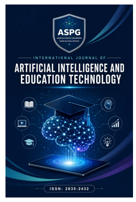
Int. J. of AI and Education Technology

♦ ISSN: 2835-2432 Vol: 06, No. 01, 2027 ♦ PP. 08–20
♦ Article DOI: https://doi.org/10.54216/IJAIET.060102



FULL LENGTH ARTICLE

SmartTutor-GPT: An AI-Driven Intelligent Tutoring System Using ChatGPT for Enhancing Personalized Learning in Online Education Platforms

Alcardo Alex Barakabitze[1,*]

[1] AI and Emerging Technologies Lab, Department of Informatics and IT, Sokoine University of Agriculture, Tanzania

Email: alcardo.alex@sua.ac.tz

Received: June 20, 2026 Revised: July 24, 2026 Accepted: August 16, 2026 ⋆ Corresponding author

ABSTRACT

In the current era, the rapid evolution of artificial intelligence has opened doors to combining education with cutting-edge AI capabilities. However, prominent online learning platforms like Coursera and Udemy have not yet tapped into this synergy. This paper proposes an intelligent tutoring system that leverages AI innovations such as ChatGPT to enhance the online learning experience and improve student performance. To achieve this, the paper use the OpenAI interface to interact with ChatGPT, employing a suite of frameworks including React, Spring Boot, and Flask for a seamless front-end and back-end service delivery. On the infrastructure front, the robust capabilities of AWS, Kubernetes, Docker, and Jenkins is employed to facilitate continuous integration and continuous deployment (CICD). To assess the effectiveness of our system, we use a combination of A/B tests, tree tests, and questionnaires. These methods are complemented by stress tests and monitoring mechanisms to ensure reliability. While the proposed system satisfied the majority of functional and non-functional requirements, the experimental group demonstrated greater learning gains than the control group, with average scores increasing from 0.50 to 3.33 (improvement = 2.83), compared with 0.66 to 3.00 (improvement = 2.33) for the control group. The lower variance in the experimental group (0.333 vs. 1.00) indicates more consistent outcomes; however, the t-test results suggest that the difference in learning outcomes was not statistically conclusive.

Keywords: AI ▪ Online learning ▪ Intelligent learning system ▪ ITS ▪ ChatGPT ▪ OpenAI ▪ digital learning ▪ digital education and innovation

1. INTRODUCTION

Education is fundamental to the socio-economic progress of societies and pivotal in alleviating poverty. Yet, millions globally remain devoid of this privilege. The United Nations Educational, Scientific and Cultural Organization (UNESCO) reports that by 2030, approximately 225 million children, adolescents, and youth will not be in school, while 750 million adults worldwide lack basic literacy [1]. These statistics emphasize the need for innovative interventions to close the educational divide. In the digital era, a significant challenge that the global educational arena grapples with is the conspicuous disparity in educational resources and opportunities. Since the early 20th century, Intelligent Tutoring Systems (ITS), sophisticated platforms that offer personalized guidance and feedback to learners without the need for human intervention, have emerged as a potentially transformative solution to bridge this disparity [2].

The advent of ITS has revolutionized the pedagogical landscape, enabling educators to disseminate educational content online, engage interactively with students, and evaluate learners' data [3]. This paradigm shift ensures a remarkable reduction of educational inequities, granting students, regardless of geographical constraints or national boundaries, unfettered access to consistent educational resources via ITS platforms. Recent advancements in machine learning and data analytics have greatly enhanced the capabilities of ITS, enabling a more personalized approach to instruction. These systems excel in autonomously analyzing students' academic progress and offering customized recommendations based on individual learning styles and knowledge levels. By evaluating a student's learning trajectory, these systems can provide targeted suggestions for improvement while considering their preferences and proficiency [4].

1.1 The Architecture of ITS

Intelligent Tutoring Systems (ITS) [2], [3] are sophisticated educational technologies developed to deliver personalized instruction and adaptive feedback to learners. These systems emulate many functions traditionally performed by human tutors by tailoring instructional approaches to individual learning needs with minimal human involvement. The fundamental objective of ITS is to improve the quality and effectiveness of learning by providing customized educational experiences that support diverse learner profiles. In contemporary education, ITS have become increasingly important due to their ability to accommodate varying student abilities, learning styles, and progress rates [5]. Their capacity to offer immediate feedback, identify knowledge gaps, and adjust instructional content dynamically contributes to improved learning outcomes. The ITS are particularly beneficial in online learning environments and large classroom settings where individualized support from instructors may be limited.

A widely recognized framework for Intelligent Tutoring Systems consists of four main components [28]: the Domain Model, Student Model, Tutor Model, and User Interface. Together, these components form the foundation of most ITS architectures. The Domain Model contains the subject-specific knowledge required for instruction, including concepts, principles, rules, and problem-solving procedures. Acting as an expert knowledge repository, it provides the basis for evaluating learner responses and identifying misconceptions or errors. By maintaining a structured representation of the learning content, the Domain Model enables the system to compare student performance against expert-level understanding and generate accurate feedback.

The Student Model is responsible for maintaining a dynamic representation of a learner's knowledge, skills, progress, and misconceptions. This component continuously analyzes learner interactions and compares them with the information stored in the Domain Model to estimate the learner's current level of understanding. Advanced computational approaches, including Bayesian networks, machine learning techniques, and educational data mining methods, are commonly employed to improve the accuracy of learner profiling and prediction of future learning needs.

The introduction of models like OpenAI's GPT[1] has demonstrated exceptional performance in tasks such as text generation, summarization, and question answering. Integrating GPT into ITS has the potential to significantly enhance the quality of interactions and personalization within the system [28]. GPT's extensive training on diverse datasets equips it to fulfill precise personalization requirements [5]. Our primary challenge is that, by configuring specific attributes and metrics, ITS can offer a more detailed analysis of learners' behaviors and preferences, and thus utilize the GPT model to provide tailored feedback [6]. This amalgamation of advanced machine learning techniques and state-of-the-art NLP capabilities holds the promise of significantly elevating the overall quality of the learning journey.

[1] https://openai.com/contributions/gpt-4/

1.2 The Contributions of this Paper

The primary objective of this system is to establish a scalable AI driven learning intelligent platform. As it dispenses pertinent educational resources, the system meticulously constructs learning profiles based on the attribute of "learning styles" [7]. By integrating the GPT model, it offers an ITS that not only supports multilingualism and multimedia but also boasts personalized adaptive feedback and interactive mechanisms. Central to the platform's goals is the aspiration to cultivate a more inclusive educational environment. It endeavors to cater to the unique requirements of every learner, guaranteeing a tailor-made, efficacious learning journey for all. This paper contributes to the advancement of Artificial Intelligence (AI)-enabled education through the design, implementation, and evaluation of an ITS powered by large language models. The key contributions are summarized as follows:

- We present SmartTutor-GPT, an AI-driven intelligent tutoring system that leverages the capabilities of ChatGPT and Generative AI technologies to provide personalized learning experiences. The SmartTutor-GPT, emulates human tutoring by offering adaptive guidance, individualized learning pathways, and real-time feedback to learners.
- We develop and implement cloud AI-enabled educational system architecture, that utilizes contemporary software engineering frameworks and technologies, including React, Spring Boot, Flask, OpenAI APIs, AWS cloud infrastructure, Kubernetes, Docker, and Jenkins-based Continuous Integration and Continuous Deployment (CI/CD) pipelines. This architecture ensures scalability, reliability, maintainability, and efficient service delivery.
- We introduce an interactive conversational learning in SmartTutor-GPT which is supported by ChatGPT, nabling learners to engage in natural language dialogue, receive instant explanations, ask questions, and obtain contextualized learning support.
- We provide a systematic evaluation of the proposed ITS using multiple assessment methodologies, including A/B testing, tree testing, user questionnaires, stress testing, and performance monitoring. This multi-dimensional evaluation offers insights into both user experience and system performance.

1.3 Significance of the SmartTutor-GPT System

The study provides empirical evidence on the opportunities and challenges of integrating Generative AI into online learning environments, demonstrating that AI-powered intelligent tutoring systems can support personalized, adaptive, and learner-centered education through continuous learner pro-

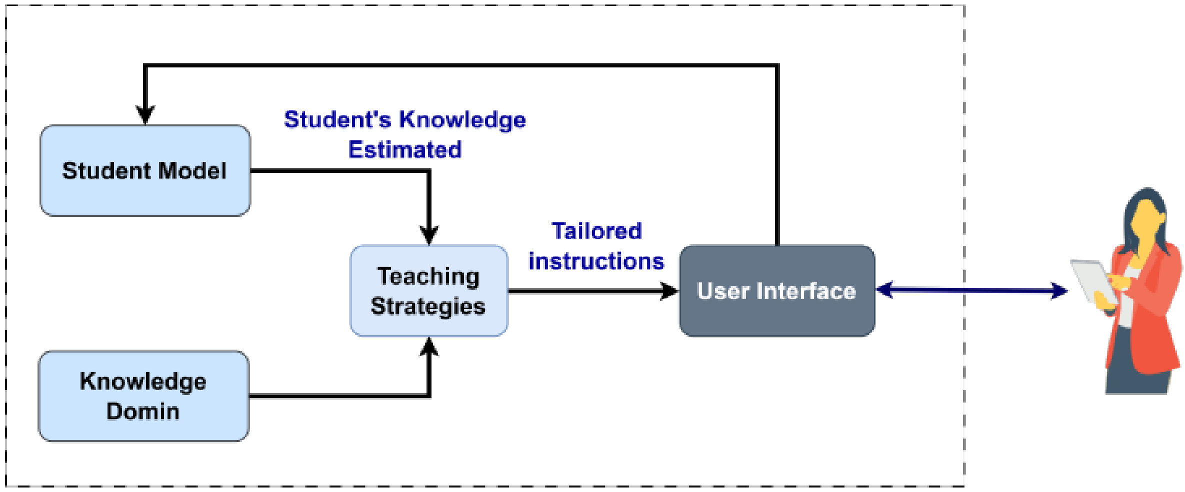


Figure 1. ITS architecture.

filing based on learning styles and behavioral patterns. The proposed platform combines multilingual communication capabilities, multimedia learning resources, and personalized content recommendations to enhance accessibility, inclusivity, and engagement for learners from diverse backgrounds. While the system successfully meets most functional and non-functional requirements and establishes a framework for next-generation intelligent e-learning platforms, the findings suggest that further research is necessary to determine its long-term impact on improving student learning outcomes.

1.4 Organization of the Paper

The rest of the paper is structured as follows: Section II: Discussion on user scenarios, highlighting the role of Intelligent Tutoring Systems (ITS) in offering personalized and adaptive learning. Section III: A review of prior work on ITS, encompassing applications in education and impacts on individual learners. Section IV: Describes the implementation details of the AI-driven online learning system, including the full technical realization from development to deployment. Section V: Describes in detail the evaluation methodology and results of the system, including system performance evaluation and user evaluation. Section VI: Summarizes the successes and shortcomings of the system and provides the direction for future work.

2. USER SCENARIOS

Tom, a college student with a keen interest in enhancing his English proficiency, aims to bolster his language skills for confident communication in international exchanges. However, during his studies, he often encounters questions that remain unanswered in real-time. Following a grammar practice session, he reviewed the answers, yet found himself puzzled by the rationale behind two incorrect responses. Unable to locate identical queries online, he had to wait for his English class three days later to seek clarification from his teacher. This disruption hampered his learning progress significantly.

As a user, Tom desires a feature to review his errors in tests, enabling him to grasp the underlying reasons for these mistakes.

Lily, a middle-aged woman, shares the aspiration of seamless communication during her travels abroad. Similarly, Mrs. Zhang, also middle-aged, seeks fluency in foreign conversations. However, they both face challenges in initiating their English journeys. Often, they invest days in online study materials only to realize the content's excessive difficulty, mismatched to their current level. This process not only squanders their time but also erodes their confidence.

A common thread among these users is their inability to afford private tutors to aid them in learning English. Taking these user needs into account, our AI Tutoring System offers free access. Users can engage in a self-assessment test before embarking on their learning journey. The AI system subsequently tailors suitable study materials based on their self-assessment results. Additionally, each learning unit concludes with an evaluation test to gauge progress. Post-assessment, users can consult the AI for clarifications on any queries, with the AI providing pertinent explanations based on user responses.

3. RELATED WORK

3.1 Related ITS Work

Abdolhossein et al [8] employed facial expression analysis to decode emotions related to learning, intervening, and engaging students through dynamic emotional avatars [8]. Wang [9] introduced a unique learning map derived from blogs, streamlining students' searches for blog-related content and guiding them towards focused learning maps, thereby promoting rapid and meaningful learning [9]. In 2013, Tanner and Danielle integrated gaming elements into the iSTART system, enhancing students' motivation to learn. They harnessed the successful characteristics of the ITS for extended interactions, augmenting student engagement and enthusiasm [10]. Oscar's CITS, enhanced with AI capabilities, caters to individual learning styles that align with learners' existing preferences. This system provides cues and suggestions, encouraging learners to independently address challenges and offering tailored feedback when errors occur [11]. Jason utilized an advanced facial expression detection system (like FaceReader 5.0), combined with self-reporting tools (such as emotional value scales), and skin conductivity measurements (using Affectivas' Q-2.0 Sensor) to concurrently assess students' emotional responses during learning [12]. Certain researchers delved into conceptualizing and constructing an ITS focused on problem-solving techniques, expanding the existing ITS framework to create the iTutor [13]. David presented an approach centered around mathematical challenges, with the system identifying students' problem-solving approaches and providing adaptive responses based on the student's decisions and the problem's constraints [14].

3.2 Technical Challenges in Implementing ITS

The implementation of ITS brings the promise of a more personalized and effective learning experience, but it also introduces several technical challenges [48]-[49]. These issues range from the integration of different technologies to ensuring the security of student data. Despite the benefits of ITS, it is essential to address these challenges to achieve successful implementation. The integration and standardization of various technologies are two of the foremost challenges in the development of ITS. The need for seamless technology integration stems from the diversity of tools and systems in the educational technology landscape, leading to potential version incompatibility and interface mismatches [15]. OpenAI's GPT-4 offers APIs that support integration with different technologies, but specific requirements may necessitate custom solutions [16]. Standardization of interfaces is also a complex issue due to variations in system designs and implementations [17]. Creating standardized interfaces is essential for smooth interaction with external services, particularly within the broader ecosystem of educational technologies ensuring smooth integration and standardization across the technology stack is crucial for the functionality and user experience of ITS.

As ITS may serve large numbers of users with fluctuating access volumes, auto-scaling becomes a key technical challenge [18]. While cloud solutions exist, balancing scalability with stability is difficult [19]. Costs, especially from APIs like GPT-4, are significant, requiring developers to balance educational benefits with expenses. Security is vital when integrating external APIs into education systems. Developers must adhere to data protection laws and prioritize student data safety [20]. Ensuring robust security measures minimizes risks like data breaches [21]. Overcoming these challenges is crucial for ITS's successful implementation, ensuring enhanced learning experiences.

3.3 Discussion of Technical Solutions

The primary target of our system is integrating the GPT model into the ITS. Studies indicate that AI-based technologies can enhance learning outcomes and motivation [22]. Specifically, AI-powered tutoring programs can boost students' academic performance and drive in educational settings [23]. In particular, ChatGPT, by offering tailored interactive support, can enrich the learning experience and heighten student engagement [24]. By exploring the integration of the GPT model with Intelligent Tutoring Systems (ITS), one can significantly address the limitations of ITS in personalized feedback and interaction. Additionally, this integration can support individualized tasks and activities aligned with students' needs and learning objectives.

In the implementation, this work envisioned two technological solutions. The first approach centers on enabling the GPT model to autonomously generate questions. Based on the questions posed by GPT, tutoring sessions for students are facilitated. The second approach emphasizes the GPT model's capability in feedback and guidance. Here, appropriate questions and prompts are relayed to the GPT model via a backend database, ensuring optimal guidance and feedback. However, based on multiple pieces of evidence, the performance of the GPT-3 model reveals certain constraints. The GPT-3 model sometimes struggles with natural language generation, occasionally producing uncontrollable content [25]. Given that contemporary language learning and testing now incorporate rich multimedia contents like audio, video, and images, the GPT model's inability to autonomously generate multimedia-format questions can only yield basic textual queries, significantly dampening students' interest [26].

Thus, in this endeavor, the work employed the second approach. Harnessing the GPT model's prowess in understanding and responding to natural language inputs, combined with its inherent self-learning capacities, the paper relay appropriate descriptors and metrics to the GPT model via a backend system [27]. This allows the model to offer tailored feedback and guidance to learners, thereby enhancing their overall educational experience.

4. IMPLEMENTATION

This section offers an overview of the implementation of Intelligent Tutoring Systems (ITS), providing detailed insights into both the frontend and backend components, data source, the integration of the GPT model, and the Continuous Integra-tion/Continuous Deployment (CICD) workflow. The structural organization of this section follows our technological stack, enabling a cohesive understanding of the separate elements that constitute the system.

4.1 Front-End

Continuing with this system, the frontend has been developed as a web service application to ensure portability and easy access for students across various operating systems and technological devices. To achieve this, the paper employed the three primary programming languages for the web: JavaScript, HTML, and CSS.

JavaScript, as our core programming language, has been complemented by NPM, serving as both an architecture and package manager. NPM facilitated the installation, removal, and testing of libraries crucial to our system. Through commandline interactions, NPM has managed our development environment, facilitated system building for publishing, and enabled code testing. This paper have harnessed the power of React, a free and open-source library, to craft a dynamic and interactive user interface. This approach has allowed us to modularize complex UI elements, creating reusable components and modules that enhance both efficiency and maintainability.

In conjunction with React, the paper have seamlessly integrated other libraries such as Axios. This library streamlines API integration by simplifying the process of making and managing requests. Additionally, Reactstrap has proven invaluable for incorporating new HTML and CSS elements.

By utilizing its components, UI design becomes more intuitive. Furthermore, the implementation of translation into the UI has been facilitated by i18Next. This involved creating distinct JSON files for the various languages involved.

Internally, this work adopted tools like Prettier and ESLint to ensure our code is clean and well-structured. Throughout the development process, stringent control over components was exercised primarily within the JavaScript files. This approach has been pivotal in maintaining code quality and consistency across the platform.

4.2 Back-End

Within the scope of this system, the backend assumes the primary responsibility of receiving and dispatching client requests, implementing business logic, and facilitating interactions with the database. Given the specific technology stack chosen, which aligns with the nature of a concise MVP development effort, the paper placed a strong emphasis on selecting open-source and cost-free frameworks and tools.

Our initial preference leaned towards the Spring Boot framework, serving as the cornerstone of our development. Its intrinsic features of automated configuration and embedded server capabilities have significantly streamlined our development process. By combining this with Maven, the focus narrows down to fundamental adjustments within the "pom.xml" configuration file and the "application" configuration file. This endeavor not only ensures version control and lifecycle management but also fosters consistency across platforms and systems.

For data persistence, the paper opted for MySQL as our database solution. This decision rests on a twofold foundation. Firstly, MySQL's InnoDB storage engine seamlessly aligns with the essential ACID properties - atomicity, consistency, isolation, and durability - ensuring transaction reliability and data in-tegrity. Additionally, MySQL supports foreign constraints, reinforcing data integrity. Secondly, MySQL is optimized for performance, delivering attributes like high throughput and low-latency read and insert operations, making it well-suited for applications handling substantial workloads. To address the potential need for query optimization and customization, this work turned to MyBatis as an Object-Relational Mapping (ORM) tool of choice. This selection is rooted in its exceptional flexibility in this domain, coupled with its capacity to handle intricate database structures and data transformations. Its smooth integration with Spring Boot further simplifies the implementation of data persistence.

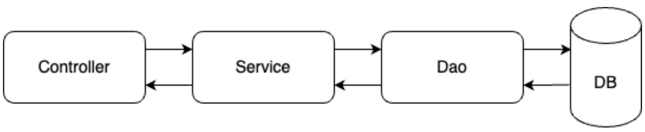


Figure 2. Three-tier architecture.

Within the operational workflow of the system, as illustrated in Figure 2, a strategic adoption of a three-tier architecture (Controller-Service-DAO) has been implemented. The Controller tier serves as the entry point for user requests, invoking relevant services and providing the frontend with the required data. The Service layer manages business logic and orchestrates communication with the data access layer (DAO). The DAO tier interfaces directly with the data repository, usually a database, to facilitate Create, Read, Update, and Delete (CRUD) operations.

Throughout the business flow, communication between the frontend and backend occurs in JSON format. For instance, during the user login process, the frontend sends a JSON-formatted request containing the username and password. This request is initially received by the Spring Boot controller, which then forwards it to the appropriate service for processing. The service layer engages in vital business operations, such as verifying account existence and authenticating decrypted passwords. Concurrently, the service layer interacts with the DAO layer by invoking methods and utilizing XML configurations for result mapping. This interaction, facilitated by MyBatis and the MySQL database, involves executing customized SQL statements to retrieve the required data.

This architecture offers several advantages. Its modular code design, achieved through hierarchical division into well-defined modules and components, reduces inter-layer coupling. This allows targeted development on specific segments with-out affecting others, enabling parallel development and swift responsiveness within the team. Additionally, this architecture sets the stage for future scalability improvements. For instance, if increased data volume and the introduction of additional cache mechanisms (e.g., Redis) become necessary, these upgrades can be implemented without altering the Controller and Service layers, ensuring smooth transitions. This approach combines practical and theoretical principles, resulting in a comprehensive implementation plan that underscores the significance of the chosen architectural and technological paradigms.

4.3 Data Source

This system sources its data from three primary avenues: user information acquired directly from users; educational videos, which have been strategically selected from YouTube; and question data, which has been inspired by and adapted from official language examinations such as IELTS, HSK, and DELE, and then stored in our database. While executing this segment, the implementation of this work encountered three primary challenges:

a) Database Design: Given the substantial volume of information at our disposal, it was imperative to adopt the Boyce Codd Normal Form (BCNF) in the database design. Such an approach mitigates redundancy and dependency in the database tables, thereby upholding a high degree of data integrity and consistency.
b) Multimedia Information Retrieval: Our database is not equipped to directly house multimedia resources, be it video, audio, or image files. To circumvent this limitation, references were retained to the locations of these resources. For videos from YouTube, the frontend makes a request to the backend, which then retrieves the video URL from the database. This is subsequently accessed by the frontend through YouTube's playback widget. As for image and audio resources, they reside on an AWS S3 server—a flat-storage device. The educational materials were uploaded to the relevant S3 directories and record the paths and filenames in the database. When the fron-

tend application requisitions specific learning materials, the backend supplies the stored path and filename, enabling the frontend to fetch the data from the S3 server via AWS authentication and the corresponding URL.

c) Transferring Multimedia Resources to the GPT Model: Given that the GPT model predominantly entertains text-based data for feedback, the paper resorted to tagging our data. For every multimedia piece that necessitates an in-depth feedback from GPT, this work append an appropriate descriptor, ensuring that GPT comprehends our inquiry optimally.

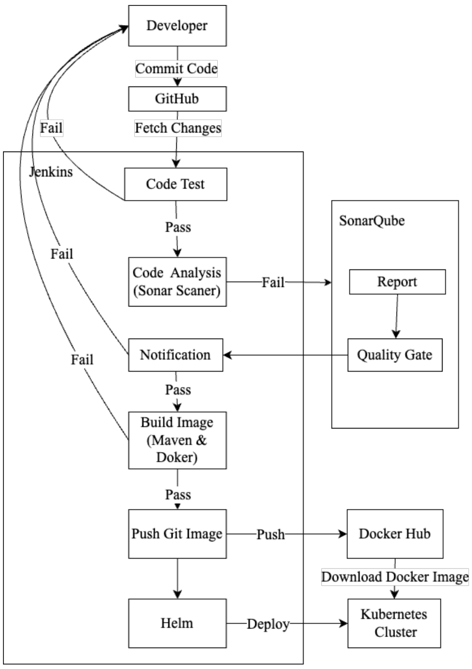


Figure 3. CI/CD pipeline.

Through this, the aim is to extract GPT feedback tailored for multimedia content.

4.4 GPT Integration

The paper have embraced a microservices architecture to implement intelligent feedback functionality through integration with the GPT model. This design approach offers the advantage of scalability and long-term maintainability. Given that our backend's primary role involves the reception and routing of requests, a lightweight and streamlined framework for API communication is essential. To address this need, this paper have opted for the Flask framework. Leveraging Flask, this work can swiftly configure an API interface that facilitates seamless interaction between the frontend and the GPT model. Upon the receipt of requests from the frontend, it can promptly engage with the GPT model to generate contextually appropriate responses. This paper harness the distinct attributes encapsulated by four dimensions - activist, reflector, theorist, and pragmatist - ascertained from the Learning Style test [7]. This strategic utilization enables the provision of tailored feedback in alignment with the individual learning styles of students. However, in light of the constraints associated with the GPT model, our interaction is confined to textual communication. In order to cater to the delivery of feedback in multimedia formats, a pragmatic solution has been implemented. This paper have associated non-textual data within our database with pertinent descriptions. This tactic enables us to manage queries and feedback pertaining to video and audio resources to a certain extent.

4.5 Cloud Implementation and Deployment

The proposed system uses AWS for cloud infrastructure, Docker for containerization and Kubernetes for managing docker containers. On top of that, the Jenkins as indicated in Figure 4 is used for continuous integration/continuous delivery.

Figure 3 is the CICD pipeline. It starts with a developer submitting the code that needs to be updated to Github, and the Jenkins server periodically checks Github for new commits. When a new commit is detected, Jenkins will start the corresponding pipeline process. Code testing is performed first, such as mvn test, and an exception is reported if a failure is detected. When the code testing is done, we use Sonar to analyze the code. The analyzed results of the code are uploaded to Sonar Qube for display. After the code is analyzed, the docker image will be built and uploaded to the docker hub after the image is uploaded, Jenkins will call the slave node (Kops server) to update the new image to the Kubernetes cluster using a rolling update.

The Kubernetes cluster runs on AWS cloud servers and is controlled using Kops. When creating the Kubernetes cluster, Kops server will automatically request EC2 instances, loadbalancer, target groups and other resources based on the command. Figure 5 is a diagram of the Kubernetes architecture. First the user accesses it through the domain name. The domain is bound to the Kubernetes loadbalancer via Route 53, and a Nignx ingress controller was used to control which front-end and back-end services the request needs to access. The services include a front-end service using the React framework, a Java back-end service using the Spring boot framework, a Python back-end service using the flask framework, and a MySQL database. For all front-end and back-end services, The ClusterIP, Kubernetes Deployment and HPA were used to manage all the pods. ClusterIP is a Kubernetes service type that manages which traffic is sent to which pods. In the architecture, ClusterIP accepts traffic from the ingress controller and passes the request to the corresponding pod.Kubernetes Deployment is a resource object that is used to define and manage a set of pods. When using Kubernetes Deployment, The number of pods, and when a running pod terminates unexpectedly, or the When a service is updated, Kubernetes Deployment starts new or stops old pods to ensure that the number of pods running in the cluster remains the same. HPA (Horizontal Pod Autoscaling) is used to autoscale the pods within the cluster. Firstly the autoscaling within the cluster is categorized into horizontal scaling and vertical scaling. Horizontal scaling scales the number of pods but does not change the CPU and memory limits of the pods. Vertical scaling does not change the number of pods, but dynamically changes the CPU and memory limits of the pods. HPA monitors the average CPU utilization of the corresponding service and when the CPU utilization reaches the value that was set (80%), new pods are started immediately to increase the load capacity of the cluster. In addition, when HPA detects low CPU utilization, it starts deleting excessive pods after five minutes. Similar to HPA, our cluster nodes

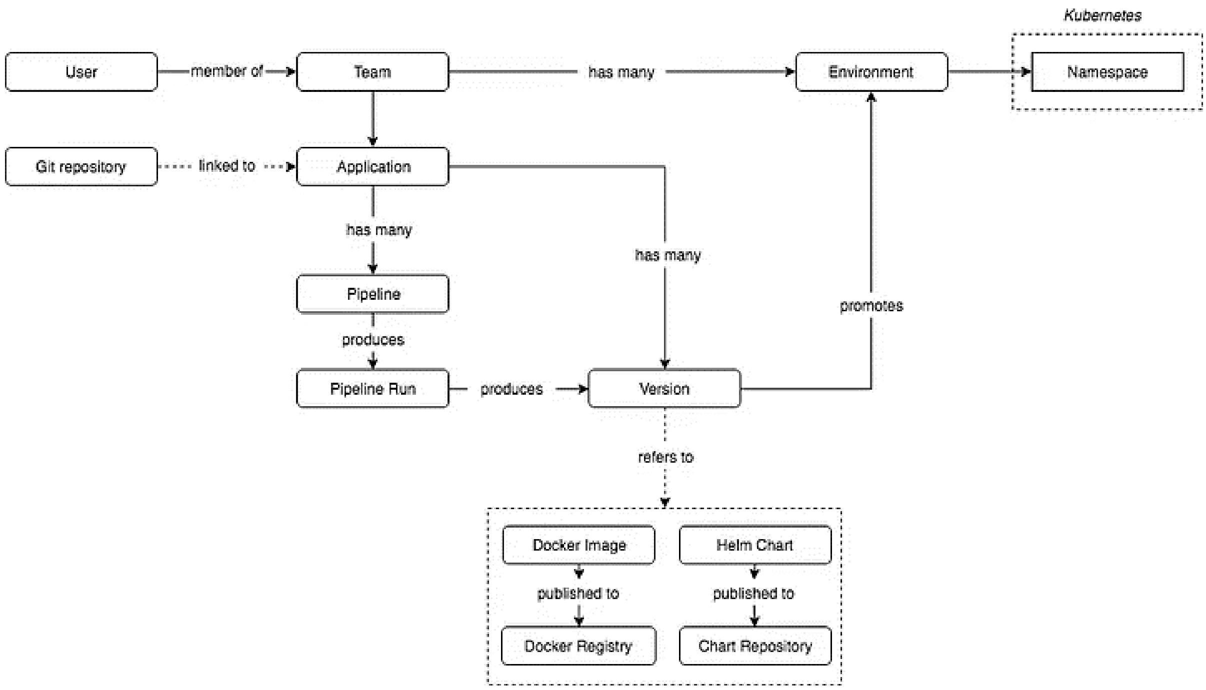


Figure 4. The workflow of Jenkins.

are also equipped with an autoscaling feature, whereby new nodes are self-started when the cluster CPU utilization is too high. When the CPU utilization drops, the cluster will reschedule the pods running on the nodes that will be deleted to other nodes within 10 minutes.

5. EVALUATION AND DISCUSSIONS

The evaluation of the system was divided into a system evaluation and a functional evaluation.

5.1 SmartTutor-GPT System Evaluation

The system evaluation consists of the following aspects: monitoring of basic metrics, scalability of the system and response time of users accessing the system.

5.1.1 Basic Metrics and Scalability

With traditional architectures (no microservices architecture), it is important to develop evaluation criteria for basic metrics such as CPU, memory, etc. due to the lack of scalability. However, since a microservices architecture is employed, the role of these base metrics is more often used as a criterion for auto-scaling. The evaluation metrics for auto-scaling are the time required for auto-scaling and whether the scaling is perceived by the user. Autoscaling is divided into node autoscaling and pod horizontal scaling. Auto-scaling of nodes takes about 2 to 3 minutes and is completely insensitive to the user since no traffic enters when node starts. For horizontal scaling of pods, the situation can be relatively complex. Each different service takes a different amount of time to start, for example, our Java backend service takes about 45 seconds to a minute or so to perform auto-scaling, which is extrapolated from the readiness time of the Http type probe. In addition, due to the use of probing technology, user traffic will be accessed after the probe detects that the service is ready, so the user will also not receive an incorrect return response. For pod scaling, some of the pods are removed when the average CPU usage is reduced to five minutes after which the pods can be reduced. The shortcoming of auto-scaling is that the CPU utilisation was set to start scaling to 80%, which is a resource-saving setting, but when a large number of requests come in, the system may perform a sudden scaling, during which the user will feel a significant increase in the response time of the service due to the queue stacking.

5.1.2 Response Time

The average response time is an important metric for measuring the system, and we use a stress test to measure the average response time of the system under normal conditions and when scaling up, respectively.

Table 1. Normal response time.

status	succ	avg rt	error
OK	100.00%	0.011	

Table 2. Busy response time.

status	succ	avg rt	error
OK	100.00%	0.0046	

5.2 Functional Evaluation

5.2.1 A/B Test

A total of six subjects were included in this test, all of whom were students of the School of Computing, UCD. Three of the subjects were assigned to the experimental group and the other three were assigned to the control group. Before the start of the experiment, members of both groups were pretested and their scores were recorded. During the experiment, the experimental group used our program for learning and the control group used the same learning materials as in the program, but did not use our service. At the end of the study, all members took a posttest and recorded their scores. Both

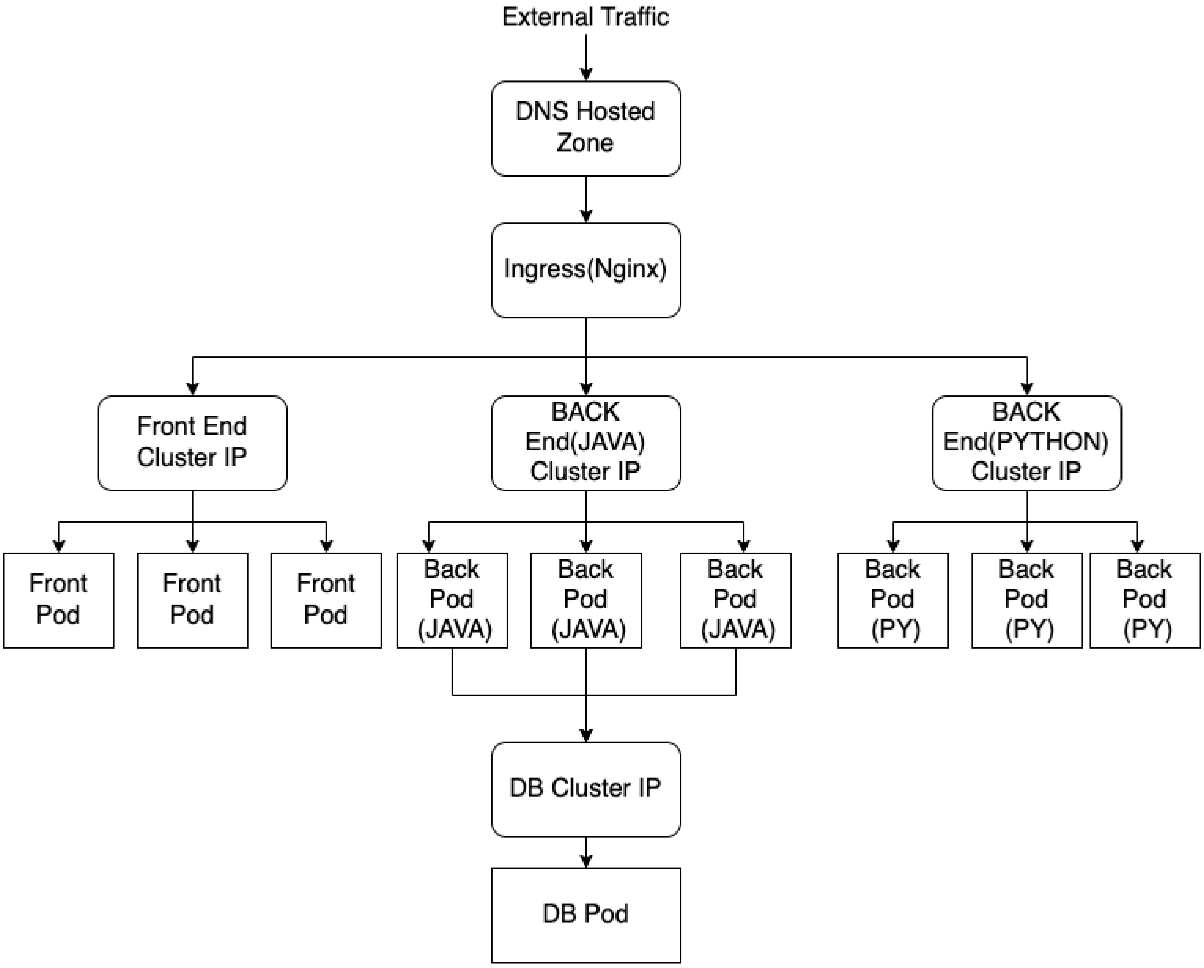


Figure 5. Architecture of Kubernetes.

the pre-test and the post-test consisted of five multiple-choice questions, and each subject answered the same questions.

We calculated the mean improvement of the subjects' pre-test and post-test and their variance for t-test. The data are shown in Table 3,

Table 3. T-test data.

Criteria	Experimental group	Control group
Average pre-test	0.5	0.66
Average post-test	3.33	3
Average improvement	2.83	2.33
variance	0.333	1

The independent samples for two-sided t-test were chosen for the following reasons: (a) The sample size was extremely small (n=3). (b) There was no overall standard (e.g., intelligence test) that could be used for comparison, and (c) We did not know whether our system would have a positive or negative impact on students (it is possible that after using the system students would not be as good as if they were self-taught). Equation (1) is the formula for the independent samples t-test:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

Table 1 shows the average response time under normal conditions. Table 1I shows the average response time when the server is experiencing auto-scaling. We can see that the average response time gets longer due to the stacking of the message queue, and it is not until the auto-scaling is complete that the average response time drops to a normal level.

The T-value of 0.75 was calculated by bringing the desired value into the formula. We chose 0.05 confidence interval as well as two sided t-test to get this critical value of 4.303 by looking up the table when the degree of freedom is 2. Since our t-value (0.75) is less than 4.303, it cannot be proved that our system has an improving effect on students' learning.

The shortcomings of this assessment method are the following two:

1. The number of participants in the experiment is too small, usually the group size for this type of experiment will be larger than five, in some cases, such as when schools promote new learning methods, a whole class will be used for the test. As we only had three people in each group taking part in the test, this made the t-value required to achieve confidence quite high. It is often difficult to prove statistically that the service we provide is effective.
2. Our test has only five multiple-choice questions, and with a small number of questions on the test, the random effects of subjects guessing their answers to questions will be greater than when there are many questions, which will make our final results inaccurate.

5.2.2 User Test

During a comprehensive user testing phase of our intelligent tutoring system, we engaged a diverse user group to ensure varied and in-depth feedback. This robust evaluation was driven by our commitment to inclusivity and a user-centred ethos.

Notably, our UI & UX design received widespread acclaim. Users lauded the intuitive design, swift page load speeds, and impeccable navigation aided by a well-structured information architecture. The comprehensive suite of tools and resources we offered was also recognized, showcasing the platform's robust functionality.

In addition, we have identified some important areas for improvement. Specifically, these encompassed shortcomings in accessibility for differently-abled users, the need for a more integrated social media experience, voiced demands for improved multilingual capabilities, and a consensus on refining the search function. In response to this invaluable feedback, we advocate a systematic accessibility review in compliance with WCAG standards, more seamless social media integration, the introduction of translation facilities aligned with user-focused localisation, and the advancement of the search module, spotlighting faceted search and predictive suggestions. With a foundation in iterative refinement, we hold every piece of feedback in high regard, using it as a beacon for our ongoing system enhancements.

5.2.3 Tree Testing

Tree Testing is an evaluation method in the field of user interface design and information architecture used to test whether the navigation and organization of information on a website or application is clear and easy to understand. The primary goal of tree testing is to verify that users can easily find specific information or perform specific tasks without being influenced by page design, style, or visual elements. We tested our system with the assistance of Optimal Workshop. Before the test, we entered the sitemap of the AI Tutoring website into Optimal Workshop's website, set up tasks to be completed by the tester, and generated links for the test. The testers were asked to find the appropriate features, and Optimal Workshop recorded and analyzed the data for each selection. As Figure 6 shown, we found that 100% of the users were able to find the correct feature, with 33% of the users not finding the feature directly.

Figure 6. Results of the tree test.

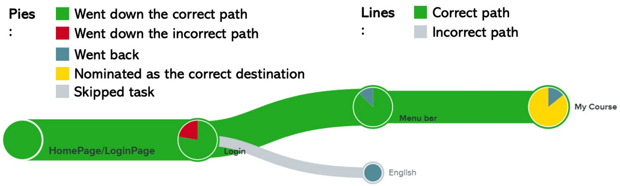


Figure 7. View of the pie tree.

The users in this section went down the incorrect path to English before they find the correct feature. Figure 7 shown that our sitemap has a good recognition degree. This shows that the sitemap of our system has good recognition and users can use it without any training and find the corresponding features. However, there is still room for improvement, for example, the next version of the system to add a newbie guide function, for the use of the process to give the appropriate tips.

5.3 Discussion of the Findings

Recent advancements in ITS have increasingly shifted from traditional model-tracing architectures toward AI-driven, data-intensive, and generative learning environments [28]. Classical ITS such as Cognitive Tutors and ANDES established the foundation for adaptive learning through structured problem-solving and knowledge tracing, but their scalability and conversational flexibility remain limited [28]. In contrast, recent ITS research emphasizes deep learning, multimodal interaction, and large language model (LLM) integration to enhance personalization and engagement. For instance, LLM-based tutoring systems have demonstrated the ability to provide contextual explanations, adaptive feedback, and dialogue-based learning interactions beyond rule-based constraints [6], [40], [44]. SmartTutor-GPT aligns with this new paradigm by leveraging GPT-based generative capabilities to deliver dynamic instructional responses and learner-specific explanations. Unlike earlier ITS that rely heavily on predefined knowledge structures, SmartTutor-GPT introduces a flexible architecture where learning content is enriched through real-time contextual augmentation, representing a shift toward generative ITS ecosystems similar to emerging frameworks reported in recent AI-in-education studies [38],[39]. From a system architecture and scalability perspective, SmartTutor-GPT adopts a microservices-based cloud-native design, which reflects current best practices in scalable educational platforms. Recent ITS infrastructures increasingly rely on Kubernetes-based orchestration, containerization, and distributed cloud services to support high concurrency and global accessibility [13], [25], [42]. Compared to monolithic ITS architectures such as earlier Cognitive Tutors, modern systems prioritize elasticity and fault tolerance to support large-scale deployments in online education environments [31] - [41]. SmartTutor-GPT demonstrates similar scalability advantages through node and pod autoscaling, enabling efficient handling of fluctuating workloads. However, consistent with findings in cloud-based ITS research, autoscaling latency remains a critical limitation, particularly under sudden traffic spikes where queue accumulation increases response time [42]-[43]. This challenge is also observed in other cloud-native e-learning systems where resource provisioning delay affects real-time user experience despite high system availability [41], [46-47]. The SmartTutor – GPT can be also integrated with QoE –centric multimedia services [51] for teaching multimedia contents and its applications [52]-[59]. Therefore, while SmartTutor-GPT demonstrates strong infrastructural resilience, further optimization of predictive scaling mechanisms is required to align with next-generation ITS performance expectations including teaching and learning using multimedia contents [56] –[59].

Table 4 summarizes the comparison of SmartTutor-GPT with traditional, adaptive ML-based, and LLM-based intelligent tutoring systems.

Table 4. Comparison of SmartTutor-GPT and traditional, adaptive ML-based, and LLM-based intelligent tutoring systems.

S/N	Dimension	Traditional ITS	Adaptive ML-Based ITS	LLM-Based ITS	SmartTutor-GPT (Proposed System)
1	Core Architecture	Rule-based + model tracing systems (fixed pedagogical rules	Probabilistic models (BKT, DKT, deep learning pipelines) for knowledge tracking	Transformer-based LLMs integrated with tutoring frameworks; prompt-based adaptation	Microservices-based cloud-native system integrating GPT API, learner modeling and scalable backend
2	Adaptivity Mechanism	Predefined step-by-step adaptation; limited flexibility	Data-driven personalization using historical learner data	Contextual adaptation via prompts and conversation history	Hybrid adaptivity: learning styles + GPT-generated dynamic feedback
3	Pedagogical Model	Cognitive task decomposition and expert systems	Statistical learning models for mastery estimation	Weak explicit pedagogy; relies on prompt engineering	Learning-style-driven adaptive pedagogy + GPT reasoning
4	Interaction Type	Structured problem-solving (step-by-step hints)	Semi-adaptive exercises and quizzes	Natural language conversational tutoring (Kasneci et al., 2023)	Fully conversational + contextual tutoring with instructional metadata injection
5	Use of AI Models	No deep AI; symbolic reasoning only	Machine learning (RNNs, Bayesian models, transformers)	Large Language Models (GPT-4, GPT-4o, LLaMA)	GPT-based LLM integrated with domain context augmentation
6	Learning Personalization	Limited to predefined learner states	Strong personalization via knowledge tracing (DKT/BKT)	Moderate personalization via prompt context	Strong personalization using learning styles and interaction history
7	Multimodality Support	Minimal (text only)	Limited multimedia integration	Increasing multimodal capability (text + image + code)	Multilingual and multimedia learning support (text, explanations, quiz augmentation)
8	Multimodality Support	Minimal (text only)	Limited multimedia integration	Increasing multimodal capability (text + image + code)	Multilingual and multimedia learning support (text, explanations, quiz augmentation)
9	Response Generation	Pre-authored responses	Template-based + probabilistic outputs	Fully generative responses (LLMs)	Context-aware GPT-generated responses with structured instructional prompts
10	Real-Time Feedback	Immediate but rule-limited feedback	Data-driven feedback based on model prediction	Real-time conversational feedback but inconsistent pedagogy	Real-time adaptive feedback with learning-style conditioning
11	Real-Time Feedback	Immediate but rule-limited feedback	Data-driven feedback based on model prediction	Real-time conversational feedback but inconsistent pedagogy	Real-time adaptive feedback with learning-style conditioning
12	Learning Outcome Evidence	Strong evidence (long-term studies)	Moderate improvements in performance	Mixed results; often engagement > measurable gains (Zawacki-Richter et al., 2024)	Positive improvement trend but not statistically significant (small sample limitation)
13	Key Strength	Strong pedagogical grounding	Accurate knowledge tracing	Highly flexible natural language tutoring	Integration of LLM, pedagogy and scalable cloud architecture
14	Key Limitation	Not scalable, rigid	Requires large labeled datasets	Weak pedagogical control, hallucination risk	Limited dataset size; weak statistical significance of learning gains

In terms of pedagogical effectiveness and learning outcomes, SmartTutor-GPT contributes to ongoing debates in AI-enhanced education regarding measurable learning gains. Traditional ITS such as Cognitive Tutors and ASSISTments have consistently shown statistically significant improvements in student performance due to structured feedback loops and validated cognitive models [45], [38]. However, recent studies on generative AI-based tutoring systems report mixed results, particularly when sample sizes are small or when evaluation instruments are limited in scope [6],[43]. The findings from SmartTutor-GPT follow this trend, where A/B testing indicates improved mean performance in the experimental group, but t-test analysis does not confirm statistical significance. This aligns with broader empirical observations that LLM-based ITS often enhance engagement and comprehension but require larger-scale longitudinal studies to validate learning impact rigorously [38]. Furthermore, limitations such as small sample size and limited assessment questions in this study reflect common methodological challenges in early-stage ITS prototyping reported in recent literature [46].

Finally, from a human-computer interaction and learner experience perspective, SmartTutor-GPT demonstrates strong alignment with recent ITS design principles emphasizing usability, accessibility, and adaptive learning experiences. Modern ITS frameworks increasingly incorporate user-centered design, multilingual support, and inclusive learning features to accommodate diverse learner populations [39]. Similar systems such as AI-enhanced MOOCs and conversational tutoring agents have reported improved learner satisfaction when intuitive interfaces and real-time feedback mechanisms are integrated [44] –[46]. SmartTutor-GPT reflects these developments through its positive user feedback regarding interface usability, navigation, and learning resource organization. However, identified limitations in accessibility, multilingual depth, and search optimization are consistent with challenges reported in recent ITS deployments, particularly those relying on generative models without structured pedagogical constraints (Li et al., 2023). The proposed future enhancements—such as emotion recognition, improved learning style adaptation, and open-source extensibility—are also consistent with next-generation ITS research directions, which emphasize affective computing, adaptive intelligence, and collaborative ecosystem development [38], [41]-[46]. The SmartTutor-GPT represents a transitional ITS architecture bridging classical adaptive tutoring systems and emerging generative AI-powered educational platforms.

6. CONCLUSION

Our SmartTutor-GPT system successfully realized the functional and non-functional requirements of an intelligent tutoring system. The unique feature of the system is to provide descriptions for all learning materials and quizzes, and make the GPT responses more relevant by sending the descriptions to the GPT API at the same time as the student asks a question. Our system still has some shortcomings. First, the combination of the Learning Styles Questionnaire and the GPT did not produce significant results; the GPT only mentioned learning styles in the responses, and there was not enough variation in the responses across learning styles. Second, the results of the t-test did not show in a statistically significant way that our system improves students' learning outcomes. The reason for this problem may be due to the small sample size; with a sample size of 3, only a large enough difference can be considered statistically significant. In addition, we still have several ideas that need to be realized in the future. First, we would like to add emotion recognition to the system for monitoring students' learning status. Alerting students when they are distracted or taking into account their learning state when they ask questions to the GPT to learn a particular segment. Secondly, we would like to include other kinds of learning style tests and combine learning styles with the student's learning process rather than the feedback provided by the AI. Finally, we want to make the system an open-source educational platform that offers the opportunity to upload custom courses, so that anyone can add courses that meet the specifications to our platform and make the AI service available to their students.

ACKNOWLEDGMENT

The author sincerely thank the reviewers for their valuable comments and constructive feedback, which significantly improved the quality of this manuscript.

REFERENCES

[1] UNESCO Institute for Statistics (UIS). Literacy. Accessed Aug. 20, 2023.

[2] A. Ni and A. Cheung, "Understanding secondary students' continuance intention to adopt ai-powered intelligent tutoring system for english learning," Education and Information Technologies, vol. 28, no. 3, pp. 3191–3216, 2023.

[3] B. Jacob, D. Berger, C. Hart, and S. Loeb, "Can technology help promote equality of educational opportunities?" RSF: The Russell Sage Foundation Journal of the Social Sciences, vol. 2, no. 5, pp. 242–271, 2016.

[4] V. Yuskovych-Zhukovska, T. Poplavska, O. Diachenko, T. Mishenina, Y. Topolnyk, and R. Gurevych, "Application of artificial intelligence in education. problems and opportunities for sustainable development," BRAIN. Broad Research in Artificial Intelligence and Neuroscience, vol. 13, no. 1Sup1, pp. 339–356, 2022.

[5] S. Grassini, "Shaping the future of education: exploring the potential and consequences of ai and chatgpt in educational settings," Education Sciences, vol. 13, no. 7, p. 692, 2023.

[6] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier et al., "Chatgpt for good? on opportunities and challenges of large language models for education," Learning and Individual Differences, vol. 103, p. 102274, 2023.

[7] P. Honey and A. Mumford, Learning styles questionnaire. Organization Design and Development, Incorporated, 1989.

[8] A. Rrafzadeh, S. Alexander, F. Dadgostar, C. Fan, and A. Bigdeli, "How do you know that i don't understand?" A look at the future of intelligent tutoring systems. Computers in Human Behavior, vol. 24, no. 4, pp. 1342–1363, 2008.

[9] K.-T. Wang, Y.-M. Huang, Y.-L. Jeng, and T.-I. Wang, "A blog-based dynamic learning map," Computers & Education, vol. 51, pp. 262–278, 08 2008.

[10] G. T. Jackson and D. S. McNamara, "Motivation and performance in a game-based intelligent tutoring system." Journal of Educational Psychology, vol. 105, no. 4, p. 1036, 2013.

[11] A. Latham, K. Crockett, and D. McLean, "An adaptation algorithm for an intelligent natural language tutoring system," Computers & Education, vol. 71, pp. 97–110, 2014.

[12] J. M. Harley, F. Bouchet, M. S. Hussain, R. Azevedo, and R. Calvo, "A multi-componential analysis of emotions during complex learning with an intelligent multi-agent system," Computers in Human Behavior, vol. 48, pp. 615–625, 2015.

[13] D. Wang, H. Han, Z. Zhan, J. Xu, Q. Liu, and G. Ren, "A problem solving oriented intelligent tutoring system to improve students' acquisition of basic computer skills," Computers & Education, vol. 81, pp. 102–112, 2015.

[14] D. Arnau, M. Arevalillo-Herráez, and J. A. González-Calero, "Emulating human supervision in an intelligent tutoring system for arithmetical problem solving," IEEE transactions on learning technologies, vol. 7, no. 2, pp. 155–164, 2014.

[15] S. Ramaila and A. J. Molwele, “The role of technology integration in the development of 21st century skills and competencies in life scsiences teaching and learning.” International Journal of Higher Education, vol. 11, no. 5, pp. 9–17, 2022.

[16] A. Tsikora, “A feasibility study of available natural language chatbot technologies,” 2023.

[17] A. Lorenz, M. Raven, and K. Blind, “The role of standardization at the interface of product and process development in biotechnology,” The Journal of Technology Transfer, vol. 44, no. 4, pp. 1097–1133, 2019.

[18] J. Yang, C. Liu, Y. Shang, B. Cheng, Z. Mao, C. Liu, L. Niu, and J. Chen, “A cost-aware auto-scaling approach using the workload prediction in service clouds,” Information Systems Frontiers, vol. 16, pp. 7–18, 2014.

[19] D. C. Marinescu, Cloud computing: theory and practice. Morgan Kaufmann, 2022.

[20] M. N. Asghar, N. Kanwal, B. Lee, M. Fleury, M. Herbst, and Y. Qiao, “Visual surveillance within the eu general data protection regulation: A technology perspective,” IEEE Access, vol. 7, pp. 111 709–111 726, 2019.

[21] S. Slade and P. Prinsloo, “Learning analytics: Ethical issues and dilemmas,” American Behavioral Scientist, vol. 57, no. 10, pp. 1510–1529, 2013.

[22] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., “Language models are few-shot learners,” Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.

[23] A. S. Patil and A. Abraham, “Intelligent and interactive web-based tutoring system in engineering education: Reviews, perspectives and development,” in Computational Intelligence for Technology Enhanced Learning. Springer, 2010, pp. 79–97.

[24] K. Srinivasa, M. Kurni, and K. Saritha, “Harnessing the power of ai to education,” in Learning, Teaching, and Assessment Methods for Con-temporary Learners: Pedagogy for the Digital Generation. Springer, 2022, pp. 311–342.

[25] M. Zhang and J. Li, “A commentary of gpt-3 in mit technology review 2021,” Fundamental Research, vol. 1, no. 6, pp. 831–833, 2021.

[26] J. Costley and C. H. Lange, “Video lectures in e-learning: effects of viewership and media diversity on learning, satisfaction, engagement, interest, and future behavioral intention,” Interactive Technology and Smart Education, vol. 14, no. 1, pp. 14–30, 2017.

[27] M. Firat, “How chat gpt can transform autodidactic experiences and open education,” Department of Distance Education, Open Education Faculty, Anadolu Unive, 2023.

[28] M. Zerkouk, M. Mihoubi, and B. Chikhaoui, “A Comprehensive Review of AI-based Intelligent Tutoring Systems: Applications and Challenges,” arXiv preprint arXiv:2507.18882, Jul. 2025.

[29] E. Kochmar et al., “Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems,” International Journal of Artificial Intelligence in Education, vol. 32, pp. 323–349, 2022.

[30] M. K. J. Carlon and J. S. Cross, “Knowledge tracing for adaptive learning in a metacognitive tutor,” Open Education Studies, vol. 4, no. 1, pp. 206–224, 2022.

[31] L. Qiu, M. Zhu, and J. Zhou, “OPKT: Enhancing knowledge tracing with optimized pretraining mechanisms in intelligent tutoring,” IEEE Transactions on Learning Technologies, vol. 17, pp. 841–855, 2024.

[32] Y. Zeng, Q. Zhang, N. Xie, and B. Yang, “Application of deep self-attention in knowledge tracing,” arXiv preprint arXiv:2105.07909, 2021.

[33] Z. Wang, X. Feng, J. Tang, G. Y. Huang, and Z. Liu, “Deep knowledge tracing with side information,” arXiv preprint arXiv:1909.00372, 2019.

[34] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu and J. -J. Vie, "Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing," 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 2018, pp. 1182-1187.

[35] M. K. J. Carlon and J. S. Cross, “Progressive knowledge tracing: Modeling learning process from abstract to concrete,” Expert Systems with Applications, vol. 238, p. 122280, 2024.

[36] H. A. S. Alrakhawi, N. Jamiat, and S. S. Abu-Naser, “Intelligent tutoring systems in education: A systematic review of usage, tools, effects and evaluation,” Journal of Theoretical and Applied Information Technology, vol. 101, no. 4, pp. 1205–1226, 2023.

[37] S. K. N. A. Rahim, A. H. Jaafar, and G. Baskaran, “Intelligent tutoring system: New criteria and evaluation to measure students’ degree of mastery,” Journal of ICT in Education, vol. 10, no. 2, pp. 113–131, 2023.

[38] UNESCO, “AI and education: Guidance for policymakers,” UNESCO Report, 2023.

[39] G. Kostopoulos, V. Gkamas, M. Rigou and S. Kotsiantis, "Agentic AI in Education: State of the Art and Future Directions," in IEEE Access, vol. 13, pp. 177467-177491, 2025.

[40] Qian, Y. Pedagogical Applications of Generative AI in Higher Education: A Systematic Review of the Field. TechTrends 69, 1105–1120 (2025).

[41] M. Zerkouk, M. Mihoubi, and B. Chikhaoui, “A comprehensive review of AI-based intelligent tutoring systems: applications and challenges,” arXiv preprint arXiv:2507.18882, 2025.

[42] Q. Li and L. Zhang, "Design and Implementation of Online Education Platform with Microservice Architecture," 2024 IEEE 15th International Conference on Software Engineering and Service Science (ICSESS), Changsha, China, 2024, pp. 173-177.

[43] S. Dwivedi et al., “Artificial intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research,” International Journal of Information Management, vol. 70, 2023.

[44] A. Scarlatos, N. Liu, J. Lee, R. Baraniuk, and A. Lan, “Training LLM-based tutors to improve student learning outcomes in dialogues,” in Artificial Intelligence in Education: Proceedings of the 26th International Conference on Artificial Intelligence in Education (AIED 2025), Lecture Notes in In Artificial Intelligence in Education. AIED 2025. Lecture Notes in Computer Science(), vol 15877. Springer, Cham.

[45] A. Gaeta, F. Orciuoli, A. Pascuzzo, and A. Peduto, “Enhancing traditional ITS architectures with large language models for generating motivational feedback,” Computers and Education: Artificial Intelligence, vol. 9, Art. no. 100433, Dec. 2025.

[46] T. Zhang, "Enhancing Efficient Personalized Learning and Educational Management in Universities Using Graph Neural Networks in Intelligent Tutoring Systems," in IEEE Access, vol. 14, pp. 44775-44797, 2026.

[47] Barakabitze, Alcardo Alex, William-Andey Lazaro, Anangisye, Ainea, Neterindwa, Mkwizu, Michael Hamza, Maziku, Hellen, Matofali, Alex Xavery, Iddi, Aziza, Sanga, Camillius, Transforming African Education Systems in Science, Technology, Engineering, and Mathematics (STEM) Using ICTs: Challenges and Opportunities, Education Research International, 2019, 6946809, 29 pages, 2019.

[48] Barakabitze, A. A., Kitindi, E. J., Sanga, C., Kibirige, G., & Makwinya, N. (2015). Exploring students’ skills and attitudes on effective use of ICTs: Case study of selected Tanzanian public secondary schools. Universal Journal of Educational Research, 3(6), 407–425.

[49] Barakabitze, A. A. (2014). The Context of Education Initiatives, Importance and Inhibitors of ICTs towards Improving Teaching and Learning in Tanzania: A Critical Literature Review. Information and Knowledge Management, 4(10), 83–91.

[50] Oubibi, M.; Fute, A.; Kangwa, D.; Barakabitze, A.A.; Adarkwah, M.A. Interactive Technologies in Online Teacher Education in Africa: A Systematic Review 2014–2024. Educ. Sci. 2024, 14, 1188.

[51] Barakabitze, A. (2020) QoE-Centric Control and Management of Multimedia Services in Software Defined and Virtualized Networks. Thesis. University of Plymouth.

[52] A. A. Barakabitze, I. -H. Mkwawa, A. Hines and R. Walshe, "QoE-Aware Dynamic Resource Management in Future Softwarized and Virtualized Networks," in IEEE Access, vol. 11, pp. 93310-93330, 2023.

[53] Alcardo Barakabitze; Andrew Hines, "QoE Management of Multimedia Services Using Machine Learning in SDN/NFV 5G Networks," in Multimedia Streaming in SDN/NFV and 5G Networks: Machine Learning for Managing Big Data Streaming, IEEE, 2023, pp.73-97.

[54] Alcardo Barakabitze; Andrew Hines, "Multimedia QoE-Driven Services Delivery Toward 6G and Beyond Network," in Multimedia Streaming in SDN/NFV and 5G Networks: Machine Learning for Managing Big Data Streaming, IEEE, 2023, pp.185-201.

[55] Alcardo Barakabitze; Andrew Hines, "Emerging Applications and Services in Future 5G Networks," in Multimedia Streaming in SDN/NFV and 5G Networks: Machine Learning for Managing Big Data Streaming, IEEE, 2023, pp.133-145.

[56] Kitindi, E. J., Alex, A. A., Sanga, C., Shabani, A., Kibirige, G., Phillip, J., & Oketchi, J. (2014). Mobile phone based payment authentication system: An intervention for customers’ bank account fraud in Tanzania. International Journal of Information and Communication Technology Research, 4(9), 325–?.

[57] A. A. Barakabitze et al., "QoE Management of Multimedia Streaming Services in Future Networks: A Tutorial and Survey," in IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 526-565, Firstquarter 2020.

[58] Barakabitze, A. A., & Walshe, R. (2022). SDN and NFV for QoE-driven multimedia services delivery: The road towards 6G and beyond networks. Computer Networks, 210, 108950.

[59] Alcardo Barakabitze; Andrew Hines, "Multimedia Streaming Services Delivery in 2030 and Beyond Networks," in Multimedia Streaming in SDN/NFV and 5G Networks: Machine Learning for Managing Big Data Streaming, IEEE, 2023, pp.203-220.