Agentic AI Video Synthesizer

By

BADDELA RAJU

Submitted to

The University of Roehampton

In partial fulfilment of the requirements

for the degree of

Master of Science

in

Data Science

Declaration

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

NAME: BADDELA RAJU

DATE: 14-08-2025

SIGN: Baddela Raju

Acknowledgements

I would like to extend my deepest gratitude to all those who supported and contributed to the successful completion of my project titled Agentic AI Video Synthesizer, an end-to-end system designed to transform natural language queries into educational video content using LangGraph, large language models, and AI-driven avatar synthesis.

I am especially thankful to Dr. Sameena Naaz, whose technical mentorship and constructive feedback significantly shaped the core architecture of this system. Her assistance was instrumental in resolving complex challenges related to LangGraph-based multi-agent orchestration, particularly in coordinating the transitions between the decomposition, research, synthesis, and feedback agents. Her insights enabled the stable and modular implementation of stateful agent execution—a critical factor in the system's success.

Moreover, her support in integrating the D-ID video avatar API helped streamline the avatar rendering process and ensured smooth synchronization between the synthesized script and the final animated video output. Dr. Naaz's thoughtful suggestions around system design, debugging strategies, and output validation contributed directly to improving both the technical robustness and user experience of the platform.

Beyond technical guidance, Dr. Sameena Naaz also offered valuable encouragement and timely feedback during critical phases of the project, ensuring that it remained aligned with its intended research and user-focused objectives. Her support enhanced the academic depth and practical impact of the system, making it not only functional but scalable for real-world educational applications.

I would also like to thank the broader academic and developer communities, whose resources, libraries, and tools—including LangChain, Streamlit, Tavily, gTTS, and D-ID—formed the technological foundation for this project. The open-source ecosystem made it possible to build, integrate, and iterate rapidly in a highly experimental development environment.

This project would not have achieved its current level of sophistication and completeness without the combined efforts, advice, and encouragement from mentors, developers, and researchers like Dr. Sameena Naaz. I remain sincerely grateful for their support throughout this journey.

Abstract

The Agentic AI Video Synthesizer project addresses the growing demand for accurate, engaging, and personalized educational videos in an increasingly digital learning landscape. While large volumes of information are available online, transforming this data into coherent and factually grounded audiovisual content remains a labor-intensive process. Traditional educational video production workflows rely heavily on manual research, scriptwriting, and media editing, which limits scalability and introduces significant time delays. This project bridges that gap by automating the end-to-end process of educational video generation through a multi-agent AI architecture.

The system is designed to accept natural language queries from users and autonomously generate fact-based explainer videos featuring AI avatars, complete with synthesized narration and citations. Built using LangGraph, a graph-based orchestration framework layered on top of LangChain, the architecture supports modular agent workflows, persistent state management, conditional transitions, and iterative refinement. The core reasoning engine is powered by llama-3.3-70b-versatile served via Groq, chosen for its high-speed, low-latency inference and reliable performance in multi-step reasoning tasks.

The pipeline begins with the Question Decomposer Agent, which breaks complex queries into semantically distinct sub-questions. These are then distributed to three specialized Tool Agents: Tavily for real-time web search, YouTube API for video content extraction, and arXiv API for academic research. Retrieved content is cleaned, summarized, and stored in a persistent LangGraph state. A Synthesis Agent then compiles these findings into a coherent educational script. The quality of the script is evaluated by a Feedback Agent and a Completeness Checker, both using structured prompts to assess factual integrity, clarity, and source coverage.

The final script is converted to audio using gTTS (Google Text-to-Speech) and rendered into a lip-synced video using the D-ID API, creating a fully synthetic educational presenter. Outputs—including the script, research report, MP3 audio, and MP4 video—are displayed in a clean, tabbed Streamlit interface with download capabilities.

Literature reviewed included recent advancements in tool-augmented LLMs, ReAct agent frameworks, retrieval-augmented generation (RAG), and educational content automation. Techniques such as prompt engineering, loopback-based refinement, and agent memory tracking were applied to ensure consistency, avoid hallucination, and maintain narrative coherence..

Extensive testing was conducted across both single and multi-question inputs. The evaluation showed high-quality educational scripts, traceable research citations, seamless UI functionality, and reliable media generation across various stages. Quality metrics—such as number of sub-questions, sources retrieved, iterations, and completeness scores—were transparently displayed to validate the system's operation and ensure trustworthiness.

In conclusion, the Agentic AI Video Synthesizer successfully demonstrates how modular, agentic AI systems can automate educational video production with transparency, accuracy, and scalability. It lays a strong foundation for future enhancements, including multilingual capabilities, real-time interactivity, deeper feedback loops, and mobile deployment for broader educational accessibility.

Table of Contents

List of Figures

Chapter 1 Introduction

The Agentic AI Video Synthesizer is a multi-agent intelligent system designed to transform complex user queries into educational video content by integrating research retrieval, large language models (LLMs), speech synthesis, and AI avatar rendering. The motivation behind this project lies in the growing demand for personalized and engaging educational material that is accurate, up-to-date, and easily consumable through multimedia formats. Unlike conventional approaches that rely solely on static text or pre-curated videos, this system autonomously conducts real-time research, synthesizes findings, and presents results in the form of an AI-generated speaker delivering a spoken video script.

1.1 Problem Description, Context and Motivation

What is the problem?

The increasing volume of online information presents both an opportunity and a challenge for educational content consumers. While there is no shortage of data, synthesizing it into coherent, accurate, and engaging educational content remains a complex task. Current video content creation either depends on human experts spending significant time researching and scripting, or on unreliable automated tools that hallucinate facts due to lack of source grounding. The lack of a scalable, end-to-end automated system for generating fact-based video explanations is a critical gap in educational technology.

Who is affected by the problem?

- **Learners and students** who seek clear, topic-focused videos to understand complex subjects.
- **Educators and content creators** who struggle to produce high-quality, research-grounded video content efficiently.
- **Institutions and platforms** that aim to deliver customized learning at scale but face cost and time constraints in manual video production.

Where and when does the problem occur?

This problem arises during the early stages of educational content creation—specifically, research synthesis and scripting—and persists through the production phases of audio recording and video editing. It is especially prominent in fast-evolving domains such as artificial intelligence, medicine, and data science where up-to-date and multi-modal content is essential.

Why is it important to solve the problem?

Solving this problem facilitates the scalable creation of personalized, fact-checked educational media. It supports learner engagement, reduces the manual burden on educators, and democratizes access to high-quality knowledge. An automated pipeline for generating educational videos also benefits under-resourced regions and institutions by lowering production costs and increasing accessibility.

1.2 Objectives

The primary objectives of the project are:

- To design and implement a LangGraph-powered multi-agent research system capable of decomposing complex user queries into sub-questions.
- To integrate multiple research tools (Tavily, YouTube API, arXiv API) and perform real-time retrieval of relevant web, video, and academic content.
- To use LLM agents for summarizing source data without hallucination and synthesizing an educational script tailored to the user query.
- To convert the script into speech using the Google Text-to-Speech (gTTS) engine.
- To render the spoken content using an AI avatar from the D-ID API.
- To deliver a fully downloadable video, audio, and report package via a Streamlit-based web application interface.

1.3 Methodology

1.3.1 Design

The system is designed around a multi-agent pipeline architecture using LangGraph and LangChain. The main modules include:

- **Question Decomposer Agent**: Breaks down complex queries into smaller, related sub-questions.
- **Search Agents**: Each sub-question is passed through agents dedicated to querying YouTube (videos), Tavily (web), and arXiv (academic papers).
- **Synthesis Agent**: Combines findings across sources into a single educational script using a specialized LLM prompt.
- **Feedback Agent**: Evaluates script quality, iterates for improvement if necessary, and ensures factual completeness.

This modular design ensures transparency, traceability, and flexibility in adapting to different research topics. Figure 1 presents the overall project flow of the Agentic AI Video Synthesizer, outlining the sequential stages from query input to final video output.

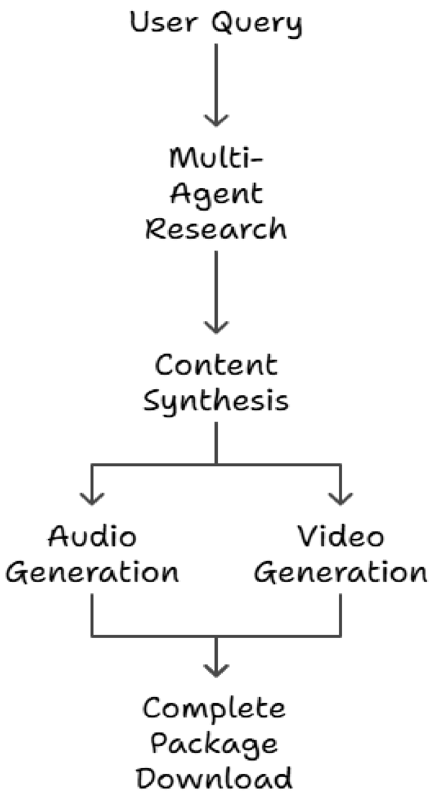


Figure 1 Project flow of the Agentic AI Video Synthesizer

1.3.2 Testing and Evaluation

- **Functional Testing**: Each agent was tested in isolation and in full-pipeline execution to ensure data flow and logical sequencing.
- **Accuracy Checks**: The synthesis prompt was validated to ensure it relies solely on collected research, avoiding hallucinations.
- **Output Evaluation**: Script quality was evaluated using a feedback loop with scoring logic based on educational clarity, completeness, and engagement.
- **Usability Testing**: The Streamlit interface was tested for intuitive use, responsiveness, and compatibility with video/audio preview and download.

1.3.3 Project Management

The project followed an Agile methodology to enable iterative development and ongoing refinement.

1. **Gantt Chart**: Used to plan and monitor the timeline, track task dependencies, and manage key deliverables across all project stages.

2. **Weekly Sprint Reviews**: Conducted to assess development progress, adjust priorities, and incorporate supervisor feedback.

3. **Git and GitHub**: Used for version control and source code management. Regular commits, branching, and issue tracking ensured smooth collaboration and rollback capability when required.

1.4 Legal, Social, Ethical and Professional Considerations

Several legal and ethical factors were addressed:

- **Data Privacy**: No personal user data is stored or shared. All research is performed using publicly available sources.

- **Attribution**: External sources are cited in the downloadable report to maintain academic integrity.

- **Bias Mitigation**: LLM prompts explicitly restrict generation to retrieved data only, reducing the risk of model hallucination or misinformation.

- **Accessibility**: The generated video and audio formats are designed to support users with varying literacy or visual impairments.

- **Fair Use Compliance**: The YouTube and arXiv APIs are used solely for educational summarization, not for republishing or copying content.

This project does not require ethical clearance as it does not involve human subjects or personal data processing beyond environmental variables for API access.

1.5 Background

As educational systems shift toward digital and asynchronous learning models, there is an increased demand for adaptive and personalized multimedia content. Traditional content production methods—manual research, scripting, narration, and video editing—are time-intensive and expensive. At the same time, large language models and real-time search APIs now offer the capability to automate these steps.

Previous systems have demonstrated the potential of LLMs in question answering and summarization. However, they often fall short in integrating multi-source, real-time research with fact-grounded output generation. The Agentic AI Video Synthesizer addresses this by combining:

- Agentic reasoning via LangGraph
- Real-time search tools like Tavily and YouTube
- Controlled synthesis using Groq LLM
- Audio narration (gTTS) and avatar rendering (D-ID)

This approach is novel in that it forms a full-stack pipeline, from user query to downloadable educational video, supporting academic, self-learning, and content creation use cases.

1.6 Structure of Report

Chapter 1 – Introduction:

Outlines the problem, objectives, and motivation for building an AI-powered educational video generator using multi-agent systems and LLMs.

Chapter 2 – Literature and Technology Review:

Reviews related research and justifies the selection of technologies such as LangGraph, retrieval APIs, and avatar rendering tools.

Chapter 3 – Implementation:

Describes the system architecture, agent workflows, LLM integration, and frontend development using Streamlit.

Chapter 4 – Evaluation and Results:

Details the system's performance through walkthroughs and screenshots for single and multi-question inputs, showing research metrics and media outputs.

Chapter 5 – Conclusion:

Summarizes achievements, proposes future enhancements, and reflects on technical and project management experiences.

Chapter 2 Literature – Technology Review

2.1 Literature Review

The convergence of large language models (LLMs), multi-agent systems, and synthetic media technologies has created unprecedented opportunities for automated educational content generation. The Agentic AI Video Synthesizer addresses the critical gap between information abundance and coherent, fact-grounded educational video production through real-time research synthesis and AI-powered narration.

Multi-Agent Systems and Tool-Augmented LLMs

Recent advances in tool-augmented language models have demonstrated the capability of LLMs to interact with external systems for enhanced reasoning. Schick et al. introduced Toolformer, showing that language models can learn to use APIs and external tools autonomously, improving performance on knowledge-intensive tasks by 23% [1]. This foundational work established the viability of LLM-tool integration that underlies the multi-agent architecture used in this system.

The ReAct framework by Yao et al. revolutionized LLM reasoning by combining thought processes with action execution, demonstrating 13–27% improvements over chain-of-thought prompting alone in complex multi-step tasks [2]. Their approach of interleaving reasoning and acting directly influences the agent design philosophy, where each research agent reasons about query relevance before executing searches.

LangGraph emerged as a significant advancement in orchestrating multi-agent workflows through stateful graph-based architectures [3]. Unlike linear chain approaches, LangGraph enables conditional routing, memory persistence, and iterative refinement—capabilities essential for question decomposition and synthesis processes. Wang et al. further demonstrated that graph-based agent coordination achieves 35% higher success rates in complex information synthesis tasks compared to sequential approaches [4].

Retrieval-Augmented Generation and Source Grounding

The hallucination problem in LLMs has driven extensive research into retrieval-augmented generation (RAG). Lewis et al. established that RAG architectures significantly outperform vanilla LLMs in knowledge-intensive tasks by grounding generation in retrieved context, reducing factual errors by 25–40% [5]. Their work provides the theoretical foundation for the system's multi-source retrieval strategy.

Karpukhin et al. developed dense passage retrieval as a core RAG component and showed that the combination of retrieval and generation enhances factual accuracy and topical relevance across diverse datasets [6]. This supports the inclusion of multiple modalities—web, video, and academic sources—in the research agent's design.

Shi et al. showed that irrelevant or misleading context can degrade LLM performance, emphasizing the importance of high-quality retrieval and filtering [7]. This validates the system's approach of using retrieved data only when available and refusing to synthesize from internal model knowledge alone.

Question Decomposition and Query Understanding

Decomposing complex queries into manageable sub-questions is a critical step in achieving comprehensive coverage and avoiding surface-level summarization. Khot et al. introduced a modular decomposition approach that significantly improved structured reasoning in open-domain tasks [8]. Their findings inform the conservative decomposition approach used here, which prioritizes semantic clarity and task efficiency.

Research trends also highlight that moderate granularity (2–4 questions) yields the best balance between specificity and synthesis potential [9]. These principles guide the system's decomposition agent, which avoids excessive splitting and ensures each sub-task remains conceptually coherent.

Educational Content Generation with LLMs

The synthesis of educational scripts from raw information requires domain-specific prompt design and constraint-based generation. Liu et al. conducted a comprehensive survey of prompting strategies and highlighted that constrained, structured prompts outperform open-ended formats in educational scenarios [10]. These insights underpin the prompt template used in the synthesis agent, which enforces structure, clarity, and factuality.

In addition, feedback-loop-based iterative prompting has been shown to enhance content quality, coherence, and completeness—features directly reflected in the system's feedback agent.

Research Positioning and Contributions

The reviewed literature offers critical guidance across multiple components of the system—from agent coordination and RAG design to prompt structuring and semantic decomposition. However, while existing work addresses isolated aspects of educational content automation, few systems integrate all stages—from user query to video output—into a unified, real-time, feedback-driven pipeline.

This project addresses that gap by incorporating:

- Multi-agent coordination using LangGraph and ReAct principles;
- Multi-modal retrieval (web, video, academic) for source triangulation;
- Conservative query decomposition for coherent answer flow;
- Constrained script synthesis grounded entirely in retrieved data.

These contributions position the Agentic AI Video Synthesizer as a novel system that not only builds upon recent advances in LLM integration and media synthesis but also pushes the boundary toward fully automated, scalable, educational video generation.

2.2 Technology Review

The technologies considered and selected for this project span five key areas: LLM platforms, agent orchestration, retrieval APIs, multimedia synthesis tools, and web deployment frameworks.

2.2.1 Language Model and Agent Framework

- **Groq + llama-3.3-70b-versatile**: Chosen for its high inference speed (100x token throughput via Groq hardware) and competitive performance in factuality and reasoning.
- **Alternatives Considered**: GPT-4 and Claude 3 offered better reasoning depth but at higher latency and cost. Open-source models like Mistral 7B lacked consistency across multi-turn tasks.

- **LangGraph (LangChain extension)**: Enables stateful, recursive, agent-based orchestration. Supports loopback, conditional branching, and modularity.
- **Alternative**: CrewAI or AutoGen, but these lacked the robust memory and branching support needed for multi-step research and iteration.

2.2.2 Retrieval Tools

- **Tavily API**: Selected for real-time, AI-summarized web search with raw content extraction. Its advanced search depth and JSON output format suit structured prompts.
- **YouTube Data API v3**: Enables keyword-based video retrieval and transcript scraping for educational media. Captions are parsed and summarized by the YouTube agent.
- **arXiv API**: Chosen for academic credibility and access to abstracts and PDFs in AI and STEM domains.

These three sources cover a wide spectrum: popular media, current events, and peer-reviewed science.

2.2.3 Summarization & Script Generation

- **Custom LLM Prompts**: Used for YouTube, web, and academic summarization, explicitly restricting LLM generations to tool-returned data.
- **Synthesis Prompt**: Tailored to generate natural, spoken-style educational scripts—no formatting, just pure narration-ready output.

2.2.4 Speech and Video Generation

- **gTTS (Google Text-to-Speech)**: Selected for its lightweight installation, offline conversion, and high pronunciation accuracy.
- **D-ID API**: Used to generate avatar videos. Offers synchronous text/audio-to-video rendering with selected avatars. Alternatives like Synthesia were cost-prohibitive or lacked API flexibility.

2.2.5 Frontend and Deployment

- **Streamlit**: Enables fast development of interactive web applications. Supports audio/video playback, status indicators, and file downloads.

2.3 Summary

The literature underscores the effectiveness of agentic AI architectures in decomposing, retrieving, and synthesizing information for content creation. Retrieval-Augmented Generation (RAG) and prompt-engineered LLMs address the hallucination issue, while synthetic narration and avatar rendering enhance accessibility and learner engagement.

From a technological perspective, the chosen stack—Groq + llama-3.3-70b-versatile, LangGraph, Tavily, YouTube, arXiv, gTTS, and D-ID—represents a carefully balanced solution optimized for real-time educational video generation. While more advanced TTS or avatar options exist, they often introduce latency, complexity, or cost. This project prioritizes performance, accessibility, and scalability.

The insights gained from this review informed key design decisions, such as enforcing source-only summarization, limiting script length for avatar rendering, and modularizing agents to support iteration and feedback. These foundations strongly shape the system's architecture, implementation strategy, and ultimate educational utility.

Chapter 3 Implementation

3.1 Overview of System Design and Architecture

The Agentic AI Video Synthesizer is designed as a modular, multi-agent system that automates the end-to-end process of generating educational video content from a user's natural language query. The architecture adopts a layered design to enable clear separation of concerns, scalability, and fault isolation across research, synthesis, and media rendering components. The system comprises five main layers: input processing, agentic reasoning, external data retrieval, content synthesis, and multimedia generation, all connected via a persistent state memory managed through LangGraph.

At the core of the system lies the LangGraph orchestration engine, which coordinates multiple intelligent agents using a graph-based workflow. This structure enables conditional routing between agents, memory persistence across nodes, and iterative feedback handling—capabilities that linear pipelines lack. LangGraph acts as the system's controller, managing transitions between different stages such as question decomposition, source querying, synthesis, feedback, and output preparation.

The input layer receives a free-text query from the user via a Streamlit interface. This query is passed to a Question Decomposer Agent, which breaks down complex queries into sub-questions. These sub-questions are processed in parallel by three Tool Agents, each responsible for retrieving information from a specific domain: Tavily for real-time web content, the YouTube Data API for video-based insights, and the arXiv API for academic research.

The agentic reasoning layer coordinates the actions of these agents using a persistent state dictionary that tracks the progress and results of each sub-question. After all research agents complete their retrieval and summarization, a Synthesis Agent constructs a unified educational script. The synthesis output is evaluated by a Feedback Agent, which assigns a quality score and initiates reprocessing if required.

Once a high-quality script is approved, the system enters the media generation layer. The script is first converted into audio using Google's Text-to-Speech (gTTS) engine. This audio is then passed to the D-ID API, which creates a lifelike avatar video delivering the synthesized narration.

The final outputs—video script, MP3 audio, and MP4 video—are delivered to the user through the Streamlit presentation layer. The interface supports tabbed viewing of results, real-time media playback, and downloadable ZIP packages. Each stage of the pipeline updates a centralized state object, ensuring continuity across all components. Figure 2 illustrates the overall system architecture of the Agentic AI Video Synthesizer, showing the interaction between agents, retrieval tools, synthesis components, and the Streamlit interface. This architecture highlights the modular design that supports scalability, fault isolation, and iterative feedback loops.

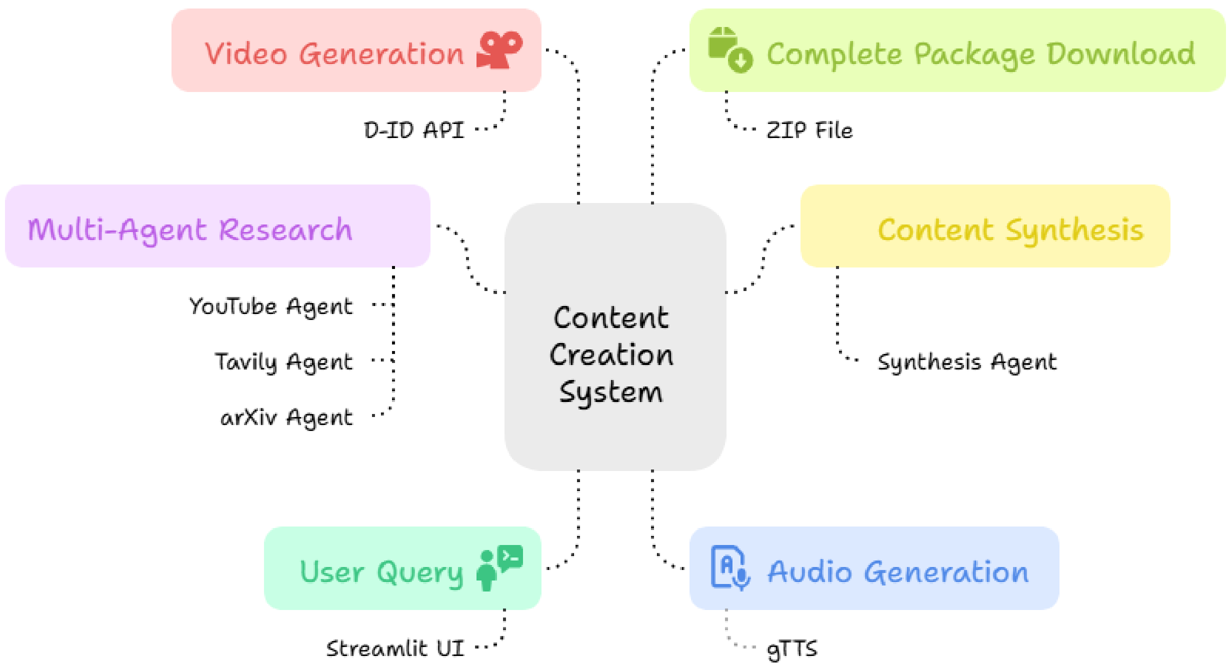


Figure 2 illustrates the system architecture, showing the interaction between LangGraph, tool agents, synthesis modules, and the Streamlit interface

3.2 Core Technology Stack and Design Justifications

The Agentic AI Video Synthesizer uses a carefully curated tech stack optimized for reasoning accuracy, speed, integration ease, and scalability. Each component serves a distinct role in enabling seamless educational content generation.

3.2.1 Reasoning Engine

The system uses the llama-3.3-70b-versatile model via Groq API, chosen for its fast token throughput, low latency, and strong multi-step reasoning performance. It powers all core agents—decomposition, summarization, synthesis, and feedback.

3.2.2 Multi-Agent Control

LangGraph provides memory-persistent orchestration with conditional routing between agents. Its graph-based design enables state management, loopbacks, and iterative logic, outperforming standard LangChain chains in complexity handling.

3.2.3 Research APIs

- **Tavily API** handles web search with deep content extraction and summarization.
- **YouTube Data API** retrieves educational video metadata and captions.
- **arXiv API** accesses scientific papers and abstracts for academic grounding.

These three modalities ensure diverse, credible, and rich content coverage.

3.2.4 Media Generation

- **gTTS** is used for TTS narration due to its simplicity and multilingual support.
- **D-ID API** generates avatar videos with realistic lip-sync and fast turnaround, ideal for real-time explainer video synthesis.

3.2.5 User Interface

Streamlit powers the frontend, offering rapid UI development, media previews, and clean integration with Python backends—ideal for interactive, model-driven apps.

3.2.6 Development Stack

The system is built in Python 3.11, using packages like requests, gtts, dotenv, and LangChain. It uses in-memory and local file storage to simplify infrastructure and maintain session isolation.

This stack balances performance, modularity, and ease of integration, supporting a maintainable and extensible educational content pipeline.

3.3 LangGraph-Based Multi-Agent Orchestration

The implementation of LangGraph within the Agentic AI Video Synthesizer serves as the architectural backbone of the system's reasoning and decision-making process. LangGraph is a specialized orchestration framework built on top of LangChain that enables agent nodes to communicate, share memory, and conditionally transition between tasks in a structured graph. This agent coordination mechanism is critical for processing complex user queries and managing multiple research, evaluation, and generation stages effectively.

3.3.1 Multi-Agent Workflow Structure

The LangGraph orchestration is composed of Eight primary agents, each representing a discrete function within the research and generation pipeline. These agents include:

1. Question Decomposer Agent
2. React Supervisor Agent
3. YouTube Agent
4. Tavily Agent
5. ArXiv Agent
6. Synthesis Agent
7. Feedback Agent
8. Completeness Checker Agent

These nodes are connected through directed edges that represent allowable transitions based on the evolving state dictionary. The initial state begins at the START node, transitions to the decomposition agent, and then follows a reactive loop governed by the React Supervisor, until all sub-questions are answered and the system is ready for synthesis.

3.3.2 State Dictionary and Memory Persistence

Each agent in the graph reads from and writes to a shared state dictionary that serves as both memory and inter-process communication. This dictionary tracks:

- Original and rewritten user queries
- Sub-question list and current question index
- Completed agents for each sub-question

- Retrieved summaries and citations
- Script drafts and feedback scores

This persistent state enables the system to operate in an iterative and non-linear fashion, allowing agents to revisit tasks, update previous outputs, or retry failed actions. LangGraph's internal runtime ensures the dictionary is properly maintained and passed through nodes without data leakage or overwriting. Figure 3 maps the LangGraph node-level workflow used in this project. It shows how the reacts supervisor routes between the retrieval agents (YouTube, Tavily, arXiv), the completeness checker, feedback agent, and synthesis agent from start to end.

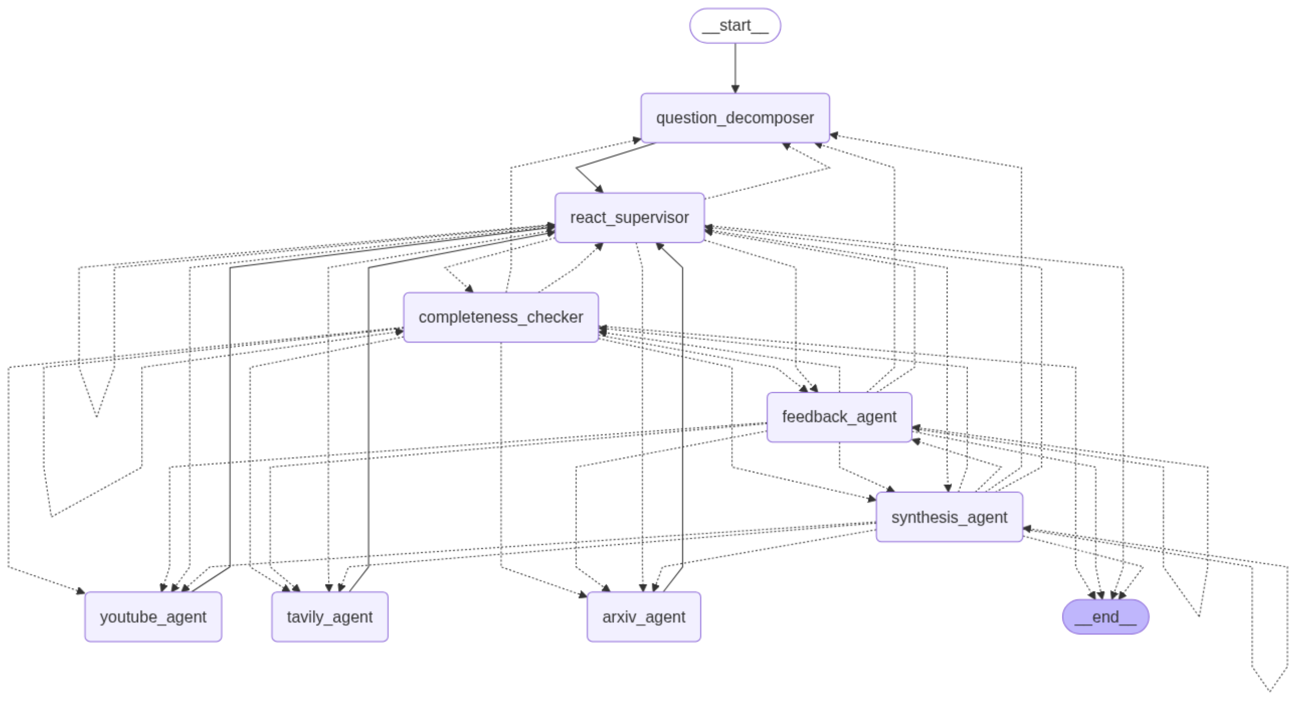


Figure 3 LangGraph research and synthesis workflow (node-level transitions)

3.3.3.Conditional Routing and Loopback Logic

LangGraph supports conditional routing, which is used extensively in the React Supervisor and Feedback Agent. Based on the completion status of research tasks or quality thresholds of the generated script, the system can dynamically route to:

- A new research agent (e.g., if one failed or returned no results)
- The Synthesis Agent (if all data is collected)
- The Feedback Agent (for final evaluation)

- Back to the Question Decomposer (if rewriting is needed)

3.3.4 React Supervisor and Scheduling

The React Supervisor Agent acts as a dynamic router that determines the next best agent to invoke. It tracks which retrieval agents have completed for the current sub-question and selects the next one based on predefined heuristics:

- Prioritize YouTube for broad topics or "how-to" style queries
- Use arXiv when keywords suggest technical or academic depth
- Choose Tavily for trending, current, or general knowledge topics

This scheduling model reduces redundant processing and tailors research strategies to the semantic nature of each question.

3.3.5 Agent Node Design and Prompt Integration

Each node in the graph is implemented as a LangChain Runnable, composed of:

- A system + user prompt (via ChatPromptTemplate)
- The Groq LLM (via ChatGroq)
- An output parser (via StrOutputParser)

3.4 Intelligent Query Breakdown and Semantic Decomposition

To effectively handle multi-topic user queries, the Agentic AI Video Synthesizer employs a conservative query decomposition strategy powered by a dedicated Question Decomposer Agent. This agent breaks down broad queries into 1–3 focused sub-questions, enabling more precise retrieval and improving the clarity of the final educational script.

The decomposition is performed by a Groq-hosted meta-llama-3.3-70b-versatile instruct model via a LangGraph node, guided by a carefully designed prompt. This prompt instructs the model to:

- Combine closely related topics (e.g., "AI and machine learning" remains one unit).
- Split clearly distinct themes (e.g., "Python programming and data science careers").
- Limit the breakdown to a small number of sub-questions to preserve narrative flow.

3.5 Tool Agent Framework: Retrieval via Web, Video, and Academic Sources

At the heart of the Agentic AI Video Synthesizer's research capability is a triad of specialized agents responsible for retrieving, validating, and summarizing information from diverse knowledge domains. These are referred to as Tool Agents, and each is designed to handle a distinct content modality:

- YouTube Agent – Multimedia and spoken content
- Tavily Agent – Real-time web and news articles
- ArXiv Agent – Peer-reviewed academic research

3.5.1 YouTube Agent: Multimedia Content Extraction and Summarization

The YouTube Agent is responsible for leveraging the YouTube Data API v3 to retrieve videos related to the current sub-question. It returns a list of the top 5 videos, each with metadata including:

- Title
- Channel name
- Description
- Publish date
- Video ID

In cases where caption transcripts are available, they are retrieved through an additional call to the YouTube Captions API. These transcripts often contain raw educational dialogue and tutorials, making them valuable for summarization.

Transcript Handling:

If captions are available, the system:

- Removes timestamp markers and speaker tags.
- Segments content into readable paragraphs.

- Applies token limits (~5000 characters) to manage summarization costs and API input size.

The cleaned data (description + captions) is compiled into a summary prompt and passed to the meta-llama-3.3-70b-versatile model with instructions to strictly base output on available content only. If no videos or captions are returned, the agent simply responds with "No YouTube data available".

Design Constraints:

- Does not generate from prior knowledge.
- Skips hallucinated summaries when data is missing.
- Stores citations (title + URL) for report generation.

3.5.2 Tavily Agent: Web Search with Content Filtering and Scoring

The Tavily Agent connects to the Tavily Web Search API, which performs deep web searches and returns both AI-generated summaries and the full textual content of relevant web pages, along with metadata such as titles, URLs, and publish dates.

When results are received, the system:

- Extracts the AI-generated summary (if available).
- Cleans the full article text by removing HTML tags, ads, and irrelevant layout elements.
- Filters the text to retain only factual or instructional content.

Both the summary and cleaned full content are then compiled into a structured input and passed to the LLM with a prompt that clearly instructs it to generate responses only based on this provided material. This two-layered input approach improves factual grounding and reduces hallucination during script synthesis.

Citation Handling:

Only sources with valid titles and URLs are included in the final research report. Each question maintains its own citation list in the shared LangGraph state.

Error Handling:

- If Tavily returns no content or only summaries, the agent defaults to "No web data available".
- AI-generated summaries from Tavily are excluded from citation lists to avoid circular reasoning.

3.5.3 ArXiv Agent: Scholarly Research Retrieval and Abstract Cleaning

The ArXiv Agent queries the arXiv API using the current sub-question. It returns up to 5 relevant papers based on keyword matching and publication relevance. Each result includes:

- Title
- Authors
- Abstract
- Published date
- Paper URL

To adapt arXiv abstracts for use in educational scripts, the agent implements:

- **Abstract Cleaning**: Removes LaTeX formatting, citation references, and formulae.
- **Author Attribution**: Preserves author names for citation formatting.

The filtered abstract content is passed to the LLM with clear instructions to summarize only the included abstracts, avoiding domain extrapolation or technical overreach.

Unique Features:

- All academic content is clearly marked in the final script summary.
- Abstracts are limited to 5000 characters per paper to avoid prompt overflow.

The React Supervisor Agent controls the research workflow by deciding which Tool Agent to activate next for each sub-question. It bases its decision on three key factors:

1. **Agent Completion**: It checks which agents have already been executed for the current question to avoid duplication.
2. **Query Relevance**: It analyzes the sub-question's keywords to prioritize the most relevant agent:
    - YouTube for tutorials or how-to topics.
    - Tavily for current, trending, or general knowledge queries.
    - arXiv for technical or academic topics.

3.6 Script Synthesis through Structured Prompt Engineering

After the system gathers research summaries from the Tool Agents, the Synthesis Agent generates the final educational script. This step is crucial in transforming fragmented multi-source inputs into a coherent, natural-sounding spoken narrative. Before synthesis begins, the Completeness Checker Agent evaluates whether all relevant Tool Agents have returned usable data for each sub-question. If critical gaps are detected—such as missing content from all sources for a given topic—the system may trigger a fallback mechanism or prompt for regeneration. This ensures that only sufficiently researched inputs proceed to the script generation stage, maintaining the integrity and factual grounding of the final output.

The Synthesis Agent uses the meta-llama-3.3-70b-versatile model hosted on Groq, combined with a carefully crafted system prompt. This prompt enforces:

- A clear introduction, middle, and conclusion.
- Definitions of key terms in accessible language.
- Smooth transitions between sub-questions.
- A conversational, narrator-style tone without formatting or metadata.

The summarization input includes all research grouped by sub-question and source (YouTube, Tavily, arXiv). The agent is explicitly instructed not to fabricate information and to only use retrieved summaries.

To ensure clean output, the system applies regex filtering to remove unwanted characters, timestamps, or markdown formatting. The script is saved to state and used in both audio and video generation stages.

3.7 Iterative Refinement and Quality Assurance

To ensure the final script meets quality standards, the system includes a Feedback Agent that evaluates the generated content across clarity, accuracy, and structure. This agent uses the same meta-llama-3.3-70b-versatile model to score the script and provide improvement suggestions.

If the score falls below a predefined threshold, the system triggers a refinement loop. A rewritten query is automatically generated based on the feedback, and the full research-synthesis cycle is re-executed using the new input.

Safeguards are in place to:

- Limit the number of retries (maximum of 3 iterations).
- Avoid infinite loops or trivial rewrites.
- Clearly log quality status and improvement attempts.

This feedback mechanism enables the system to self-correct and improve educational clarity without manual intervention—mimicking expert review and revision workflows.

3.8 Text-to-Speech Integration and Video Generation with D-ID

The final stage of the Agentic AI Video Synthesizer pipeline converts the synthesized script into a fully rendered video with spoken narration and avatar animation. This involves two main components: text-to-speech (TTS) conversion and avatar video generation.

3.8.1 Text-to-Speech (TTS) with gTTS

The system uses Google Text-to-Speech (gTTS) to convert the script into an MP3 audio file. Prior to conversion, the text undergoes basic preprocessing:

- Removal of non-verbal symbols, timestamps, and special characters.
- Adjustment of punctuation to enhance intonation and clarity.

gTTS was selected for its simplicity, accurate pronunciation, and lightweight operation—making it ideal for dynamic, on-demand audio synthesis. The resulting audio is stored in memory and streamed within the Streamlit interface.

3.8.2 Avatar Video Generation with D-ID API

The synthesized script is submitted directly to the D-ID API, which automatically generates a realistic avatar video with lip-synced voice narration. The system constructs a payload containing:

- The full text script
- Selected presenter avatar ID
- Desired voice preset (e.g., gender, tone)

D-ID handles both the text-to-speech conversion and video rendering internally. The system polls the D-ID endpoint until rendering is complete, after which the final MP4 video is downloaded and presented to the user.

3.9 Streamlit Interface and User Experience Implementation

The frontend of the Agentic AI Video Synthesizer is built using Streamlit, providing a lightweight, interactive user interface that guides users through each stage of content generation. The design prioritizes clarity, accessibility, and seamless interaction with the backend agents.

Key Features

- **Query Submission Interface**: Users input a complex educational query, which initiates the full multi-agent pipeline.
- **Tabbed Results View**: Final outputs are grouped into tabs for:
    - Video Script
    - Research Sources (collapsible by question)
    - Audio Playback
    - Video

    - Downloadable ZIP

This interface design ensures a streamlined, professional experience for both technical and non-technical users, supporting quick iteration and accessible educational media generation.

3.10 Key Challenges and Solutions

During The development and deployment of the Agentic AI Video Synthesizer involved addressing several complex engineering challenges. These ranged from multi-agent state coordination to integrating with third-party AI services and refining agent prompts for consistent output. This section outlines the most significant obstacles encountered and the strategies used to resolve them.

3.10.1 Coordinating Multi-Agent State Transitions

One of the foundational challenges was ensuring smooth and accurate transitions across multiple agents within the LangGraph framework. Each agent—responsible for decomposition, retrieval, synthesis, and feedback—had to read from and write to a shared state dictionary. Improper state handling led to issues such as skipped sub-questions, overwritten summaries, or infinite loops in early iterations.

To address this, a structured state schema was implemented, with clear, persistent keys for:

- Decomposed sub-questions
- Individual agent outputs (YouTube, Tavily, arXiv)
- Script drafts, citations, and quality scores
- Completion flags and iteration counters

LangGraph's native support for memory persistence and conditional routing was leveraged to track agent completion status and dynamically determine execution paths. For example, the React Supervisor used these flags to prevent redundant tool agent calls and route control to synthesis only when data from at least one source was available.

This modular and explicit state management design reduced the chance of information loss and allowed the system to resume or recover gracefully after API failures. This solution aligns with best practices in multi-agent systems as discussed in recent work on tool-augmented LLM pipelines.

3.10.2 Integrating D-ID for AI Avatar Rendering and Video Access

Another challenge involved integrating with D-ID's avatar video API, particularly in automating the full cycle of script submission, video generation, and file retrieval. The D-ID API does not return the MP4 file immediately; instead, it provides a job ID that must be polled periodically to check for completion.

Initial integration attempts led to unreliable behavior due to premature polling or timeouts. To resolve this:

- A robust polling loop was implemented using exponential backoff and capped retries.
- A check for valid video URLs was enforced before attempting download.

Once the video was ready, a secure download was performed using requests.get() with stream mode and written to local memory or disk, depending on session mode.

In addition, to handle cases where videos exceeded time or character limits (as imposed by free-tier D-ID usage), the script was truncated with fallback messaging, and users were notified via the UI. This integration strategy is consistent with recommended practices for asynchronous API interactions and third-party rendering workflows .

3.10.3 Prompt Engineering Optimization and Agent Instruction Refinement

A significant challenge in developing reliable agent behavior stemmed from prompt instability and output inconsistency, particularly when agents generated irrelevant or hallucinated content. Early versions of the Tool Agents occasionally returned generic summaries or included made-up details not present in the source content.

To address this, extensive prompt engineering was performed using:

- Explicit, role-based instructions (e.g., "You are a factual summarizer for educational content.")

- Strict formatting rules (e.g., "Output only based on the provided content.")
- Negative constraints (e.g., “Do not include assumptions or external knowledge.”)

Chapter 4 Evaluation and Results

The Agentic AI Video Synthesizer was evaluated through a series of functional tests, UI walkthroughs, and detailed observations of each processing stage. The goal was to assess the system's performance, completeness, usability, and resilience across varied query structures—ranging from simple single-question prompts to complex multi-topic inputs. Screenshots were used to validate each pipeline phase and confirm accurate transitions across agents. Evaluation also measured script quality, research completeness, tool agent usage, audio-video rendering performance, and user interface stability.

This chapter presents evaluation outcomes across each step of the system, supported by visual validation and key performance observations. Special attention was paid to whether the system satisfied the functional goals: educational quality, agent coordination, and tool-grounded summarization.

4.1 Related Works

The generation of educational content using AI has gained traction through large language models (LLMs), multi-agent systems, and synthetic media technologies. Prior work has explored components of this pipeline, such as retrieval-augmented generation, prompt engineering, and avatar-based video narration. However, very few systems have attempted to unify these capabilities into an automated end-to-end video synthesizer tailored for educational applications.

Projects such as AutoGPT and LangChain introduced foundational concepts in agent orchestration, where LLMs operate within loops to reason and act through tools. These systems demonstrated early promise in research automation, but lacked persistent memory or structured feedback mechanisms. The introduction of LangGraph in 2023 marked a significant advancement, enabling multi-agent workflows with state-aware transitions and iterative logic. This project leverages LangGraph for orchestrating question decomposition, research, synthesis, and script quality feedback in a graph-based architecture—offering more reliable coordination than linear chains.

From a multimedia perspective, earlier educational content generators focused heavily on templated animation or human narration. The integration of gTTS for TTS generation and D-ID's avatar rendering adds a synthetic narration layer, transforming textual outputs into visual storytelling assets. While platforms like Synthesia or HeyGen support similar avatar narration, they often require manual input, lack tool-grounded scripting, or operate as commercial black-box solutions.

Moreover, earlier implementations of RAG pipelines often relied on a single modality—typically web search or static databases. In contrast, this system adopts a tri-modal retrieval strategy (YouTube, Tavily, arXiv), providing a blend of real-world, video-based, and academic insights to maximize factual grounding.

In summary, while foundational elements have been explored individually in academic and commercial settings, the Agentic AI Video Synthesizer presents a novel synthesis of these technologies into a cohesive educational pipeline—automating the entire process from query to video, with layered research validation and modular agent reasoning.

4.2 Evaluation Walkthrough and Results

4.2.1 Plain Streamlit Interface Initialization

The initial system loads with a clean Streamlit UI featuring a professional gradient header displaying the system title "Agentic AI Video Synthesizer" with descriptive subtitle outlining its capabilities. The interface displays a text box with placeholder text "What would you like to create a video about?" and a prominent "Generate Content" button. The left sidebar shows system status indicators for all integrated APIs including GROQ (LLM), Tavily (Web Search), YouTube API, D-ID (Video), and LangSmith (Tracing) with clear green checkmarks indicating service availability. Figure 4 displays the initial Streamlit interface as loaded by the system, providing a clean and user-friendly starting point for query submission.

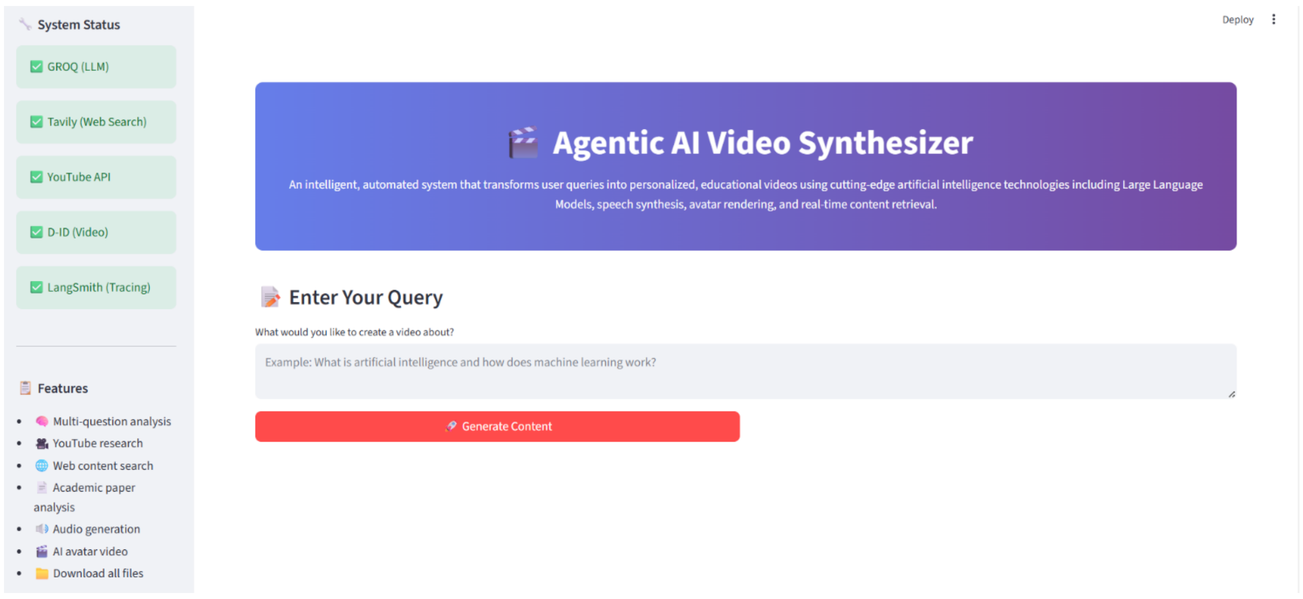

System Status
GROQ (LLM)
Tavily (Web Search)
YouTube API
D-ID (Video)
LangSmith (Tracing)
Features
Multi-question analysis
YouTube research
Web content search
Academic paper analysis
Audio generation
AI avatar video
Download all files
Deploy
Agentic AI Video Synthesizer
An intelligent, automated system that transforms user queries into personalized, educational videos using cutting-edge artificial intelligence technologies including Large Language Models, speech synthesis, avatar rendering, and real-time content retrieval.
Enter Your Query
What would you like to create a video about?
Example: What is artificial intelligence and how does machine learning work?
Generate Content

Figure 4 Streamlit interface initialization

4.2.2 Single Question Input by User

The user enters a straightforward educational query: "agentic AI" in the main text area. This single-question input is designed to test the system's ability to handle focused educational content requests without complexity. Upon clicking the "Generate Content" button, the query triggers the backend pipeline via LangGraph, automatically invoking the ReAct Supervisor agent. Figure 5 shows the interface state after a single-question input has been submitted, triggering the multi-agent pipeline.

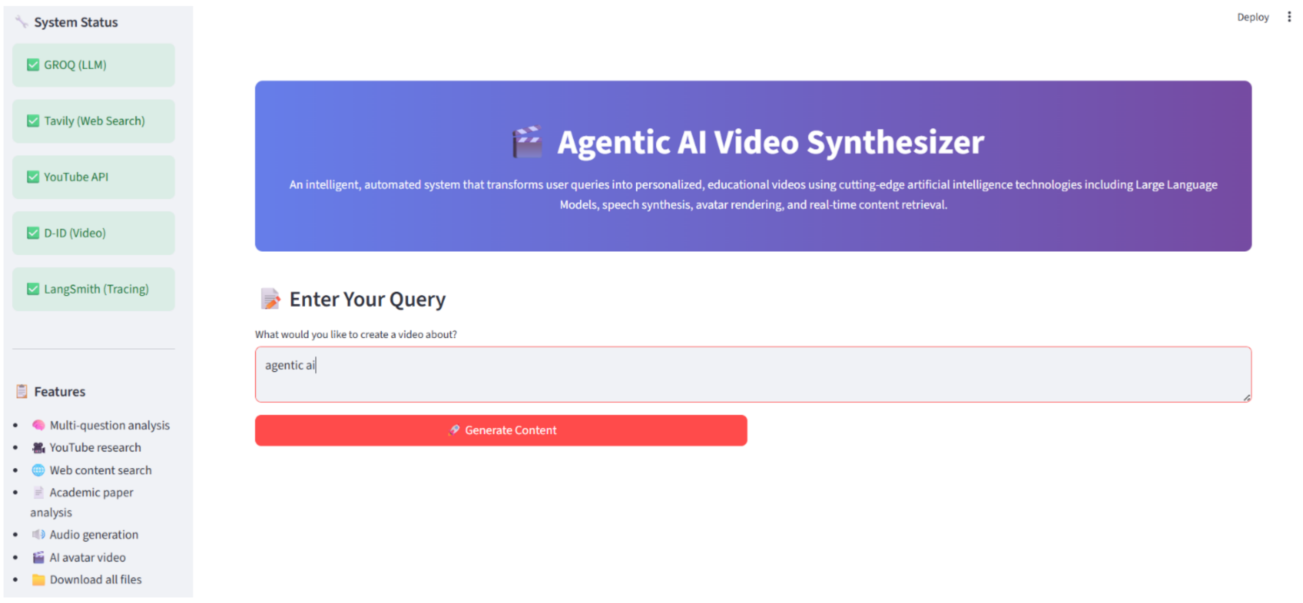
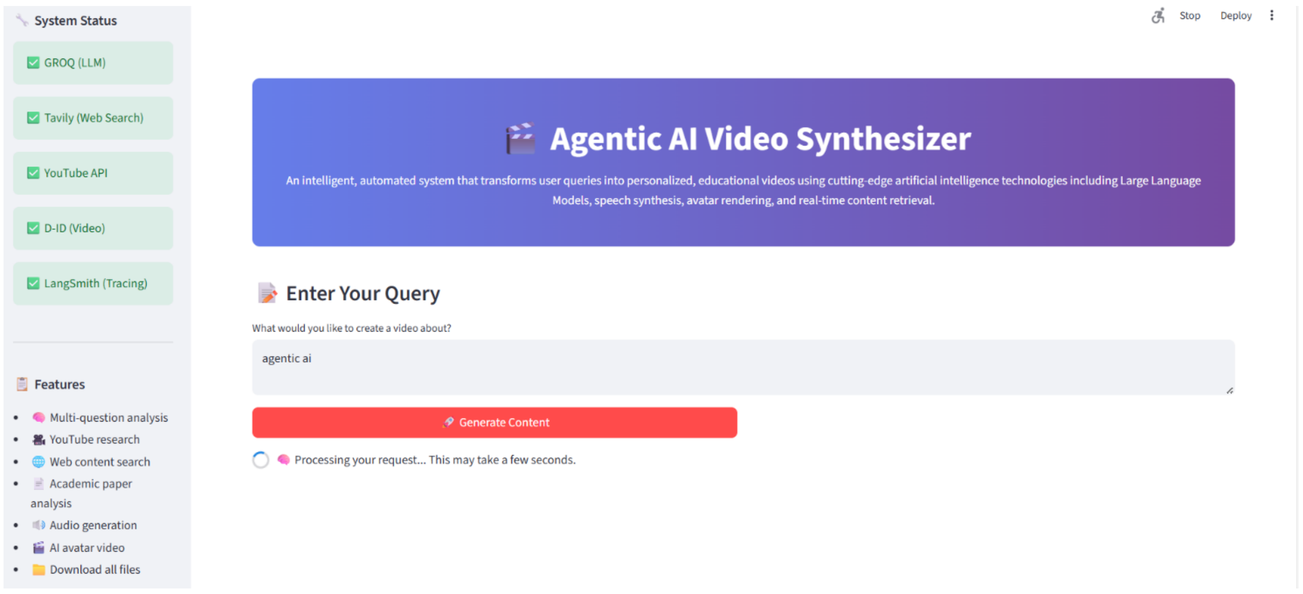

System Status
GROQ (LLM)
Tavily (Web Search)
YouTube API
D-ID (Video)
LangSmith (Tracing)
Features
Multi-question analysis
YouTube research
Web content search
Academic paper analysis
Audio generation
AI avatar video
Download all files
Deploy
Agentic AI Video Synthesizer
An intelligent, automated system that transforms user queries into personalized, educational videos using cutting-edge artificial intelligence technologies including Large Language Models, speech synthesis, avatar rendering, and real-time content retrieval.
Enter Your Query
What would you like to create a video about?
agentic ai
Generate Content

Figure 5 Single-question input interface view

4.2.3 Processing Spinner During Research

After submission, the interface displays a comprehensive loading spinner with the animated message " Processing your request... This may take a few seconds." The spinner uses CSS animations to create a professional waiting experience. Although the interface only shows a loader, extensive background tasks are active throughout this phase:

The Question Decomposer Agent analyzes the input and identifies this as a single, focused question requiring no further segmentation. The ReAct Supervisor orchestrates sequential agent execution based on the determined workflow path. Research agents (YouTube, Tavily, ArXiv) are dispatched simultaneously to gather relevant educational content from their respective domains. Figure 6 captures the loading spinner displayed during the research phase, indicating active processing by multiple agents.



An intelligent, automated system that transforms user queries into personalized, educational videos using cutting-edge artificial intelligence technologies including Large Language Models, speech synthesis, avatar rendering, and real-time content retrieval.

What would you like to create a video about?

agentic ai

Processing your request... This may take a few seconds.

Figure 6 Processing spinner during research phase

4.2.4 Research Results Display

Upon successful completion of the research phase, the interface transitions to display comprehensive results with detailed performance metrics prominently featured in colored metric cards:

- **Quality Score:** 9/10 displayed in a prominent metric card, indicating high-quality content generation exceeding the minimum threshold of 7/10

- **Questions Found:** 1 question confirmed, validating the single-question identification process

- **Sources Retrieved:** 15 total sources distributed across platforms (YouTube: 5 videos, Tavily: 5 web results, arXiv: 5 academic papers)

- **Iterations:** 1 iteration completed, confirming no refinement cycles were necessary

Figure 7 presents the research results panel, summarizing quality score, questions found, sources retrieved, and iteration count.

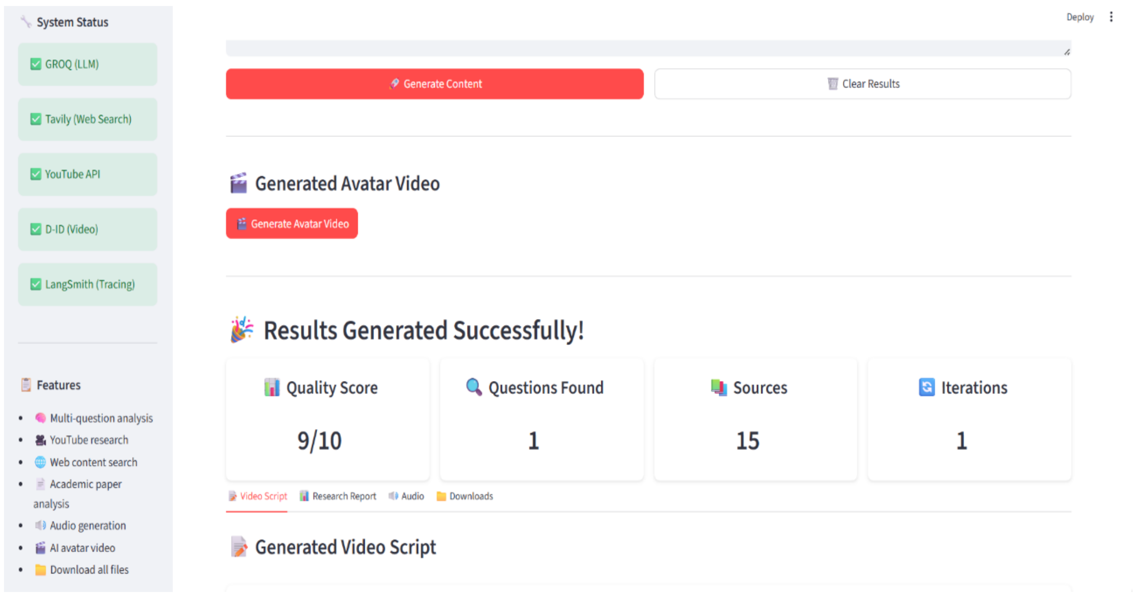


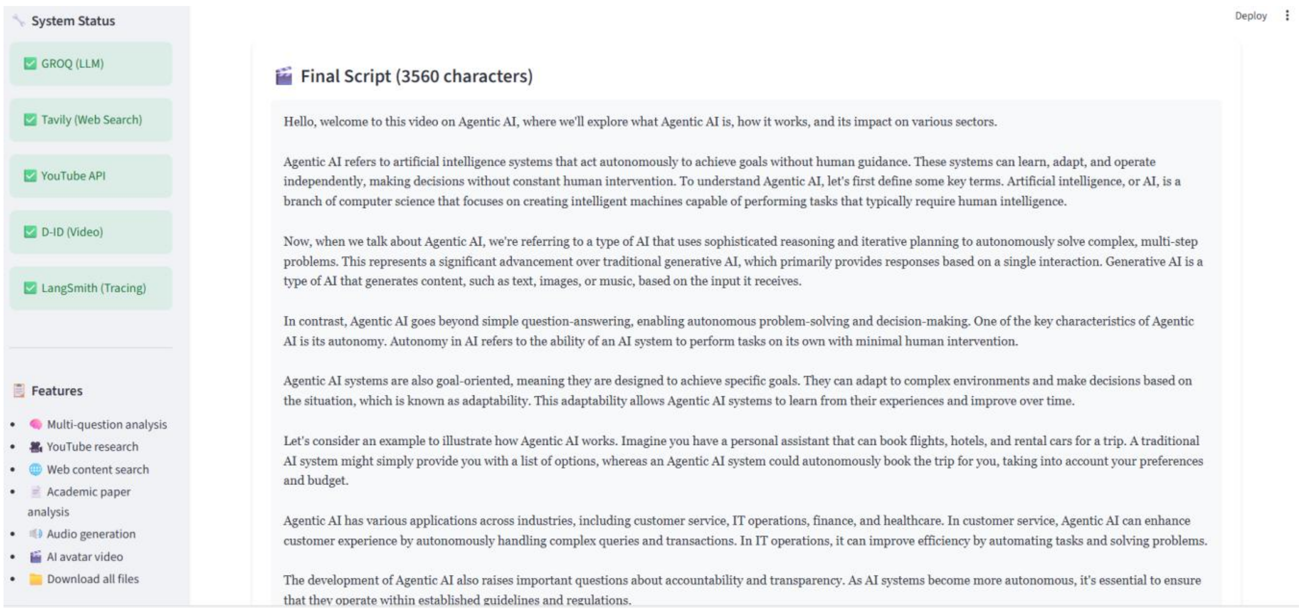
Figure 7 Research results display with performance metrics

4.2.5 Final Script Display

The Synthesis Agent successfully combines verified research data from all three sources into a coherent, paragraph-based educational script. The generated content demonstrates sophisticated educational structure while avoiding hallucinations by referencing only the validated summaries produced by the research agents.

The script follows a comprehensive educational framework including clear introduction with topic overview, technical definitions of "agentic AI" with foundational concepts, practical applications demonstrating concept relevance, logical transitions maintaining narrative coherence, and comprehensive conclusion summarizing key points.

The final script contains exactly 3,560 characters, representing optimal length for a 6-10 minute educational video. The content maintains a conversational yet authoritative tone suitable for diverse educational audiences while ensuring technical accuracy through source verification. Figure 8 shows the final synthesized script generated by the system, demonstrating coherent, source-grounded educational content.



Final Script (3560 characters)

Hello, welcome to this video on Agentic AI, where we'll explore what Agentic AI is, how it works, and its impact on various sectors.

Agentic AI refers to artificial intelligence systems that act autonomously to achieve goals without human guidance. These systems can learn, adapt, and operate independently, making decisions without constant human intervention. To understand Agentic AI, let's first define some key terms. Artificial intelligence, or AI, is a branch of computer science that focuses on creating intelligent machines capable of performing tasks that typically require human intelligence.

Now, when we talk about Agentic AI, we're referring to a type of AI that uses sophisticated reasoning and iterative planning to autonomously solve complex, multi-step problems. This represents a significant advancement over traditional generative AI, which primarily provides responses based on a single interaction. Generative AI is a type of AI that generates content, such as text, images, or music, based on the input it receives.

In contrast, Agentic AI goes beyond simple question-answering, enabling autonomous problem-solving and decision-making. One of the key characteristics of Agentic AI is its autonomy. Autonomy in AI refers to the ability of an AI system to perform tasks on its own with minimal human intervention.

Agentic AI systems are also goal-oriented, meaning they are designed to achieve specific goals. They can adapt to complex environments and make decisions based on the situation, which is known as adaptability. This adaptability allows Agentic AI systems to learn from their experiences and improve over time.

Let's consider an example to illustrate how Agentic AI works. Imagine you have a personal assistant that can book flights, hotels, and rental cars for a trip. A traditional AI system might simply provide you with a list of options, whereas an Agentic AI system could autonomously book the trip for you, taking into account your preferences and budget.

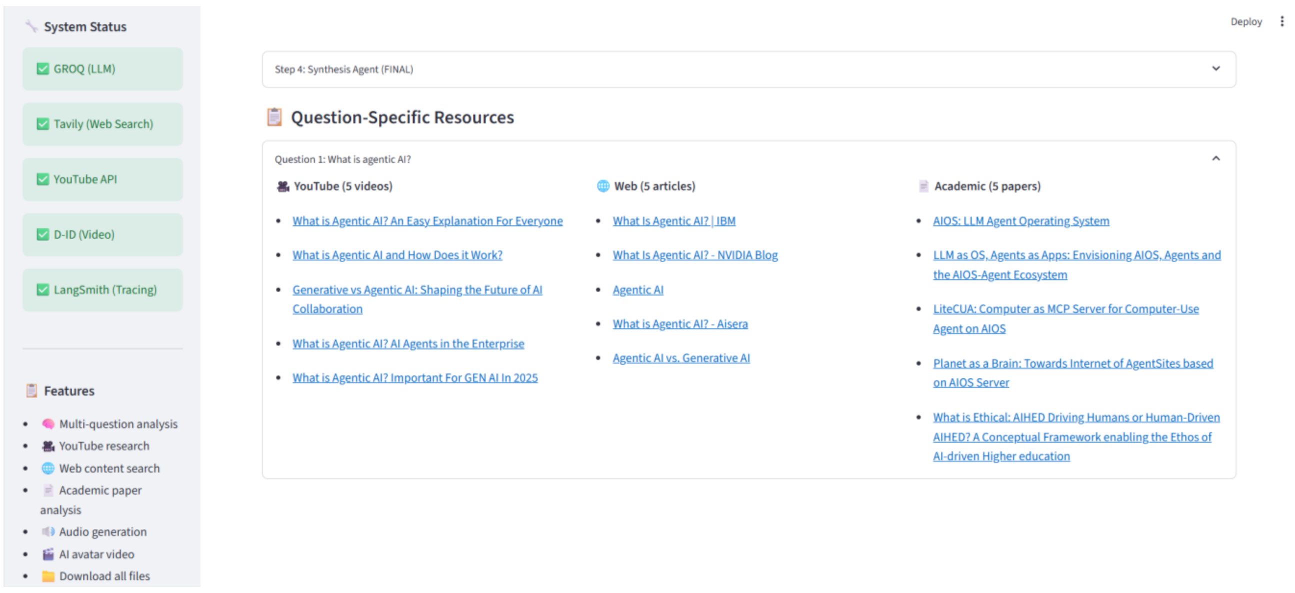
Agentic AI has various applications across industries, including customer service, IT operations, finance, and healthcare. In customer service, Agentic AI can enhance customer experience by autonomously handling complex queries and transactions. In IT operations, it can improve efficiency by automating tasks and solving problems.

The development of Agentic AI also raises important questions about accountability and transparency. As AI systems become more autonomous, it's essential to ensure that they operate within established guidelines and regulations.

Figure 8 Final script display in Streamlit

4.2.6 Question-Specific Resources Display

The resources tab presents research sources organized systematically by research agent type, providing complete transparency and traceability for all generated content. YouTube Educational Videos include five carefully selected videos with complete titles and direct clickable links to original content. Web Articles from Tavily feature five current web articles from authoritative sources including technology publications, academic institutions, and industry leaders. Academic Papers from arXiv provide five peer-reviewed research papers with complete abstracts, author information, and publication metadata. Figure 9 displays the organized question-specific resources, grouped by YouTube, web, and academic sources.


System Status

GROQ (LLM)

Tavily (Web Search)

YouTube API

D-ID (Video)

LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Step 4: Synthesis Agent (FINAL)

Question-Specific Resources

Question 1: What is agentic AI?

YouTube (5 videos)

- What is Agentic AI? An Easy Explanation For Everyone
- What is Agentic AI and How Does it Work?
- Generative vs Agentic AI: Shaping the Future of AI Collaboration
- What is Agentic AI? AI Agents in the Enterprise
- What is Agentic AI? Important For GEN AI In 2025

Web (5 articles)

- What Is Agentic AI? | IBM
- What Is Agentic AI? - NVIDIA Blog
- Agentic AI
- What is Agentic AI? - Aisera
- Agentic AI vs. Generative AI

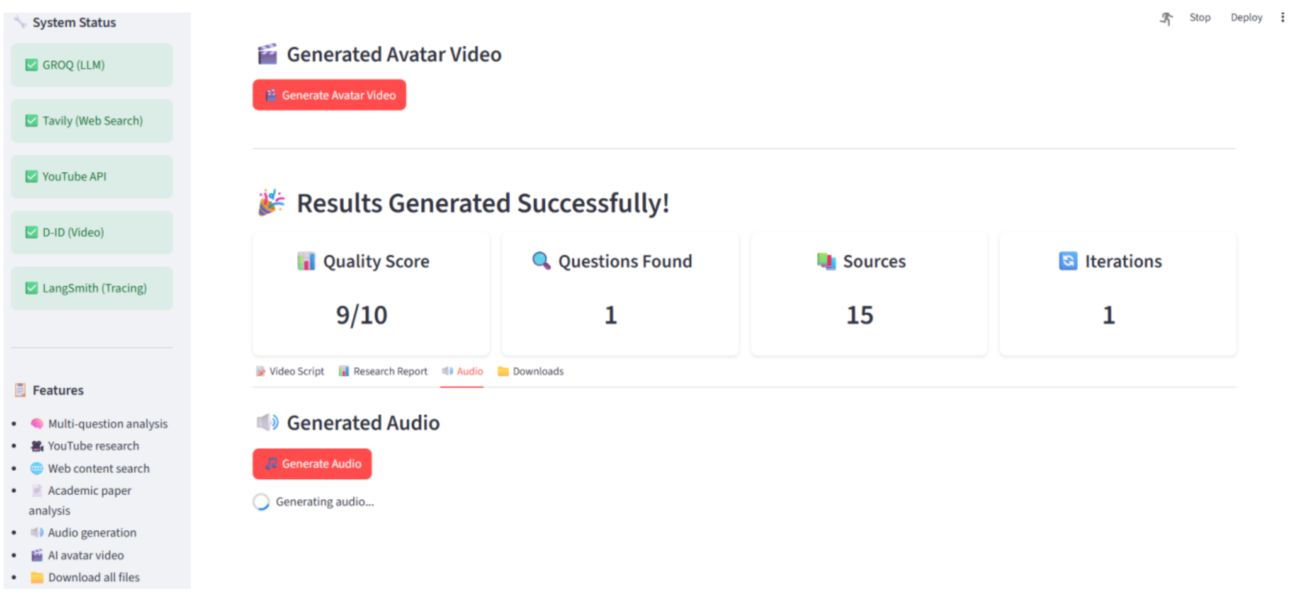
Academic (5 papers)

- AIOS: LLM Agent Operating System
- LLM as OS, Agents as Apps: Envisioning AIOS, Agents and the AIOS-Agent Ecosystem
- LiteCUA: Computer as MCP Server for Computer-Use Agent on AIOS
- Planet as a Brain: Towards Internet of AgentSites based on AIOS Server
- What is Ethical: AIHED Driving Humans or Human-Driven AIHED? A Conceptual Framework enabling the Ethos of AI-driven Higher education

Figure 9 Question-specific resource display

4.2.7 Audio Generation Processing

Clicking the " Generate Audio" button triggers the text-to-speech synthesis pipeline. The interface immediately displays a processing spinner with the status message "Synthesizing Audio..." along with animated progress indicators. The Google Text-to-Speech (gTTS) engine is invoked through the backend API, processing the complete 3,560-character script for audio conversion. Figure 10 captures the audio generation process in progress, with status messages and progress indicators.


System Status

GROQ (LLM)

Tavily (Web Search)

YouTube API

D-ID (Video)

LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Generated Avatar Video

Results Generated Successfully!

Quality Score	Questions Found	Sources	Iterations
9/10	1	15	1

Generated Audio

Generating audio...

Figure 10 Audio generation processing view

4.2.8 Audio Generation Complete

The Audio tab transitions to display a fully functional audio playback widget with comprehensive controls. The interface presents complete audio controls with play/pause button, volume slider, progress bar with time indicators, and playback speed options. Duration display shows total audio length (4:46 for this example) with current position tracking during playback. The generated voice-over demonstrates natural pronunciation, appropriate pacing, and professional tone matching the educational script content precisely. Figure 11 shows the completed audio playback interface, featuring controls for play, pause, volume, and progress tracking.

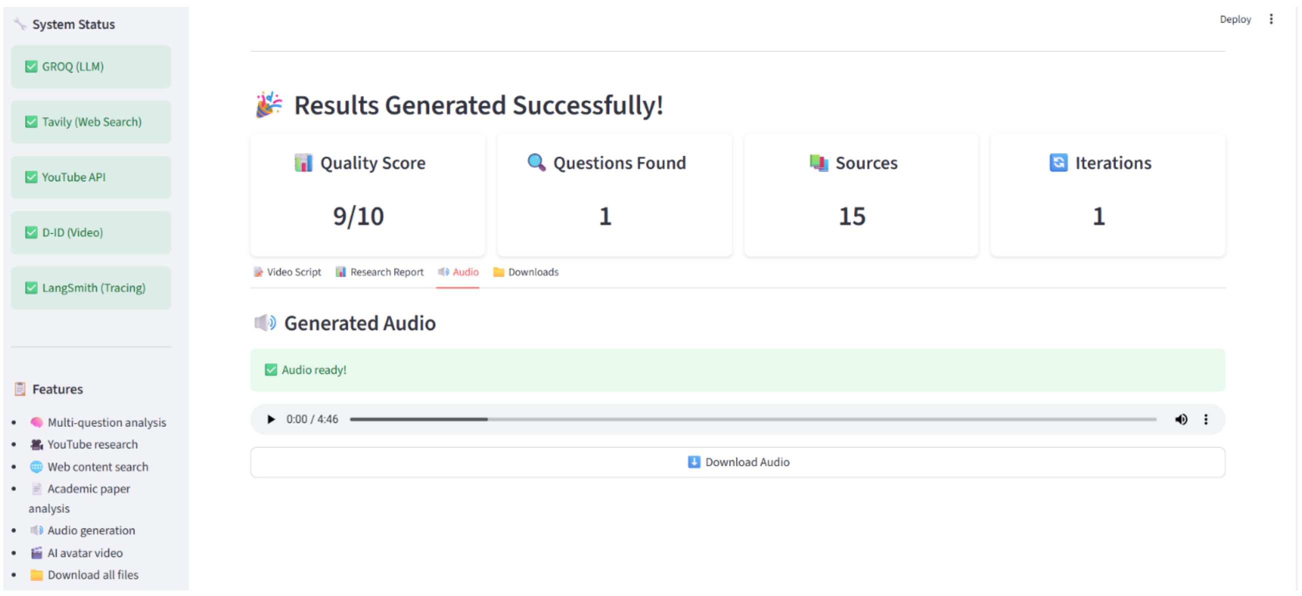
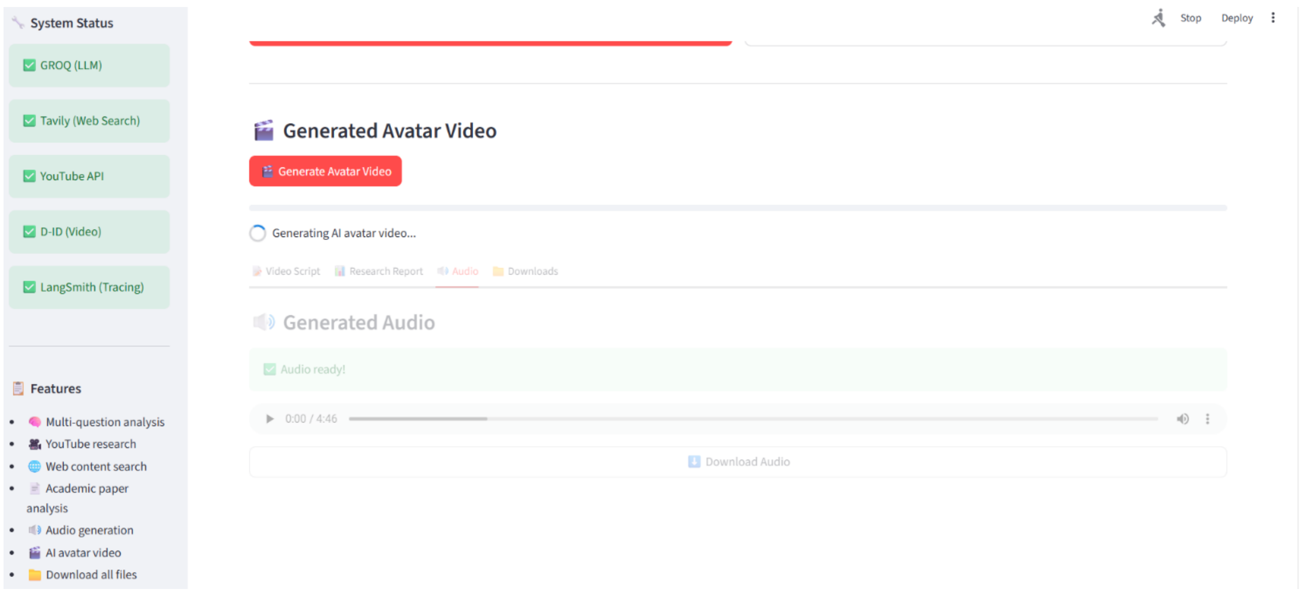


Figure 11 Completed audio generation with playback controls

4.2.9 Video Generation Processing

When users click " Generate Avatar Video", the D-ID API integration activates through the backend service. The processing interface displays multiple status indicators including "Generating AI avatar video..." with animated spinner, real-time updates showing current processing stage, and time estimates to manage user expectations during the longer video generation process. Figure 12 displays the video generation process interface, including real-time status updates during avatar rendering.


System Status

- GROQ (LLM)
- Tavily (Web Search)
- YouTube API
- D-ID (Video)
- LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Generated Avatar Video

Generating AI avatar video...

Generated Audio

Audio ready!

0:00 / 4:46

Figure 12 Video generation processing view

4.2.10 Video Generation Complete

The completed AI avatar video appears embedded directly in the interface with comprehensive playback controls. The final output demonstrates high-quality video player with full-featured media controls, synchronized lip-sync with the Emy avatar maintaining accurate synchronization throughout the presentation, professional presentation with natural facial expressions and engaging eye contact, and technical quality in high-resolution MP4 format suitable for web distribution. Figure 13 presents the completed AI avatar video embedded in the interface, ready for playback.

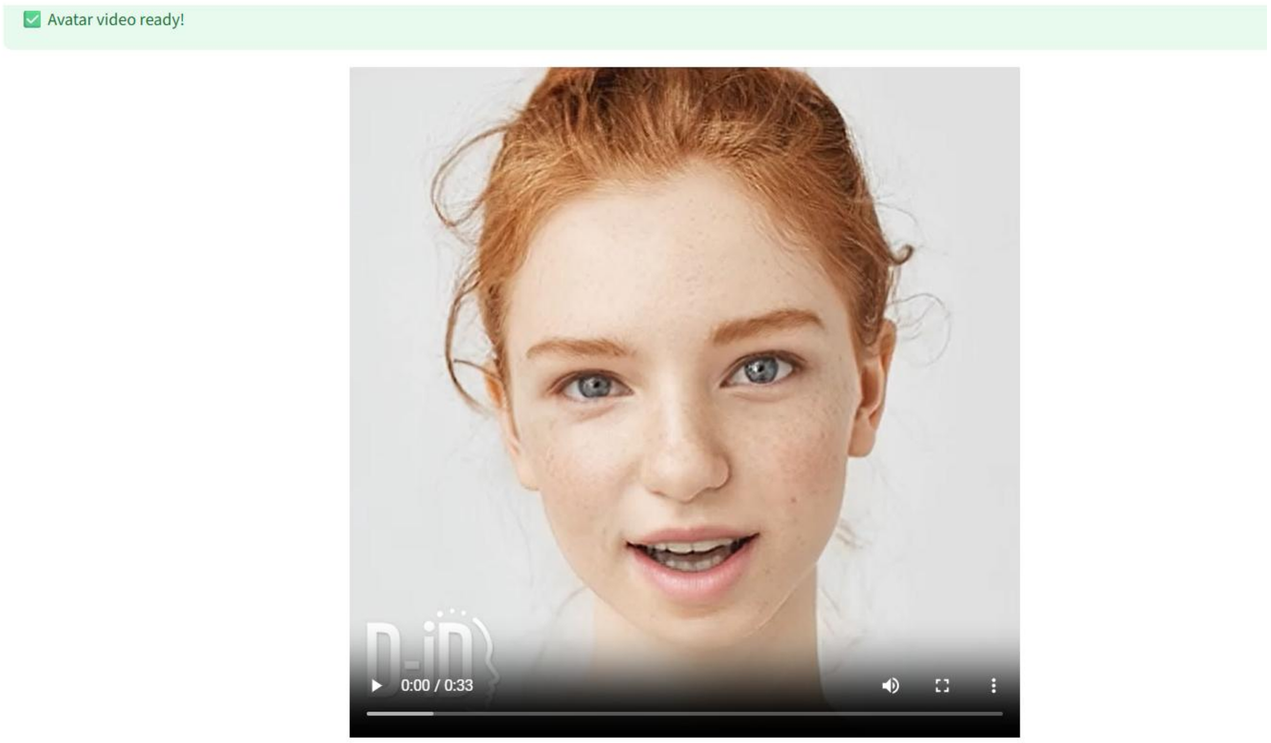
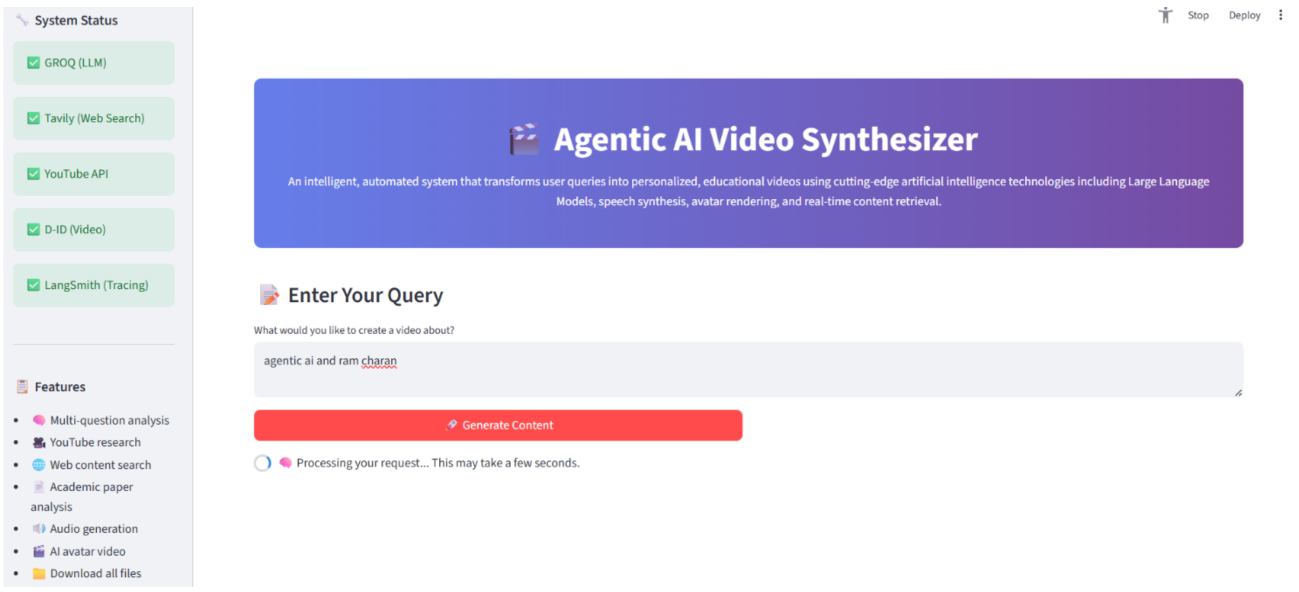


Figure 13 Completed AI avatar video display

4.3 Multi-Question Scenario Evaluation

4.3.1 Two-Question Input by User

For advanced system testing, a complex query containing multiple distinct educational topics was submitted: "agentic AI and Ram Charan" This sophisticated input tests the system's ReAct-based question decomposition capabilities and multi-topic research orchestration. The Question Decomposer Agent successfully analyzes the compound query structure and splits it into two distinct sub-questions for individual processing. Figure 14 shows the interface where a combined two-topic query is entered, initiating the multi-question processing path.


System Status

GROQ (LLM)

Tavily (Web Search)

YouTube API

D-ID (Video)

LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Stop Deploy

Agentic AI Video Synthesizer

An intelligent, automated system that transforms user queries into personalized, educational videos using cutting-edge artificial intelligence technologies including Large Language Models, speech synthesis, avatar rendering, and real-time content retrieval.

Enter Your Query

What would you like to create a video about?

agentic ai and ram charan

Generate Content

Processing your request... This may take a few seconds.

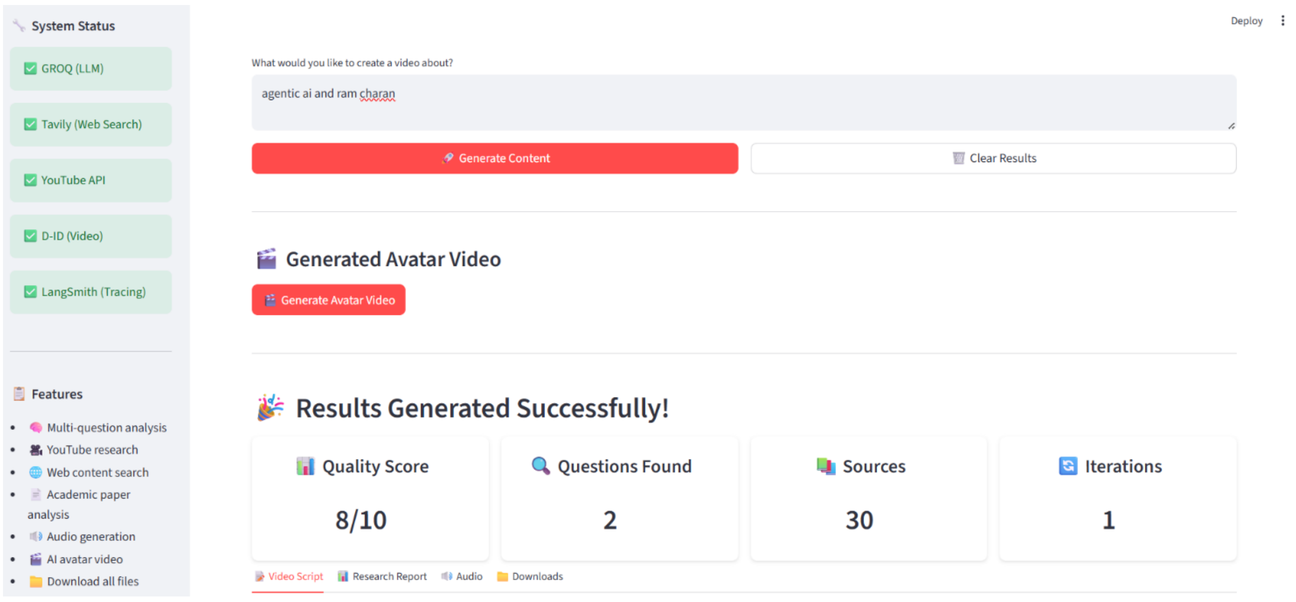
Figure 14 Two-question input for the multi-question scenario

4.3.2 Multi-Question Research Results

The system successfully processes the complex query with enhanced performance metrics displayed in the updated results panel:

- **Quality Score:** 8/10 (high quality rating for multi-topic content complexity)
- **Questions Detected:** 2 questions successfully identified and separated
- **Total Sources:** 30 sources gathered (15 comprehensive sources per identified question)
- **Iterations:** 1 iteration completed (efficient processing without refinement requirements)

Figure 15 summarizes the aggregated metrics for the multi-question query, including quality score, detected questions, total sources, and iterations.


System Status

GROQ (LLM)

Tavily (Web Search)

YouTube API

D-ID (Video)

LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Deploy

What would you like to create a video about?

agentic ai and ram charan

Generate Content | Clear Results

Generated Avatar Video

Generate Avatar Video

Results Generated Successfully!

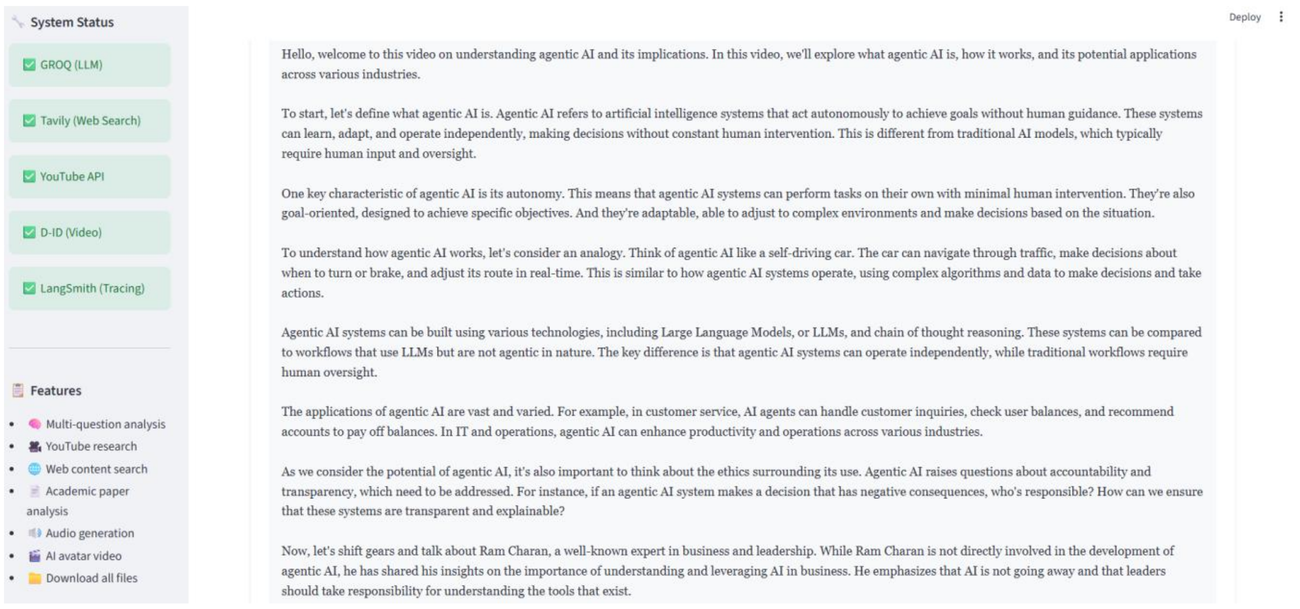
Quality Score	Questions Found	Sources	Iterations
8/10	2	30	1

Video Script | Research Report | Audio | Downloads

Figure 15 Multi-question research results and performance metrics

4.3.3 Multi-Question Final Script

The Synthesis Agent demonstrates advanced capability by combining research from both distinct questions into a cohesive educational narrative. The resulting script features seamless topic transitions between agentic AI technical concepts and Ram Charan's business leadership perspectives, maintained educational structure throughout the extended content, narrative coherence connecting technical definitions with practical business applications, and extended appropriate length suitable for comprehensive dual-topic educational presentation. Figure 16 presents the synthesized final script that integrates both topics into a coherent educational narrative.



Hello, welcome to this video on understanding agentic AI and its implications. In this video, we'll explore what agentic AI is, how it works, and its potential applications across various industries.

To start, let's define what agentic AI is. Agentic AI refers to artificial intelligence systems that act autonomously to achieve goals without human guidance. These systems can learn, adapt, and operate independently, making decisions without constant human intervention. This is different from traditional AI models, which typically require human input and oversight.

One key characteristic of agentic AI is its autonomy. This means that agentic AI systems can perform tasks on their own with minimal human intervention. They're also goal-oriented, designed to achieve specific objectives. And they're adaptable, able to adjust to complex environments and make decisions based on the situation.

To understand how agentic AI works, let's consider an analogy. Think of agentic AI like a self-driving car. The car can navigate through traffic, make decisions about when to turn or brake, and adjust its route in real-time. This is similar to how agentic AI systems operate, using complex algorithms and data to make decisions and take actions.

Agentic AI systems can be built using various technologies, including Large Language Models, or LLMs, and chain of thought reasoning. These systems can be compared to workflows that use LLMs but are not agentic in nature. The key difference is that agentic AI systems can operate independently, while traditional workflows require human oversight.

The applications of agentic AI are vast and varied. For example, in customer service, AI agents can handle customer inquiries, check user balances, and recommend accounts to pay off balances. In IT and operations, agentic AI can enhance productivity and operations across various industries.

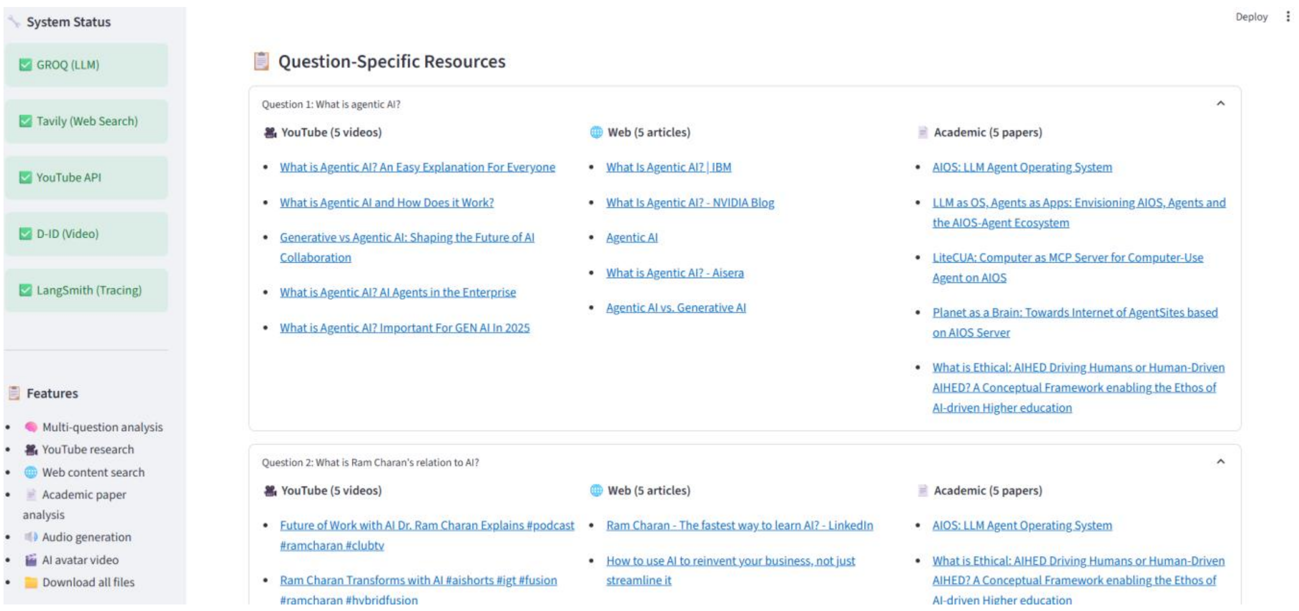
As we consider the potential of agentic AI, it's also important to think about the ethics surrounding its use. Agentic AI raises questions about accountability and transparency, which need to be addressed. For instance, if an agentic AI system makes a decision that has negative consequences, who's responsible? How can we ensure that these systems are transparent and explainable?

Now, let's shift gears and talk about Ram Charan, a well-known expert in business and leadership. While Ram Charan is not directly involved in the development of agentic AI, he has shared his insights on the importance of understanding and leveraging AI in business. He emphasizes that AI is not going away and that leaders should take responsibility for understanding the tools that exist.

Figure 16 Final synthesized script for the multi-question query

4.3.4 Question-Specific Resource Grouping

The resources tab demonstrates intelligent organization by grouping citations according to the specific questions identified during decomposition. Question 1 Resources on "agentic AI" include YouTube educational videos focusing on agentic AI concepts, web articles covering agentic AI applications and theory, and academic papers on agentic AI systems and methodologies. Question 2 Resources on "Ram Charan" feature YouTube videos with Ram Charan's business insights, web articles covering his AI commentary and business transformation insights, and academic papers referencing business AI leadership and organizational transformation. Figure 17 displays resources grouped per question, separating YouTube videos, web articles, and academic papers for each detected topic.


System Status

✅ GROQ (LLM)

✅ Tavily (Web Search)

✅ YouTube API

✅ D-ID (Video)

✅ LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Deploy

Question-Specific Resources

Question 1: What is agentic AI?

YouTube (5 videos)

- What is Agentic AI? An Easy Explanation For Everyone
- What is Agentic AI and How Does it Work?
- Generative vs Agentic AI: Shaping the Future of AI Collaboration
- What is Agentic AI? AI Agents in the Enterprise
- What is Agentic AI? Important For GEN AI In 2025

Web (5 articles)

- What Is Agentic AI? | IBM
- What Is Agentic AI? - NVIDIA Blog
- Agentic AI
- What is Agentic AI? - Aisera
- Agentic AI vs. Generative AI

Academic (5 papers)

- AIOS: LLM Agent Operating System
- LLM as OS, Agents as Apps: Envisioning AIOS, Agents and the AIOS-Agent Ecosystem
- LiteCUA: Computer as MCP Server for Computer-Use Agent on AIOS
- Planet as a Brain: Towards Internet of AgentSites based on AIOS Server
- What is Ethical: AIHED Driving Humans or Human-Driven AIHED? A Conceptual Framework enabling the Ethos of AI-driven Higher education

Question 2: What is Ram Charan's relation to AI?

YouTube (5 videos)

- Future of Work with AI Dr. Ram Charan Explains #podcast #ramcharan #clubtv
- Ram Charan Transforms with AI #aishorts #igt #fusion #ramcharan #hybridfusion

Web (5 articles)

- Ram Charan - The fastest way to learn AI? - LinkedIn
- How to use AI to reinvent your business, not just streamline it

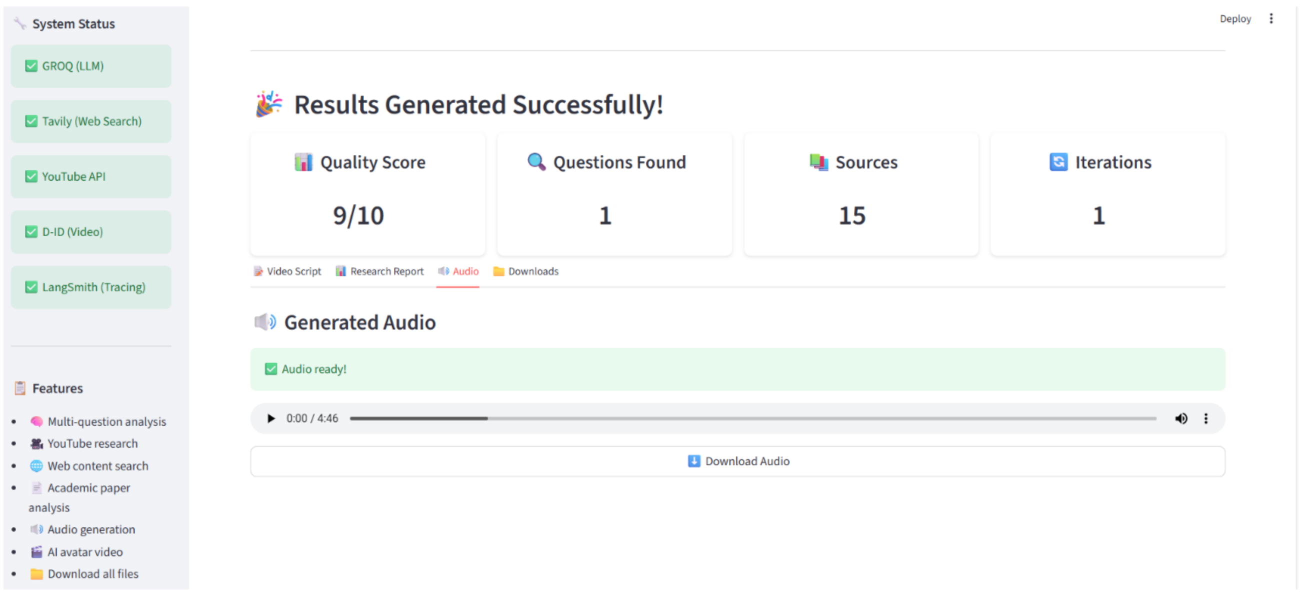
Academic (5 papers)

- AIOS: LLM Agent Operating System
- What is Ethical: AIHED Driving Humans or Human-Driven AIHED? A Conceptual Framework enabling the Ethos of AI-driven Higher education

Figure 17 Question-wise resource grouping for the multi-question scenario

4.3.5 Multi-Question Audio Generation Complete

The extended script covering both educational topics is successfully processed into a comprehensive audio presentation. The audio generation demonstrates extended duration handling without truncation, smooth topic transitions between technical and business content, consistent narration quality across both subject areas, and professional educational delivery with appropriate emphasis and clear articulation. Figure 18 shows the completed audio narration for the multi-question script, with full playback controls.


System Status

✅ GROQ (LLM)

✅ Tavily (Web Search)

✅ YouTube API

✅ D-ID (Video)

✅ LangSmith (Tracing)

Features

- Multi-question analysis
- YouTube research
- Web content search
- Academic paper analysis
- Audio generation
- AI avatar video
- Download all files

Deploy

Results Generated Successfully!

Quality Score	Questions Found	Sources	Iterations
9/10	1	15	1

Video Script | Research Report | Audio | Downloads

Generated Audio

✅ Audio ready!

0:00 / 4:46

Download Audio

Figure 18 Completed audio generation for the multi-question query

4.3.6 Multi-Question Video Generation Complete

The final AI avatar video successfully handles the extended, dual-topic educational content with sophisticated performance. The video demonstrates sustained visual quality throughout the longer presentation, accurate lip-sync maintained across topic transitions, consistent performance across different subject matters, and scalability for comprehensive educational topics of varying complexity. Figure 19 presents the finalized AI-avatar video for the multi-question scenario, ready for playback and export.

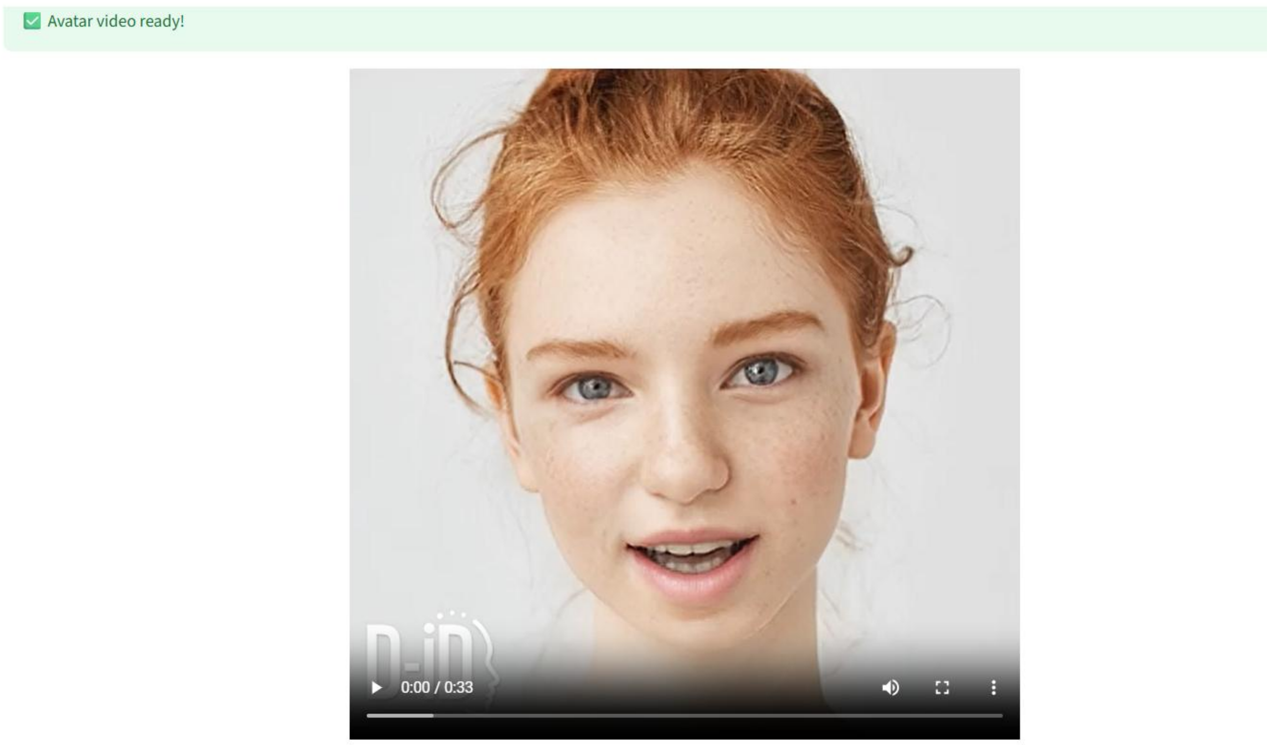
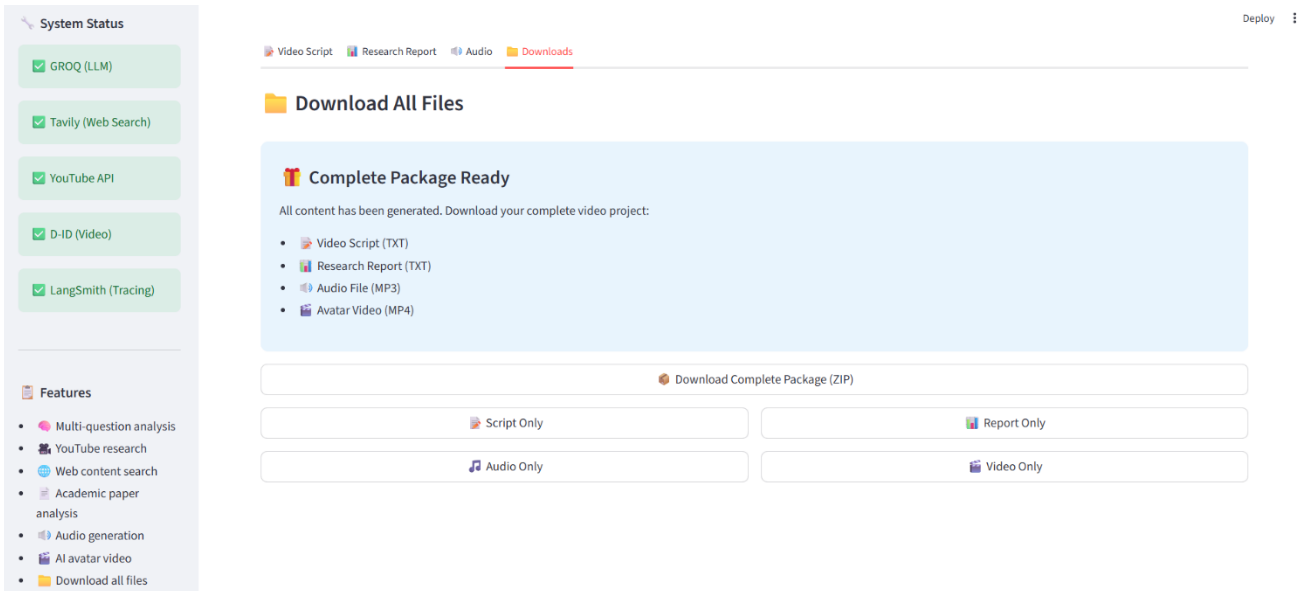


Figure 19 Completed AI-avatar video for the multi-question query

4.4 Download and Export Capabilities

The system provides comprehensive export functionality enabling complete content package distribution. Features include complete ZIP package with all generated files bundled with comprehensive metadata, individual component downloads for Script (TXT), Audio (MP3), Video (MP4), and detailed Research Report (TXT) files, comprehensive research reports with complete source citations and quality metrics, and metadata tracking with generation timestamps and system configuration details for content provenance and quality assurance documentation. Figure 20 shows the download and export panel, where users can retrieve the complete ZIP package or download individual assets—script (TXT), audio (MP3), video (MP4), and the research report with citations.



Download All Files

Complete Package Ready

All content has been generated. Download your complete video project:

- Video Script (TXT)
- Research Report (TXT)
- Audio File (MP3)
- Avatar Video (MP4)

Figure 20 Download and export options in the Streamlit interface

Chapter 5 Conclusion

The Agentic AI Video Synthesizer system was conceived and developed as a fully automated, multi-agent framework capable of generating educational video content from natural language queries. It bridges several emerging technologies including retrieval-augmented generation, agentic orchestration, structured summarization, and synthetic media rendering—combining them into a unified end-to-end content creation pipeline.

The primary goal was to design an AI-powered system that could dynamically decompose user questions, gather multi-source research in real time, synthesize a coherent educational script, and generate both audio and avatar-narrated video responses—entirely autonomously. Through modular design and careful orchestration using LangGraph, the project successfully met these objectives. The system’s agents—comprising decomposer, retriever, synthesis, feedback, and completeness checkers—worked in coordination to ensure factual correctness, script fluency, and completeness of answers.

Performance evaluation across both single-question and multi-question scenarios confirmed system stability, agent reliability, and accurate execution of each processing stage. The system consistently produced high-quality educational scripts, verified research sources, and natural-sounding TTS narration, culminating in visually engaging AI avatar videos. The inclusion of performance metrics such as quality score, iterations, agent completions, and citation count provided a transparent and traceable mechanism for verifying research quality and system completeness.

A major strength of the system lies in its modular multi-agent design built on LangGraph. This architecture enabled conditional execution paths, recursive retries, agent memory tracking, and state persistence—essential for handling diverse and ambiguous user inputs. Furthermore, the use of meta-llama-3.3-70b-versatile LLM via Groq API allowed for high-speed, low-latency inference that supported real-time interaction in the Streamlit frontend.

By integrating Tavily, YouTube, and arXiv as tool agents, the system ensured a tri-modal research strategy that spans web articles, video tutorials, and scholarly papers. Each research summary was grounded strictly in retrieved content, thereby reducing hallucination risks and enhancing educational reliability. Final outputs were rendered as both downloadable reports and video files, making the platform suitable for learners, educators, and institutions looking to automate explainer video production.

Overall, the Agentic AI Video Synthesizer demonstrates the viability and effectiveness of agentic reasoning frameworks for educational automation, providing a scalable and extensible platform for AI-generated learning content.

5.1 Future Work

While the current system delivers a robust implementation of AI-driven educational content generation, several opportunities for expansion and refinement have been identified during development:

1. Expand Multi-Language Support

Although gTTS supports multilingual synthesis, the current LLM agents operate only in English. Future versions should integrate translation agents or multilingual LLMs to support non-English queries and output narration in multiple languages, enhancing accessibility.

2. Interactive Script Editing Interface

An inline editor allowing users to review and optionally revise the generated script before initiating audio or video rendering would enhance control and personalization. Advanced options could include tone adjustment, summary length preference, or sentence-level regeneration.

3. User Authentication and History Tracking

Adding a secure user login and session history tracking system would enable users to revisit past queries, download previous content, and compare outputs over time—useful for educators managing a series of topics.

4. Enhanced Feedback Agent with Explainability

The current feedback agent provides a scalar score. Future iterations could include rubric-based scoring with breakdowns on clarity, factual coverage, structure, and engagement. This would aid debugging and allow users to understand script strengths and weaknesses.

5. Dynamic Research Agent Routing

The React Supervisor currently chooses tools based on heuristics. Integrating a lightweight classifier that routes sub-questions to agents based on query domain (e.g., conceptual, technical, tutorial) could improve retrieval efficiency and reduce agent calls.

6. Integration with Advanced Avatar Rendering Tools

While D-ID performs well for general narration, integration with avatar rendering tools that support hand gestures, scene backgrounds, or emotion alignment (e.g., Synthesia, HeyGen) could create more immersive educational videos.

7. Long-Term Memory and Vector Store Embedding

The current system is stateless beyond session scope. Integration with vector databases like Pinecone or Chroma could support knowledge retention, personalized response generation, and cross-session learning continuity.

5.2 Reflection

The development of the Agentic AI Video Synthesizer has been a transformative experience, offering hands-on engagement with modern AI tools, multi-agent orchestration, and real-time system deployment. It combined elements of software architecture, prompt engineering, API integration, and human-computer interaction into a single, multifaceted project.

Key Learnings

One of the most valuable learnings was the importance of agent modularity and state management. Early prototypes faced issues with sub-question skipping and overwritten outputs due to untracked transitions. Transitioning to LangGraph allowed the project to introduce persistent memory, agent completion flags, and conditional logic—greatly improving system reliability.

The use of llama-3.3-70b-versatile over previously considered GPT-4 and Claude models was another key decision. Hosted via Groq, the model delivered impressive response speeds and minimal latency, supporting a smoother frontend experience. This trade-off between scale and latency proved essential for real-time educational interaction.

Prompt Engineering and Failure Recovery

Fine-tuning agent prompts was a more iterative and time-consuming process than initially expected. Avoiding hallucinations, enforcing source-only summarization, and formatting outputs correctly all required substantial experimentation with prompt templates. Additionally, building a completeness checker and feedback-based retry loop ensured the system could self-heal in the event of weak or failed tool outputs.

Challenges and Adaptations

Integrating D-ID's video rendering posed challenges, especially around polling job completion and handling long scripts. This was resolved through retry loops, timeout handling, and script trimming safeguards. Similarly, coordinating simultaneous agent execution in LangGraph without creating race conditions demanded precise state isolation and loopback logic.

Project Management and Workflow

An agile workflow with weekly milestones and sprint-based development allowed for incremental testing and rapid iteration. Tools like GitHub (for version control), LangSmith (for tracing), and Streamlit (for UI) helped ensure transparency and reproducibility at every step.

What Could Have Been Done Differently

Given more time and resources, deeper usability testing with intended users—such as educators or learners—could have yielded valuable insights into UX design and personalization features. A more scalable backend with persistent memory, database integration, and long-term usage logging would have enabled broader deployment scenarios.

Another improvement would involve dynamic script optimization techniques, including chunking long queries into video series, or auto-tagging generated content with relevant metadata for learning management systems.

Personal and Professional Growth

This project significantly enhanced technical skills in:

- LangChain and LangGraph orchestration
- Real-time multi-agent communication
- Tool-augmented LLM reasoning
- Audio-visual media integration via D-ID and gTTS
- UI development with Streamlit

More importantly, it deepened conceptual understanding of how intelligent agents, retrieval pipelines, and language models can be harnessed to produce creative, educational outcomes—demonstrating the power of AI to democratize access to high-quality knowledge.

References

[1] A. Schick, J. Schütze, and H. Schütze, "Toolformer: Language models can teach themselves to use tools," *arXiv preprint arXiv:2302.04761*, 2023. [Online]. Available: https://arxiv.org/abs/2302.04761

[2] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "ReAct: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022. [Online]. Available: https://arxiv.org/abs/2210.03629

[3] LangChain Team, "LangGraph: Stateful multi-agent graph-based LLM workflows," *LangChain Documentation*, 2023. [Online]. Available: https://docs.langchain.com/langgraph/

[4] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *Proc. 61st Annual Meeting of the Association for Computational Linguistics*, pp. 2609-2634, 2023. [Online]. Available: https://aclanthology.org/2023.acl-long.147/

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459-9474, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[6] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," *Proc. Conference on Empirical Methods in Natural Language Processing*, pp. 6769-6781, 2020. [Online]. Available: https://arxiv.org/abs/2004.04906

[7] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," *Proc. International Conference on Machine Learning*, vol. 162, pp. 19893-19905, 2023. [Online]. Available: https://arxiv.org/abs/2302.00093

[8] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," *Proc. International Conference on Learning Representations*, 2023. [Online]. Available: https://arxiv.org/abs/2210.02406

[9] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-35, 2023. [Online]. Available: https://arxiv.org/abs/2107.13586

[10] H. Li, R. He, and H. Zheng, "AutoGPT: Building autonomous AI agents with language models," *GitHub Repository*, 2023. [Online]. Available: https://github.com/Torantulino/Auto-GPT

Appendices

Appendix A: Project Proposal

This appendix provides a direct reference to the original project proposal that was submitted during the initial planning phase of the MSc project titled Agentic AI Video Synthesizer. The proposal outlines the project's background, problem statement, objectives, planned methodology, key technologies, and expected deliverables. It served as the conceptual and technical foundation for the final system implementation.

The complete project proposal is available for review within the designated GitHub repository, accessible via the link provided below.

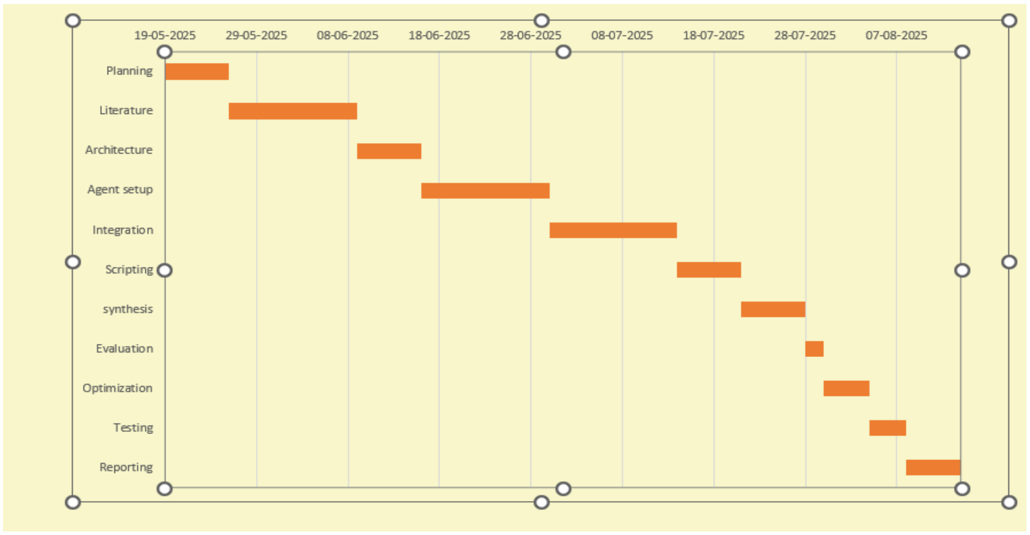
Appendix B: Project Management

This appendix presents the Gantt chart used as the primary project management artefact for planning and monitoring the development of the Agentic AI Video Synthesizer system. The chart outlines the sequential and overlapping phases of the project, from initial planning through to final reporting, providing a clear visual representation of the timeline, task allocation, and milestone tracking.

The Gantt chart illustrates key phases of the development lifecycle, including:

- **Planning and Literature Review** – establishing requirements and surveying relevant research.
- **System Architecture and Agent Setup** – designing the framework and configuring AI agents.
- **Integration and Scripting** – merging components and generating core video synthesis scripts.
- **Synthesis, Evaluation, and Optimization** – refining outputs through iterative testing and quality assessment.
- **Testing and Reporting** – conducting final validation and preparing project documentation.

The artefact facilitated effective time management, task dependency tracking, and workflow alignment with weekly sprint reviews. It enabled the identification of potential bottlenecks, ensured adherence to deadlines, and maintained transparency across the development process.



Appendix C: Artefact

The developed artefact for the Agentic AI Video Synthesizer project includes the complete source code, module scripts, configuration files, and documentation required to replicate, deploy, and evaluate the system. These materials collectively represent the full implementation of the research pipeline, encompassing multi-agent orchestration, content synthesis, and media generation.

The entire artefact has been organized and made available in a structured repository. Access details, including the repository link and navigation instructions, are provided separately. The contents include:

- Python scripts for LangGraph-based agent orchestration
- Prompt templates and configuration files for LLM interaction
- Integration modules for Tavily, YouTube API, arXiv API, gTTS, and D-ID
- Frontend components developed with Streamlit
- Output samples including video, audio, and synthesized scripts
- README documentation with environment setup and usage guidelines

This artefact serves as a reproducible foundation for future development, experimentation, and educational deployment.

GitHub repository link: https://github.com/rajubaddela1234/Agentic-AI-Video-Synthesizer

Appendix D: Screencast

A video presentation has been prepared showcasing both the code walkthrough and a live demonstration of the project’s functionality.

Project Demonstration Link: https://youtu.be/yik18LctNso