DOI: https://doi.org/10.36910/6775-2524-0560-2026-63-04
UDC 004.4:004.8:004.5
Povstiana Yuliia, Ph.D., Associate Professor
https://orcid.org/0000-0001-5426-4157
Samchuk Lyudmila Ph.D., Associate Professor
https://orcid.org/0000-0003-2516-045X
Lishchyna Nataliia, Ph.D., Associate Professor
https://orcid.org/0000-0002-5200-536X
Boiko Lev, Ph.D.,
https://orcid.org/0009-0001-0117-8551
Antoniuk Pavlo, Master's Student
https://orcid.org/ 0009-0004-6565-0108
Lutsk National Technical University, Lutsk, Ukraine

ARCHITECTURE AND EXPERIMENTAL EVALUATION OF A CROSS-PLATFORM MOBILE APPLICATION FOR ADAPTIVE LEARNING USING LARGE LANGUAGE MODELS

Povstiana Y., Samchuk L., Lishchyna N., Boiko L., Antoniuk P. Architecture And Experimental Evaluation Of A Cross-Platform Mobile Application For Adaptive Learning Using Large Language Models. The paper addresses the design and experimental evaluation of the architecture of a cross-platform mobile application for adaptive foreign language learning using large language models. An architectural approach based on a dedicated AI Integration Layer is proposed, enabling the separation of business logic and improving the reliability of interaction with external AI services. The system implements adaptive content generation considering individual user characteristics. Special attention is given to performance optimization through the implementation of a multi-level caching mechanism, which reduced AI service usage costs by 74 % and decreased response latency. An experimental evaluation of system performance demonstrated stable operation under a load of up to 100 concurrent users with an average response time of 280-500 ms, and identified a degradation threshold at 160-170 users. The obtained results confirm the effectiveness of the proposed approach and its applicability for developing modern mobile educational systems based on large language models.
Keywords: cross-platform mobile application, adaptive learning, large language models, artificial intelligence, software architecture, caching, performance, learning personalization.

Повстяна Ю.С., Самчук Л.М., Ліщина Н.М., Бойко Л.С., Антонюк П.О. Архітектура та експериментальне дослідження кросплатформного мобільного застосунку адаптивного навчання з використанням великих мовних моделей. У статті розглянуто питання розробки та експериментального дослідження архітектури кросплатформного мобільного застосунку адаптивного навчання іноземних мов із використанням великих мовних моделей. Запропоновано архітектурний підхід із виділенням окремого шару інтеграції штучного інтелекту (AI Integration Layer), що забезпечує ізоляцію бізнес-логіки та підвищує стабільність взаємодії із зовнішніми AI-сервісами. Реалізовано механізм адаптивної генерації навчального контенту з урахуванням індивідуальних характеристик користувача. Особливу увагу приділено оптимізації продуктивності системи за рахунок впровадження багаторівневого кешування, що дозволило знизити витрати на використання AI-сервісів на 74 % та зменшити латентність відповіді. Проведено експериментальне дослідження продуктивності, яке показало стабільну роботу системи при навантаженні до 100 одночасних користувачів із середнім часом відповіді 280-500 мс та визначило поріг деградації при навантаженні 160-170 користувачів. Отримані результати підтверджують ефективність запропонованого підходу та доцільність його застосування для створення сучасних мобільних освітніх систем із використанням великих мовних моделей.
Ключові слова: кросплатформний мобільний застосунок, адаптивне навчання, великі мовні моделі, штучний інтелект; архітектура програмних систем, кешування, продуктивність, персоналізація навчання.

Statement of the Scientific Problem. The modern development of information technologies, particularly mobile platforms and artificial intelligence, is driving a transformation in approaches to the organization of the educational process. This is especially evident in the field of foreign language learning, where traditional methods are gradually being supplemented or replaced by intelligent adaptive systems. The increasing availability of mobile devices, together with the advancement of large language models, creates the prerequisites for the formation of personalized learning environments capable of adapting educational content in real time.

At the same time, despite the considerable number of studies in the field of mobile-assisted language learning and the application of artificial intelligence in education, the issue of effective integration of large language models into mobile systems – taking into account performance, scalability, and cost-efficiency requirements – remains insufficiently explored. This determines the relevance of developing new architectural approaches to the design of intelligent educational applications.

Analysis of Research and Publications. Recent studies in the field of mobile-assisted language learning indicate a significant and growing interest in the use of mobile technologies within the educational process. In particular, the systematic review [1], which analyzed 72 scientific publications, revealed a substantial increase in the number of studies after 2019 and confirmed the effectiveness of mobile-assisted language learning in enhancing student motivation and engagement. At the same time, the authors emphasize that most existing solutions are based on pre-designed or partially adaptive content, while many studies rely on limited sample sizes and short durations, which complicates the generalization of results.

Similar conclusions are presented in [2], where the impact of mobile learning on user engagement, anxiety levels, and learning outcomes is examined. It was found that most modern mobile applications are focused on standard learning scenarios and gamification techniques without deep adaptation to individual user characteristics, which limits the effectiveness of personalized learning.

Further development in this area is associated with the application of artificial intelligence, particularly large language models (LLMs), which enable real-time generation of educational content. Contemporary research demonstrates that large language models have significant potential for personalizing the learning process, automating task generation, and adapting educational content [3]. At the same time, studies on the integration of LLMs into intelligent learning systems show an expansion of their functional capabilities, particularly in terms of generating adaptive feedback and constructing individualized learning scenarios [4]. However, as highlighted in [5], the use of large language models is associated with several technical limitations, including high computational costs, response latency, and scalability challenges, which necessitate the development of specialized architectural solutions.

Additionally, study [6] confirms the positive impact of artificial intelligence technologies on language learning effectiveness, particularly in terms of increased engagement and improved learning outcomes. Nevertheless, these works are primarily focused on pedagogical aspects and insufficiently address engineering challenges, such as optimizing interaction with AI services, ensuring system stability, and reducing computational costs.

Thus, the analysis of current research indicates that, despite significant progress in the development of mobile educational systems and the application of large language models, there remains a need for architectural approaches that ensure effective integration of AI into mobile applications while meeting performance, scalability, and economic efficiency requirements.

In accordance with the identified limitations of existing approaches, this study proposes an architectural solution for integrating large language models into a cross-platform mobile application using a dedicated AI Integration Layer and mechanisms for optimizing interaction with AI services.

The scientific novelty of the study lies in the development of an architectural approach to integrating large language models into cross-platform mobile applications through a dedicated AI Integration Layer, which ensures the isolation of business logic and improves system reliability.

In contrast to existing approaches, the proposed solution combines adaptive content generation with a multi-level caching mechanism that enables a significant reduction in computational costs and response latency, while maintaining system scalability under dynamic load conditions.

The aim of the study is to design and experimentally validate the architecture of a cross-platform mobile application for adaptive foreign language learning using large language models, aimed at ensuring the personalization of educational content, improving system performance, and optimizing the costs associated with interaction with AI services.

Presentation of the Main Material and Justification of the Research Results. Within the scope of the study, a cross-platform mobile application for adaptive foreign language learning using artificial intelligence technologies was developed. The application is based on a hybrid client–server architecture with a dedicated layer for the integration of large language models.

The paper proposes an approach to integrating large language models into mobile educational systems through the use of a dedicated AI Integration Layer, which ensures the isolation of business logic and enhances the stability of interaction with external services. In addition, the caching mechanism for generated results has been improved, making it possible to reduce the costs associated with the use of AI services. The overall system architecture is shown in Figure 1.

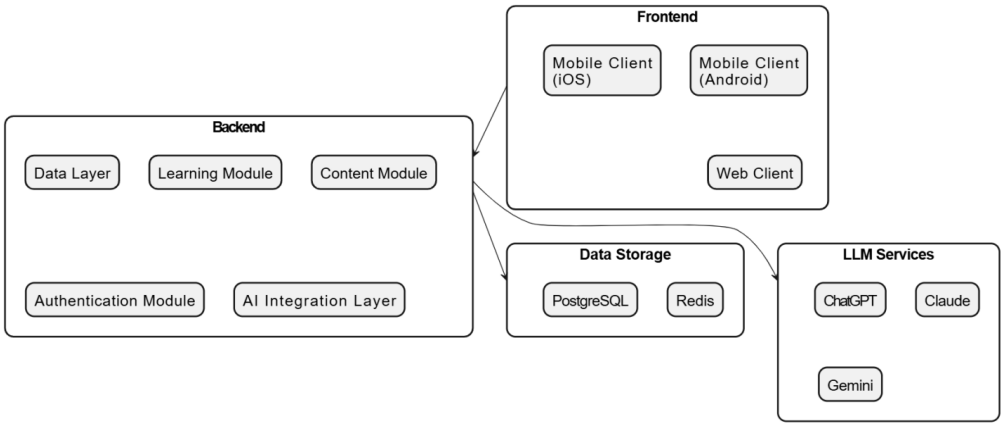

Fig. 1. Architecture of a Cross-Platform Adaptive Learning System with AI Integration

As illustrated in Figure 1, the system is built using a three-tier architecture with a dedicated artificial intelligence integration layer. This approach enables clear separation of responsibilities between system components, reduces the dependency of business logic on external AI services, and ensures independent scalability of individual layers. The introduction of the AI Integration Layer also improves system stability by localizing potential failures of external services.

Access security is implemented through authentication and authorization mechanisms based on JWT and refresh tokens. This approach ensures secure interaction between the client and the server, as well as efficient session management. The generalized authentication process is presented in Figure 2.

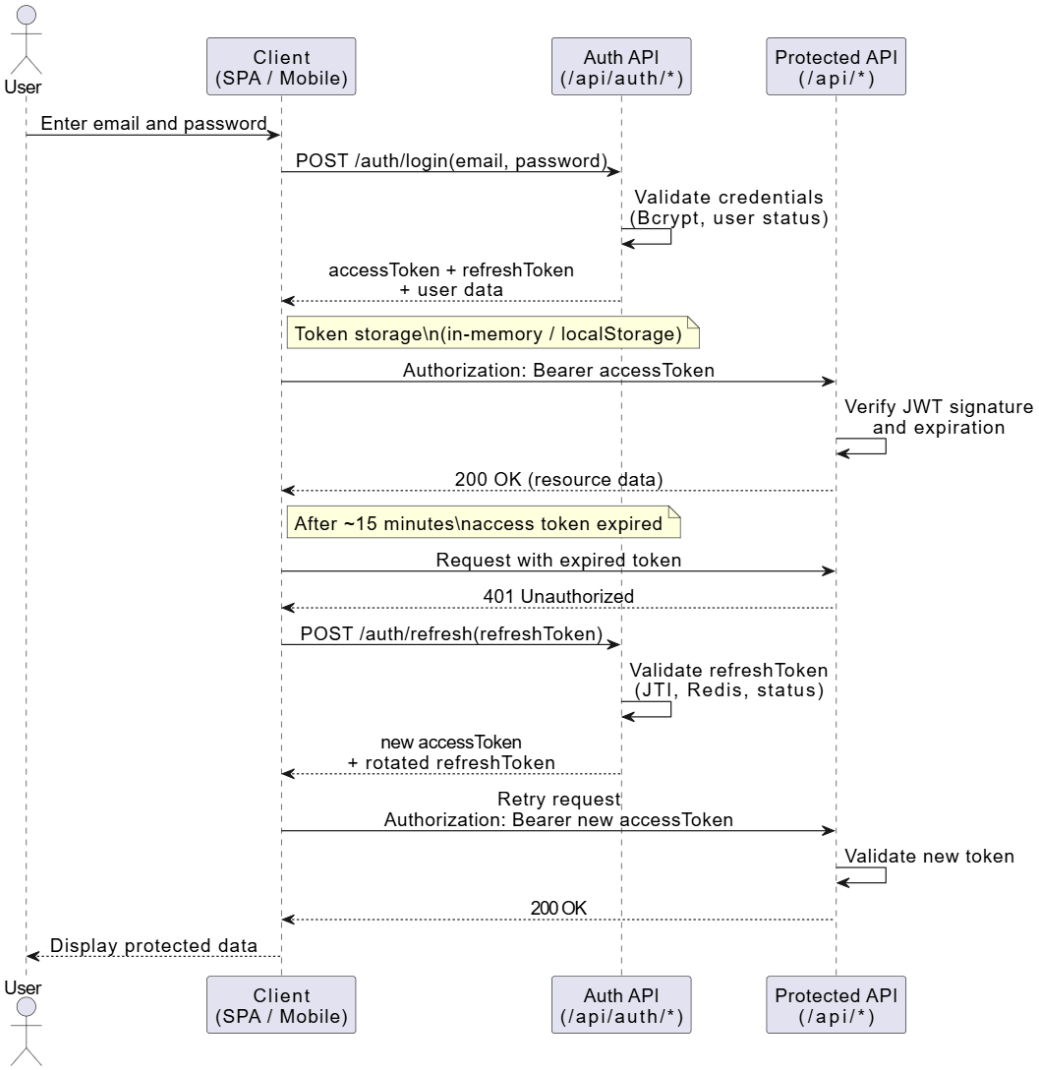

Fig. 2. Generalized Sequence of the Authentication Process Using JWT and Refresh Tokens

The client layer provides user interaction and implements the learning interface. The server layer is responsible for request processing, business logic implementation, and interaction with databases. The artificial intelligence integration layer performs the generation of educational content, analysis of user responses, and management of interaction with large language models.

The system was implemented using a modern technology stack. The client side was developed using TypeScript, Vue.js 3, and NativeScript, ensuring cross-platform compatibility and native performance of the mobile application. Pinia was used for state management. The server side was implemented in Java 17 using Spring Boot 3.2 and Spring Security, enabling efficient organization of business logic and a high level of security. PostgreSQL 16 was used as the database management system, while Redis 7.2 was applied for caching. Integration with artificial intelligence was performed via Google Vertex AI (Gemini). The system infrastructure supports containerization and scalability using Docker and Kubernetes.

A key element of the system is the educational content generation subsystem, which is implemented based on large language models. The process of forming a learning task includes generating a request considering the user profile, processing the response from the AI service, and returning the result to the client. The sequence diagram of this process is shown in Figure 3.

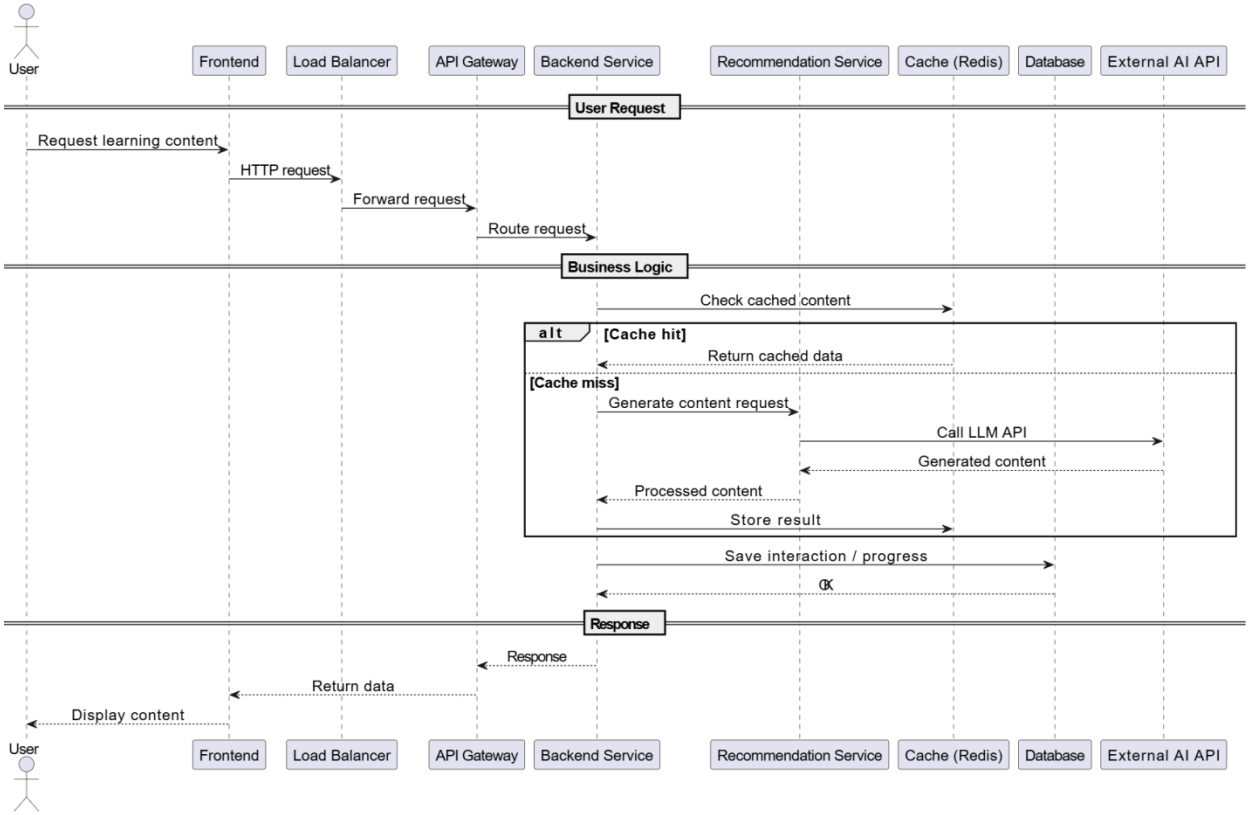

Fig. 3. Sequence Diagram of Learning Task Generation Using the AI Integration Layer

The use of the AI Integration Layer enables adaptive generation of educational content by taking into account the learning history, user knowledge level, and previous mistakes. This ensures personalization of the learning process and improves its effectiveness.

The client part of the system is built using a component-based approach, which ensures flexibility and scalability of the interface. The central element is the LessonSessionView component, which coordinates the interaction of child components within a learning session. The component hierarchy is shown in Figure 4.

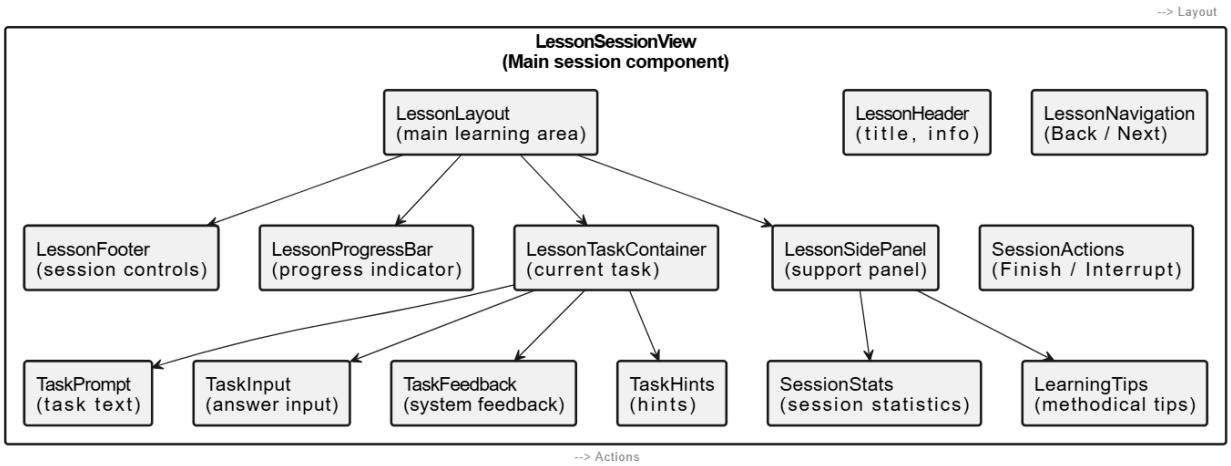

Fig. 4. Component Hierarchy of the Learning Session (LessonSessionView and Child Components)

This approach allows for component reuse, simplifies interface maintenance, and improves user experience.

To optimize system performance and reduce the costs of using AI services, a multi-level caching system has been implemented, including the use of Redis for short-term storage and a database for long-term caching of generated results. This makes it possible to reuse generated content and significantly reduce the number of requests to external APIs, which led to a reduction in AI service costs by approximately 74% and decreased system response latency.

To further evaluate the effectiveness of the applied caching mechanisms, the impact of the cache hit rate on actual AI API costs was analyzed over several optimization iterations. An increase in the proportion of successful cache hits resulted in a reduction in the number of requests to external AI services, thereby lowering the overall cost of request processing.

To formalize the impact of caching mechanisms on the economic efficiency of the system, the following mathematical model was used:

$$C = (1 - H) \cdot N \cdot C\_api \quad (1)$$

where C – is the total cost of using AI services,

H – is the cache hit rate,

N – is the number of requests,

C_api – is the average cost of a single AI API request

This relationship shows that costs are directly proportional to the share of requests not served by the cache.

To evaluate caching efficiency, the following savings indicator was introduced:

$$E = (C\_no\_cache - C\_cache) / C\_no\_cache \cdot 100\% \quad (2)$$

The obtained results confirm that an increase in the cache hit rate leads to a significant reduction in costs. The dynamics of these changes are presented in Table 1.

Table 1 – Dynamics of Cache Hit Rate and Corresponding AI Service Costs

Week	Cache Hit Rate %	Estimated Costs, USD
1	42	320
2	58	210
3	68	165
4	76	125
8	78	136 *(stable value)*

As shown in Table 1, an increase in the cache hit rate is accompanied by a substantial reduction in AI service costs. The greatest effect is observed at the initial stages of optimization, while further increases

in the cache hit rate lead to cost stabilization. This confirms the effectiveness of caching as a mechanism for cost optimization and performance improvement.

The experimental evaluation of system performance showed that it operates stably under a load of up to approximately 100 concurrent users, with an average response time of 280-500 ms. Analysis of the 95th percentile response time (p95) made it possible to determine the system performance degradation threshold.

The system response time can be formalized as follows:

$$T = (1 - H) \cdot T\_AI + H \cdot T\_cache + T\_network \quad (3)$$

where *T_AI* – is the processing time of the AI service;
T_cache – is the time required to retrieve a response from the cache;
T_network – represents network latency.

This model demonstrates that increasing the proportion of cached requests significantly reduces the average system response time.

The corresponding results are shown in Figure 5.

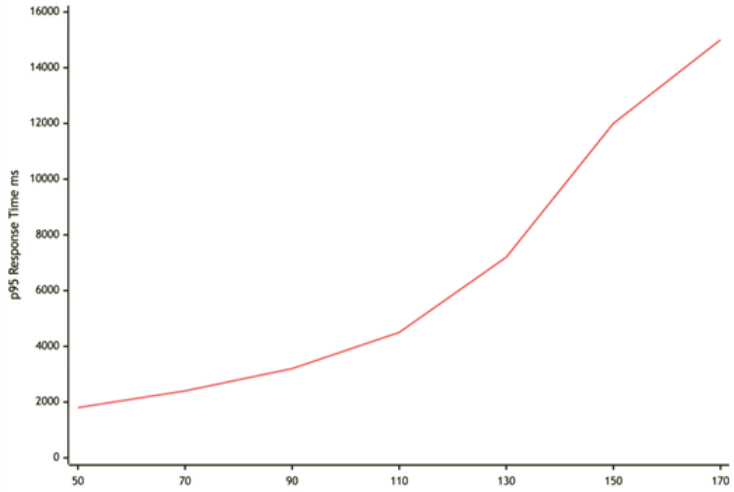

Fig. 5. Dependence of Response Time (p95) on the Number of Concurrent Users

The obtained results indicate that when the load exceeds 100 concurrent users, an exponential increase in response time is observed, while the critical failure threshold is reached at approximately 160-170 users. This justifies the need for horizontal scaling mechanisms to ensure stable system operation. The obtained response times meet the requirements for interactive mobile applications, as exceeding the 1-second threshold leads to a significant deterioration in user experience.

To generalize the dependence of response time on system load, the following model was used:

$$T\_response = T\_base / (1 - U / U\_max) \quad (4)$$

where *U* – is the number of concurrent users
U_max – is the maximum system load.

The model explains the exponential growth of response time as the system approaches its critical load level.

Conclusions and Prospects for Further Research. In this study, the architecture of a cross-platform mobile application for adaptive foreign language learning using large language models was developed and experimentally evaluated. The proposed approach involves the introduction of a dedicated AI Integration Layer, which ensures the isolation of business logic, improves the reliability of interaction with external services, and simplifies system scalability.

The implemented architecture enables adaptive generation of educational content based on individual user characteristics. The introduction of multi-level caching reduced the number of requests to AI services and decreased costs by 74 % compared to a system without caching.

Experimental results demonstrated stable system performance under a load of up to 100 concurrent users, with an average response time of 280-500 ms. It was determined that further load increase leads to exponential growth in latency, which justifies the use of horizontal scaling.

Thus, the proposed solution ensures effective integration of large language models into mobile educational systems by combining learning personalization, performance, and economic efficiency, and differs from existing approaches by incorporating a dedicated integration layer and mechanisms for optimizing interaction with AI services.

Prospects for further research include the development of advanced personalization mechanisms based on user behavior analytics, the integration of multimodal AI models, as well as the improvement of scalability approaches through adaptive load balancing and distributed processing.

References

1. Guo, P., Jeyaraj, J. J., & Razali, A. B. (2024). A systematic review of collaborative mobile-assisted language learning (C-MALL) practices. Humanities and Social Sciences Communications. https://doi.org/10.1057/s41599-024-03940-3.
2. Puri, V., et al. (2025). The effects of mobile-based language learning on learners' engagement, anxiety, and achievement. Computers & Education: Artificial Intelligence. https://www.sciencedirect.com/science/article/pii/S0001691825007139, Doi: 10.1016/j.actpsy.2025.105400.
3. Kasneci, E., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. https://doi.org/10.1016/j.lindif.2023.102274.
4. Gaeta, M., et al. (2025). Enhancing intelligent tutoring systems with large language models. Computers & Education: Artificial Intelligence. https://doi.org/10.1016/j.caeai.2025.100433.
5. Yan, L., et al. (2023). Practical and ethical challenges of large language models in education. https://doi.org/10.1111/bjet.13370.
6. Torres, G., & Kahveci, M. (2025). Artificial intelligence in language learning: A meta-analysis. Computers & Education: Artificial Intelligence. https://doi.org/10.1016/j.caeai.2025.100522.

Історія статті:
Отримано: 12.05.2026 Доопрацьовано: 16.05.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026