Project Synopsis

on

AI-Driven Knowledge Synthesis Platform

in partial fulfilment of requirements for the degree of

Bachelor of Technology

in

Computer Science & Engineering

(AI specialization with IBM)

Submitted by:

Umang Goswami (22100BTCSAII11062)

Shreyansh Gupta (22100BTCSAII11052)

Naman Mathur (22100BTCSAII11027)

Under the guidance of

Prof. Neeraj Mehta

Department of Computer Science & Engineering

Shri Vaishnav Institute of Information Technology

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

January – June 2025

Table of Contents

ABSTRACT

This project aims to come up with a unified AI-driven system that processes diverse input formats, like PDFs, presentations, e-books, research papers, spreadsheets, and web articles, using Retrieval-Augmented Generation (RAG) pipelines and NLP models to create tailored outputs like summaries, quizzes, flashcards, and podcasts.

The AI-Driven Knowledge Synthesis Platform is the solution to the challenged manual content curation for the education and professional workflows through the automation of unstructured data, the structure, and collaborative learning resort into multimodal that is possible. The procedures for study materials, reports, or training modules are usually time-consuming and lack personalization. They are often leading to cognitive overload.

The platform is powered by LangChain for document processing, Hugging Face Transformers for semantic analysis, and vector databases (FAISS) for efficient knowledge retrieval. Its modular architecture provides scalability and customization for various use cases, which can be deployed on Microsoft Azure.

The solution is addressing two distinct domains: education (automating the process and preparing the course material for institutions); corporate training (reporting the programs and upskilling programs).

The platform actualizes the fusion of raw data and actionable insights by enabling content creation enhancement up to 70%, and hence, allowing users to concentrate on higher-order tasks. The power of multimodal outputs, such as podcasts and interactive slides, makes it a platform that can be adapted to modern learning and productivity needs.

1. INTRODUCTION

Access to knowledge has vastly increased, but making meaningful decisions remains a daunting challenge. In academic and professional environments, people often waste their precious time on manual content curation as this may in, which may involve selecting papers, and articles to be included in training modules or reports.

Existing tools, such as a summarization tool that gives an abstract of a document using basic APIs or a template-based platform, cannot establish relationships between different documents or adapt the result to user needs. This issue is one of the many that have to be solved in artificial intelligence and knowledge management: the scarcity of scalable, multimodal synthesis systems.

Some of the recent NLP technologies, such as transformer-based architectures including BERT and GPT-4 or Retrieval-Augmented Generation (RAG) framework, have made it possible for computers to physics and generate text similar to the human one. Superficially, though, most systems are trained on a single task like document summarization or chatbot conversation.

The very few systems are mostly conventional and focused on the narrow aspects of the problem, while the most difficult part is the ones which are designed to be flexible. For instance, platforms like Quizlet depend largely on user-generated content. In contrast, business giants like SharePoint do not have access to such sophisticated and automated toolkits.

This endeavour is part of the rapidly growing space of AI-augmented productivity platforms, the main aim of which is the development of a tool that can not only handle various types of inputs but also generate outputs dynamically in education and professional workflow. The work described in this paper is part of the whole AI productivity tools offering a bridge between the academia and the industry practicality.

The search for semantic, module NLP pipes and the generation of the multi-modal generation are the frontier of the research and application of this system. The advancement in scalable knowledge synthesis research has been achieved while it gives companies the needed ability in terms of effectiveness and adaptability.

2. PROBLEM DOMAIN

The problem domain of this project revolves around the inefficiencies and limitations in current methods of knowledge synthesis and content creation. Manual curation of content is not only time-consuming but also prone to inconsistencies, making it difficult to produce high-quality, personalized outputs like study guides, training modules, or reports.

The creation of this solution is driven by three core problems plaguing current knowledge synthesis workflows:

i. Inefficient Manual Content Curation;

ii. Limitations of Existing Tools especially in Cross-Document Analysis, and, Personalization of outputs; and,

iii. Absence of Multimodal Outputs

These problems lead to broken workflows, lower productivity, and mental strain for users. Take educators and students putting together course materials as an example – They have to switch between many tools to make summaries, presentations, mind-maps, practice tests, and quizzes. In the same way, professionals find it hard to pull together reports from different data sources.

Current AI-powered tools like Quizlet (which deals with user-made flashcards) or Grammarly (which focuses on checking grammar) tackle just small parts of this issue leaving the bigger problem unsolved.

2.1. Objectives of the Proposed Work

i. Automate the conversion of unstructured data into structured, multimodal outputs.

ii. Enable cross-format and cross-document knowledge synthesis.

iii. Dynamically tailor outputs to user needs (e.g., learning styles, professional standards).

iv. Ensure scalability for institutional and enterprise deployment.

3. SOLUTION DOMAIN

The AI-Driven Knowledge Synthesis Platform adopts a modular, end-to-end architecture designed to automate content curation and deliver tailored, multimodal outputs. The solution comprises four core components:

3.1. Document Processing & Knowledge Extraction

i. **Multi-Format Ingestion:** A unified parser converts PDFs, spreadsheets, and web articles into structured text using Python libraries. LangChain orchestrates document preprocessing, splitting content into semantically meaningful chunks.

ii. **Semantic Embedding:** Hugging Face and LangChain Transformers generate vector embeddings for each chunk which shall be stored in a vector database. This enables efficient similarity searches across documents.

3.2. Cross-Document Knowledge Synthesis

i. **Retrieval-Augmented Generation (RAG):** User queries trigger a hybrid search (keyword + vector similarity) to retrieve relevant document chunks. These chunks are fed into a fine-tuned transformer model to generate context-aware outputs.

ii. **Cross-Format Linking:** Metadata tagging ensures outputs like summaries or reports integrate insights from heterogeneous sources.

3.3. Multimodal Output Generation

Dynamic Content Engine:

i. **Summaries:** Extractive and abstractive summarization pipelines prioritize key concepts based on user roles

ii. **Quizzes & Flashcards:** A question-generation model creates MCQs and flashcards

iii. **Podcasts:** Text-to-speech APIs convert summaries into audio, with adjustable pacing and language.

3.4. Deployment

The platform may be containerized using Docker and deployed on Microsoft Azure

3.5. Technical Innovation

i. **Unified Processing Pipeline:** Combines RAG, cross-document linking, and multimodal generation in a single workflow.

ii. **Plagiarism-Safe Outputs:** All generated content references source document embeddings, ensuring traceability and reducing redundancy.

4. SYSTEM DOMAIN

4.1. Backend

i. **Core Language:** Python, chosen for its extensive NLP/ML libraries and compatibility with AI frameworks.

ii. **Document Processing and NLP Models:** LangChain orchestrates RAG pipelines for document chunking, embedding, and retrieval; LangChain Transformers handle text generation, summarization, and question-answering.

iii. **Vector Database:** FAISS (Facebook AI Similarity Search)

4.2. Frontend

Gradio serves as the lightweight backend server, with HTML, CSS, and JavaScript for dynamic user interactions. For data visualization purposes Chart.js shall be used.

4.3. Development Environment and Version Control

i. **IDE:** PyCharm (Professional Edition) for backend python development, and, WebStorm for professional frontend development

ii. **Version Control:** Git and GitHub

4.4. Deployment Environment

The platform is hosted on Microsoft Azure enabling seamless deployment and management of the application. Azure's built-in security protocols and global server

infrastructure ensure reliable access for users worldwide, while auto-scaling capabilities handle fluctuating demand without manual intervention.

5. APPLICATION DOMAIN

The platform is designed for two primary domains: education and professional training.

In academic settings, it empowers educators to automate course material creation (e.g., lecture slides, quizzes) and enables students to generate personalized study aids (flashcards, audio summaries).

For professionals, it streamlines the synthesis of reports, compliance documents, and interactive training modules from diverse data sources like spreadsheets and web articles. Its adaptability allows customization for niche use cases, such as research labs compiling literature reviews or corporations developing onboarding programs.

6. EXPECTED OUTCOME

The proposed platform is designed to deliver measurable improvements in knowledge synthesis and content creation workflows. Key outcomes include:

- **Automated Content Generation:** AI-driven creation of summaries, quizzes, and flashcards, reducing manual curation time by 70%.
- **Cross-Format Knowledge Synthesis:** Unified insights from PDFs, spreadsheets, and articles for holistic analysis.
- **Multimodal Outputs:** Podcasts, slides, and quizzes tailored to diverse learning and professional needs.

7. REFERENCES

[1] “Azure AI Search Documentation,” 25 February 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/search/. [Accessed 10 March 2025].

[2] “FAISS GitHub Wiki,” Meta AI Research, 24 February 2025. [Online]. Available: https://github.com/facebookresearch/faiss/wiki. [Accessed 10 March 2025].

[3] “Flask Documentation (3.1.x),” Pallets, 5 January 2025. [Online]. Available: https://flask.palletsprojects.com. [Accessed 10 March 2025].

[4] “Gradio Documentation,” Gradio, 9 March 2025. [Online]. Available: https://www.gradio.app/docs. [Accessed 10 Marcj 2025].

[5] “LangChain Documentation,” LangChain, Inc., 30 January 2025. [Online]. Available: https://python.langchain.com/docs/introduction/. [Accessed 10 March 2025].

[6] “LangChain HuggingFace Integrations Documentations,” LangChain, Inc., 16 October 2024. [Online]. Available: https://python.langchain.com/docs/integrations/providers/huggingface/. [Accessed 10 March 2025].

[7] “LangChain Python API Reference,” LangChain Inc., 2 March 2025. [Online]. Available: https://python.langchain.com/api_reference/. [Accessed 10 March 2025].