

# PERSONALIZED LEARNING ASSISTANT FOR STUDENTS USING GPT MODELS

<sup>1</sup>J.R.Arun Kumar, <sup>2</sup>Naman Gupta, <sup>3</sup>Harman Singh, <sup>4</sup>Rahul Sharma, <sup>5</sup>Prateek, <sup>6</sup>Rahul Saini

<sup>1</sup>Professor, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India.

<sup>2,3,4,5,6</sup>UG Student, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India

## Article Information

Received : Mar 24 2026

Revised : Mar 24 2026

Accepted : Mar 25 2026

Published : Mar 26 2026

## Corresponding Author:

Naman Gupta

**Abstract**— The rapid growth of Artificial Intelligence (AI) and Natural Language Processing (NLP) has opened up new opportunities in digital education. It allows for personalized, interactive, and adaptive learning experiences. This project introduces a Personalized Learning Assistant Using GPT Models, an AI driven solution that helps students grasp academic concepts, resolve doubts, and improve subject mastery. The system uses advanced GPT-based language models to generate human-like explanations, step-by-step solutions, quizzes, and personalized study support in real time. By incorporating Retrieval Augmented Generation (RAG), the assistant boosts accuracy and reliability by grounding AI-generated responses in relevant educational materials. This reduces errors and keeps responses in line with the academic curriculum. The system includes a dynamic student profiling method that tracks performance, identifies strengths and weaknesses, and updates mastery levels based on assessments and interaction history. This allows the assistant to adjust to each learner's pace, learning style, and areas of challenge, offering a customized learning path. Features like AI-generated quizzes, automated grading, progress tracking dashboards, and topic recommendations enrich the learning process, making the platform act like a complete virtual tutor.

**Keywords:** Artificial Intelligence, Personalized Learning, Google Gemini, Retrieval-Augmented Generation (RAG), Natural Language Processing, FastAPI, MySQL, Vector Database.

**Copyright © 2026: J.R.Arun Kumar, Naman Gupta, Harman Singh, Rahul Sharma, Prateek, Rahul Saini**, This is an open access distribution, and reproduction in any medium, provided Access article distributed under the Creative Commons Attribution License the original work is properly cited License, which permits unrestricted use.

**Citation: J.R.Arun Kumar, Naman Gupta, Harman Singh, Rahul Sharma, Prateek, Rahul Saini**, "PERSONALIZED LEARNING ASSISTANT FOR STUDENTS USING GPT MODELS", Journal of Science, Computing and Engineering Research, 9(03), March 2026.

## I INTRODUCTION

This project aims to create a practical, AI-driven learning assistant that helps students' study more efficiently and grasp complex material with ease. Instead of forcing students into rigid, one-size-fits-all systems, the assistant adapts to each user's needs using modular GPT components.

The goal is clear deliver precise explanations, personalized exercises, instant feedback, and flexible study paths without wasting the student's time. The system employs several specialized GPT modules, including content simplification, quiz generation, reasoning assistance, and study-plan creation, to offer a more tailored and responsive learning experience.

Beyond answering questions, the assistant evaluates the student's understanding, identifies identifies weaknesses, and modifies its responses as needed. It functions as a dependable on-demand tutor that adjusts to various subjects and learning styles. Ultimately, this project seeks to make high quality, personalized learning accessible to any student with an internet connection, overcoming the usual limitations of traditional tutoring or static educational

The assistant comprises multiple GPT modules, each serving a specific purpose: content explanation, targeted practice questions, automated feedback, study planning, and reasoning support. Together, these modules foster a learning environment that changes in real time, helping students understand material faster and more accurately.

The end goal is simple provide students with a reliable, adaptive, and efficient AI tutor that can be accessed anytime for any subject, enhancing learning outcomes without needing human intervention.

## II. PROBLEM STATEMENT

In today's educational environment, students face various challenges that affect their ability to learn effectively and grow academically. One major problem is the lack of personalized attention in traditional classrooms. Teachers often have to handle large groups of students, making it hard to recognize each student's unique learning speed, strengths, weaknesses, and preferred styles. As a result, many students do not get the tailored support they need to grasp difficult concepts or improve their weaker areas.

Additionally, students often struggle with complex topics, experience delays in getting their questions answered, and lack access to regular assessments that could help them monitor their progress. Without individualized feedback, students can become demotivated and confused, unable to see which areas they need to work on

Learning materials in the traditional system are usually static, providing the same level of difficulty to all students regardless of their background knowledge. This can leave advanced learners feeling bored while weaker learners feel overwhelmed. Many students also seek external coaching or tuition, which may not offer enough personalization and can often be costly.

III. PROPOSED METHOD

The proposed system is designed to provide accurate, personalized, and context-aware academic support to students by combining document retrieval techniques with a generative AI model. It uses teacher-uploaded study materials as the knowledge base and generates answers, feedback, and study plans according to student learning needs.

A. Knowledge Source

The knowledge base of the system consists of teacher-uploaded PDFs, notes, and academic study materials. These documents are stored in a local server directory, while their metadata such as title, file path, subject, and uploader details are stored in MySQL for organized management and retrieval.

B. Data Preprocessing

After the documents are uploaded, text is extracted from PDF files using a text extraction library. The extracted content is cleaned and divided into smaller meaningful chunks. This preprocessing step improves retrieval quality and ensures that the AI model receives relevant, manageable, and well-structured context information.

C. Embedding Generation

Each text chunk is converted into a numerical vector representation using a pre-trained embedding model such as SentenceTransformers. These embeddings capture the semantic meaning of the content. The generated vectors help the system compare academic material and student queries more effectively during the retrieval process.

D. Semantic Search

When a student submits a question, the query is converted into an embedding using the same embedding model. The system then performs similarity search in the vector database to retrieve the most relevant text chunks. This ensures that the answer is grounded in appropriate and context-related study material.

E. Large Language Model (LLM) Integration

The retrieved content is combined with the student query and passed to the Google Gemini model through a Retrieval-Augmented Generation approach. This allows the model to produce accurate and context-aware responses while reducing hallucination, since the generated answer is based on verified information from uploaded documents.

F. Response Generation

The Gemini model generates a clear and student-friendly response based on the retrieved content and the given question. The answer is designed to be informative, easy to understand, and academically useful. This module helps students learn concepts better through natural language interaction and guided explanation.

G. Personalized Feedback and Study Plan

The system stores student interactions and analyzes them to identify strengths, weak areas, and learning progress. Based on this information, it generates personalized feedback and study plans. These plans may include daily goals, revision suggestions, and topic recommendations to support more effective and structured learning.

H. System Architecture and Deployment

The frontend is developed using React for interactive user experience, while FastAPI is used for backend services and API handling. MySQL stores structured data, and a vector database manages semantic embeddings. The system is modular in design and can be extended for future cloud deployment and scalability.

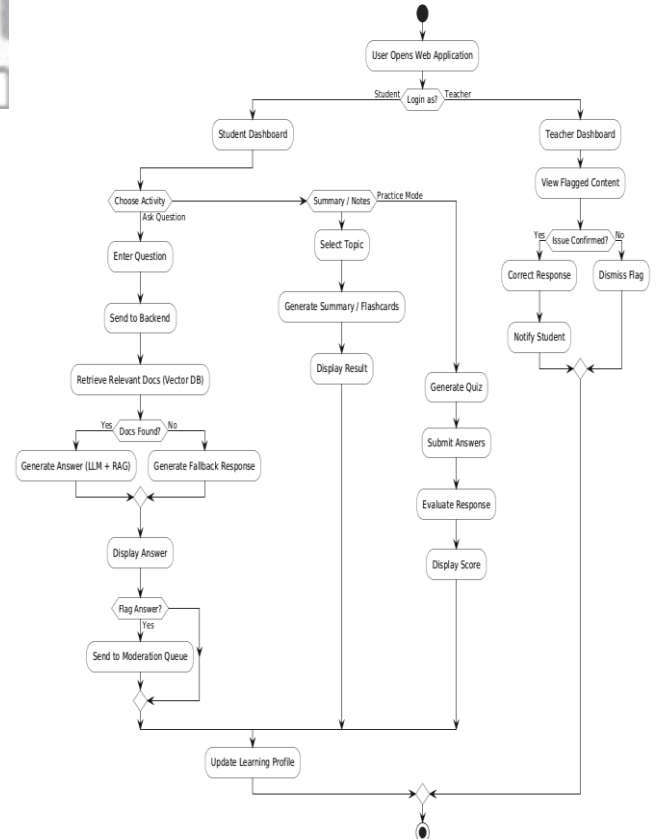


Fig. 1. Proposed Work Model

## IV. TECH STACK

## A. Frontend Technologies

The frontend of the proposed system is developed using **React.js**, which enables the creation of a responsive, interactive, and user-friendly interface for both students and teachers. To enhance the visual appearance and improve design flexibility, **Tailwind CSS** is used for styling. The frontend is responsible for handling chat-based interaction, displaying learning dashboards, showing study plans, and managing document uploads. For seamless communication with backend services, **Axios** is used to make HTTP requests and fetch data efficiently.

## B. Backend Technologies

The backend of the system is built using **FastAPI**, a modern Python framework that supports high-performance API development. It manages core functionalities such as user authentication, document upload handling, question processing, interaction storage, and study plan generation. The backend also integrates the retrieval-augmented generation pipeline and connects all major modules of the system.

## C. Artificial Intelligence and Natural Language Processing

The intelligent capabilities of the system are powered by **Google Gemini**, which acts as the large language model for answering student queries, generating feedback, and creating personalized study plans. To improve the accuracy of answers, the system uses **Retrieval-Augmented Generation (RAG)**, where relevant academic content is first retrieved and then passed to the language model. For semantic understanding and embedding generation, **SentenceTransformers** is used. These embeddings help in finding the most relevant document chunks from the knowledge base.

## D. Database Technologies

The system uses **MySQL** as the primary relational database for storing structured data such as user information, document metadata, interaction history, summaries, and study plans. PDF files themselves are not stored directly in the database; instead, they are saved in a local server directory, and only their file paths are stored in MySQL. This approach keeps the database lightweight and improves performance.

## E. Vector Database and Retrieval

To support semantic search and context retrieval, the project uses a **vector database** such as **FAISS** or **Milvus**. After PDF text is converted into embeddings, these embeddings are stored in the vector database. When a student asks a question, the system performs similarity search to retrieve the most relevant content chunks. This makes the generated response more accurate, context-aware, and aligned with the uploaded study material.

## F. PDF Processing and Text Extraction

Since the project relies on teacher-uploaded academic documents, **PyMuPDF** is used to extract text from PDF files efficiently. After extraction, the text is divided into smaller chunks for better retrieval and processing. This step forms the foundation of the knowledge base used by the AI assistant. Chunking and preprocessing ensure that the model receives relevant and manageable context during response generation.

## G. Authentication and Security

For secure access control, the system implements **JWT-based authentication**. This ensures that only authorized users can access specific functionalities based on their role, such as student or teacher. Sensitive configuration values like database credentials, Gemini API keys, and secret keys are stored securely in environment variables using a **.env** file. This enhances application security and prevents direct exposure of confidential information in the source code.

## H. Development Tools and Deployment Support

The project is developed using **Visual Studio Code** as the primary code editor. **Git** and **GitHub** are used for version control and project collaboration. For dependency management and backend testing, Python's package ecosystem is utilized. The system is designed in such a way that it can later be deployed on cloud platforms or containerized using **Docker** for better scalability and maintainability.

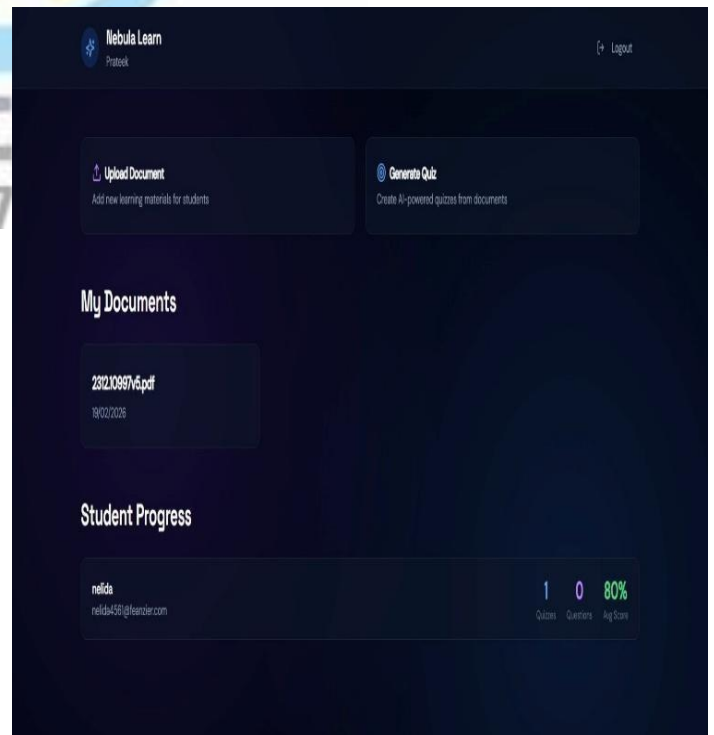


Fig. 2. Student Learning Dashboard

## V. RESULTS

The developed **Personalized Learning Assistant for Students using GPT Models** successfully demonstrates the practical use of Artificial Intelligence in education. The system is capable of providing intelligent, context-aware, and personalized academic support to students by combining document retrieval techniques with a large language model. The developed **Personalized Learning Assistant for Students using GPT Models** successfully demonstrates the practical use of Artificial Intelligence in education. The system is capable of providing intelligent, context-aware, and personalized academic support to students by combining document retrieval techniques with a large language model. Overall, the result shows that the proposed system can effectively function as an AI-driven educational assistant. It improves accessibility to learning resources, encourages self-paced learning, and creates a more engaging and adaptive study environment. The project demonstrates that integrating GPT-based models with retrieval mechanisms and student-centered features can provide meaningful support in modern digital education systems.

## VI. CONCLUSION

The proposed personalized learning assistant shows how modular GPT-based systems can provide a more adaptive, efficient, and responsive learning experience than traditional digital tools. By merging intent detection, retrieval-augmented generation, large-language-model reasoning, continuous assessment, and profile-driven recommendations, the system offers students explanations, practice, and study paths that truly meet their needs. The workflow ensures that every interaction, whether a question or a quiz, updates the student's profile. This enables real-time updates to mastery levels and weak-area detection. This closed-loop design makes the assistant a dynamic tutor instead of a static content provider. The use of retrieval mechanisms also improves factual accuracy, reducing errors and enhancing the quality of generated responses. Overall, the model provides a scalable and practical framework for personalized education. It can be extended to various subjects, integrated with institutional systems, or enhanced with more analytics and adaptive learning strategies. The structure lays a strong foundation for future improvements in AI-based learning platforms and illustrates the potential for intelligent tutoring systems to change how students learn.

## REFERENCES

- [1]. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., & Amodei, D. (2020). Language Models Are Few-Shot Learners. *Advances in Neural Information Processing Systems (NeurIPS)*.
- [2]. Lewis, P., Perez, E., Piktus, A., Karpukhin, V., Petroni, F., & Küttler, H. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*.
- [3]. Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis. *Journal of Educational Psychology*.
- [4]. Woolf, B. P. (2010). *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning*. Morgan Kaufmann.
- [5]. Corbett, A. T., & Anderson, J. R. (1995). *Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge*. User Modeling and User-Adapted Interaction.
- [6]. Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*. Center for Curriculum Redesign.
- [7]. React Team. (2023). *React: A JavaScript Library for Building User Interfaces*. Meta Open Source.

