International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 12, Issue 2 (2026)
https://doi.org/10.31436/ijpcc.v12i2.720

Enhancing Student Performance through EduMate: An AI-Powered Smart Tutoring System

Zobayer Md Ahsanul Mahbub, Mahamat Ali Adam, Norsaremah Salleh*

Department of Computer Science, Kulliyyah of ICT, IIUM

*Corresponding author: norsaremah@iium.edu.my
(Received: 7th June 2026; Accepted: 3rd July, 2026; Published on-line: 30th July, 2026)

Abstract— In the modern educational landscape, students possess unique learning styles, yet many institutions continue to rely on a one-size-fits-all teaching approach. This traditional methodology often results in struggling students falling behind while advanced learners remain unchallenged. This paper presents EduMate, an AI-powered personalized tutoring system designed to adapt to individual learning styles, proficiency levels, and academic progress. Utilizing a combination of Machine Learning (ML) and Large Language Models (LLM), EduMate functions as a web-based platform that offers tailored lessons, dynamically generated quizzes, and real-time AI conversational feedback. The system features dedicated interfaces for students, teachers, and administrators, effectively bridging learning gaps while equipping educators with actionable insights into student performance.

Keywords— Personalized learning, smart tutoring system, educational technology.

I. Introduction

Education is the cornerstone of human development, yet traditional educational systems struggle to keep pace with rapid technological advancements. Traditional teaching methods predominantly treat all students as identical learners, despite the reality that each student has distinct learning styles, strengths, and pacing requirements (Luckin et al., 2016) [1]

Recently, Artificial Intelligence (AI) has emerged as a transformative tool in education. According to Holmes, Bialik, and Fadel (2019), AI enables personalized instruction, allowing students to learn at their own pace and preferred style [2]. Implementations of AI in education offer innovative ways to support students, making the learning process highly effective and engaging [3][4].

The quality of education in large classrooms is frequently compromised because teachers lack the time and resources to provide individualized attention [5]. Consequently, struggling students often go unnoticed and quietly fall behind. Furthermore, every learner possesses unique requirements; visual learners may need diagrams, while others may require interactive exercises or more time to grasp concepts [6]. When these diverse needs are unmet, students experience frustration and a loss of academic interest. Additionally, many students hesitate to ask questions due to shyness, creating cumulative learning gaps [7]. While various online learning platforms exist, most fail to offer genuine personalization, delivering static content without adjusting to the learner's knowledge level [8]. An intelligent, affordable, and flexible AI-powered tutoring system is urgently needed to provide personalized learning experiences that function akin to a private human tutor [9].

The primary objective of this project is to develop a smart tutoring system that utilizes artificial intelligence to support personalized learning by adjusting content and teaching styles based on student behavior and performance. To achieve this, the specific objectives are:

i) To study and understand different learning styles and their effects on student performance.

ii) To design and develop an AI-based system that collects data on student learning behavior and uses it to generate customized learning paths;

iii)To integrate dynamic features such as adaptive quizzes, real-time feedback, and progress tracking;

iv)To test the system with real users to analyze its effectiveness in improving learning outcomes and;

v) To ensure the resulting platform is highly accessible and easy to use for students from all backgrounds.

The proposed system is designed to facilitate personalized learning for a distinct, multi-level user base. The primary target audience includes secondary school and university students who require tailored educational content that adapts to their unique learning speed and preferences. Additionally, the system is intended for teachers and educators who seek smart technological assistance to monitor student analytics, reduce manual workloads, and offer targeted interventions. Finally, the platform serves educational institutions looking for affordable, scalable, and AI-driven tutoring solutions to elevate overall academic quality.

https://doi.org/10.31436/ijpcc.v12i2.720

This paper is organized as follows: Section II describes the Literature review. Section III describes the methodology whereas Section IV presented the analysis of results. Section V presented discussion of the findings. Finally, Section VI concludes this study.

II. Literature Review

The integration of Artificial Intelligence in education has opened new possibilities for personalized learning [9]. Several notable systems have been developed to utilize AI for tutoring and academic support. One notable platform is known as Khanmigo (by Khan Academy) [10]. It is an AI-driven virtual tutor developed in collaboration with OpenAI. It uses a conversational AI model to engage with students naturally, guiding them through problem-solving steps rather than directly providing answers. Squirrel AI Learning is another platform that pioneered adaptive learning platform from China [11]. It employs machine learning algorithms to build individualized learning paths based on detailed data gathered from student interactions.

MATHia (by Carnegie Learning) has been developed as an intelligent tutoring system [12]. It focused on mathematics that uses cognitive modeling and AI to simulate the behavior of a one-on-one human tutor. Another application is known as Syntea, which has been developed by IU International University of Applied Sciences [13]. It is an AI-powered tutoring assistant designed for university students, offering on-demand academic assistance using Natural Language Processing (NLP).

Each of these platforms brings valuable benefits to learners and educators, which demonstrate the potential of AI in education. Khanmigo guide students with problem-solving steps rather than giving direct answers. It supports critical thinking and student independence [10]. Its conversational style is less intimidating, making it highly suitable for shy students and helping them build problem-solving skills that are useful beyond specific subjects. Meanwhile, Squirrel AI Learning is a platform that excels in fine-grained personalization by detecting minor patterns in student behavior and adapting in real-time [11]. This precision allows for highly targeted interventions, reducing time wasted on already-mastered content and building student confidence. MATHia, on the hands, provides sharp, research-based mathematical instruction supported by detailed analytics. This allows teachers to identify and address learning challenges efficiently, effectively bridging the gap between traditional teaching and one-on-one tutoring. Finally, Syntea is designed for higher education and it offers flexible and immediate academic support that is especially useful for distance and adult learners. It empowers students to study independently by providing instant explanations across a wide range of subjects.

Despite their strengths, these systems exhibit distinct limitations that create barriers to widespread, general-purpose use. For instance, Khanmigo operates as a fully cloud-based service requiring constant internet access, which limits its usefulness in low-resource environments and poses a risk of students becoming overly reliant on the AI without fully engaging in deeper learning [10]. Similarly, Squirrel AI Learning relies heavily on extensive data collection from students, raising ethical and privacy concerns regarding data usage. Because it was originally designed for the Chinese education system, its global adaptations also still face challenges with diverse user interfaces and curriculum alignment. Furthermore, platforms like MATHia are exclusively focused on mathematics, restricting their usefulness for students seeking support in other academic subjects. Moreover, MATHia's interface can be overwhelming for younger or less tech-savvy users, and it lacks interactive or fun features, which can make learning feel tedious [12].

Finally, the effectiveness of systems like Syntea is strictly limited to the internal curriculum of its host institution, IU International University. This lack of broader scope restricts its usefulness for external students and limits its ability to support diverse academic needs outside of that specific environment. EduMate addresses these gaps by offering an affordable (open-source foundations), multi-subject (Science, English, Computer Science, Math), real-time adaptive platform with a highly simplified, accessible user interface.

III. Methodology

To build a flexible and responsive AI system, Extreme Programming (XP) was adopted as the core development methodology [14]. XP relies on rapid creation in smaller phases, emphasizing fast feedback, frequent testing, and user-centric design. The system was broken down into manageable modules (e.g., personalized quizzes, LLM chatbot, dashboard), allowing the team to code in short cycles, test immediately, and refine the interface continuously based on user input. Fig. 1 shows the development step employed in this study.

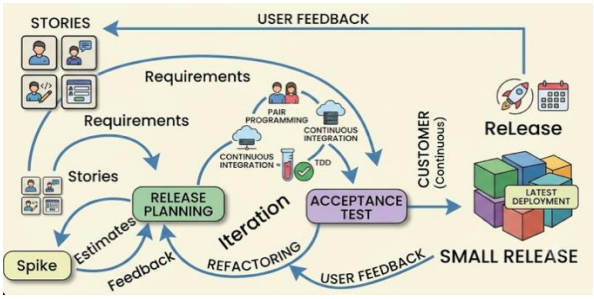

Fig. 1 Extreme Programming process

https://doi.org/10.31436/ijpcc.v12i2.720

In this study, XP was used to ensure development teams working in short, iterative cycles, typically lasting one to three weeks. Each iteration begins with the user describing desired features through "user stories," which we can then estimate for effort. The developers and users jointly negotiate which stories will be completed in the upcoming iteration, ensuring the most valuable features are prioritized and delivered in small, functional increments.

During development, XP focuses on a collaborative and technical practices to maintain quality and adaptability [14]. These core practices include pair programming, where two developers work together at a single workstation to continuously review code and solve problems; test-driven development (TDD), where automated unit tests are written before any production code; and continuous integration, requiring developers to integrate code multiple times into a universal test suite to catch errors immediately. Furthermore, the user remains actively involved as a part of the development team, providing feedback and conducting acceptance testing to verify that each small release meets their needs. Through this iterative process of listening, testing, coding, and designing, we are able to produce high-quality software that remains aligned with the customer's evolving requirements.

A. *Questionnaire Design*

To better understand the needs, preferences, and challenges faced by students in their learning journey, we conducted a survey questionnaire as part of the initial research for EduMate. The goal of the survey was to assess how students perceive current educational tools, their struggles with traditional learning methods, and their openness to AI-based personalized education. The survey was distributed online among IIUM undergraduate students in Gombak Campus. The survey questions comprise of the following questions:

- What is your preferred learning style?
- How often do you use online learning platforms? (e.g. Coursera, Udemy, Khan Academy)
- Have you ever used an AI-based learning tool before? (e.g. ChatGPT, Deepseek)
- Would you be interested in using a system that suggests lesson, quizzes, and exercises based on your learning style and performance?
- What features would you find helpful in an AI-powered tutoring system?

B. *Requirements Specification*

System requirements were driven directly by user needs. EduMate consists of the following functional requirements.

TABLE I
FUNCTIONAL REQUIREMENTS

Req ID	Description
FR-AUTH-01	The system shall allow users to register an account.
FR-AUTH-02	The system shall allow users to log in and log out securely.
FR-ROLE-01	The system shall support role-based access for Student, Teacher, and Admin.
FR-COURSE-01	The system shall allow teachers to create, update, and delete courses.
FR-ENR-01	The system shall allow teachers to enroll students into a course.
FR-ENR-02	The system shall restrict course access so only enrolled students can view course content.
FR-LESSON-01	The system shall allow teachers to create and manage lessons under a course.
FR-LESSON-02	The system shall allow teachers to upload learning materials (files/links) for lessons.
FR-QUIZ-01	The system shall allow teachers to create and manage quizzes and questions.
FR-QUIZ-02	The system shall allow students to attempt quizzes and submit answers
FR-FB-01	The system shall display quiz results and feedback after submission
FR-PROG-01	The system shall store quiz attempts and update student progress records
FR-TDASH-01 `	The system shall provide a teacher dashboard showing student performance and progress analytics.
FR-CHAT-01	The system shall provide an AI tutor chat feature for student questions and explanations.

C. *Technologies Used*

The technology stack was chosen to maximize AI integration and system scalability. Python was selected as the primary programming language due to its robust ML libraries [15]. Hugging Face Transformers (DistilBERT) and the Gemini API were utilized for lightweight, real-time natural language processing and chatbot capabilities [16]. The frontend was developed using React and Bootstrap for a responsive interface, paired with FastAPI for asynchronous backend API management [17][18][19]. Firebase and MySQL were integrated for real-time data syncing and structured relational data storage [20][21].

D. *Logical Design*

The logical design of EduMate describes how the system works before full implementation. It identifies the main users (student, teacher, and admin), key processes such as lessons, quizzes, feedback, and progress updates, and how these processes interact through the frontend, backend, and database. This design helps keep the system structured,

https://doi.org/10.31436/ijpcc.v12i2.720

reduces development errors, and ensures all features match the project requirements.

The class diagram for EduMate shows how the main parts of the system are structured and connected. At the centre is the User class, which is the base for three types of users: Student, Teacher, and Admin. Each of them has specific roles. For instance, students take quizzes and receive feedback, teachers create lessons and track progress, and admins manage users and content. The system includes Lesson, Quiz, and Question classes that represent the learning materials, and a Feedback class that gives students their results. To make the system smart and personalized, an AI Engine class generates lessons, quizzes, and feedback based on each student's performance. There is also a Chatbot that answer students' questions and a progress tracker to monitor learning progress. This structure helps organize the system clearly so that every part works together to create a smooth and personalized learning experience.

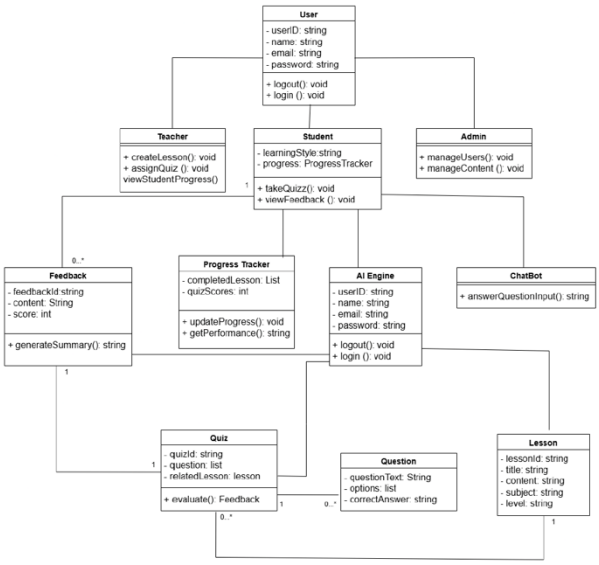

Fig. 2 Class Diagram

E. Frontend and Backend Integration

The frontend was developed using React and Bootstrap. React was used to build the pages as components. Bootstrap was used for layout and styling. It keeps the design consistent. It also supports responsive display. EduMate includes public pages such as Home, Login, and Registration. It also provides separate views for Student, Teacher, and Admin. After a user logs in, the system shows the correct dashboard based on the selected role. The frontend stores the login token and sends it with future requests. This allows the user to access protected modules. Examples include Courses, Lessons, Quizzes, AI Help, Notifications, and Profile. When the user clicks a button like "View Lesson" or "Start Quiz", the frontend sends a request to the backend and waits for the response.

The backend was developed using FastAPI. It handles the main business logic of the system. It provides API endpoints for login, user role checking, course and lesson retrieval, quiz operations, and progress updates. The backend validates each request before returning 55 data. It also ensures that students cannot access teacher or admin functions. The same rule applies to other roles.

IV. Results

A total of 54 students has responded to the survey. The findings from the survey revealed that 72.2% of respondents prefer visual, auditory, or reading/writing learning styles, with visual learning being the most favored at 33.3%. Consequently, 72.2% of students reported that they sometimes or often struggle in class because traditional teaching methods do not align with how they learn best. The findings also highlighted a strong readiness for digital and AI solutions, showing that nearly 89% of students regularly use online learning platforms and 72.2% have previously interacted with AI tools. Notably, 100% of the surveyed students expressed interest in using an AI-based educational system that personalizes lessons and quizzes to their performance. When asked about specific system features, the majority of students requested personalized learning paths (61.1%), progress tracking dashboards (61.1%), and instant feedback on quizzes (50%).

The implemented EduMate prototype successfully delivered specialized modules for distinct user roles:

- Student Dashboard: Features visual mastery tracking, enrolled course summaries, and a "Weak Topics" identifier to guide focus. Students access Personalized Quizzes, which adjust to their knowledge level to reduce stress, and receive instant feedback upon submission
- AI Help Chatbot: An integrated LLM-powered chat interface where students can ask complex academic questions and receive conversational, natural-language explanations without waiting for human intervention
- Teacher Dashboard: Provides educators with tools to upload PDF/text lessons, assign dynamic quizzes generated automatically by the AI based on lesson text, and monitor specific student metrics
- Admin Dashboard: Centralizes system monitoring, displaying active API/Database connections, and facilitating the management of user roles and course catalogs

The Landing page is the first screen displayed to users (see Fig. 3). It introduces EduMate and explains the main purpose of the platform. It also provides a clear "Get Started" button and navigation links to sections such as About,

https://doi.org/10.31436/ijpcc.v12i2.720

Features, Workflow, and Impact. From this page, users can move to the login process.

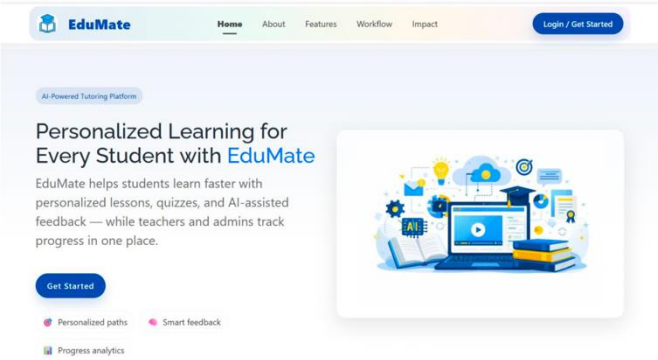

Fig. 3 Landing Page

The student interface is designed to support learning and practice. The student dashboard gives a quick summary. It shows mastery, enrolled courses, lesson count, and weak topics. Students can open Courses and Lessons to access materials (see Fig 4-5). They can also go to Quizzes and pick a lesson to attempt a quiz. The AI Help page lets students ask questions during study. It returns short explanations to support learning. These features help students follow their progress over time.

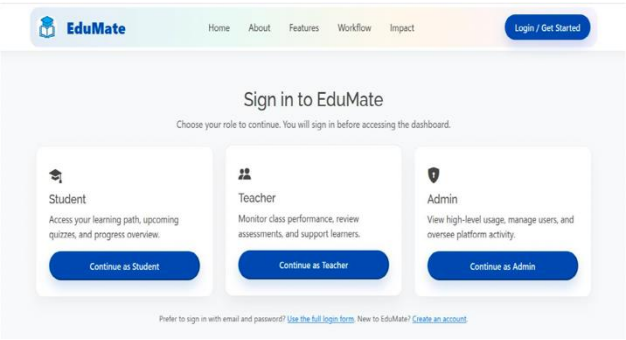

Fig. 4 User Sign-in Page

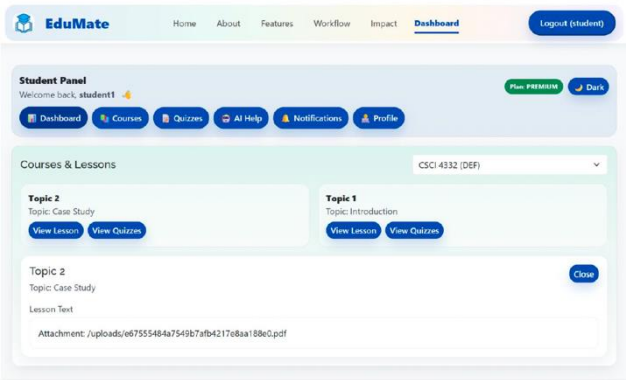

Fig. 5 Course and Lessons Page

System testing was done after the core modules of EduMate were implemented. The aim was to confirm that the system works as expected. It also checks that the system is stable during normal use. Testing was carried out from the user point of view. This means the testers followed real tasks that a student, teacher, or admin would do. The testing covered the full flow of the system. It starts from the Home page and sign-in role selection. It continues with registration and login. After login, it checks if the correct dashboard is shown. It also verifies that each role only sees the modules allowed for them. For example, students should not see admin functions. Teachers should not access admin management pages. This helps confirm role-based access control. Functional testing was applied to the key modules. For students, the tests include viewing courses, opening lesson materials, attempting quizzes, and using AI Help. For teachers, the tests include uploading lesson materials and generating quizzes. For admins, the tests include viewing system statistics and managing users and courses. During testing, the expected results were compared with the actual results. Any mismatch was recorded. After that, fixes were applied and the test was repeated. This approach helps ensure EduMate meets the main project requirements and provides a smooth experience for users.

The test plan uses task-based functional testing. Each tester followed a set of common tasks in EduMate. The tasks were based on the main modules shown in the interface. Testing involved different roles such as Student, Teacher, and Admin. All the test case includes pre-conditions, dependencies, and step-by-step actions. The expected output and actual output were recorded. The test status was marked as Pass or Fail. This method helps verify that EduMate meets the key project requirements and works smoothly for users.

V. Discussion

The deployment of EduMate demonstrates that integrating lightweight AI frameworks (DistilBERT) and external LLM APIs (Gemini) into a web-based educational platform can effectively bridge the personalization gap prevalent in traditional schooling. As identified in our survey, students are actively seeking adaptive tools. We found that from this study, EduMate meets this demand by dynamically parsing academic strengths and weaknesses to construct targeted learning paths. In contrast to existing monolithic systems that are often restricted by high subscription costs or single-subject focuses [22].

A critical factor in EduMate's success was the strategic selection of its technology stack. Python was chosen as the primary programming language due to its rich machine learning libraries, clean syntax, and superior flexibility

compared to languages like Java or C++. To power the system's natural language processing and real-time intelligent responses, the Hugging Face DistilBERT framework was integrated. DistilBERT, a lighter and faster version of the original BERT model, proved ideal for balancing high performance with efficiency, especially in an educational environment with limited computational resources [19]. Furthermore, integrating the Gemini API enabled a highly interactive conversational AI chatbot, which is capable of breaking down complex academic topics into simple, natural language explanations for students. To ensure a responsive and scalable platform, the architecture was built using React and Bootstrap for the frontend, FastAPI for asynchronous backend operations, and a hybrid database approach utilizing both Firebase (for real-time syncing) and MySQL (for structured relational data).

Despite the successful implementation, the project faced several developmental and operational constraints. Because the project was developed under a fixed semester schedule and with limited funding, the scope was restricted strictly to core functional modules, relying heavily on free or affordable open-source tools. Operationally, the system's heavy reliance on cloud-based services and external APIs (like Gemini) means it requires a highly stable internet connection. Users in low-resource environments with slow internet may experience noticeable delays when opening lesson files or waiting for AI-generated feedback. Additionally, system testing was conducted with a relatively small group of participants; while this provided valuable baseline feedback, it may not fully represent the diverse technical and educational needs of a broader global user base.

Future enhancements will focus on deepening AI personalization, improving teacher controls, and boosting student motivation. The current system's mastery logic will be refined to analyze a student's repeated mistakes and weak topics over longer periods, enabling the AI to automatically recommend highly specific practice exercises. To maintain high academic standards, future iterations will provide teachers with advanced oversight features, allowing them to preview and manually edit AI-generated quizzes before they are published to the student dashboard. Finally, to foster long-term student engagement and consistent study habits, subsequent versions of EduMate will introduce gamification elements, such as motivation badges, learning streaks, and milestone achievements.

VI. Conclusions

The EduMate project successfully conceptualized, designed, and implemented an AI-powered personalized tutoring system capable of addressing the limitations of one-size-fits-all education. By leveraging Extreme Programming and a modern Python/React technology stack, the platform provides secure, role-based environments for students, teachers, and administrators. Testing and survey results conclusively indicate that the system's dynamic quizzes, real-time AI conversational support, and automated progress tracking successfully facilitate a more engaging, tailored, and supportive learning experience. EduMate establishes a robust, accessible foundation for the future of AI-assisted, personalized education.

Acknowledgment

We would like to thank the participants who have responded to the survey and involved in giving feedback on EduMate.

Conflict of Interest

The authors declare that there is no conflict of interest.

Author(s) Contribution Statement

All authors contributed equally to this work.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Statement

This study did not require ethical approval

References

[1] R. Luckin, W. Holmes, M. Griffiths, and L. B. Forcier, *Intelligence Unleashed: An Argument for AI in Education*, London, U.K.: Pearson Education, 2016.

[2] W. Holmes, M. Bialik, and C. Fadel, Artificial Intelligence in Education: Promises and Implications for Teaching and Learning. Center for Curriculum Redesign, 2021.

[3] Y. D. Assmar, R. Contreras-Masse, M. V. H. Dozal, I. O. H. Andrade, and L. F. A. González, "Application of Artificial Intelligence in Personalized Education," in *Advances in Computational Intelligence. MICAI 2025 International Workshops* (Lecture Notes in Computer Science, vol. 16264), L. Martínez-Villaseñor *et al.*, Eds., Cham, Switzerland: Springer, 2026, pp. 110–122, doi: 10.1007/978-3-032-17930-2_10.

[4] X. Liu and B. Zhong, "Integrating generative Artificial Intelligence into student learning: A systematic review from a TPACK perspective," *Educational Research Review*, vol. 49, Art. no. 100741, 2025.

[5] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education," International Journal of Educational Technology in Higher Education, vol. 16, art. 39, 2019.

[6] B. J. Zimmerman, "Becoming a self-regulated learner: An overview," Theory Into Practice, vol. 41, no. 2, pp. 64–70, 2002.

[7] O. Henkel, H. Horne-Robinson, N. Kozhakhmetova, and A. Lee, "Effective and Scalable Math Support: Experimental Evidence on the Impact of an AI-Math Tutor in Ghana," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and*

Blue Sky, Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 373–381.

[8] G. Kwid, N. Sarty, and D. Yang, 'A Review of AI Tools: Definitions, Functions, and Applications for K-12 Education', ACRT, vol. 3, Nov. 2024, doi: 10.5772/acrt.20240048.

[9] A. Fortuna, F. Prasetya, A. D. Samala, S. Rawas, S. Criollo-C, D. Kaya, M. Raihan, W. Andriani, D. Safitri, and R. A. Nabawi, "Artificial intelligence in personalized learning: A global systematic review of current advancements and shaping future opportunities," *Social Sciences & Humanities Open*, vol. 12, p. 102114, 2025, doi: 10.1016/j.ssaho.2025.102114.

[10] Khan Academy, "Introducing Khanmigo," 2025. [Online]. Available: https://www.khanacademy.org/khan-labs

[11] Squirrel AI Learning, "Squirrel AI," 2025. [Online]. Available: https://squirrelai.com/

[12] Carnegie Learning, "MATHia: Intelligent math software," 2025. [Online]. Available: https://www.carnegielearning.com/solutions/math/mathia

[13] IU International University of Applied Sciences, "Syntea: Your digital study assistant," 2023. [Online]. Available: https://www.iu.org

[14] A. Salem, "Extreme Programming (XP) methodology," *Scrum-Master.org*, Mar. 13, 2024. [Online]. Available: https://scrum-master.org/en/extreme-programming-xp-a-beginners-guide-to-the-agile-method/. [Accessed: May 30, 2026].

[15] Python Software Foundation, "About Python," 2025. [Online]. Available: https://www.python.org/doc/essays/blurb/

[16] Google, "Gemini API documentation," n.d. [Online]. Available: https://ai.google.dev/

[17] React, "React documentation," n.d. [Online]. Available: https://react.dev/learn

[18] Bootstrap, "Bootstrap documentation," n.d. [Online]. Available: https://getbootstrap.com/docs/

[19] FastAPI, "FastAPI documentation," n.d. [Online]. Available: https://fastapi.tiangolo.com/

[20] Firebase, "Firebase documentation," Google Developers, 2025. [Online]. Available: https://firebase.google.com/docs

[21] SQLite, "SQLite documentation," n.d. [Online]. Available: https://www.sqlite.org/docs.html

[22] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett, "Cognitive Tutor: Applied research in mathematics education," Psychonomic Bulletin & Review, vol. 14, no. 2, pp. 249–255, 2016.