Student Portal with AI Assistant

Martin Petkeš, Radoslav Vargic
Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19, Bratislava, Slovakia
xpetkes@stuba.sk
radoslav.vargic@stuba.sk

Abstract - **This article explores the design and development of a modular web application – a student portal enhanced with an artificial intelligence (AI) assistant based on the Retrieval-Augmented Generation (RAG) approach. The system aims to provide students with a centralized platform for accessing study-related information and receiving contextual assistance through AI. The solution includes role management, ensuring that features are accessible according to role permissions (e.g., student, professor, administrator). The modular architecture ensures flexibility, scalability and the potential for future enhancements.**

Keywords **- AI; LLM; RAG; Assistant; Portal; Education; LangChain; Vector database**

I. INTRODUCTION

In recent years, the integration of artificial intelligence (AI) into educational systems has significantly transformed the way students access information and manage their academic responsibilities. The rapid evolution of Large Language Models (LLMs), such as OpenAI's GPT models, has made it possible to create intelligent systems capable of understanding natural language, generating human-like responses and assisting users in a conversational manner.

One particularly promising method that enhances the capabilities of LLMs is Retrieval-Augmented Generation (RAG). RAG combines an LLM with an information retrieval component, allowing the system to access external data sources and generate more accurate, context-aware answers. This approach is especially useful in domains where up-to-date or user-specific information is needed – such as education or customer support.

The goal of this project is to design and implement a modern web-based portal that enables students to perform essential tasks, such as viewing schedules or upcoming events, communicating with professors or other students. The main focus though, is on receiving AI-generated assistance. The AI assistant is designed to respond to student queries.

This article is structured as follows:

Section II provides an overview of related systems and technologies currently in use,

Section III describes the design choices, architecture and implementation of the proposed system,

Section IV evaluates the usability and functionality of the system in a test environment,

Section V discusses future improvements and possible extensions of the platform.

II. STATE OF THE ART

LLMs have been employed to enhance personalized learning experiences and provide intelligent tutoring systems. Their ability to process and generate contextually relevant responses allows for the creation of AI-driven assistants that can support students in their academic journey. LLMs, with natural language technology at their core, align seamlessly with the education industry's development and adapt to the vast changes in intelligent education. These models have the potential to support and enhance various aspects of the learning experience, making education more accessible, engaging, and effective [1].

Taking the development of LLMs in China as an example, the Spark Desk by iFLYTEK [4], the ERNIE Bot by Baidu [5], and the "MathGPT" [6] by TAL have accumulated data from years of experience in the education industry [3]. Fig.1 shows some examples of utilizing LLMs in the education sector.

Company	Product	Application	Website
Baidu	ERNIE Bot	Literary creation, Mathematical, Chinese understanding, Multimodal generation	https://yiyan.baidu.com/
Alibaba	Tongyi Qianwen	Covering language, hearing, multimodal and other fields	https://qianwen.aliyun.com/
Tal Education Group	MathGPT	Specialize in the field of mathematics	https://www.mathgpt.com/
Iflytek Co.,Ltd	Spark Desk	Text generation, Multimodal interaction, trivia	https://xinghuo.xfyun.cn/?ch=blzc_iYNVDy
NYSE: DAO	Zi Yue	Virtual speaking coach, Dictionary pen, "AI Box" ……	https://aicenter.youdao.com/#/home
SenseTime	INTERN2.5	Image description, Visual question answering, Visual reasoning	https://intern-ai.org.cn/home
Tsinghua University	ChatGLM	Intelligent question and answer dialog, Efficient access to information to solve problem	https://chatglm.cn/

Figure 1. Examples of LLMs in education [1]

A significant advancement in this domain is the development of Retrieval-Augmented Generation systems. RAG offers several advantages augmenting traditional methods of text generation, especially when dealing with factual information or data-driven responses. LLMs are limited to their pre-trained data. This leads to outdated and potentially inaccurate responses. RAG overcomes this by providing up-to-date information to LLMs. RAGs usually retrieve facts via search, and modern search engines now leverage vector databases to efficiently retrieve relevant documents (Fig. 2). Vector databases store documents as embeddings in a high-dimensional space, allowing for fast and accurate retrieval based on semantic similarity.

In the study of improving assesment of tutoring practices using Retrieval-Augmented generation, the RAG-based prompt not only showcased more accurate performance in evaluating tutoring practices and lower financial costs compared to other prompts but also laid the groundwork for broader applications in tutor skill assessment [7].

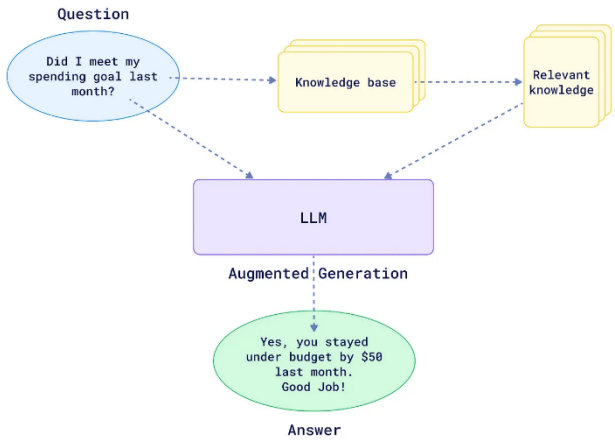

Figure 2. RAG wokflow simplified. [8]

There are several widely adopted frameworks and technologies that facilitate the development of RAG systems, particularly for chatbot applications in both educational and enterprise contexts. These frameworks offer robust pipelines for integrating LLMs with retrieval mechanisms, enhancing both the accuracy and contextual relevance of AI-generated responses. Langchain is one of the most popular open-source frameworks for building applications powered by LLMs and RAG. It supports integration with various language model providers and vector databases. Haystack, RAGFlow, STORM and the list goes on [9].

These frameworks play a vital role in the practical implementation of RAG-based systems. By leveraging vector databases, semantic search, and real-time document retrieval, these tools enable the development of assistants that can provide students with highly accurate and relevant support throughout their academic experience.

III. Design and Implementation

A. Benefit of the solution

As mentioned in the State of the Art section, large language models have already shown significant promise in personalized education, content recommendation, and intelligent tutoring. I believe their true potential in this field is realized when paired with RAG.

A key benefit of this solution lies in its adaptability and scalability. Students receive personal assistance, professors can use it for communication or answering FAQs, and administrators may leverage it for automating repetitive support tasks. The use of AI within the portal is not intended to replace human interaction but to streamline and enrich the student experience by offering 24/7 support, quicker access to information, and the ability to query personal or institutional documents in natural language.

B. Conceptual design of the solution

The design of the student portal with an integrated AI assistant is based on the principles of modularity, scalability and user-role differentiation. At the conceptual level, the system is intended to serve as a centralized digital environment that provides students, professors, and administrators with relevant academic information and support based on their specific roles and permissions.

The architecture is divided into these logical components:

Frontend Interface. The user interface is built using a modern web development framework that ensures responsiveness, speed, and interactivity. It allows users to navigate the system, view academic resources, and communicate with the AI assistant through a dynamic, chat-based experience.

Cloud Services and Authentication. The frontend is connected to a cloud-based backend-as-a-service platform that handles user authentication, role-based access control, real-time data synchronization, and file storage. This enables secure user sessions, document uploads, and retrieval of personalized data.

Backend API. The backend is implemented as a lightweight and efficient web service, exposing RESTful APIs for frontend communication. It acts as the intermediary between the user interface, data storage, and AI components, managing request processing and business logic.

Data Layer. The system employs a hybrid data approach. Structured academic data such as user profiles, messages and chat histories are stored in a scalable NoSQL database, while unstructured documents are securely stored and linked to the retrieval system.

Vector Storage. Uploaded documents and other relevant texts are converted into semantic representations and stored in a vector database. These embeddings are used for fast and accurate retrieval based on similarity to user queries, forming the foundation of the AI assistant's contextual understanding.

This conceptual design ensures that each user interacts with the system in a way that is relevant to their academic role while benefiting from AI-driven support that enhances productivity, engagement, and access to personalized information.

C. Practical implementation of the portal

The implementation of the student portal followed a modern, modular approach, integrating cloud-based services, an interactive web frontend, and an AI-powered backend.

The frontend of the application was developed using Svelte - a reactive JavaScript framework optimized for performance and user interactivity. It is connected to a Firebase platform that handles user authentication, real-time data synchronization and file storage. The interface includes several components that are dynamically loaded after the user signs in with email and password. The entry point of the application is the authentication page (Fig. 3), which provides users with the ability to register, log in, and recover forgotten credentials via their email. The design focuses on simplicity and clarity, ensuring a smooth user experience for first-time and returning users.

Figure 3. Login and Registration Interface.

Successful authentication redirects users to a dashboard, where they can access features relevant to their role within the system. This main dashboard (Fig. 4) serves as the central navigation hub for the entire system. This area is designed to provide access to the core functionalities and applications available within the system, depending on the user's role.

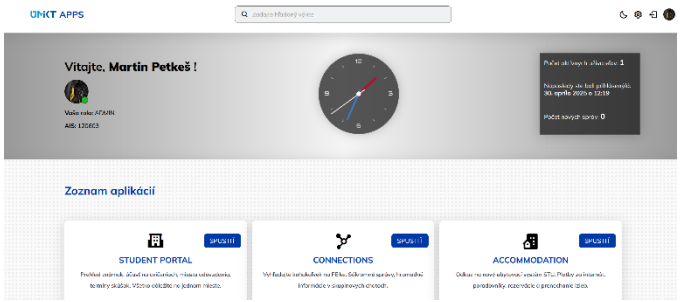

Figure 4. Dashboard of the system.

The routing between various application modules is handled dynamically using a reactive routing system, which responds to changes in the application state. Access control is enforced on the frontend based on the user's session state, ensuring that unauthorized users are not granted access to protected routes. State management is facilitated using reactive stores and local variables, which track session status and user-specific preferences in real time.

Navigation bar fixed on the top provides access to system settings and a toggle button for dark mode. In addition to basic profile information, users can update their profile picture, view the number of active users currently online, switch their status and check the timestamp of their last login.

A simple note-taking feature is also available within the dashboard, enabling users to create quick reminders or personal to-do items directly from the home interface. These notes are stored per user and can be updated or removed at any time.

At the center lies the applications overview - a visually structured list of available applications. One of the key modules accessible from this list is the student portal, which includes an integrated AI assistant that supports natural language interaction and personalized document-based help. The dashboard is responsive and modular, allowing for seamless integration of new features and applications in the future, while maintaining a clean and user-friendly layout.

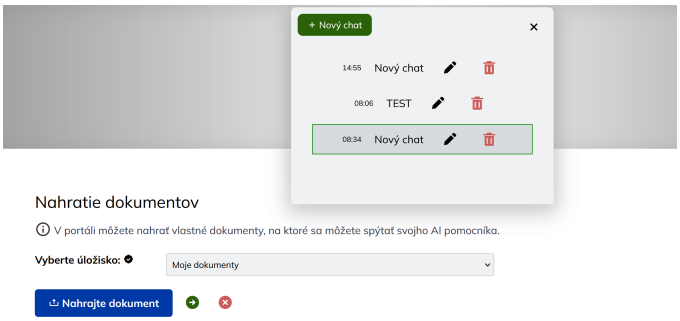

Figure 5. Document upload section of the Student Portal.

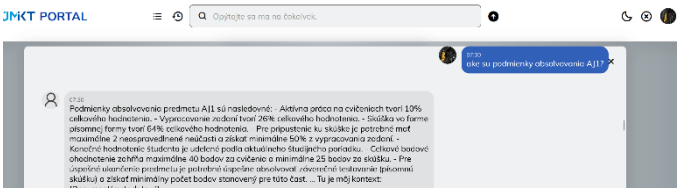

Figure 6. AI assistant chat window.

The Student Portal accessible from user dashboard is designed as an interactive workspace where users can upload documents (Fig. 5), interact with an AI assistant, manage their conversations, and collaborate with others through document sharing. The central feature of the portal is the AI chat interface (Fig. 6), which enables natural language interaction between the user and the assistant. The assistant can answer questions based on both general knowledge and user-provided documents. In addition to user uploads, administrators can upload global documents such as official study guides or school policies, which are then accessible to all users for AI-assisted querying.

The backend was implemented using a lightweight, high-performance Python framework – FastAPI. It handles requests from the frontend and processes user data. A key component of the AI assistant is the RAG pipeline, which is implemented using the LangChain framework. LangChain facilitates seamless orchestration between language models, vector databases, and external tools, enabling complex workflows such as document-based question answering.

To support semantic document search, the system utilizes ChromaDB, a high-performance open-source vector database. Uploaded documents are first converted into text chunks, which are then embedded into high-dimensional vector space. For natural language understanding and generation, the system integrates OpenAI's LLMs. These models are responsible for interpreting user input and generating relevant responses.

IV. Evaluation

From a user experience perspective, the application is intuitive and accessible. It has a clean user interface, combined with interactive chat functionality and is designed to be used by multiple roles.

A unique collaborative feature is the ability to share uploaded documents with other students. When a document is shared, the recipient gains the ability to ask the AI assistant questions based on the shared material. This promotes collaborative study, peer assistance, and shared access to relevant academic content. A teacher uploads course guidelines and allows students to ask the assistant questions about deadlines, assessment criteria, or reading materials. Alternatively, a student preparing for an exam uploads their course notes and asks clarifying questions.

While the system shows strong potential, certain limitations must be acknowledged. For instance, the quality of AI responses depends on the clarity and completeness of the uploaded documents and relies on third-party LLMs and authentication system. Also, the retrieval process needs to be optimized to provide correct embeddings for language models.

V. Future Work

Future development of the portal will focus on improving the AI assistant's contextual understanding, expanding its capabilities such as multilingual support and enhancing personalization features. This includes refining the retrieval process, optimizing prompt engineering, and better handling of complex, multi-turn conversations. Mobile version of the portal is also considered for development to increase accessibility. Another direction for future work is to expand the modular structure of the system, enabling the integration of additional applications within the platform. This would allow institutions or administrators to customize the portal further by adding tools tailored to specific academic workflows.

VI. Summary and Conclusion

This article presented the design and implementation of a modular student portal enhanced with an AI assistant based on retrieval-augmented generation.

The evaluation section emphasized the practical value of this application in academic settings. The portal improves access to information, supports collaboration, and enables students to receive instant, relevant answers to their study-related queries. By offering a personalized, interactive, and secure environment, it enhances the overall learning experience.

In conclusion, the presented system demonstrates a meaningful step toward smarter, more accessible education through the integration of artificial intelligence. With planned improvements which were described in *Section V,* the platform is well-positioned for future development and broader adoption.

Acknowledgement

This paper was supported by NEXT (ERASMUS-EDU-2023-CBHE-STRAND-2), CYB-FUT (ERASMUS+), and EULiST (ERASMUS).

References

[1] Hanyu Xu, Wensheng Gan, Zhenlian Qi, Jiayang Wu, Phillip S. Yu, 2024. Large Language Models for Education: A Survey https://arxiv.org/pdf/2405.13001

[2] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, Qingsong Wen, 2024. Large Language Models for Education: A Survey and Outlook https://arxiv.org/pdf/2403.18105

[3] Zou, L., Zhang, S., Cai, H., Ma, D., Cheng, S., Wang, S., Shi, D., Cheng, Z., Yin, D., 2021. Pre-Trained Language Model Based Ranking in Baidu Search, in: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ACM. pp. 4014–4022. https://arxiv.org/pdf/2105.11108

[4] iFlytek Large Language model, available online https://xinghuo.xfyun.cn/, accessed on 29.4.2025

[5] Baidu chatbot, available online https://yiyan.baidu.com/, accessed on 29.4.2025

[6] Tal Education Group, MathGPT chatbot, available online https://math-gpt.org/ , accessed on 29.4.2025

[7] Zifei Han, Jionghao Lin, Ashish Gurung, Danielle R. Thomas, Eason Chen, Conrad Borchers, Shivang Gupta, Kenneth R. Koedinger, 2024. Improving Assesment of Tutoring Practices using Retrieval-Augmented Generation https://arxiv.org/pdf/2402.14594

[8] https://cloud.google.com/use-cases/retrieval-augmented-generation

[9] Petrus, S. (2024). Top 10 RAG Frameworks & GitHub Repos to Check Out in 2024. https://sebastian-petrus.medium.com/top-10-rag-frameworks-github-repos-2024-12b2a81f4a49