AI BASED SMART TUTOR

Prof. Vinutha Raghu[1], Shravanth R Naik[2], Sandeep Chavhan[3], Nandish N[4], Sanjay N[5]

[1]Assistant Professor, ISE, Acharya Institute of Technology, Bengaluru, Karnataka, India
[2]B.E Student, ISE, Acharya Institute of Technology, Bengaluru, Karnataka, India
[3]B.E Student, ISE, Acharya Institute of Technology, Bengaluru, Karnataka, India
[4]B.E Student, ISE, Acharya Institute of Technology, Bengaluru, Karnataka, India
[5]B.E Student, ISE, Acharya Institute of Technology, Bengaluru, Karnataka, India

---***---

Abstract

This paper explores how an AI-based online tutor addresses the limitations of traditional tutoring by providing personalized, scalable, and accessible learning experiences. Utilizing natural language processing and machine learning, it offers dynamic subject material, interactive quizzes, and adaptive learning paths. Features like real-time progress tracking and automated doubt resolution enhance inclusivity, particularly in underserved regions. The system also supports exam preparation and skill-based training through predictive analysis. By processing vast educational data, AI-driven tutoring has the potential to reshape education, making it more equitable and connected. This research examines its design, implementation, and impact, referencing prior work in AI-driven education.

Keywords: Smart Tutor, Python, Flask, Gemini API, PostgreSQL

1. Introduction

The concept of AI-based education systems has gained substantial momentum with advancements in artificial intelligence, enabling the creation of highly personalized and effective learning platforms. An AI-based online tutor leverages these technologies to provide learners with curated content, interactive lessons, and customized feedback, ensuring a more engaging and efficient learning experience compared to traditional methods. This technological intervention not only complements existing education systems but also addresses significant gaps in accessibility, adaptability, and scalability.

Unlike conventional education systems, which often struggle to address the individual needs of learners, AI tutors analyze user preferences, learning pace, and proficiency to tailor their teaching strategies. This adaptability enhances user satisfaction while improving knowledge retention and skill acquisition. Furthermore, the system ensures inclusivity by offering multilingual support, catering to diverse learning styles, and delivering content that aligns with individual.

Recent developments in this field include the integration of real-time doubt resolution, intelligent note summarization, and interactive learning flowcharts. These features, based by AI, ensure that learners receive immediate support and a clearer understanding of complex concepts. Moreover, AI-driven platforms have expanded to support diverse educational applications, such as competitive exam preparation, vocational training, and corporate upskilling, showcasing their flexibility and utility in various contexts. The integration of gamification techniques, such as quizzes, achievement badges, and leader boards, further boosts user engagement and motivates learners to achieve their academic or professional goals.

A significant milestone in AI education systems is the deployment of scalable architectures that cater to diverse user bases while maintaining a personalized approach. For instance, advanced natural language processing models, such as GPT-based frameworks, enhance the system's ability to understand and respond to user queries effectively. Additionally, machine learning algorithms analyze user interactions to provide data-driven insights into learning progress and recommend optimal learning paths. Cloud-based infrastructures and data encryption techniques ensure that the platform remains accessible, secure, and efficient across devices and regions, particularly in areas with limited technological resources.

The potential impact of AI-based online tutors extends beyond individual learning experiences to societal and economic levels. These platforms can bridge the gap between urban and rural learners, reduce dropout rates, and contribute to workforce readiness. For example, in underserved regions where access to qualified educators is scarce, AI tutors can serve as a reliable alternative, delivering high-quality education and skill development at scale.

This paper delves into the architecture of an AI-based online tutor, highlighting its core components, including user profiling algorithms, AI-based content curation, and real-time learning analytics. By leveraging these tools, the tutor provides an interactive and adaptive learning environment, fostering inclusivity and bridging educational gaps in regions where traditional systems fall short. The platform also incorporates features such as predictive analytics to identify areas of improvement, automated progress tracking to motivate learners, and voice-based assistance for hands-free learning experiences. Building on existing work in AI education, this study emphasizes practical implementation and real-world impact, positioning AI-based tutors as transformative agents in the education sector. These platforms not only redefine traditional teaching methods but also pave the way for equitable and innovative learning systems, empowering students and professionals worldwide. Additionally, this paper explores the ethical considerations and challenges associated with implementing AI in education, such as data privacy, algorithmic bias, and the digital divide, providing a roadmap for future developments in this domain.

2. Problem Definition

Traditional online learning platforms offer accessibility and convenience but face significant challenges that hinder effective and personalized learning experiences. These challenges include:

- Lack of Personalization: Existing platforms often provide generic learning content that does not adapt to individual learners' levels, preferences, or progress. This one-size-fits-all approach limits effectiveness.
- Overwhelming Information: Students are frequently bombarded with excessive information, making it difficult to focus on key concepts and relevant material.
- Limited Interaction: Traditional platforms lack mechanisms for interactive and engaging learning experiences, which are crucial for retaining knowledge and developing critical thinking skills.
- Inadequate Assessment and Feedback: Most systems fail to offer meaningful, real-time assessments and constructive feedback tailored to a learner's needs.
- Accessibility Barriers: Students in remote or underprivileged areas often face challenges in accessing quality learning resources due to technological or infrastructural limitations.

The proposed solution is an AI-based online tutor application designed to address these challenges by leveraging advanced technologies, including AI, machine learning, and cloud computing. The system provides a personalized, efficient, and engaging learning experience through the following key features:

- Dynamic Personalization: The AI system tailors learning content and paths based on individual progress, knowledge level, and learning pace.
- Content Summarization: Using AI algorithms, the tutor summarizes vast notes into concise and meaningful formats, focusing on critical concepts and topics.
- Interactive Learning Aids: Features such as flowcharts, quizzes, and guided tutorials enhance interaction and comprehension.
- Real-Time Assessments: The system generates practice questions and offers instant feedback to help learners identify and address weaknesses.
- Accessibility and Inclusivity: By utilizing cloud-based infrastructure, the platform ensures learning resources are available anytime, anywhere, bridging gaps for remote learners.

The architecture of the AI-based online tutor is designed to provide a seamless, efficient, and user-centric learning and experience. The system integrates AI algorithms, cloud-based storage, and interactive user interfaces to deliver adaptive and personalized education.

3. Architecture

The architecture of the AI-based online tutor is designed to provide a personalized, scalable, and efficient learning experience. It is built using a microservices architecture, with each component serving a specific purpose to ensure seamless integration and responsiveness. The core of the system is a Django web framework, which handles the user interface, data processing, and serves as the main application server. The system integrates with various APIs to fetch subject notes, summarize content, generate important questions, and provide interactive learning aids such as flowcharts.

and

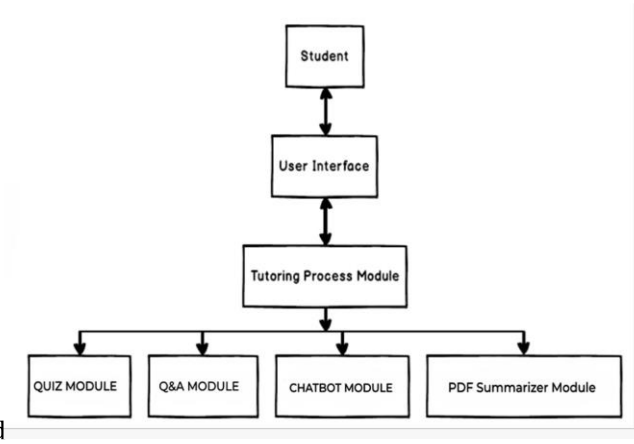

Figure 1: AI-Based Online Tutor Infrastructure

To ensure that the learning experience is both dynamic and personalized, the system leverages Google Cloud's machine learning models for text summarization

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 Apr 2025 www.irjet.net p-ISSN: 2395-0072

and question generation. The backend employs a PostgreSQL database to store user data, learning progress, and educational content. This allows the AI tutor to recommend study material based on the student's level and preferred learning time. The user's profile, including their current subject knowledge and learning preferences, is stored in the database, which helps in providing tailored recommendations.

The system utilizes a RESTful API to interact with the front-end, ensuring smooth communication between the database, machine learning models, and the user interface. For scalability and performance, the AI models and data processing are handled by Google Functions, which dynamically allocate resources based on usage and ensure high availability.

Upon students input their subject of interest, current level, and the time they wish to dedicate to learning. The AI then fetches the corresponding subject notes from the database and processes them to generate summaries, important questions, and other learning aids. Flowcharts and diagrams are dynamically created to illustrate complex concepts, providing a visual learning experience. Notifications and reminders are also integrated, prompting students about upcoming lessons or tasks. The architecture emphasizes seamless user interaction, scalability, and adaptability by integrating multiple cloud-based tools. By utilizing machine learning, natural language processing, and cloud services, the AI-based online tutor offers a comprehensive, adaptive learning solution designed for modern education.

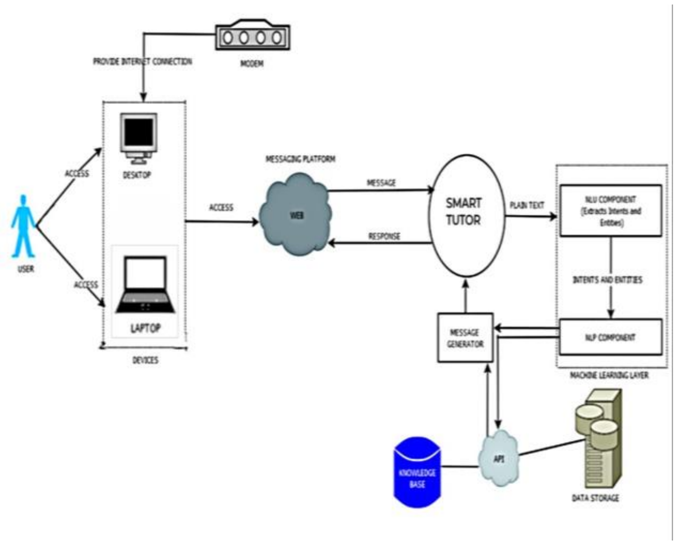

Figure 2: System Architecture

The AI-based online tutor system follows a well-defined architecture to ensure smooth and efficient functionality. In this system, the flow of data and interaction between components is carefully orchestrated, ensuring scalability and flexibility. The architecture is designed to handle various tasks like fetching notes, summarizing content, generating questions, and providing interactive learning aids using machine learning and cloud services.

Flow of the above Fig:

- Student Inputs Learning Preferences: The student logs into the system and inputs their subject, current learning level, and the amount of time they wish to dedicate to learning.
- AI Fetches Learning Material: The AI fetches the relevant subject notes and material from the database based on the student's inputs. These are then processed for summarization and question generation.
- Content Processing & Summarization: Using NLP models, the AI processes the notes, generating concise summaries, important questions, and learning aids like flowcharts and diagrams.
- Interactive Learning Interface: The frontend, built with modern web technologies like HTML, CSS, and JavaScript, displays the summarized content, interactive flowcharts, and key questions. Students can interact with the content through a user-friendly interface.
- Notifications & Reminders: The AI-based tutor sends notifications to the student about upcoming lessons, reminders, and rescheduled sessions based on their preferences.
- Student Progress & Feedback: The system tracks student progress and feedback. Based on the performance and feedback, it adjusts the material provided and recommends further learning paths.

Internal Structure

- Frontend: A combination of HTML, CSS, and JavaScript provides an interactive and user-friendly interface. The frontend communicates with the backend via RESTful APIs to fetch learning materials and interact with the AI.
- AI Models (Google Cloud): The AI models, such as text summarization and question generation, are hosted on Google Cloud Functions. These models process the data fetched from the database and generate learning material dynamically.
- Backend (Django): The backend handles user authentication, interacts with the PostgreSQL database, and coordinates the AI model interactions. It also manages the user profiles and learning data.
- PostgreSQL Database: The database stores user profiles, learning progress, subject material, and other necessary data to ensure a personalized experience.
- Google Cloud Functions: These handle the machine learning operations such as summarizing notes,

generating questions, and providing learning recommendations.

Steps Involved in Implementation:

- Frontend Development: Use frameworks like React.js or Vue.js to build an intuitive user interface. Students will use this interface to input learning preferences and interact with the learning content.

- AI Model Integration: Leverage Google Cloud's NLP APIs to perform text summarization, question generation, and recommendation services. Ensure these services are accessible through backend APIs.

- Backend API (Django): Set up the Django framework as the backend to manage user requests, interact with the database, and serve the AI-generated content. Use RESTful APIs to fetch content from the database and process it with AI models.

- PostgreSQL Integration: Store subject notes, user data, and learning progress in PostgreSQL. This ensures that the AI can provide personalized content and track student performance.

- Notification System: Implement a notification system to send learning reminders and session updates based on the student's schedule.

- Testing & Debugging: Thoroughly test the system for bugs, performance issues, and AI model accuracy. Use tools like Django's test framework and Google Cloud's debugging tools.

- Deployment: Deploy the AI models and backend to Google Cloud, while hosting the frontend on a cloud platform like Netlify or a decentralized platform like IPFS for scalability and accessibility.

4.Different Modules

The Online Smart AI Tutor integrates multiple AI-driven modules that work together to create a personalized, efficient, and interactive learning experience. The Quiz Module allows users to select a topic or subject for the quiz. Upon selection, the frontend sends a request to the Flask backend, which queries the Gemini API to fetch quiz questions related to the chosen topic. The backend stores the questions temporarily in a session and sends them to the frontend. Once the user completes the quiz, their answers are sent back to the backend, where the system compares them to the correct answers from the Gemini API. The system then calculates the score and provides instant feedback, helping users identify areas for improvement. This feature allows quizzes to be dynamic and adapt to the user's progress and knowledge level.

The Chatbot Module is a real-time virtual assistant that facilitates continuous user interaction. When a user submits a query or message, the frontend sends the query to the Flask backend. The backend then forwards the query to the Gemini API, which processes the input and generates a response based on context. This response is sent back to the frontend and displayed to the user in real time. By using Natural Language Processing (NLP), the chatbot ensures that the answers provided are context-sensitive, making the interaction feel conversational and natural. This module fosters a dynamic learning environment by offering instant clarification on concepts and queries.

The Q&A Module allows users to ask specific questions about a subject or topic. After the user submits a question, the frontend sends it to the Flask backend, which forwards the request to the Gemini API. The API analyzes the question and returns the most relevant, accurate answer. This system ensures that the answers provided are not only concise but also deeply informative, enabling students to better understand complex topics and resolve doubts quickly. Additionally, the backend can retrieve answers from a pre-existing FAQ database, offering another layer of assistance to the user.

The PDF Summarizer Module is particularly useful for students dealing with large volumes of text in educational PDFs. The user uploads a PDF document, which is sent to the Flask backend for text extraction. The backend processes the document and sends the extracted text to the Gemini API, which generates a summarized version by identifying key points and reducing the content to more manageable chunks. The summarized content is then sent back to the frontend, allowing users to access an easy-to-understand overview of their materials. Users can also annotate the summarized PDF, making it a more interactive study tool.

Each of these modules, from the dynamic quiz system to the interactive chatbot and the document summarizer, is designed to work seamlessly together. These features empower users with instant access to tailored content, immediate clarification of doubts, and the ability to navigate and engage with learning materials more efficiently. By integrating AI models like the Gemini API, the platform offers a comprehensive, adaptive, and scalable solution for personalized education, making learning more accessible, engaging, and effective.

5. Result and Analysis

The implementation of the "AI Based Online Tutor" successfully addresses the challenges of traditional tutoring systems by leveraging advanced AI algorithms and integrating them with an intuitive, user-friendly interface. The system efficiently fetches subject notes from a database and delivers them in a concise and easily digestible format. The AI's ability to summarize content, provide important questions, and guide students through learning with visual aids like flowcharts ensures that the learning experience is personalized and effective.

The system's backend, built with Django and PostgreSQL, handles user data securely, while the frontend provides an engaging experience with simple navigation and interactive elements. The integration of AI to recommend study materials based on the user's learning level and goals further enhances the tutor's personalized approach, increasing the likelihood of student engagement and retention.

Testing of the AI tutor system demonstrated its ability to deliver accurate and relevant study content based on user input. The AI model's effectiveness in summarizing notes and suggesting learning paths was verified through real-time interaction with students. While the system operates seamlessly under normal conditions, there are areas for improvement, particularly in terms of the AI's ability to provide highly customized feedback

Further analysis suggests that while the AI-based approach improves efficiency and user engagement, a few challenges remain. Scaling the system to handle a large volume of users while maintaining the performance of the AI model and database queries may require additional optimization. Moreover, there is room to enhance the AI's ability to engage users in a more interactive manner, with features such as voice interaction or real-time Q&A capabilities. Additionally, ensuring a seamless experience across different devices and platforms remains a key focus for future improvements.

The system, overall, demonstrates the potential of AI-driven learning solutions, offering a cost-effective and scalable method of delivering high-quality education. Future work will focus on improving the adaptability of the AI system, expanding its range of subjects, and exploring new methods for enhancing user engagement. Integrating real-time analytics to monitor student progress and incorporating adaptive learning techniques will further refine the tutor's ability to meet diverse learning needs.

The AI-based tutor also has the potential to integrate multi-modal learning, including interactive videos and exercises, to cater to diverse learning styles. By incorporating real-time feedback mechanisms, it can provide more personalized guidance, further improving user engagement. Additionally, incorporating peer collaboration features can create a sense of community, enriching the learning experience.

The system could also explore multilingual capabilities, broadening its accessibility across different regions. Continuous monitoring and adaptation to user needs will enhance the system's ability to deliver relevant and accurate content. In the future, expanding the tutor's capabilities to cover a wider range of subjects will make it a more versatile educational tool.

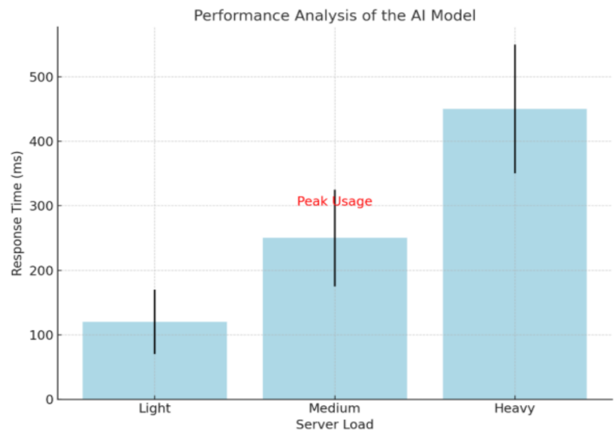

Figure 3: Performance analysis of the AI model. The variance in response times is attributed to server load and AI processing during peak usage.

The resource consumption of the system, including computational cost for processing user input and generating responses, was evaluated under various conditions:

AI Model Execution: Running the AI model to analyze and summarize subject notes typically consumes significant processing power. For example, generating a summary of a subject's notes requires multiple computations, with each summary processing consuming around 500–1,000 milliseconds per query, depending on the complexity of the subject material.

Data Retrieval: Fetching data from the PostgreSQL database to generate relevant study materials costs approximately 50–100 milliseconds per query. However, this can increase if the query involves filtering or retrieving large datasets.

User Interactions: Each user interaction (e.g., submitting a learning request or querying the tutor) generally takes around 1-2 seconds to process, including database retrieval, AI processing, and content delivery.

These system operations were evaluated for cost efficiency in terms of resource utilization, which is crucial for scaling the platform. Further refinements are necessary to optimize server-side processes and reduce latency in real-time interactions.

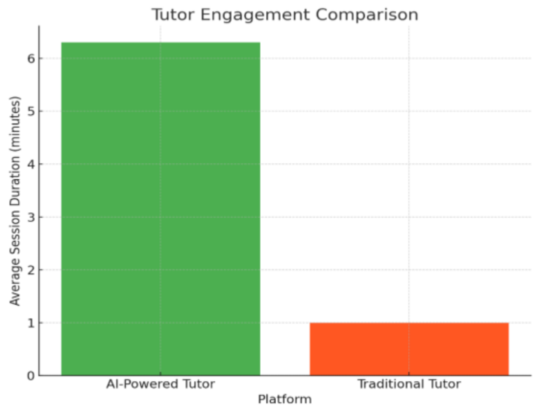

Figure 4: This Fig Comparison of tutor engagement and user satisfaction levels.

The results of user engagement studies highlight the strengths of the AI tutor in terms of user satisfaction and learning efficiency. Users reported higher satisfaction levels with the personalized guidance provided by the AI, which directly impacts the overall learning experience. Additionally, the AI tutor's ability to suggest relevant materials and create a structured learning path contributes to a more engaging experience compared to traditional methods. Despite a slightly higher upfront investment in setting up the AI system, the long-term benefits of scalable and cost-effective tutoring services make the AI-based solution a competitive alternative in the education market.

6.Conclusion

The Online Smart AI Tutor integrates advanced AI features to create a highly interactive and personalized learning experience. A key component is the AI-based Chatbot, which acts as a 24/7 virtual assistant, answering user queries, explaining concepts, and guiding learners through their studies using natural language processing (NLP).

Another essential feature is PDF Handling, allowing users to upload, read, and interact with educational PDFs. The AI extracts key information, summarizes content, highlights important points, and enables annotations, making large documents more manageable. The Q&A System provides subject-specific answers and an FAQ section for quick access to commonly asked questions. This feature enhances understanding by addressing knowledge gaps and offering detailed explanations.

Additionally, the Quizzes feature generates dynamic assessments tailored to the learner's progress. Adaptive quizzes with instant feedback help users identify weaknesses, reinforce learning, and improve retention.

These features collectively create an engaging, AI-driven educational environment that enhances learning efficiency and accessibility.

7. Future Work

Future enhancements for the AI-based online tutor will focus on key areas to improve its effectiveness and accessibility. One priority is adaptive learning, where the tutor evolves based on user interactions using real-time feedback and continuous learning techniques for better personalization. Another crucial development is multi-modal content integration, incorporating videos, interactive quizzes, and voice-based feedback to create an immersive learning experience. Scalability will also be a major focus, leveraging cloud-based services like AWS, GCP, or Azure to handle a growing user base efficiently. Additionally, privacy and data security will be essential, ensuring compliance with regulations like GDPR and implementing robust encryption to protect user data. These advancements will enhance the tutor's impact, making education more personalized, scalable, and secure.

8. References

[1] H. Shahri, M. Emad, N. Ibrahim, R. N. B. Rais, and Y. Al-Fayoumi, "Elevating Education through AI Tutor: Utilizing GPT-4 for Personalized Learning," 15th Annual Undergraduate Research Conference on Applied Computing (URC), 2024, IEEE.

[2] M. Alam, M. S. Alam, S. O. Siddique, N. Hasan, M. M. H. Tajwar, M. K. Rahman, and M. Rahman, "Quantifying Attention Levels in Individualized Online Tutoring: A Case of One-on-One Sessions," International Conference on Computational Intelligence, Networks and Security (ICCINS), 2023, IEEE.

[3] I. Pesek, N. Nosović, and M. Krašna, "The Role of AI in the Education and for the Education," 11th Mediterranean Conference on Embedded Computing (MECO), 2022, IEEE.

[4] A. Islam, R. Ali, G. Singh, B. Islam, A. Islam, and S. Hossain, "An Evaluation of AI-Enhanced Collaborative Learning Platforms," International Conference on Communication, Computer Sciences and Engineering (IC3SE), 2024, IEEE.

[5] A. Abduljabbar, N. Gupta, L. Healy, Y. Kumar, J. J. Li, and P. Morreale, "A Self-Served AI Tutor for Growth Mindset Teaching," 5th International Conference on Information and Computer Technologies (ICICT), 2022, IEEE.

[6] E. Frankford, C. Sauerwein, P. Bassner, S. Krusche, and R. Breu, "AI-Tutoring in Software Engineering Education: Experiences with Large Language Models in Programming Assessments," IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2024, IEEE.

[7] W.-H. Kim and J.-H. Kim, "Individualized AI Tutor Based on Developmental Learning Networks," IEEE Access, vol. 8, 2020, IEEE.

[8] A. Baillifard, M. Gabella, P. B. Lavenex, et al., "Effective Learning with a Personal AI Tutor: A Case Study," Education and Springer, 2024.