

“ELEVATE AI”

Enhanced Learning Experience & Virtual Assistant Tool for Education

¹Prof. Chandani Lachake, ²Rushikesh Wani, ³Shivraj Ghodake, ⁴Pritesh Chaudhari

^{2,3,4} Undergrad. Student, Dept. of Computer Engineering

¹Associate Prof, Dept. of Computer Engineering, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra

Abstract:

AI tools are now widely used by students for studying and exam preparation. However, most of these tools are general-purpose and do not follow university-specific answer formats, syllabus structure, or marking schemes. Because of this, students may understand concepts but still fail to score well in exams.

To solve this, we propose ELEVATE AI, an AI-based system designed specifically for exam-oriented learning. The system generates structured answers using syllabus data, textbooks, previous year questions, and predefined diagrams. It also introduces a practice-based approach where students write answers and get feedback by comparing with ideal model answers.

In addition, voice interaction is supported using speech-to-text and text-to-speech, allowing the system to act like a tutor. The backend uses a multi-agent architecture along with Retrieval-Augmented Generation and high-speed inference through Groq with LLaMA 3.3.

The system focuses not just on understanding but also on answer writing and presentation, helping students improve both learning and scoring.

Index Terms:

Artificial Intelligence, Large Language Models, Retrieval-Augmented Generation, Exam-Oriented Learning, Intelligent Tutoring System, Previous Year Question Analysis, Educational Chatbot

Introduction

Artificial Intelligence has changed how students' study, especially with the rise of Large Language Models. Tools like ChatGPT and Google Gemini allow students to quickly understand concepts and generate answers.

But there is a problem.

These tools are general-purpose. They give correct answers, but not in the format expected in exams. They don't follow university-specific structure, don't focus on marking schemes, and don't highlight what actually matters for scoring.

Another issue is that they ignore things like:

- previous year questions
- answer presentation
- diagrams
- structured writing

Because of this, students still have to modify answers manually or may lose marks even after studying properly.

So, the gap is clear. Students don't just need correct answers. They need answers written the way examiners expect.

To solve this, we propose ELEVATE AI, a system designed specifically for exam preparation. It focuses on:

- syllabus alignment
- structured answers
- PYQ importance
- answer writing practice

The goal is simple: not just learn, but write better answers in exams.

Literature Survey:

Vaswani et al. (2017): Attention Is All You Need (Transformer Architecture):-

With the advancement of Natural Language Processing, newer models were developed to overcome the limitations of earlier approaches like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). These models had issues such as sequential processing of data and limited ability to capture long-range dependencies in text. The transformer architecture introduced by Vaswani et al. solved these

problems using an attention mechanism, which allows the model to focus on all words in a sentence at the same time while maintaining their relationships. This made it possible to understand context more effectively and significantly improved the performance of NLP systems.

Devlin et al. (2018): BERT – Bidirectional Encoder Representations from Transformers:-

Before BERT, most NLP models processed text in a single direction, either from left to right or right to left, which limited their understanding of context. Devlin et al. introduced BERT, which uses bidirectional processing to understand the full context of a word by looking at both its previous and next words. This was made possible due to the transformer architecture. As a result, BERT improved tasks such as question answering and sentence understanding by capturing deeper contextual relationships in text.

Brown et al. (2020): GPT-3 – Language Models are Few-Shot Learners:-

Further advancements in language models were demonstrated by GPT-3, which introduced the concept of scaling models to a very large size. Brown et al. showed that with enough parameters, a model can perform multiple tasks using only a few examples without requiring task-specific training. This gave rise to general-purpose language models that can generate human-like text and handle a wide variety of tasks. However, such models are not specifically optimized for structured or domain-specific outputs like academic exam answers.

OpenAI / Educational Studies (2023): Applications of ChatGPT in Education:-

With the rise of large language models, tools like ChatGPT and Gemini have become widely used by students for learning purposes. These tools help in understanding concepts, simplifying difficult topics, and providing quick answers. They also support interactive learning through continuous conversation. However, despite these advantages, they are designed as general-purpose systems and do not focus on exam-specific requirements. They often generate answers that are too descriptive or unstructured and do not follow the format expected in university exams.

Problem Statement:

Students today widely use AI tools like ChatGPT and Gemini for academic preparation because they help in quickly understanding concepts and generating answers. However, these tools are built as general-

purpose systems and do not fully understand the specific requirements of university exams. Even when the answers are factually correct and conceptually accurate, they may not follow the expected structure, terminology, or presentation style defined by the syllabus and marking scheme. They also lack awareness of important aspects like previous year questions, standard answer formats, and the use of diagrams, which play a key role in scoring. Because of this, students often have to spend extra time modifying the answers or may lose marks despite knowing the correct content. Therefore, there is a need for a system that can generate structured, exam-oriented, and syllabus-aware answers tailored to academic requirements.

Proposed Methodology:

1. Overall Approach

The proposed system follows an approach where a Large Language Model is combined with a Retrieval-Augmented Generation (RAG)-like setup using structured academic data such as syllabus, textbooks, previous year questions, predefined diagrams, and expected model answer formats. When a student inputs a question, the system processes it with reference to this data and generates an answer that is aligned with exam expectations. Additionally, it identifies whether the question has appeared in previous exams and presents relevant details along with an appropriate diagram, making the response both informative and exam-oriented.

2. Multi-Agent Architecture

The system is divided into multiple modules, where each agent is assigned a specific task to improve consistency and performance. One agent is responsible for generating answers strictly based on university-specific data, while another checks whether the question matches any previous year questions. A separate agent selects the most relevant diagram from a predefined set, and another ensures that the final response follows the expected model answer format used in exams. This modular approach helps in producing more structured and reliable outputs, while also avoiding the need to generate diagrams dynamically, which can be inconsistent in quality.

3. AI Model + Groq Integration

The system uses high-performance inference through the Groq API along with the LLaMA 3.3 (70B) model to handle response generation. Instead of running models locally, Groq provides faster and more efficient processing, which helps in reducing response time while maintaining output quality. This makes the system more responsive and practical for real-time use by students. The use of a powerful

model combined with optimized inference ensures that the generated answers are both accurate and detailed, while still being delivered quickly.

4. Structured Prompting

To ensure that the answers generated are suitable for exams, structured prompting is used. Instead of allowing the model to generate free-flow answers, the prompt is designed in such a way that the output follows a fixed format, including sections like theory, applications, advantages, and disadvantages wherever applicable. This helps in maintaining consistency across responses and makes the answers more aligned with what is expected in university exams. It also reduces the need for students to manually restructure the answer before writing it.

5. PYQ Integration

The system includes a dataset of previous year questions, which is used to check whether the user's query has appeared in past exams. When a question is entered, it is compared with the stored PYQ data, and if a match is found, the system returns details such as the year and marks associated with that question. This helps students identify important topics and understand exam patterns. Even if the question is not an exact match, the system still attempts to find similar questions to provide useful context.

6. Diagram Retrieval System

To include visual understanding, the system uses a predefined set of diagrams categorized topic-wise. Based on the user's question, the model selects the most relevant diagram topic, and the corresponding image is displayed from stored assets. This approach ensures that the diagrams are accurate and consistent, unlike dynamically generated images which may vary in quality. Including diagrams in answers helps students improve presentation and can contribute to better scoring in exams.

7. Answer Writing Practice

The system also includes a model answer dataset built from optimized answers of previous year questions. Instead of directly showing answers, the system presents a question to the student and requires them to write the answer on their own. Once submitted, the response is compared with the ideal model answer, and the system provides feedback based on missing points, structure, and overall presentation. This approach helps shift learning from passive reading to active

writing, which improves retention, enhances answer writing skills, and allows students to better understand what is expected in exams.

8. Voice Tutor Integration

The system also supports voice-based interaction, allowing it to function like a tutor. It uses Deepgram for speech-to-text conversion and Sarvam AI for text-to-speech output. With this, students can speak their answers instead of typing, and the system processes the input, evaluates it, and responds with feedback. This makes the learning experience more interactive and natural, while also helping students practice verbal explanation and understanding of concepts.

9. Frontend-Backend Flow

The system follows a simple frontend-backend architecture where the user interacts through a web-based interface built using HTML, CSS, and JavaScript. When a question is submitted, it is sent to the backend through an API call handled by FastAPI. The backend processes the request using different agents and the AI model, and the final structured response is sent back to the frontend. The interface then displays the answer along with any additional information such as PYQ details and diagrams, providing a smooth and interactive experience for the user.

System Architecture:

The system follows a full-stack architecture consisting of frontend, backend, AI processing layer, and a structured data layer. The user interacts with the system through a web-based interface where queries are entered in a chat format or through voice input. These requests are sent to the backend, processed using different modules, and a structured response is generated and displayed back to the user. The architecture is designed to ensure smooth data flow and modular handling of tasks while also supporting both learning and practice-based interactions, including tutor-like guidance.

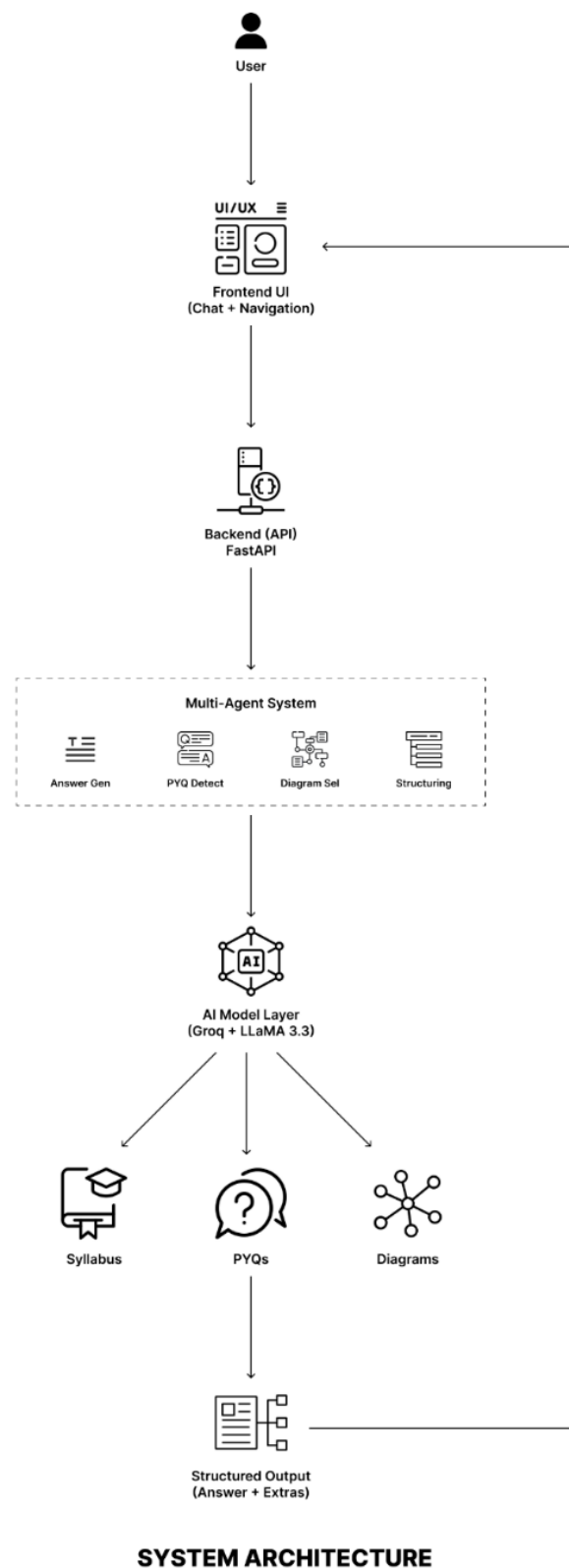
The frontend is built using HTML, CSS, and JavaScript, providing a chat-based interface similar to modern AI tools. It includes features such as a sidebar for subject selection, navigation tabs for chat, notes, and exam preparation, along with a dedicated practice mode where users can attempt answers. It also supports voice input and audio output, allowing users to interact with the system like a tutor. The frontend communicates with the backend using Fetch API, sending user queries or responses and receiving outputs in real time. The design is kept simple so that students can use it without any difficulty.

The backend is implemented using FastAPI and acts as the main processing unit of the system. It receives requests through API endpoints and routes them to different modules based on the type of interaction. The backend follows a multi-agent approach, where separate agents handle tasks such as answer generation, previous year question detection, diagram selection, answer evaluation, and conversational guidance. The system also manages voice-based inputs by converting speech into text before processing. The evaluation module compares user-written or spoken answers with model answers and generates feedback. This separation of tasks improves consistency and overall system performance.

The AI processing layer uses the Groq API along with the LLaMA 3.3 (70B) model to generate and evaluate responses. The model receives structured prompts from the backend and produces outputs in a predefined format. It is also responsible for comparing student answers with ideal answers, identifying missing points, and guiding improvements. In tutor mode, the model maintains a conversational flow, explains mistakes, and responds in a way similar to a human tutor. The use of Groq ensures faster inference, making the system responsive for real-time interaction.

The system also includes a data layer which contains syllabus-based content, textbook references, previous year questions, predefined diagrams, and model answer datasets. This data is used during processing to ensure that responses are aligned with academic requirements. Instead of relying only on the model's internal knowledge, the system uses this structured data to guide outputs, following a Retrieval-Augmented Generation approach.

The overall flow of the system starts when the user enters or speaks a query in the frontend. The request is sent to the backend, where it is processed by different agents. The PYQ agent checks for previous exam relevance, the answer generation agent creates a structured response, and the diagram agent selects the appropriate diagram. In practice mode, the system presents a question, collects the user's answer, and the evaluation module compares it with the model answer to provide feedback. In tutor interaction, the system continues the conversation, explains concepts, and guides the student step by step. The final response is then sent back to the frontend and displayed in a structured and easy-to-understand format.



SYSTEM ARCHITECTURE

Conclusion:

In this work, we proposed ELEVATE AI, an AI-based educational assistant designed specifically for exam-oriented learning. Unlike general-purpose tools, the system focuses on generating structured answers that are aligned with university syllabus, expected formats, and marking schemes. By integrating features such as previous year question detection, diagram support, and structured response generation, the system helps students not only understand concepts but also learn how to present answers effectively in exams.

The use of a multi-agent architecture along with a RAG-based approach ensures that the responses are consistent, relevant, and grounded in academic data. Additionally, the integration of high-performance inference through Groq improves the overall responsiveness of the system, making it suitable for real-time usage. This combination of techniques allows the system to bridge the gap between general AI tools and actual academic requirements.

Overall, ELEVATE AI provides a more focused and practical solution for students preparing for exams, helping them improve both their understanding and answer presentation. While there are still areas for improvement, the proposed system demonstrates how AI can be adapted to better suit domain-specific needs in education.

References:

- [1] V. J. Shute and J. Psotka, "Intelligent Tutoring Systems: Past, Present, and Future," in Handbook of Research on Educational Communications and Technology, 1994. Available at: https://www.researchgate.net/publication/235108246_Intelligent_Tutoring_Systems_Past_Present_and_Future
- [2] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017. Available at: <https://arxiv.org/abs/1706.03762>
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of NAACL-HLT, 2019. Available at: <https://arxiv.org/abs/1810.04805>
- [4] T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), 2020. Available at: <https://arxiv.org/abs/2005.14165>
- [5] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems (NeurIPS), 2020. Available at: <https://arxiv.org/abs/2005.11401>
- [6] OpenAI, "Applications of ChatGPT in Education: Opportunities and Challenges," 2023. Available at: https://www.researchgate.net/publication/373668841_The_opportunities_and_challenges_of_ChatGPT_in_education

