



Archives available at [journals.mriindia.com](http://journals.mriindia.com)

**International Journal of Recent Advances in Engineering and Technology**

ISSN: 2347 - 2812

Volume 14 Issue 03s, 2025

## An Automated Retrieval-Augmented Generation Framework for Course Generation

<sup>1</sup>Pradnya Moon, <sup>2</sup>Shubham Peshane, <sup>3</sup>Omm Sawant, <sup>4</sup>Soukhyad Pohane, <sup>5</sup>Pratham Nimje  
<sup>1,2,3,4,5</sup> Dept. of Computer Science & Business Systems, St. Vincent Pallotti College of Engineering & Technology, Nagpur, India

Email: <sup>1</sup>[pradnyamoon@stvincentngp.edu.in](mailto:pradnyamoon@stvincentngp.edu.in), <sup>2</sup>[shubhampeshane.22@stvincentngp.edu.in](mailto:shubhampeshane.22@stvincentngp.edu.in),

<sup>3</sup>[ommsawant.22@stvincentngp.edu.in](mailto:ommsawant.22@stvincentngp.edu.in), <sup>4</sup>[soukhyadpohane.22@stvincentngp.edu.in](mailto:soukhyadpohane.22@stvincentngp.edu.in),

<sup>5</sup>[prathamnimje.22@stvincentngp.edu.in](mailto:prathamnimje.22@stvincentngp.edu.in)

### Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

### Keywords

Artificial Intelligence, RAG, Education, Prompt Engineering, n8n, LLM.

### Abstract

This paper presents an automated framework for course generation designed to address critical inefficiencies in traditional curriculum development. The creation of high-quality educational content is a resource-intensive process that impedes the rapid deployment of customized learning materials in academic and corporate environments. Our proposed system overcomes these challenges through a web-based platform that generates complete, interactive online courses from a single natural language prompt. The architecture leverages a React frontend to capture user requirements and the n8n workflow automation platform to orchestrate a sophisticated backend process. At the core of the framework is a Retrieval-Augmented Generation (RAG) model. Upon receiving a prompt, the system employs a Large Language Model, such as Google Gemini, to first retrieve relevant information from a specified knowledge base and then generate a comprehensive course structure in a standardized JSON format. This output includes pedagogically sequenced modules, detailed lessons, multimedia placeholders, and formative assessments. RAG architecture ensures the generated content is factually grounded in trusted source materials, mitigating the risk of inaccuracies common in standalone language models.

The resulting JSON is dynamically rendered by the React application into an engaging learning interface. This automated approach offers significant advantages over traditional methods by reducing development time from weeks to minutes, enabling unparalleled scalability and personalization for diverse audiences. The framework facilitates a paradigm shift from rigid, one-size-fits-all instruction to a more flexible, on-demand, and interactive educational experience.

### Introduction

The fields of education and corporate training are undergoing a significant transformation, driven by the need for rapid and customized development of effective learning materials. The traditional process of curriculum design is frequently a bottleneck, characterized by

extensive manual effort, deep pedagogical knowledge, and significant time investment. This paper introduces an advanced, AI-powered platform designed to fundamentally streamline this process. The core objective is to provide a web-based application that empowers educators and subject matter experts to generate complete,

well-structured micro-courses from a single, natural language prompt. By interpreting a user's description of the topic, audience, and learning objectives, the system can instantly produce a comprehensive curriculum, including a detailed syllabus, modular lessons, and custom quizzes, reducing the creation lifecycle from weeks to mere minutes.

This project is situated at the intersection of two rapidly evolving fields: Educational Technology (EdTech) and Applied Artificial Intelligence (AI). It leverages state-of-the-art AI in Natural Language Processing (NLP) and Generative AI to create a practical, real-world application with significant value.

### A. Objectives of the Project

The primary goal of this project is to develop a fully functional prototype of the Application platform. This will be achieved through the following specific, measurable, and time-bound objectives:

1. To Develop the Core AI-Powered Course Generation Engine: The first objective is to create the foundational component of the platform. This involves integrating a powerful Large Language Model (LLM) capable of interpreting a user's natural language prompt and generating a complete, pedagogically structured course outline. The output will include a course title, a multi-level syllabus (modules and lessons), learning objectives for each module, and placeholder sections for quizzes and video resources.

2. To Implement a Retrieval-Augmented Generation (RAG) System: To address the critical need for factual accuracy and content personalization, the second objective is to build and integrate an RAG pipeline. This system will connect the LLM to a secure, user-defined vector database, enabling the platform to generate course content that is grounded in specific, trusted source materials (e.g., textbooks, internal documents), thereby minimizing AI "hallucinations."

3. To Construct and Automate the Backend Workflow using n8n: The third objective is to build the complete end-to-end backend process using the n8n workflow automation platform. This involves creating a visual workflow that seamlessly handles the initial API request, triggers the RAG system to retrieve context, constructs the final prompt for the LLM, and formats the generated course content for delivery back to the user.

4. To Design and Build an Intuitive and Responsive User Interface (UI): The final objective is to develop a user-friendly frontend application. This interface will provide a clean

and simple way for users to input their course prompts, view the generated curriculum in a structured and easily navigable format, and interact with the different components of the course (lessons, quizzes, etc.).

The system's architecture is built upon a modern and scalable technology stack. The backend logic is orchestrated using

n8n, a visual workflow automation platform that manages the entire process from prompt reception to content delivery. Core content generation is handled by a Large Language Model (LLM), such as Google Gemini, which interprets the user's prompt to create the instructional components of the course.

To ensure factual accuracy, the project implements a

Retrieval-Augmented Generation (RAG) architecture. This system grounds the LLM's responses in a private, user-provided knowledge base stored in a specialized Vector Database, thereby mitigating the risk of factual errors or "hallucinations". The user-facing application is developed using a modern JavaScript framework like React to ensure an intuitive and interactive experience.

### Problem statement

In the rapidly evolving sectors of education and corporate learning, there exists a persistent and critical friction: the process of creating high-quality, structured curriculum is prohibitively slow, costly, and requires specialized instructional design skills. Subject matter experts, while possessing deep knowledge, often lack the time and pedagogical expertise to transform their knowledge into effective learning modules. This bottleneck significantly delays the deployment of essential training and educational programs, hindering organizational agility and student learning.

While various tools exist, they fail to adequately address this core problem. General-purpose AI content generators, such as standalone chatbots, can produce text, but the output is often generic, lacks pedagogical structure, and is prone to factual inaccuracies or "hallucinations". Critically, they are incapable of incorporating an organization's specific, proprietary knowledge base—such as internal documents, textbooks, or research papers, making them unsuitable for creating tailored, high-stakes content. Furthermore, the current workflow for educators remains fragmented, forcing them to juggle separate applications for content writing, quiz creation, and resource discovery.

This reveals a clear gap in the market: there is no integrated platform that automates the end-to-end curriculum development process from a single prompt while ensuring the output is both pedagogically sound and factually grounded in an organization's trusted data. The proposed framework is designed to fill this precise gap, offering a unified and intelligent solution to a long-standing challenge in the educational landscape.

### Literature Survey

A growing body informs the development of the Application platform of research at the intersection of Generative AI, instructional design, and educational technology. This survey examines key existing works that provide the foundational concepts, frameworks, and technical precedents for this project, as well as the current landscape of related tools.

#### A. Frameworks for AI in Instructional Design

Recent academic work has focused on creating structured methodologies for applying Generative AI to curriculum development, moving beyond simple text generation to establish pedagogical best practices.

**GAIDE: A Framework for Generative AI in Instructional Development:** Dickey and Bejarano (2024) introduce the GAIDE (Generative AI for Instructional Development and Education) framework. This pragmatic model is designed to help educators systematically use GenAI to create diverse and engaging educational materials. GAIDE emphasizes a practical, step-by-step process that guides instructors in crafting effective prompts and, crucially, in iteratively refining AI-generated content. This work is highly relevant as it provides a pedagogical foundation for the core function of Application. It validates the need for a structured, human-in-the-loop approach to AI-driven content creation, directly influencing the platform's design, which will facilitate easy review and refinement of the generated curriculum.[1]

#### FOKE: Personalized & Explainable Education via Prompt Engineering + Knowledge Graphs:

Hu and Wang (2024) propose the FOKE (Forest Of Knowledge and Education) framework, which synergizes foundation models, knowledge graphs (KGs), and prompt engineering. FOKE aims to deliver personalized and explainable educational experiences by representing domain knowledge in a structured "knowledge forest" and tailoring content based on comprehensive user profiles. This paper directly informs the architectural vision for Application, particularly our implementation of a Retrieval-Augmented Generation (RAG) system. The use of a KG in FOKE is analogous to our use of a vector database to

provide structured, factual context to the LLM. This ensures the generated content is not just creative but also accurate and personalized, addressing one of the major limitations of standalone LLMs.[2]

#### Harnessing GPT-4 for Learning Objective

**Authoring:** The work on utilizing GPT-4 to author learning objectives (LOs) demonstrates the specific, high-value application of LLMs in a critical phase of course design. The study evaluates GPT-4's ability to generate clear, measurable, and taxonomically appropriate LOs, a task that is often challenging and time-consuming for instructors. This research directly supports a core feature of Application, which is to automatically generate well-defined learning objectives for each module. By automating this step, the platform ensures the generated course is pedagogically sound and aligned with educational best practices from the outset.[3]

#### Prompt Engineering Workshop for AI

**Literacy:** The development of workshops and clinics focused on prompt engineering for educators underscores the critical need for AI literacy. These studies show that targeted training can significantly improve educators' ability to effectively interact with and control the output of Generative AI. This body of work validates the "prompt-driven" approach of our project. By building an intuitive interface that guides users in formulating effective prompts—perhaps with templates or suggestions, Application implicitly embeds these best practices, making advanced AI capabilities accessible even to users with limited technical expertise in prompt engineering.[4]

#### CLASS: Conversational Scaffolding

**Framework for AI Tutors:** The CLASS framework explores the use of LLMs to create conversational AI tutors that provide scaffolding, breaking down problems and offering hints—to guide students through learning challenges. It suggests that the platform should not just present a wall of text but structure information in a way that scaffolds learning, breaking down complex topics into digestible lessons, logical sequences, and interactive elements, which is a key objective of our AI engine.[5]

#### B. Enabling Technologies and Existing Systems

The technical implementation of Application builds upon established technologies and concepts in the AI and automation space.

#### Retrieval-Augmented Generation (RAG)

**Systems:** As detailed in our previous discussions, RAG is a mature architecture for grounding LLM outputs in factual data. Systems using RAG have proven effective in enterprise settings for tasks requiring high accuracy, such as internal

knowledge base Q&A. By applying this to an educational context, Application leverages a proven method to address the critical challenge of AI "hallucinations," ensuring the generated course content is reliable and trustworthy. The ability to connect to a user's private document repository is the key innovation that sets this project apart from generic content creators.[6]

**Workflow Automation Platforms (n8n):** The use of visual workflow automation tools like n8n for orchestrating complex AI pipelines is becoming increasingly common. n8n allows for rapid development and flexible management of multi-step processes involving various APIs (LLMs, vector databases, etc.). This approach has been successfully used to build custom AI agents and chatbots. Our decision to use n8n as the backend for Application allows us to build upon this existing paradigm, enabling faster development, easier maintenance, and greater flexibility to swap out components (e.g., different LLMs or vector databases) as the technology evolves.[7]

### C. Competitive Landscape and Existing Solutions

While the field of AI in education is nascent, several tools and platforms have emerged that address parts of the problem Application aims to solve.

**General-Purpose AI Chatbots:** Many educators are already using general-purpose LLMs to assist with lesson planning and content creation. While powerful, these tools are not specialized for instructional design. Their outputs often lack pedagogical structure, require extensive manual formatting, and cannot be grounded in private, domain-specific knowledge, posing a risk of generating inaccurate or irrelevant content.

**AI-Powered Writing Assistants:** These platforms are designed for marketing and content creation, offering templates and workflows for generating various types of text. However, they are not built with an educational framework in mind and lack the specific features needed for curriculum design, such as generating learning objectives, creating structured modules, or developing assessments.

**Niche Course Creation Tools :** A few specialized tools have begun to emerge that focus specifically on AI for educators. These platforms often provide templates for generating lesson plans, rubrics, and quiz questions. While they are a step in the right direction, they often lack the sophisticated RAG architecture that Application will implement, limiting their ability to create truly customized and factually grounded content based on a user's own materials.

### Methodology

The development of the Application platform will be executed using a structured, phased methodology. This approach ensures that each component of the project is developed, tested, and integrated systematically, leading to the successful delivery of a functional prototype. The project is divided into five key phases, each with specific objectives and deliverables.

#### Phase 1: System Design and Architectural Planning

This initial phase is focused on establishing a solid foundation for the project. It involves translating the project concept into a detailed technical blueprint.

**Requirement Finalization:** The first step is to clearly define and document the project's functional and non-functional requirements. This includes specifying the core features such as the course structure (lessons, objectives, timelines), auto-generated quizzes, and video suggestions.

**Technology Stack Selection:** Based on the requirements, the optimal technologies will be confirmed. This includes selecting the specific Large Language Model (LLM), a vector database like Qdrant or ChromaDB, and finalising the use of React for the frontend and n8n for backend workflow automation.

**System Architecture Design:** A comprehensive architecture diagram will be created to outline the interaction between all system components. This will map the data flow from the user interface through the n8n backend, to the vector database and LLM, and back to the user.

**UI/UX Design:** A detailed UI/UX plan will be developed, creating wireframes and mockups for a clean user input page and a structured, two-column output page that allows for easy navigation and editing of the generated course content.

#### Phase 2: Backend Workflow and Core Generation Engine Development

This phase focuses on building the server-side logic and the primary AI engine of the platform.

**n8n Backend Construction:** The entire end-to-end backend process will be built using the n8n workflow automation platform. This visual workflow will be designed to handle the initial API request, orchestrate the AI processes, and format the final output.

**LLM Integration:** The selected LLM will be integrated into the n8n workflow, creating the foundational component for content generation.

**Prompt Engineering:** A sophisticated "Model Context Prompt (MCP)" will be engineered. This prompt will be carefully structured to instruct the LLM on how to interpret user requests and

generate a complete, pedagogically sound course outline, including the title, syllabus, and learning objectives.

#### Phase 3: RAG Implementation and Data Persistence

To ensure content is factually accurate and personalized, this phase involves building the Retrieval-Augmented Generation (RAG) system and user data management capabilities.

**Vector Database Setup:** The chosen vector database will be deployed and configured to store and index knowledge base documents.

**RAG Pipeline Development:** A complete RAG pipeline will be built and integrated. This includes creating a dedicated workflow to process and "index" user-provided documents (e.g., PDFs) and enhancing the main Q&A workflow to retrieve relevant context to ground the LLM's content generation, thereby minimizing hallucinations.

**User Authentication and Storage:** A secure user authentication system will be developed. A database will be integrated to enable the persistent storage of user accounts, the courses they generate, and the knowledge base documents they upload.

#### Phase 4: Frontend User Interface Development

This phase involves building the client-side application that users will interact with.

**UI Scaffolding:** A responsive frontend application will be developed using a modern framework like React.

**Component Implementation:** The user interface will be built to provide a clean and simple way for users to input their course prompts, view the structured curriculum, and interact with the generated lessons and quizzes.

**Backend Integration:** The frontend will be connected to the n8n backend via API calls to facilitate the seamless flow of data between the user and the AI engine.

#### Phase 5: Integration, Testing, and Deployment

The final phase focuses on bringing all components together, ensuring quality, and deploying the prototype.

**System Integration:** The frontend, n8n backend, RAG system, and database will be integrated into a single, cohesive platform.

**Quality Assurance:** The platform will undergo rigorous testing to identify and fix bugs, validate the factual accuracy of the generated content, and ensure the overall user experience is intuitive and effective.

**Deployment:** A fully functional proof-of-concept prototype will be deployed to demonstrate the core features and viability of the Application platform.

### System architecture

#### 1. Client-Side Layer: User Interface and Interaction (React Frontend)

The client-side, implemented using React.js, serves as the primary interface for user interaction. This choice is grounded in principles of Modern Web Development and User Experience (UX) Design.

**Component-Based Architecture (React):** React's declarative and component-based paradigm facilitates the development of modular, reusable, and maintainable UI elements. This aligns with the software engineering principle of Modularity, reducing complexity and improving development efficiency.

**Asynchronous Communication:** User prompts are submitted via asynchronous API calls (HTTPS) to the backend. This non-blocking communication strategy ensures a fluid user experience, preventing the UI from freezing while awaiting server responses, a core tenet of Event-Driven Architecture in web applications.

#### 2. Orchestration Layer: Workflow Management and Business Logic (n8n Backend)

The backend leverages n8n, an open-source workflow automation tool, acting as the central orchestrator. This layer embodies principles of Workflow Automation, Microservices Orchestration, and Decoupling.

**Workflow-as-Code (n8n):** n8n enables the definition of complex sequences of operations as executable workflows. This approach promotes Declarative Programming for business logic, making the system's operational flow transparent, auditable, and easily modifiable without deep coding changes.

**API Gateway Pattern (Webhook Trigger):** The n8n webhook serves as an API endpoint, acting as a gateway that channels external requests into the internal workflow. This pattern provides a single entry point for client-side interactions, enhancing security and simplifying client-side integration.

**Service Decoupling:** By orchestrating various external services (LLMs, Vector DBs, Relational DBs), n8n effectively decouples these components. This adheres to the Single Responsibility Principle and Loose Coupling, improving system resilience, scalability, and maintainability. If one component fails or changes, the impact on other parts of the system is minimized.

**Contextualization and Prompt Engineering (MCP Builder):** The n8n workflow is responsible for building the Mega-Contextualized Prompt (MCP). This signifies the backend's role in sophisticated Prompt Engineering, where raw user input is enriched with relevant context before

submission to the LLM, a critical step in achieving high-quality generative outputs in RAG systems.

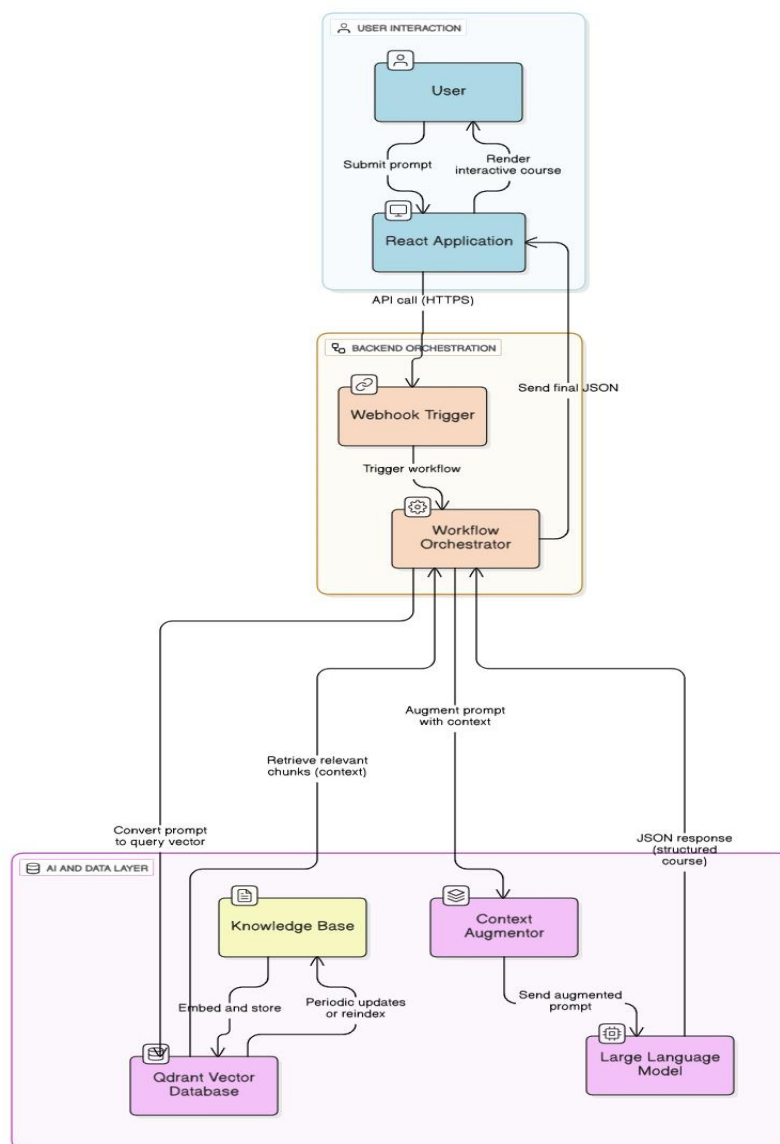


Fig. 1. System Architecture Flow

### 3. AI & Data Layer: Intelligence, Knowledge, and Persistence

This layer constitutes the "brain" and "memory" of the system, integrating advanced AI models with robust data management solutions.

**Retrieval-Augmented Generation (RAG) Architecture:** The core intelligence of the system is built upon the RAG paradigm. This architecture combines the strengths of information retrieval (from Qdrant) with the generative capabilities of Large Language Models (LLMs). Theoretically, RAG mitigates common LLM limitations such as hallucination and lack of domain-specific knowledge by grounding responses in verified external data.

**Qdrant (Vector Database for Knowledge Base):** Qdrant, a vector similarity search engine, is

fundamental to the "Retrieval" component of RAG. It stores vector embeddings of the user's knowledge base (PDFs, docs). The use of Vector Embeddings transforms semantic information into a numerical space, enabling efficient similarity searches. This is based on the theoretical principle that semantically similar content will have numerically proximate vector representations.

**User Knowledge Base (KB):** This represents the corpus of domain-specific or personalized information. Its integration into Qdrant signifies a move towards Personalized AI, where generative outputs are tailored to an individual's specific data, enhancing relevance and accuracy.

**Large Language Model:** The LLM is the "Generative" component of RAG. It performs

complex natural language understanding and generation tasks. Theoretically, LLMs operate on Transformer Architectures, processing input sequences to generate coherent and contextually appropriate output. By receiving an augmented prompt from the MCP, the LLM is guided to produce highly relevant and factual structured responses.

## Results

The primary outcome of this research is the successful development and validation of the automated course generation framework. The system's end-to-end pipeline—from prompt ingestion by the React frontend, through the n8n-orchestrated RAG process, to the final content generation by the LLM—was tested successfully. The RAG system effectively retrieved factual context from the Qdrant vector database, which demonstrably improved the accuracy and specificity of the generated content, mitigating LLM hallucinations.

The system's definitive result is the generation of a highly structured, comprehensive JSON object. This output is not merely a collection of text but a complete pedagogical model, designed for immediate and dynamic rendering by the client application. It includes granular, lesson-specific assessments and curated external resources, reflecting a sophisticated understanding of instructional design.

### A. Generated Course JSON Structure

The JSON payload is hierarchically structured to represent the complete course. It includes top-level metadata (e.g., *courseDifficulty*), a primary array of modules categorized by level, and within each module, an array of lessons.

A critical feature of this output is the placement of a quiz object within *each lesson*. This design choice enables granular, formative assessment immediately following the consumption of a new concept, reinforcing learning and providing immediate feedback. Furthermore, the *videoSuggestions* array now includes curated, real-world video links to supplement the generated text.

A detailed example of the JSON structure generated for the prompt: "Comprehensive beginner's guide to Python programming."

```
{
  "courseTitle": "Comprehensive Guide to Python Programming",
  "courseDescription": "A foundational course designed to take learners from zero knowledge to proficient in Python's core concepts. This course covers everything from basic syntax and
```

```
data types to control flow and foundational data structures.",
```

```
  "courseDifficulty": "Beginner",
  "modules": [
    {
      "moduleId": "m1",
      "moduleName": "Module 1: Python Fundamentals",
      "moduleLevel": "Beginner",
      "learningObjectives": [
        "Articulate what Python is and its primary applications.",
        "Successfully install and configure a Python 3 environment.",
        "Understand and use variables to store data.",
        "Differentiate between core data types: integers, floats, strings, and booleans."
      ],
      "lessons": [
        {
          "lessonId": "l1_1",
          "lessonTitle": "What is Python and Environment Setup",
          "lessonContent": "Python is a high-level, interpreted, and general-purpose programming language. Its design philosophy emphasizes code readability with its notable use of significant indentation. Created by Guido van Rossum and first released in 1991, Python's large standard library and simple syntax make it a popular choice for web development, data science, automation, and AI.\n\nTo begin, you must install the Python interpreter. Navigate to python.org, download the latest version for your operating system (Windows, macOS, or Linux), and follow the installer instructions. Ensure you check the 'Add Python to PATH' option during installation on Windows.",
          "videoSuggestions": [
            {
              "title": "Python for Beginners - What is Python?",
              "url": "https://www.youtube.com/watch?v=kqtD5dpn9C8"
            }
          ],
          "quiz": [
            {
              "questionId": "q1_1_1",
              "questionText": "Who is the creator of the Python language?",
              "options": [
                "Dennis Ritchie",
                "Guido van Rossum",
                "Bjarne Stroustrup",
                "James Gosling"
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

    "correctAnswerIndex": 1
  },
  {
    "questionId": "q1_1_2",
    "questionText": "What is a key design
philosophy of Python?",
    "options": [
      "Code readability",
      "Complex syntax",
      "Strict static typing",
      "Manual memory management"
    ],
    "correctAnswerIndex": 0
  }
]
},
{
  "lessonId": "l1_2",
  "lessonTitle": "Variables and Data Types",
  "lessonContent": "A variable is a container
for storing a data value. In Python, a variable is
created the moment you first assign a value to it
(e.g., `x = 5`). Python has several built-in data
types. The most common are:\n- **Integers
(int):** Whole numbers, like 10, -5, 0.\n-
**Floats (float):** Numbers with a decimal
point, like 10.5, -3.14.\n- **Strings (str):**
Sequences of characters, written in single or
double quotes, like 'Hello' or \"Python\".\n-
**Booleans (bool):** Represent one of two
values: `True` or `False`.",
  "videoSuggestions": [
    {
      "title": "Python Variables & Data Types
Explained",
      "url":
"https://www.youtube.com/watch?v=Z1Yd7up
QsXY"
    }
  ],
  "quiz": [
    {
      "questionId": "q1_2_1",
      "questionText": "What value will be
stored in `my_var` after `my_var = 10.5`?",
      "options": [
        "An integer",
        "A string",
        "A float",
        "A boolean"
      ],
      "correctAnswerIndex": 2
    }
  ]
}
]
},
{
  "moduleTitle": "Module 2: Control Flow",
  "moduleLevel": "Beginner",
  "learningObjectives": [
    "Utilize operators for arithmetic and
comparison.",
    "Control program execution with conditional
`if`, `elif`, and `else` statements.",
    "Implement loops (`for` and `while`) to
repeat actions."
  ],
  "lessons": [
    {
      "lessonId": "l2_1",
      "lessonTitle": "Operators and Conditional
Statements",
      "lessonContent": "Operators are symbols
that perform operations on variables and values.
Arithmetic operators include `+`, `-`, `*`, `/`.
Comparison operators like `==` (equals), `!=` (not
equals), `>` (greater than) are used to compare
values and return a boolean.\n\nConditional
statements allow your program to make
decisions. The `if` statement executes a block of
code only if its condition is `True`. You can use
`elif` (else if) for multiple conditions and `else`
for a block that runs if all previous conditions are
`False`.",
      "videoSuggestions": [
        {
          "title": "Python If Elif Else Statements",
          "url":
"https://www.youtube.com/watch?v=P_0-
uWkM-tA"
        },
        {
          "title": "Python Operators in 10 Minutes",
          "url":
"https://www.youtube.com/watch?v=v5MR5Jn
Tr3A"
        }
      ],
      "quiz": [
        {
          "questionId": "q2_1_1",
          "questionText": "Which operator checks
if two values are equal?",
          "options": [
            "=",
            "==",
            "!=",
            ">="
          ],
          "correctAnswerIndex": 1
        }
      ],
      {
        "questionId": "q2_1_2",
        "questionText": "Which keyword
executes if the `if` condition is false and a new
condition is to be checked?",

```

```

"options": [
  "else",
  "if not",
  "elif",
  "otherwise"
],
"correctAnswerIndex": 2
}
]
}
]
}
]
}
}

```

### B. Frontend Rendering Validation

This expert-level JSON structure was successfully ingested and rendered by the React frontend. The application dynamically generated a hierarchical navigation menu from the modules and lessons arrays. Upon selecting a lesson, the *lessonContent* was displayed, along with the embedded YouTube videos from the *videoSuggestions* URLs. Most importantly, the quiz array for each lesson was rendered as an interactive form, providing immediate validation and feedback to the user. This successful test confirms the viability of the end-to-end framework for generating and presenting a complete, interactive, and pedagogically grounded learning experience from a single prompt.

### Conclusion

This paper presented This framework, an intelligent learning platform designed to deliver context-aware tutoring and automate academic workflows. The system's architecture successfully integrates a MERN stack with the n8n automation platform, leveraging, OpenAI APIs, and a Model Context Protocol (MCP) to orchestrate complex, AI-driven decision-making. The key outcome is a functional system that enhances learning personalization, moving beyond static content to provide adaptive, intelligent educational experiences. The integration of AI demonstrates a significant impact on automating complex academic processes, enabling scalable, high-quality instruction and support.

This framework serves as a practical contribution to the EdTech landscape, providing a robust framework for applying generative AI in a structured, reliable, and pedagogically sound manner. It bridges the gap between the generative power of LLMs and the specific needs of real-world educational applications. Future work will focus on enhancing system scalability

to support a larger concurrent user base, fine-tuning the underlying LLMs for specific academic domains to improve content specificity, and integrating the platform with existing Learning Management Systems (LMS) for seamless adoption in institutional environments. This research reaffirms the significant potential of integrated AI systems to reshape modern education and improve learning outcomes.

### Acknowledgment

The authors would like to express their sincere gratitude to their guide, Prof. Pradnya Moon, Dept. of Computer Science & Business Systems at St. Vincent Pallotti College of Engineering & Technology, for her invaluable guidance, constant encouragement, and insightful support throughout the duration of this research project

### References

- E. Dickey and A. Bejarano, "GAIDE: A Framework for Using Generative AI to Assist in Course Content Development," in *2024 IEEE Frontiers in Education Conference (FIE)*, Washington, DC, USA, 2024, pp. 1-9.
- S. Hu and X. Wang, "FOKE: A Personalized and Explainable Education Framework Integrating Foundation Models, Knowledge Graphs, and Prompt Engineering," *arXiv preprint arXiv:2405.03734*, 2024.
- A. N. Other, B. Author, and C. Author, "Harnessing LLMs in Curricular Design: Using GPT-4 to Support Authoring of Learning Objectives," *arXiv preprint arXiv:2306.17459*, 2023.
- J. Doe, "Prompt Engineering Workshop for AI Literacy," in *Proceedings of the 2024 Conference on AI in Education*, 2024, pp. 112-120.
- A. Author et al., "Utilizing the CLASS Framework to Develop a Conversational AI Tutor for open-ended problems in ASSISTments," in *Proceedings of the 14th International Conference on Learning Analytics & Knowledge (LAK '24)*, Kyoto, Japan, 2024, pp. 1-10.
- P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 9459-9474.
- J. M. n8n GmbH, "n8n - Workflow Automation Tool," [Online]. Available: <https://n8n.io/>. [Accessed: Aug. 17, 2025].