Multimodal Education Creator

Course Name: Generative AI

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Sr.no	Student Name	Enrolment Number
1	ABHIKET THAKUR	EN22CS301030
2	ADITYA PALIWAL	EN22CS301055
3	AARUSHI MITTAL	EN24CS3T10026
4	AKASH KUSHWAHA	EN22CS301083
5	AKSHAY DASHORE	EN22CS301096
6	AKSHITA RATHORE	EN22CS301100

Institution Name: Medi-caps University – Datagami Skill Based Course

Group: 04D1
Project No: GAI-04
Industry Mentor Name: Aashruti Shah
University Mentor Name: Prof. Ajaj Khan

Department of Computer Science & Engineering
Faculty of Engineering
MEDICAPS UNIVERSITY, INDORE- 453331
Jan - May 2026

1. Problem Statement & Objectives

1.1 Problem Statement

With the rapid growth of digital educational resources, students are exposed to vast amounts of unstructured information such as PDFs, notes, online articles, and textbooks. However, finding precise, topic-specific explanations from large datasets is time-consuming and inefficient.

Traditional keyword-based search systems often:

- Fail to understand semantic meaning
- Return irrelevant or incomplete results
- Do not generate structured explanations
- Require manual content filtering

There is a need for an intelligent system that can understand context, retrieve relevant information, and generate structured educational content automatically. There is a need for an automated system that can:

- Uses Natural Language Processing (NLP)
- Implements Vector Embeddings for semantic understanding
- Utilizes Retrieval-Augmented Generation (RAG)
- Retrieves relevant content using similarity search
- Generates structured, topic-based educational explanations

1.2 Project Objectives

The primary objective of this project is to design and develop an AI-powered Educational Content Creator that leverages Natural Language Processing (NLP), Vector Embeddings, and Retrieval-Augmented Generation (RAG) to provide accurate, context-aware, and structured educational content based on user queries.

The specific objectives of the project are as follows:

- To develop a system capable of understanding the semantic meaning of user input rather than relying solely on keyword matching.
- To implement a vector-based similarity search mechanism for efficient retrieval of relevant educational content.
- To integrate a Retrieval-Augmented Generation (RAG) framework for combining retrieved knowledge with AI-generated explanations.
- To generate concise, structured, and topic-focused educational content automatically.
- To design a modular and scalable system architecture that allows future enhancements and easy maintenance.

- To ensure efficient data storage and retrieval using a vector database.
- To provide a user-friendly interface for seamless interaction with the system.
- To optimize system performance in terms of accuracy, response time, and resource utilization.

Overall, the project aims to enhance learning efficiency by delivering intelligent, context-aware educational assistance through modern AI techniques.

1.3 Scope of the Project

The scope of this project is to design and implement an AI-powered Educational Content Creator that enables semantic search and automated content generation using Natural Language Processing (NLP), Vector Embeddings, and Retrieval-Augmented Generation (RAG). The system focuses on processing structured educational datasets such as PDFs, lecture notes, and textual study materials to generate accurate and context-aware explanations based on user queries.

The project includes the development of a backend system for text processing, embedding generation, vector storage, similarity-based retrieval, and AI-driven response generation. It also includes a user-friendly interface that allows users to input topics and receive structured educational content in return. The system supports modular architecture, making it scalable and extendable for future enhancements.

The scope covers:

- Semantic search over predefined educational datasets
- Automated note generation based on retrieved content
- Vector database implementation for efficient similarity search
- REST-based API communication between frontend and backend
- Structured and concise educational content generation

However, the project does not include:

- Real-time internet browsing or dynamic web scraping
- Integration with full Learning Management Systems (LMS)
- User authentication and role-based access control (unless extended)
- Handling of sensitive personal data

Overall, the project is limited to creating a standalone intelligent educational assistant that enhances knowledge retrieval and learning efficiency using AI-based techniques.

2. Proposed Solution

The proposed solution is an AI-powered Educational Content Creator that utilizes Natural Language Processing (NLP), Vector Embeddings, and Retrieval-Augmented Generation (RAG) to provide intelligent, context-aware educational assistance.

2.1 Key features

The key features of the proposed system are as follows:

- **Semantic Search Capability:** The system understands the contextual meaning of user queries rather than relying solely on keyword matching. This enables accurate retrieval of relevant information from large educational datasets.
- **Retrieval-Augmented Generation (RAG):** The system combines similarity-based retrieval with AI-driven text generation. Relevant content is first retrieved from the dataset, and then a structured explanation is generated using a language model.
- **Vector-Based Similarity Search:** Educational content is converted into embeddings and stored in a vector database. When a user submits a query, similarity search techniques (e.g., cosine similarity) are used to retrieve the most relevant content chunks.
- **Automated Educational Content Generation:** The system generates concise, structured, and topic-specific explanations, making it useful for quick revision and conceptual understanding.
- **Modular and Scalable Architecture:** The application is designed with separate modules for input processing, embedding generation, retrieval, and content generation. This modular design allows easy maintenance and future upgrades.
- **Efficient Data Handling:** Chunking techniques are used to process large documents into manageable segments, improving retrieval accuracy and performance.
- **REST-Based API Integration:** The system uses RESTful APIs to enable smooth communication between the frontend interface and backend AI modules.
- **User-Friendly Interface:** A simple and interactive web interface allows users to enter topics and receive AI-generated educational content in an easy-to-read format.
- **Extensibility:** The system can be extended in the future to include features such as:

    - Multi-subject dataset integration
    - User authentication
    - Performance analytics
    - Cloud deployment

2.2 Overall Architecture / Workflow

The proposed system follows a modular architecture based on the Retrieval-Augmented Generation (RAG) framework. It integrates data preprocessing, vector storage, semantic retrieval, and AI-based content generation to deliver structured educational responses. The workflow ensures efficient processing, accurate retrieval, and context-aware output generation.

High-Level Architecture

The system consists of the following major components:

1. **User Interface (Frontend)** – Accepts topic input from the user and displays generated educational content.
2. **API Layer (Backend Server)** – Handles requests, coordinates processing modules, and returns responses.
3. **Embedding Module** – Converts text into vector representations.
4. **Vector Database** – Stores embeddings and performs similarity search.

5. **Retrieval Module** – Fetches the most relevant content chunks based on semantic similarity.
6. **Content Generation Module (LLM)** – Generates structured explanations using retrieved content.
7. **Dataset Storage** – Stores educational materials (PDFs, notes, text files).

System Workflow

Step 1: Dataset Preparation

- Educational documents are collected and preprocessed.
- Large documents are divided into smaller chunks.
- Each chunk is converted into vector embeddings.
- Embeddings are stored in the vector database.

Step 2: User Query Input

- The user enters a topic through the web interface.
- The request is sent to the backend API.

Step 3: Query Embedding Generation

- The user query is converted into a vector embedding using the embedding model.

Step 4: Similarity Search

- The vector database compares the query embedding with stored embeddings.
- Top-K most relevant content chunks are retrieved.

Step 5: Content Generation

- Retrieved chunks are passed to the language model.
- The system generates a structured, concise educational explanation.

Step 6: Response Delivery

- The generated content is formatted.
- The response is sent back to the frontend and displayed to the user.

Information Movement Across Layers:

User Layer → Presentation Layer

- User submits topic or prompt
- Request is captured by the web interface

Presentation Layer → Processing Layer

- Input is validated and preprocessed
- Cleaned text is forwarded for embedding generation

Processing Layer → Data Layer

- Query is converted into embedding vector
- Vector is compared with stored content embeddings
- Top-K relevant content chunks are retrieved

Data Layer → Processing Layer

- Retrieved contextual information is assembled
- Structured educational content is generated

Processing Layer → Presentation Layer

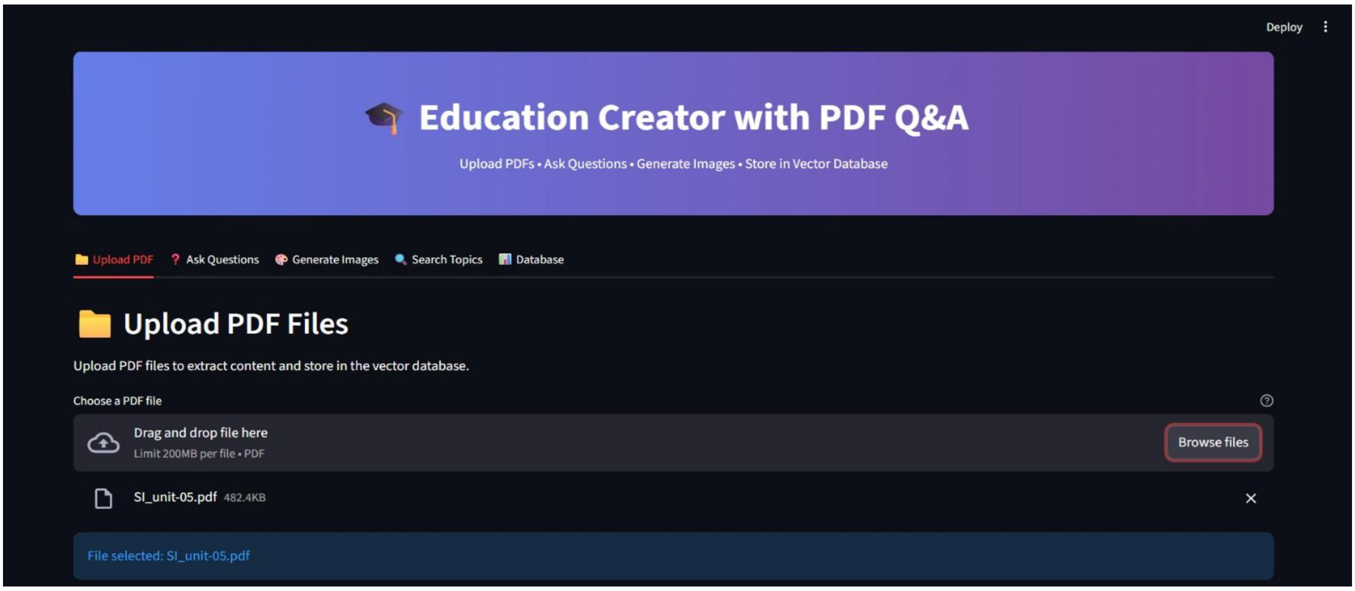
- Final formatted output is returned
- Content is displayed to the user

2.3 Tools & Technologies Used

- **Programming Language:** Python
- **Backend Framework:** FastAPI / Flask
- **Natural Language Processing Libraries**: Hugging Face Transformers, Sentence-Transformers, NLTK
- **Vector Database**: FAISS
- **Frontend Technologies:** HTML, CSS, JavaScript
- **Development & Version Control**: VS Code, Git, GitHub

3. Results & Output

3.1 Screenshots / outputs



🎓 Education Creator with PDF Q&A

Upload PDFs • Ask Questions • Generate Images • Store in Vector Database

Upload PDF Files

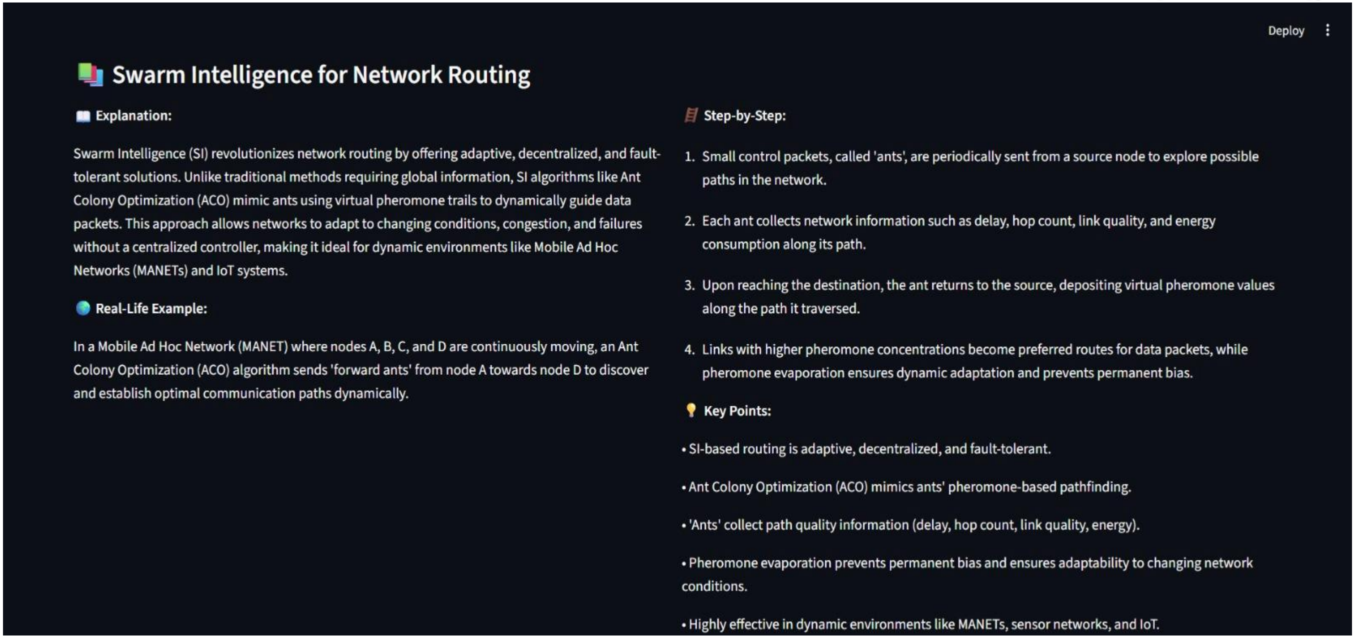
Upload PDF files to extract content and store in the vector database.

Choose a PDF file

Drag and drop file here
Limit 200MB per file • PDF

SI_unit-05.pdf 482.4KB

File selected: SI_unit-05.pdf



📚 Swarm Intelligence for Network Routing

📖 **Explanation:**

Swarm Intelligence (SI) revolutionizes network routing by offering adaptive, decentralized, and fault-tolerant solutions. Unlike traditional methods requiring global information, SI algorithms like Ant Colony Optimization (ACO) mimic ants using virtual pheromone trails to dynamically guide data packets. This approach allows networks to adapt to changing conditions, congestion, and failures without a centralized controller, making it ideal for dynamic environments like Mobile Ad Hoc Networks (MANETs) and IoT systems.

🌍 **Real-Life Example:**

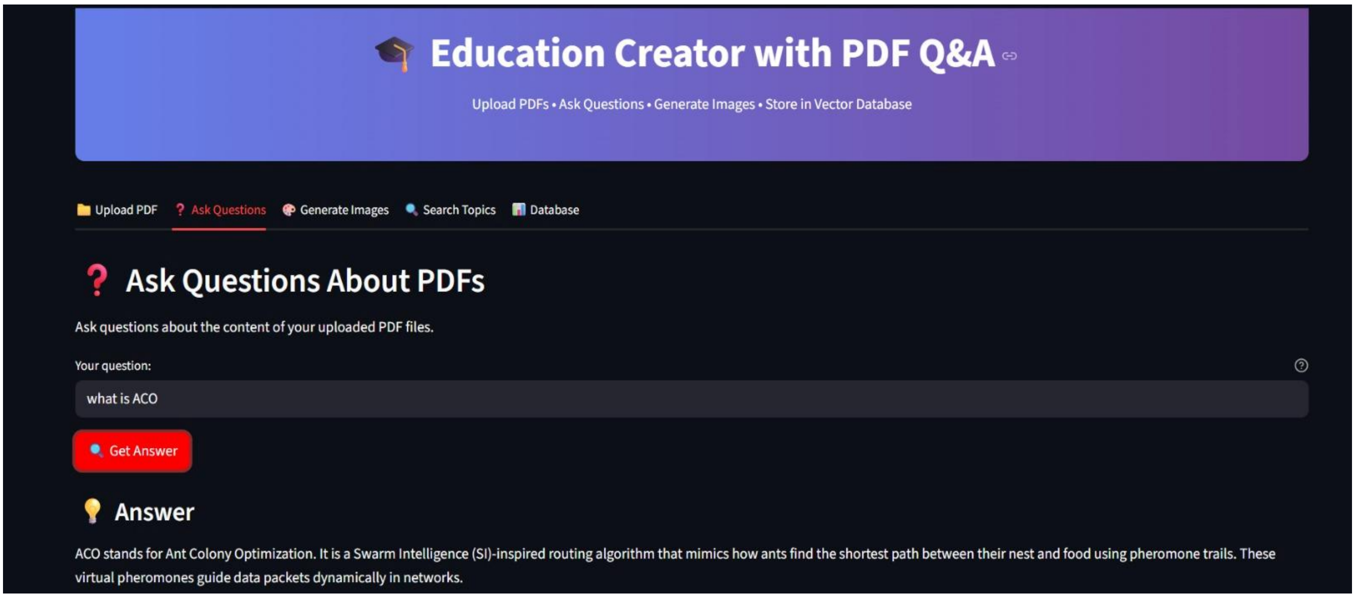
In a Mobile Ad Hoc Network (MANET) where nodes A, B, C, and D are continuously moving, an Ant Colony Optimization (ACO) algorithm sends 'forward ants' from node A towards node D to discover and establish optimal communication paths dynamically.

🪜 **Step-by-Step:**

1. Small control packets, called 'ants', are periodically sent from a source node to explore possible paths in the network.
2. Each ant collects network information such as delay, hop count, link quality, and energy consumption along its path.
3. Upon reaching the destination, the ant returns to the source, depositing virtual pheromone values along the path it traversed.
4. Links with higher pheromone concentrations become preferred routes for data packets, while pheromone evaporation ensures dynamic adaptation and prevents permanent bias.

💡 **Key Points:**

- SI-based routing is adaptive, decentralized, and fault-tolerant.
- Ant Colony Optimization (ACO) mimics ants' pheromone-based pathfinding.
- 'Ants' collect path quality information (delay, hop count, link quality, energy).
- Pheromone evaporation prevents permanent bias and ensures adaptability to changing network conditions.
- Highly effective in dynamic environments like MANETs, sensor networks, and IoT.



🎓 Education Creator with PDF Q&A

Upload PDFs • Ask Questions • Generate Images • Store in Vector Database

❓ Ask Questions About PDFs

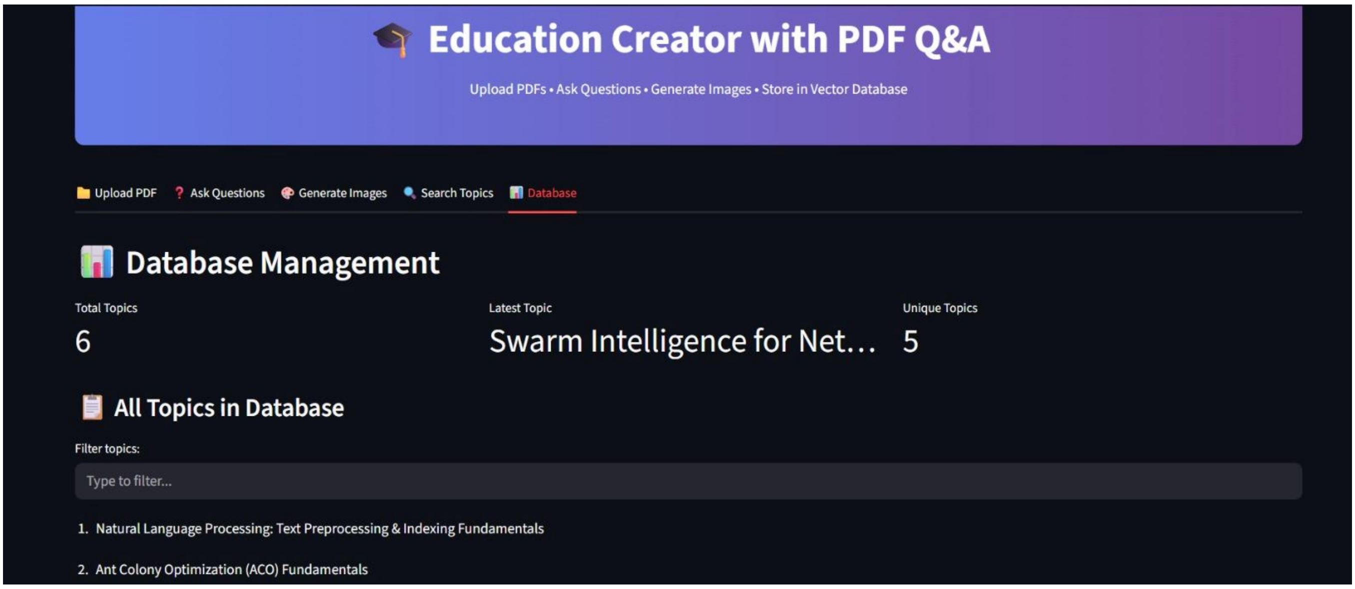
Ask questions about the content of your uploaded PDF files.

Your question:

what is ACO

💡 Answer

ACO stands for Ant Colony Optimization. It is a Swarm Intelligence (SI)-inspired routing algorithm that mimics how ants find the shortest path between their nest and food using pheromone trails. These virtual pheromones guide data packets dynamically in networks.



🎓 Education Creator with PDF Q&A

Upload PDFs • Ask Questions • Generate Images • Store in Vector Database

📊 Database Management

Total Topics: 6

Latest Topic: Swarm Intelligence for Net…

Unique Topics: 5

📋 All Topics in Database

Filter topics:

Type to filter...

1. Natural Language Processing: Text Preprocessing & Indexing Fundamentals
2. Ant Colony Optimization (ACO) Fundamentals

3.2 Reports / dashboards / models

AI Models Used

The system utilizes the following models:

- **Embedding Model** – Converts text data and user queries into dense vector representations for semantic similarity comparison.
- **Similarity Search Model (Vector Index)** – Performs cosine similarity or nearest-neighbor search to retrieve relevant content chunks.
- **Language Model (LLM)** – Generates structured and context-aware educational explanations using retrieved content.

These models collectively implement the Retrieval-Augmented Generation (RAG) framework.

System Reports

The system can generate or maintain basic reports such as:

- Query logs (frequently searched topics)
- Retrieval performance statistics
- Response time monitoring
- Dataset processing summary
- Embedding generation status

These reports help in evaluating system efficiency and improving retrieval accuracy.

Dashboard (Optional Enhancement)

A monitoring dashboard can be implemented to display:

- Total number of queries processed
- Most searched topics
- Average response time
- System health status
- Dataset size and embedding count

The dashboard provides real-time insights into system performance and usage trends.

3.3 Key outcomes

The successful completion of this project resulted in the following key outcomes:

- Developed a fully functional AI-Powered Educational Content Creator using NLP and RAG architecture.
- Implemented semantic search using vector embeddings instead of traditional keyword-based methods.
- Integrated a vector database for efficient similarity-based retrieval of relevant content.
- Successfully combined retrieval mechanisms with AI-driven content generation for structured educational output.
- Designed a modular and scalable system architecture for easy maintenance and future enhancements.
- Built RESTful APIs to enable smooth frontend-backend communication.
- Improved understanding of real-world AI system design and deployment workflow.
- Gained hands-on experience with embedding models, vector databases, and Retrieval-Augmented Generation frameworks.

4. Conclusion

This project successfully developed an **AI-Powered Educational Content Creator** using Natural Language Processing (NLP), Vector Embeddings, and Retrieval-Augmented Generation (RAG). The system overcomes the limitations of traditional keyword-based search by enabling semantic understanding and context-aware educational content generation. By integrating similarity-based retrieval with AI-driven text generation, the application delivers structured and accurate topic-based explanations.

During the development process, a modular architecture was implemented, including dataset preprocessing, embedding generation, vector storage, retrieval mechanisms, and REST-based API communication. The project provided practical exposure to semantic search, vector databases, RAG implementation, and AI system design.

Overall, this work demonstrates the effective application of modern AI technologies to enhance educational content retrieval and learning efficiency, while also strengthening technical and problem-solving skills.

5. Future Scope & Enhancements

- **Multi-Domain Expansion**: Currently, the system operates on predefined educational datasets. In the future, it can be expanded to support multiple subjects, competitive exam preparation materials, technical documentation, and domain-specific knowledge bases.
- **Real-Time Web Integration**: The system can be enhanced to integrate real-time web search capabilities, enabling dynamic content updates and access to the latest educational resources.
- **User Authentication & Personalization**: Future versions can include:
  - User login and registration
  - Personalized learning history
  - Saved notes and bookmarks
  - Recommendation systems based on user activity
- **Advanced Analytics Dashboard**: An analytics module can be added to:
  - Track frequently searched topics
  - Monitor system usage statistic
  - Analyze content performance
  - Improve retrieval accuracy through feedback
- **Cloud Deployment & Scalability**: The system can be deployed on cloud platforms such as AWS, Azure, or Google Cloud to:
  - Support large-scale users
  - Enable load balancing

  - Ensure high availability

- **Improved Retrieval Techniques**: Enhancements may include:
  - Hybrid search (combining keyword + vector search)
  - Fine-tuned embedding models
  - Domain-specific LLM fine-tuning
  - Optimized Top-K retrieval strategies
- **Multilingual Support**: The system can be extended to support multiple languages, making it accessible to a wider audience and enabling regional language education assistance.
- **Voice-Based Interaction**: Speech-to-text and text-to-speech integration can enable voice-based learning assistance, improving accessibility.
- **Integration with Learning Platforms**: Future versions may integrate with:
  - Learning Management Systems (LMS)
  - Online assessment platforms
  - Automated quiz generation systems