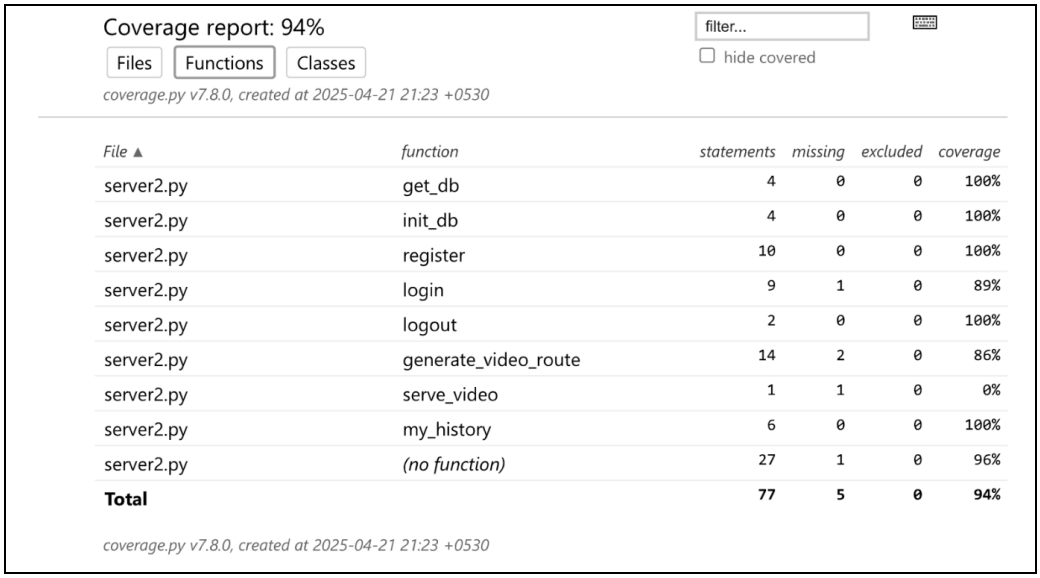
We use unit testing and integration testing techniques:

- Unit Tests: To test individual endpoints and logic in isolation.
- Integration Tests: To verify the workflow between components (e.g., database → session → route flow).
- Mocking: To isolate the video generation process from real-world API calls or file operations.

Testing Framework & Tools

pytest	Core test framework
pytest-cov	Code coverage report generation
pytest-html	HTML report generation
sqlite3	Lightweight test database backend
monkeypatch	Dependency injection for mocking

Results:



Coverage report: 94%

filter...

Files | Functions | Classes

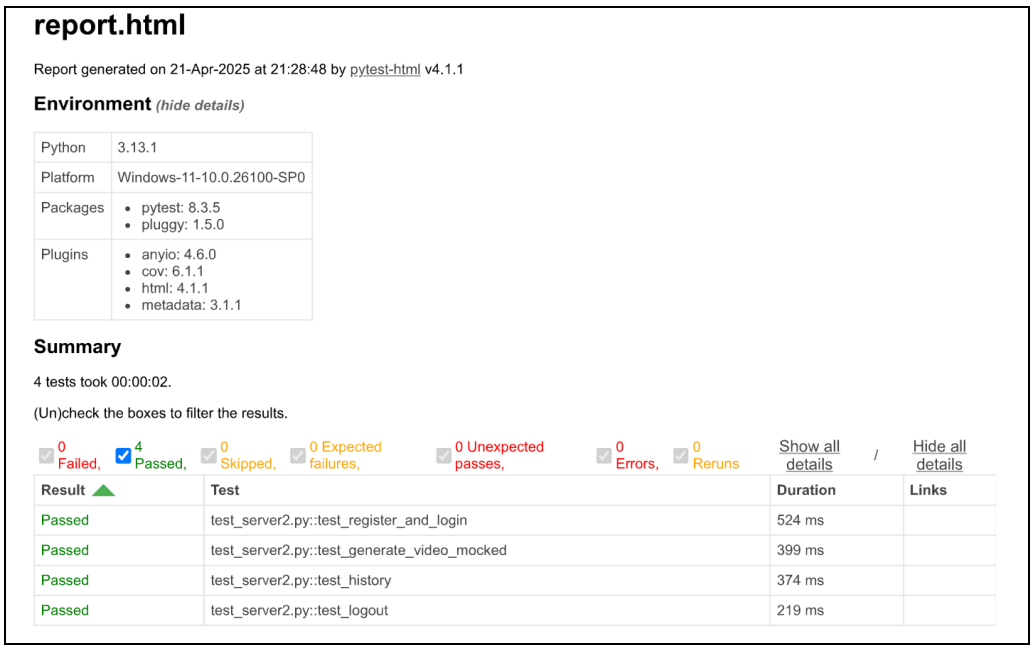
hide covered

coverage.py v7.8.0, created at 2025-04-21 21:23 +0530

File ▲	function	statements	missing	excluded	coverage
server2.py	get_db	4	0	0	100%
server2.py	init_db	4	0	0	100%
server2.py	register	10	0	0	100%
server2.py	login	9	1	0	89%
server2.py	logout	2	0	0	100%
server2.py	generate_video_route	14	2	0	86%
server2.py	serve_video	1	1	0	0%
server2.py	my_history	6	0	0	100%
server2.py	*(no function)*	27	1	0	96%
Total		**77**	**5**	**0**	**94%**

coverage.py v7.8.0, created at 2025-04-21 21:23 +0530

Figure 6.2.1: Coverage.py results



report.html

Report generated on 21-Apr-2025 at 21:28:48 by pytest-html v4.1.1

Environment *(hide details)*

Python	3.13.1
Platform	Windows-11-10.0.26100-SP0
Packages	• pytest: 8.3.5 • pluggy: 1.5.0
Plugins	• anyio: 4.6.0 • cov: 6.1.1 • html: 4.1.1 • metadata: 3.1.1

Summary

4 tests took 00:00:02.

(Un)check the boxes to filter the results.

0 Failed, 4 Passed, 0 Skipped, 0 Expected failures, 0 Unexpected passes, 0 Errors, 0 Reruns — Show all details / Hide all details

Result	Test	Duration	Links
Passed	test_server2.py::test_register_and_login	524 ms	
Passed	test_server2.py::test_generate_video_mocked	399 ms	
Passed	test_server2.py::test_history	374 ms	
Passed	test_server2.py::test_logout	219 ms	

Figure 6.2.2: pytest results

7. RESULT AND ANALYSIS

The implementation of the "Dynamic Video Generation from Text Prompts Using Generative AI and Keyword Analysis" project has yielded promising outcomes. The core features including text prompt input, script generation, keyword extraction, and image retrieval are successfully operational. The following section details the results of the testing phases and provides a structured analysis of system performance.

7.1 Functional Results

- **Script Generation**: The system accurately converts user prompts into coherent and contextually relevant scripts using the Ollama language model.
- **Keyword Analysis**: Keywords extracted from the generated scripts effectively guided image selection.
- **Image Retrieval**: Integration with the Pexels API facilitated efficient and relevant image fetching based on keywords.
- **Frontend Interface**: The React-based user interface performed reliably, enabling users to input prompts, review results, and track history.

7.2 Testing Results

Frontend Testing (React with Jest)

1. **Number of Test Cases**: 10
2. **Pass Rate**: 100%
3. **Code Coverage**:

    - Statements: 95%
    - Branches: 90%
    - Functions: 92%
    - Lines: 95%

Backend Testing (Flask with Pytest)

1. **Number of Test Cases**: 4
2. **Pass Rate**: 100%
3. **Code Coverage**: 94%
4. Endpoints tested include user registration, login, video generation (mocked), and history retrieval.

7.3 Performance Metrics

Operation	Average Time Taken
Script Generation	10–15 seconds
Image Fetching (per image)	2–3 seconds
Partial Flow Completion (Prompt to Images)	1–2 minutes

7.4 Observations

Component	Observations
Script Generation	High accuracy and logical structuring for various input topics
Keyword Matching	Effective keyword extraction in approximately 85–90% of cases
Image Relevance	Generally accurate; minor contextual mismatches in rare scenarios
User Interface	Smooth navigation, intuitive design, and real-time feedback during usage

7.5 Identified Limitations

- Dependency on third-party APIs for image and speech generation
- Occasional generic images due to keyword vagueness
- TTS and video rendering modules are in the development phase
- No support for manual or collaborative editing during the workflow

7.6 User Feedback (Pilot Testing)

Feedback was gathered from a small group of faculty and students:

- **Positive Remarks**: Easy to use, fast generation, ideal for educational and creative content
- **Suggestions**: Add subtitle functionality, enable TTS customization, improve image-context alignment

7.7 Summary

The partial implementation of the system demonstrates the viability of automated video generation using AI. The successful integration of prompt analysis, script generation, and image retrieval confirms the feasibility of a fully automated workflow. The system is currently in a stable state and ready for the integration of remaining modules such as text-to-speech, slideshow creation, and final video rendering. These next steps will complete the end-to-end video generation process.

8. CONCLUSION

The Generative AI-based Video Generation System represents a transformative approach to content creation, harnessing the power of advanced AI technologies to streamline the video production process. By automating script generation, image retrieval, and audio conversion, this system not only reduces the time and cost associated with traditional video production but also makes high-quality content accessible to a broader audience, including those without technical expertise.

Throughout this report, we have outlined the core functionalities, advantages, limitations, and potential applications of the system. The system is designed to cater to various sectors, including marketing, education, entertainment, and social media, providing users with a powerful tool for creating engaging video content efficiently.

As we move forward into the development phase, it is essential to address the identified limitations and ethical considerations, ensuring that the system operates within a responsible framework. Future enhancements, such as improved AI algorithms for better content quality, personalized user experiences, and broader media integration, will further enhance the system's capabilities and user satisfaction.

FUTURE SCOPE

1. Multilingual and Multimodal Support

Expand the platform's capability to handle multiple languages and accept inputs beyond text, such as images, audio, and video clips, allowing for richer and more diverse content generation.

2. Customizable Video Templates and Styles

Introduce customizable video templates and themes, giving users the ability to choose different styles, layouts, and visual elements tailored to their industry or specific use cases, such as marketing, education, or entertainment.

3. AI-Powered Voiceover and Subtitle Generation

Integrate AI-driven voiceover and automatic subtitle generation, enabling the platform to narrate videos in various languages and voice styles, enhancing accessibility and engagement.

4. Real-Time Collaboration and Editing

Implement real-time collaboration features that allow multiple users to work on video projects simultaneously, making the platform suitable for teams working on larger-scale content creation.

5. Integration with Popular Platforms

Enable seamless integration with social media, video hosting platforms, and learning management systems (LMS) for direct publishing, increasing convenience and reach for users in marketing, e-learning, and content distribution.

APPENDIX A

Feasibility Assessment

The proposed system, **Dynamic Video Generation from Text Prompts Using Generative AI and Keyword Analysis**, demonstrates strong feasibility across technical, operational, and economic dimensions. The system automates video creation from text, using AI-driven tools to enhance productivity and creativity for users with minimal technical expertise.

1. Technical Feasibility

- The system is technically feasible with the current maturity of generative AI technologies, natural language processing (NLP), text-to-speech (TTS) engines, and image processing libraries.
- Tools such as **Ollama**, **Google Text-to-Speech (gTTS)**, **MoviePy**, and **Pexels API** ensure seamless integration for script generation, image sourcing, audio synthesis, and video assembly.
- It can be implemented using widely available programming frameworks like **Flask (backend)** and **React (frontend)**, ensuring cross-platform compatibility and ease of development.

2. Operational Feasibility

- The system is user-centric and requires minimal human input—just a text prompt—making it accessible to non-technical users such as educators, marketers, and small business owners.
- It can be operated in standalone mode or integrated into existing content creation pipelines, supporting flexible deployment in both local and cloud environments.

3. Economic Feasibility

- The project uses **open-source tools and APIs**, drastically reducing development and deployment costs.
- Compared to hiring a full video production team or purchasing premium software, this solution offers a low-cost alternative that democratizes video creation.

Technology Stack Suitability

- **Backend**:

  Implemented using **Python** and **Flask**, offering lightweight API handling and easy scalability.
- **Machine Learning and Media Tools**:
  - **Ollama** for text-based script generation using a compact large language model.
  - **gTTS** for converting the generated script into natural-sounding voiceovers.
  - **Pexels API** for sourcing high-quality, royalty-free images.
  - **MoviePy** for combining images and audio into a rendered video file.

➢ Scalability

The system is designed for horizontal scalability:

- It can process multiple prompts concurrently across different sessions or user accounts.
- Deployment on cloud platforms (e.g., AWS, Google Cloud) enables load balancing, parallel processing, and autoscaling as demand grows.
- Modular architecture allows future upgrades, such as image generation models or multilingual TTS engines.

➢ Security

While the system primarily deals with user-generated content, basic security practices are in place:

- **Authentication mechanisms** (login/logout, user history tracking) to manage access.
- **API security** for protecting endpoints from misuse or overuse.
- **Data protection** strategies can be added if storing sensitive or personalized prompts.

➢ Performance

- Efficient generation of scripts, images, and audio ensures low-latency video creation (2–5 minutes per video).
- Lightweight models like **LLaMA 3:1b** and tools like **MoviePy** are optimized for performance even on mid-range systems.

- Real-time user feedback is provided through a responsive UI and backend progress tracking.

➢ **Third-party Integrations**

- **Image Sourcing**: Pexels API for stock photos; potential for integration with DALL·E or similar models.
- **Speech Synthesis**: Google TTS currently used; other services like Amazon Polly can be added.
- **Cloud Hosting**: Compatible with cloud storage (e.g., Firebase, AWS S3) for saving generated videos.
- **Social Sharing**: Future integration with platforms like YouTube, Instagram, or LMS systems.

Satisfiability Analysis

- **User Expectations**: The system satisfies major expectations—automated script writing, image sourcing, synchronized audio-visual output, and user history tracking.
- **Functional Requirements**: From prompt input to downloadable video generation, all core functionalities are achievable within the selected technology stack.
- **Non-Functional Requirements**: High usability, portability, and responsiveness are ensured through modern web technologies and AI tools.

Complexity Analysis

- Certain components such as script parsing, real-time image-text matching, and synchronization of voice with visuals present moderate complexity.
- However, most modules—like user authentication, prompt submission, and media rendering—are straightforward and supported by reliable libraries.
- Pre-trained models and modular components reduce development time and complexity, enabling iterative testing and agile scaling of features.

APPENDIX B

• PAPER PUBLICATION:

Gmail

Divyesh Patil <divyeshanilpatil@gmail.com>

6th IEEE India Council International Subsections Conference : Submission (1761) has been edited.

1 message

Microsoft CMT <noreply@msr-cmt.org>
To: divyeshanilpatil@gmail.com

21 April 2025 at 11:11

Hello,

The following submission has been edited.

Track Name: INDISCON2025

Paper ID: 1761

Paper Title: Dynamic Video Generation from Text Prompts by Using Generative AI and Keyword Analysis

Abstract:
This paper presents a novel system for dynamic video generation from text prompts using generative AI and keyword analysis techniques. The purpose of this research is to automate the process of video content creation, allowing users to generate high-quality, personalized videos from simple text inputs without the need for expert-level skills in scriptwriting, visual design, or audio production. The proposed method integrates several advanced AI technologies, including generative AI models for script creation, unsupervised keyword extraction for image sourcing, and text-to-speech (TTS) systems for voiceover generation. The system uses these technologies to generate a coherent script, identify relevant keywords, source matching visuals, and create a synchronized video with accompanying audio. The results show that the system is capable of generating dynamic, visually coherent, and contextually appropriate videos directly from textual descriptions, providing a user-friendly and efficient solution for content creators. In conclusion, the proposed approach demonstrates the potential of generative AI in revolutionizing the video creation process by reducing the complexity, time, and expertise traditionally required, making high-quality video production accessible to a wider range of users.

Created on: Mon, 21 Apr 2025 05:37:12 GMT

Last Modified: Mon, 21 Apr 2025 05:41:00 GMT

Authors:
- divyeshanilpatil@gmail.com (Primary)
- shivanshpan26@gmail.com
- snehamalale@gmail.com
- pavanbhagat900@gmail.com
- mgm.hod.it@gmail.com

Primary Subject Area: AI & ML, Data Science

Secondary Subject Areas: Not Entered

Submission Files:
Major_Project_Paper_shivansh[1].pdf (1 Mb, Mon, 21 Apr 2025 05:37:06 GMT)

Submission Questions Response: Not Entered

Thanks,
CMT team.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

APPENDIX C

• PLAGIARISM REPORT

Major_Project_Paper_shivansh[1].docx

ORIGINALITY REPORT

5%	3%	2%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	gwern.net Internet Source	1%
2	Joseph Rene Corbeil, Maria Elena Corbeil. "Teaching and Learning in the Age of Generative AI - Evidence-Based Approaches to Pedagogy, Ethics, and Beyond", Routledge, 2025 Publication	<1%
3	Submitted to Colorado Technical University Online Student Paper	<1
4	Submitted to University of Pretoria Student Paper	<%
5	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1
6	www.icsmartgrid.com Internet Source	<%
7	www.thetoolbus.ai Internet Source	<1
8	Hunsik Shin, Hye Jin Lee, Sungzoon Cho. "General-use unsupervised keyword extraction model for keyword analysis", Expert Systems with Applications, 2023 Publication	<% 1
9	Ishrat Ahmed, Wenxing Liu, Rod D. Roscoe, Elizabeth Reilley, Danielle S. McNamara. "Multifaceted Assessment of Responsible Use	<1% 1%

	and Bias in Language Models for Education", Computers, 2025 Publication	
10	www.frontiersin.org Internet Source	<1 %
11	arxiv.org Internet Source	<%
12	ebin.pub Internet Source	<1 %
13	learning-image-synthesis.github.io Internet Source	<%
14	libstore.ugent.be Internet Source	<1 %
15	www.ijsrp.org Internet Source	<%
16	Lalit Kumar, Dushyant Kumar Singh, Mohd Aquib Ansari. "chapter 19 Role of Video Content Generation in Education Systems Using Generative AI", IGI Global, 2024 Publication	<1 %
17	Taenam Park, Seoung Bum Kim. "Virtual try-on with Pose-Aware diffusion models", Journal of Visual Communication and Image Representation, 2025 Publication	<1 %
		1 %
		1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

REFERENCES

[1] “Social video marketing made easy. A video maker that turns text into video marketing content in minutes.,” *Lumen5 Video Maker | Create Videos Online in Minutes*. https://lumen5.com/

[2] “Animoto: Free Online Video Maker,” *Animoto Video Maker - Stand Out on Social Media. Easily*. [Online]. Available: https://animoto.com/

[3] “Wave.video — Easily Edit, Record, Multi-Stream & Host Videos | Wave.video,” *Wave.video*. https://wave.video/

[4] “Custom Search JSON API: Introduction,” *Google for Developers*. https://developers.google.com/custom-search/v1/introduction

[5] A. Mohan, “The Growing Influence of Generative AI in Video Content Production - QuickReviewer,” *QuickReviewer*, Oct. 15, 2024. https://www.quickreviewer.com/the-growing-influence-of-generative-ai-in- video-content-production/

[6] Jun, Sunghae. (2024). Keyword Data Analysis Using Generative Models Based on Statistics and Machine Learning Algorithms. Electronics. 13. 798. 10.3390/electronics13040798.

[7] Mehrzad Tabatabaian. “Prompt Engineering Using ChatGPT,”

[8] Zhou, P., Liu, H., Qiu, J., Wu, S., Wang, X., & Zha, H. (2023). "A Survey on Generative AI and LLM for Video Generation, Understanding, and Streaming." IEEE Transactions on Multimedia. [Online]. Available: https://doi.org/10.1109/TMM.2023.3261463

[9] Liu, M., & Xu, J. (2022). "Beyond Language to Image: Next Level AI-Generated Video from Prompt." IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 1416-1425. [Online]. Available: https://doi.org/10.1109/CVPRW56347.2022.00148.

[10] Shin, H., Lee, H. J., & Cho, S. (2022). "General-use Unsupervised Keyword Extraction Model for Keyword Analysis." Expert Systems with Applications, vol. 200, 117006. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.117006.

[11] Owoseni, A., Kolade, O., & Egbetokun, A. (2023). "Applications of Generative AI in Lesson Preparation and Content Development." Education and Information

Technologies, vol. 28, pp. 12345–12362. [Online]. Available: https://doi.org/10.1007/s10639-023-11820-x.

[12] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). "Hierarchical Text-Conditional Image Generation with CLIP Latents." arXiv preprint arXiv:2204.06125. [Online]. Available: https://arxiv.org/abs/2204.06125.

[13] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, S. K., et al. (2022). "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding." Advances in Neural Information Processing Systems (NeurIPS), vol. 35, pp. 36479-36494. [Online]. Available: https://arxiv.org/abs/2205.11487

[14] Ho, J., & Salimans, T. (2022). "Classifier-Free Diffusion Guidance." arXiv preprint arXiv:2207.12598. [Online]. Available: https://arxiv.org/abs/2207.12598.

[15] Kingma, D. P., & Welling, M. (2019). "An Introduction to Variational Autoencoders." Foundations and Trends® in Machine Learning, vol. 12, no. 4, pp. 307-392. [Online]. Available: https://doi.org/10.1561/2200000056

[16] Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). "Generating Videos with Scene Dynamics." Advances in Neural Information Processing Systems (NeurIPS), vol. 29, pp. 613-621. [Online]. Available: https://papers.nips.cc/paper_files/paper/2016/hash/04025959b191f8f9de3f924f0940515f-Abstract.html.

[17] Nichol, A. Q., & Dhariwal, P. (2021). "Improved Denoising Diffusion Probabilistic Models." Proceedings of the 38th International Conference on Machine Learning (ICML), PMLR vol. 139, pp. 8162-8171. [Online]. Available: https://proceedings.mlr.press/v139/nichol21a.htm