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), the Project Lead

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

Project Synopsis on

AI-Driven Knowledge Synthesis Platform

Submitted by: Umang Goswami (22100BTCSAII11062), the Project & Team Lead; Shreyansh Gupta (22100BTCSAII11052), Team Member; Naman Mathur (22100BTCSAII11027), Team Member

Degree Program: B.Tech. Computer Science & Engineering (AI specialization with IBM); Year: III; Semester: VI; Section: U

ABSTRACT

The AI-Driven Knowledge Synthesis Platform addresses the inefficiencies in manual content curation for educational and professional workflows by automating the conversion of unstructured data into structured, multimodal learning resources. Traditional methods of creating study materials, reports, or training modules are time-consuming and lack personalization, often leading to cognitive overload.

This project proposes a unified AI-powered system that processes diverse input formats – PDFs, presentations, ebooks, research papers, spreadsheets, and web articles – using Retrieval-Augmented Generation (RAG) pipelines and NLP models to generate tailored outputs like summaries, quizzes, flashcards, and podcasts.

The platform leverages LangChain for document processing, Hugging Face Transformers for semantic analysis, and vector databases (Pinecone/FAISS) for efficient knowledge retrieval. Its modular architecture supports scalability and customization for diverse use cases, making it deployable on Microsoft Azure.

The solution targets two key domains: education (automating course material creation for institutions) and corporate training (accelerating report generation and upskilling programs).

By bridging the gap between raw data and actionable insights, the platform reduces content creation time by up to 70%, enabling users to focus on higher-order tasks. Its integration of multimodal outputs, such as podcasts and interactive slides, ensures adaptability to modern learning and productivity demands.

1. INTRODUCTION

The rapid growth of digital information has created a paradox: while access to knowledge is unprecedented, synthesizing unstructured data into actionable insights remains a significant challenge. In educational and professional settings, users spend excessive time manually curating content from PDFs, research papers, and web articles into formats like study guides, training modules, or reports.

Existing tools, such as basic summarization APIs or template-driven platforms, lack the ability to contextualize cross-document relationships or adapt outputs to user-specific needs. This inefficiency underscores a critical gap in the intersection of artificial intelligence and knowledge management: the absence of scalable, multimodal synthesis systems.

Recent breakthroughs in NLP, particularly transformer-based architectures (e.g., BERT, GPT-4) and Retrieval-Augmented Generation (RAG), have enabled machines to comprehend and generate human-like text. However, most implementations focus on narrow tasks like single-document summarization or chatbot interactions.

Few address the challenge of aggregating insights from heterogeneous formats (PDFs, spreadsheets, etc.) or generating diverse outputs (quizzes, podcasts) in a unified framework. For instance, platforms like Quizlet rely on user-generated content, while enterprise tools like SharePoint lack AI-driven automation.

This project positions itself within the emerging field of AI-augmented productivity, aiming to bridge these gaps by developing a platform that not only processes multi-format inputs but also dynamically tailors outputs to pedagogical and professional workflows. This work contributes to the broader field of AI-driven productivity tools, offering a solution that bridges academic rigor with real-world applicability.

By integrating semantic search, modular NLP pipelines, and multimodal generation, the system advances research in scalable knowledge synthesis while addressing real-world demands for efficiency 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 issues result in fragmented workflows, reduced productivity, and cognitive overload for users. For example, educators and students compiling course materials must juggle multiple tools to create summaries, presentations, mind-maps, mock-test, and quizzes; while professionals struggle to synthesize reports from disparate data sources.

Existing AI-driven tools, such as Quizlet (limited to user-generated flashcards) or Grammarly (focused on grammar checks), address only narrow slices of this problem, leaving the broader challenge unresolved.

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 (e.g., FLAN-T5 or GPT-3.5) 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; Hugging Face or LangChain Transformers handle text generation, summarization, and question-answering.

iii. **Vector Database:** FAISS, ElasticSearch, Pincone, etc

4.2. Frontend

i. **Web Interface:** Flask serves as the lightweight backend server, with HTML, CSS, and JavaScript for dynamic user interactions.

ii. **Interactive Outputs:** Chart.js for data visualization; and, Azure Text-to-Speech API for podcasts

4.3. Development Enviornment 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.

4.5. Hardware Requirements

i. **Development:** OS: Windows 11 Home Single Language ; CPU: 12th Gen Intel(R) Core (TM) i7-12650H 2.30 GHz ; Installed RAM: 16.0 GB ; CPU Architecture: 64-bit OS, x64-based processor ;

ii. **Deployment:** The Azure Free Tier is leveraged ensuring cost-efficiency during the initial stages. If the product is publicly released and experiences high demand or if users request premium subscriptions, the Azure Basic or Premium tiers may be assigned to handle heavy worldwide production workloads as required.

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.
- **Scalable Deployment:** Seamless transition from Azure Free Tier to enterprise-grade infrastructure as demand grows.

7. REFERENCES

FAISS. (2025). *FAISS GitHub Wiki*. Retrieved from https://github.com/facebookresearch/faiss/wiki

Flask. (2025). *Flask Web Framework Documentation*. Retrieved from https://flask.palletsprojects.com

Gradio. (2025). *Gradio Documentation*. Retrieved from https://www.gradio.app/docs

Hugging Face. (2025). *Chat UI Documentation*. Retrieved from https://huggingface.co/docs/chat-ui/index

Hugging Face. (2025). *Evaluate Library Documentation*. Retrieved from https://huggingface.co/docs/evaluate/index

Hugging Face. (2025). *Hugging Face Generative AI Services (HUGS) Documentation*. Retrieved from https://huggingface.co/docs/hugs/index

LangChain. (2025). *LangChain API Reference*. Retrieved from https://python.langchain.com/api_reference/

LangChain. (2025). *LangChain Documentation*. Retrieved from https://python.langchain.com

LangChain. (2025). *LangChain HuggingFace Integrations*. Retrieved from https://python.langchain.com/docs/integrations/providers/huggingface/

Microsoft Azure. (2025). *Azure AI Search Documentation*. Retrieved from https://learn.microsoft.com/en-us/azure/search

Pinecone. (2025). *Vector Database Documentation*. Retrieved from https://www.pinecone.io/docs

SentenceTransformers. (2025). *SentenceTransformers Documentation*. Retrieved from https://sbert.net/