Shtefan Nataliia, Senior Lecturer, Department of Software Engineering, Odesa Polytechnic University, co-author: Oleksandr Kamniev, Odesa, Ukraine, https://orcid.org/0000-0001-6301-6534

INTELLIGENT TEACHER ASSISTANT FOR AN EDUCATIONAL PLATFORM

Abstract

The article considers the problem of automating interaction between students and teachers in an educational platform using artificial intelligence technologies. The relevance of the study is determined by the increasing amount of educational content, the growing workload of teachers, and the need to provide students with timely access to explanations of educational materials. Existing learning management systems and educational platforms provide tools for organizing courses, testing, and communication, but they do not always support deep personalization of an AI assistant for a specific course and the teacher's own materials.

The purpose of the article is to present the approach to developing an intelligent teacher assistant for an educational platform that supports the creation of a personalized AI teacher, processing of students' questions, and generation of reports on learning activity and performance. The proposed system is based on a client-server architecture and includes a web interface, server-side logic, a database, and an AI interaction module. The client part is implemented using HTML5, CSS3, JavaScript and React.js, while the server part is based on Node.js and Express.js. The system stores information about users, classes, educational materials, questions, answers, and student performance.

The article describes the subject area, functional capabilities of the system, architectural decisions, implementation features, and testing approach. The main

functional scenarios include registration and authorization, class management, creation of an AI teacher, uploading educational materials, student interaction with the AI assistant, assessment of students, and generation of analytical reports. Functional testing based on decision tables and test cases confirmed the correctness of the main usage scenarios. The experimental evaluation showed an improvement in the convenience of interaction, speed of obtaining answers, accuracy of AI responses, and overall efficiency of learning interaction.

Keywords: artificial intelligence, educational platform, intelligent assistant, AI teacher, web application, React.js, Node.js, Express.js, client-server architecture, automation of learning.

Problem statement

The modern educational process is characterized by the active use of digital learning platforms, electronic educational materials, online communication tools, and distance learning technologies. The transition to blended and online learning formats has significantly increased the amount of information processed by both teachers and students. At the same time, the number of repetitive questions from students is growing, especially when learning new topics, completing homework, preparing for tests, or working with large volumes of educational materials.

A teacher is often forced to answer similar questions repeatedly, provide additional explanations, monitor student progress, update educational content, and analyze the level of students' understanding of the material. These processes require a considerable amount of time and reduce the teacher's ability to focus on more complex pedagogical tasks, such as improving course content, preparing individual consultations, and developing practical assignments. The problem becomes especially relevant in higher education, where one teacher may work with several groups and a large number of students.

Existing educational platforms provide important tools for organizing learning, such as course management, file sharing, testing, and communication. However, their functionality is often limited when it comes to personalized automated assistance. A conventional learning management system can store materials and tasks, but it does not always provide students with immediate explanations based on the specific materials uploaded by the teacher. General-purpose chatbots and AI tools can answer questions, but they are not always connected to the structure of a particular course, the teacher's methodology, or the context of the educational materials.

Therefore, there is a need to develop an intelligent teacher assistant that can be integrated into an educational platform and provide automated support for students. Such a system should allow the teacher to create a personalized AI teacher based on educational materials, enable students to ask questions at any convenient time, and provide the teacher with analytical information about student activity and performance. The use of artificial intelligence in this context makes it possible to reduce the teacher's workload, increase the availability of educational materials, and improve the quality of interaction within the educational process.

Analysis of recent research and publications

The development of educational platforms is closely related to the digitalization of education and the growing use of artificial intelligence in learning environments. Modern learning management systems are designed to organize educational content, manage groups of students, provide access to tasks and tests, and support communication between participants in the educational process. Among the most common solutions are Google Classroom, Moodle, Khan Academy, and other platforms focused on online learning and course management [1–4].

Google Classroom, especially in combination with AI-based tools such as

Gemini for Education, provides instruments for organizing classes, distributing learning materials, and supporting students in the Google Workspace environment [1; 2]. Its advantages include integration with Google services, convenient course management, and support for certain AI-assisted educational tasks. However, the AI functionality in such platforms is mainly focused on auxiliary operations, such as generating content or supporting general learning activities, rather than creating a separate AI teacher for a specific course [2].

Moodle is one of the most widely used learning management systems in educational institutions. It supports course creation, testing, assessment, analytics, and integration of additional modules [3]. The flexibility of Moodle is one of its main advantages, but the platform often requires technical knowledge for configuration and maintenance. In addition, the implementation of AI-based support usually depends on external plugins or additional customization, which may complicate deployment and use [3].

Khan Academy provides access to educational content and supports interactive learning for a wide audience [4]. The platform is useful for self-study and provides structured materials for different subjects. However, it is less focused on personalized interaction between a particular teacher and students within a specific course. Its functionality is mainly designed for access to prepared educational content rather than for creating an AI assistant based on the teacher's own materials [4].

Recent trends in educational technology show that artificial intelligence can be used not only for content generation, but also for adaptive learning, automated feedback, analysis of student activity, and personalized recommendations [2; 10]. At the same time, the implementation of AI tools in education requires careful consideration of reliability, data security, transparency of generated answers, and the role of the teacher in controlling educational content [10; 11]. The most effective approach is not to replace the teacher with

artificial intelligence, but to provide a support tool that automates routine operations and helps students receive quick explanations [10].

The analysis of existing solutions shows that there is a gap between general learning management systems and universal AI chatbots [1–4; 10]. Learning management systems provide structure and educational content management, while AI chatbots provide conversational interaction. The proposed intelligent teacher assistant combines these two approaches by connecting AI interaction with classes, educational materials, student questions, and teacher analytics. This creates the basis for a more personalized and practically useful educational platform.

Purpose of the article

The purpose of the article is to present the design and implementation approach for an intelligent teacher assistant for an educational platform that improves interaction between students and teachers through automation of answers to students' questions, personalized processing of educational materials, and generation of analytical reports.

Presentation of the main material of the study

The proposed software system is an educational web application that provides interaction between three main components of the learning process: the teacher, the student, and the AI teacher. The teacher creates a class, uploads educational materials, configures the AI teacher, monitors student activity, and analyzes performance reports. The student joins the class using a unique code, studies the educational materials, asks questions to the AI teacher, receives explanations, and views homework assignments. The AI teacher acts as an automated tool that generates answers based on the information provided by the teacher.

The subject area of the system includes several key entities: users, classes, educational materials, questions, answers, grades, homework assignments, and

reports. A user may have the role of a student or a teacher. A class is a virtual learning environment that connects the teacher with a group of students. Educational materials are uploaded by the teacher and used to personalize the AI teacher. Questions and answers represent the main unit of interaction between a student and the AI assistant. Reports provide the teacher with information about student activity and learning progress.

The main functional capabilities of the system include registration and authorization, role-based access, class creation, student joining by class code, creation of an AI teacher, uploading of educational materials, student interaction with the AI teacher, viewing homework assignments, assessment of students, and report generation. These functions provide a complete cycle of educational interaction within the platform.

The registration and authorization module is necessary to identify users and provide access according to their roles. During registration, the user creates an account and selects or receives the appropriate role. During authorization, the system verifies the entered credentials and opens the corresponding interface for a student or teacher. Role separation is important because students and teachers have different functional capabilities and different access levels.

The class management module allows the teacher to create a virtual class and generate a unique code for student access. The student enters this code and joins the corresponding class. This mechanism simplifies the organization of groups and makes it possible to connect students with the correct AI teacher and educational materials. Each class may contain its own learning resources, list of students, homework assignments, and performance indicators.

One of the key components of the system is the module for creating a personalized AI teacher. The teacher fills in a questionnaire, provides course-related information, uploads learning materials, and initiates the creation of the AI assistant. The system processes the provided information and configures the

AI teacher so that it can answer students' questions in the context of the specific course. This approach differs from general AI chatbots because the assistant is connected to the educational materials and the teacher's course structure.

The student interaction module provides a chat interface where the student can ask questions about the material. The system receives the question, sends it for processing, forms an answer, and displays the result to the student. If the question is unclear or cannot be processed correctly, the system asks the student to clarify the request. This scenario makes learning support more accessible, because the student can receive explanations not only during classes or consultations, but also at any convenient time.

The homework module provides students with access to assignments, deadlines, and descriptions. The student can open the homework section, view available tasks, and select a specific assignment for more detailed information. If there are no active assignments, the system displays a corresponding message. This functionality supports better organization of independent work and makes the educational process more structured.

The assessment and reporting module is intended for teachers. The teacher can view the list of students, enter grades or comments, save assessment results, and generate reports. The reports may include information about student activity, interaction with the AI assistant, completed tasks, and learning progress. Such analytical functionality allows the teacher to identify students who need additional support and to adjust the educational process.

The architecture of the system is based on the client-server model. The client side is implemented as a web interface that provides interaction with the user. It is responsible for displaying pages, forms, tables, chats, class information, homework assignments, and reports. The client side is implemented using HTML5, CSS3, JavaScript, and React.js. React.js was chosen because it supports component-based development, dynamic user interfaces, and

convenient state management.

The server side processes requests from the client application, checks user access rights, interacts with the database, processes educational materials, and sends requests to the AI service. Node.js and Express.js are used for the server side. This choice is justified by the efficiency of JavaScript-based server development, support for REST API, and the ability to build scalable web applications. The interaction between the client and server parts is implemented through HTTP requests and REST API endpoints.

The database stores structured information required for the operation of the platform. The main database entities include users, classes, educational materials, questions, answers, grades, homework assignments, and reports. The use of a relational database makes it possible to ensure data consistency, define relationships between entities, and perform queries for analytics and reporting. MySQL is used as the database management system.

The AI interaction module is responsible for processing students' questions and generating answers. It receives the question from the server, takes into account the educational context, and returns a response that is displayed to the student. An important requirement for this module is that the answer should be based on the materials provided by the teacher. This reduces the risk of irrelevant answers and increases the usefulness of the assistant for a specific course.

From the point of view of non-functional requirements, the system must be reliable, efficient, secure, maintainable, and portable. Reliability means stable operation of the system, correct error handling, and preservation of user data. Efficiency requires fast processing of user requests and convenient access to the main functions. Security requires authentication, authorization, secure password storage, and protection of personal data. Maintainability is achieved through modular organization of the code and clear separation of client, server, database, and AI components. Portability is supported by the web-based nature of the

application and the possibility of using it on different devices.

The software implementation includes several important user flows. The first user flow is registration and authorization. The second user flow is teacher interaction with the system: class creation, AI teacher configuration, material uploading, student assessment, and report viewing. The third user flow is student interaction: joining a class, viewing assignments, asking questions, and receiving AI-generated explanations. These flows were used to organize the interface and check the completeness of functional implementation.

The testing of the system was performed using functional testing methods. The purpose of testing was to verify that the system correctly implements the main user scenarios and responds properly to both valid and invalid input data. The testing approach included decision tables and test cases. Decision tables were used to formalize combinations of input conditions and expected results, while test cases were used to check specific scenarios, such as successful registration, incorrect authorization, class joining, AI teacher creation, question processing, homework viewing, and report generation.

Special attention was paid to testing the interaction between the student and the AI teacher. This functionality is central to the proposed system because it directly affects the usefulness of the platform. The tests checked whether the system accepts a student’s question, processes it, generates an answer, and displays the result correctly. Additionally, cases with unclear or incomplete questions were considered, where the system should ask the student to clarify the request.

The experimental evaluation of the system was conducted with the participation of 15 users, including students and teachers. The purpose of the experiment was to evaluate the influence of the developed system on the convenience of working with educational materials, the speed of obtaining answers, the accuracy of AI responses, and the overall effectiveness of

educational interaction. Users evaluated the system according to several criteria using a five-point scale.

The results showed that after using the system, the convenience of the interface increased from 2.9 to 4.7 points. The speed of obtaining answers increased from 2.6 to 4.8 points. The accuracy of answers increased from 3.1 to 4.5 points. The convenience of interaction with classes increased from 3.0 to 4.6 points. The overall efficiency of learning interaction increased from 3.2 to 4.7 points. These results confirm that the use of an intelligent teacher assistant can improve the organization of educational interaction and reduce delays in obtaining explanations.

The practical value of the system is that it can be used as an additional tool for teachers and students in an educational platform. For teachers, it reduces the number of repetitive questions, simplifies class management, and provides analytical information about student activity. For students, it improves access to explanations, supports independent learning, and makes interaction with educational materials more convenient. The system does not replace the teacher but extends the teacher's capabilities by automating routine communication and analytical tasks.

At the same time, the developed system has certain limitations. The quality of AI answers depends on the quality and completeness of the educational materials uploaded by the teacher. If the materials are incomplete, poorly structured, or contain contradictions, the AI teacher may provide less accurate answers. Another limitation is the need to ensure data security, especially when working with personal information, grades, and educational records. In addition, further development should include more advanced analytics, integration with external learning management systems, and tools for verifying the correctness of generated answers.

The further development of the system may include integration with

existing educational platforms, support for additional file formats, improved semantic search in educational materials, automatic generation of tests and quizzes, advanced dashboards for teachers, and mechanisms for evaluating the quality of AI responses. It is also advisable to add functionality for tracking the most frequent student questions, identifying difficult topics, and recommending additional materials based on student activity.

Conclusions

The article presents an approach to the development of an intelligent teacher assistant for an educational platform. The relevance of the proposed solution is determined by the need to automate routine interaction between students and teachers, reduce the teacher's workload, and improve access to educational materials. The analysis of existing platforms showed that many of them provide tools for course management, testing, and communication, but do not fully support the creation of a personalized AI teacher based on the teacher's own materials.

The proposed system combines class management, educational material processing, AI-based question answering, student assessment, and reporting. The client-server architecture provides a clear separation of responsibilities between the web interface, server logic, database, and AI interaction module. The use of React.js, Node.js, Express.js, and MySQL makes it possible to build a scalable and maintainable web application.

Functional testing confirmed the correctness of the main user scenarios, including registration, authorization, class joining, AI teacher creation, question processing, homework viewing, assessment, and report generation. The experimental evaluation demonstrated an improvement in the convenience of using the interface, speed of obtaining answers, accuracy of AI responses, convenience of working with classes, and overall efficiency of educational interaction.

The developed intelligent teacher assistant can be used to optimize the educational process, support students in independent learning, and provide teachers with analytical information. Further research and development should focus on improving the accuracy of AI responses, expanding integration capabilities, enhancing security, and developing advanced analytics tools for monitoring student progress.

References

15. Google Classroom. Official website. Available at: https://classroom.google.com/

16. Google for Education. Gemini for Education. Available at: https://edu.google.com/

17. Moodle. Official website. Available at: https://moodle.org/

18. Khan Academy. Official website. Available at: https://www.khanacademy.org/

19. React. Official documentation. Available at: https://react.dev/

20. Node.js. Official documentation. Available at: https://nodejs.org/

21. Express.js. Official documentation. Available at: https://expressjs.com/

22. MySQL. Official documentation. Available at: https://dev.mysql.com/doc/

23. MDN Web Docs. HTML, CSS, JavaScript documentation. Available at: https://developer.mozilla.org/

24. OpenAI. API Documentation. Available at: https://platform.openai.com/docs/

25. ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE).

26. Sommerville I. Software Engineering. 10th ed. Pearson, 2016.