UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI
UNDERGRADUATED STUDENT

REPORT

USTH SCIENTIFIC RESEARCH FOR STUDENTS 2025 COMPETITION

by

Nguyen Tuan Anh - BA11-007
Information and Communication Technology

Thesis Title:

A middleware framework to support scientific generation using an AI approach

Supervisor:

MSc. Le Nhu Chu Hiep

Hanoi, 2025

Comments from the supervisor on the scientific contributions of the report

I, MSc.Le Nhu Chu Hiep, certify that the UIH report of Mr.Nguyen Tuan Anh is qualified to be presented in the UIH Contest 2025.

Hanoi, July 15th, 2025

Supervisor's signature

Abstract

This thesis presents the development of a mobile application named AI PELaX, which integrates an AI model with a LaTeX rendering middleware to assist users—especially students and researchers—in solving scientific problems and visualizing mathematical content directly on their phones. The app is built using Flutter for cross-platform mobile development and communicates with a Flask-based backend that handles AI prompt processing and LaTeX code rendering.

The core idea of the project is to allow users to send a natural-language question related to scientific or mathematical content. This question is then processed by an AI model (such as OpenR1) to generate a response that may contain LaTeX expressions or even TikZ diagrams. These LaTeX components are extracted by the middleware, compiled into images using TeXLive, and sent back to the mobile app for immediate display within the chat interface.

Compared to traditional AI systems like ChatGPT or Claude, which only return TikZ Code as plain text, this system adds a unique layer by automatically converting TikZ code into images, improving readability and learning efficiency. The system was successfully implemented with key features such as prompt input, LaTeX and TikZ code extraction, server-side rendering, and real-time display in the app.

Several challenges were faced during the development process, including rendering delays and data loss on app restart, which are discussed in the report. Future work includes adding offline history, supporting more AI models, and improving performance.

This project demonstrates a practical and scalable way to integrate AI, LaTeX rendering, and mobile interfaces, contributing to the growing need for smarter educational tools in the digital era.

Keywords: AI-generated content; LaTeX rendering; TikZ visualization; mobile application; Flutter; Flask backend; prompt engineering; educational technology; scientific problem solving

---

Chapter 1: Introduction

1.1 Global Context

In today's world, the need to find and combine scientific information from different online sources is growing rapidly, especially in fields like education, research, and technology development. As scientific knowledge becomes more accessible online, students, researchers, and developers often rely on online tools to search, understand, and visualize this information quickly and effectively. However, current tools are not always designed with ease of use in mind, especially for users who lack technical backgrounds or advanced skills in formatting scientific content.

Artificial Intelligence (AI), especially Large Language Models (LLMs), is changing the way people interact with knowledge. AI-based tools like ChatGPT, Claude, or Gemini are now capable of generating structured scientific content in the form of text, mathematical formulas, diagrams, and charts[1, 2, 3]. These tools are increasingly used in universities, research labs, and even high schools to support learning, writing, and exploration. The ability to produce complex academic content using natural language queries offers great promise, but it also introduces new challenges in how this content is presented and used.

1.2 Literature Review

Traditional search engines such as Google Search or academic databases like Google Scholar help users access large collections of scientific documents. However, these tools often lack features for summarizing, integrating, or visually presenting scientific content in an interactive and understandable way [4]. For instance, while they return relevant documents, users still have to manually read, extract, and interpret information.

Recent studies have explored the potential of LLMs in scientific research assistance. These models can return well-structured responses, including code, equations, and even LaTeX-based graphics. However, one major limitation remains: when the models return TikZ (a LaTeX-based tool for creating diagrams), it appears as plain text [5]. Users must copy this code into a LaTeX editor like Overleaf or a local compiler to visualize the content. This process requires technical setup, and is not friendly for beginners.

Moreover, the effectiveness of LLMs depends heavily on the quality of prompts given by users. Many users—especially students or casual learners—struggle to write prompts that generate clear and accurate scientific answers. This limits the accessibility of AI tools in educational and research settings.

1.3 Main Questions and Objectives

The above challenges raise an important question: How can we make AI-generated scientific content—especially TikZ diagrams—more accessible and easier to use for everyone, including non-experts?

To address this, our project proposes the development of a middleware system. This system will automatically detect TikZ code in the output of AI tools and convert it into visual images without any manual steps. Users will no longer need to compile LaTeX themselves. Instead, they will see the diagrams directly, making it easier to understand and use the information.

The main objectives of the project include:

- Building a lightweight middleware tool that detects and renders TikZ code automatically.
- Ensuring compatibility with outputs from popular LLMs such as ChatGPT, DeepSeek or Claude.
- Making the tool usable on both desktop and mobile platforms.
- Reducing barriers for students and researchers unfamiliar with LaTeX.
- Improving the overall learning and research experience by making scientific diagrams more accessible and intuitive.

Chapter 2: Objectives

The main goal of this internship is to develop a mobile application that helps users interact with scientific content generated by AI in a visual and user-friendly way. The strategy is to build a middleware system that automatically converts LaTeX TikZ code into images, integrates with multiple AI models via API, and offers an intuitive interface —especially for non-technical users.

This system is titled **AI PELaX** (Artificial Intelligence for Processing and Exploring LaTeX), which aims to bridge the gap between AI-generated scientific content and real-time visual interaction, particularly focusing on LaTeX TikZ diagrams and mathematical expressions.

Chapter 3: Materials and Methods

The mobile application was built using the Flutter framework, targeting Android devices. Flutter was chosen for its cross-platform capabilities and fast UI development[6]. The app provides users with a simple and guided interface to compose prompts, submit questions to AI models, and view the returned scientific answers—including formulas and diagrams.

For AI integration, the app communicates with third-party LLMs such as ChatGPT (OpenAI), Claude (Anthropic), and DeepSeek (OpenR1) via RESTful APIs. The user's prompt is sent from the app to the selected model. Once the AI returns an answer, the app scans the response to detect any LaTeX TikZ code using regular expression patterns and keyword heuristics.

When TikZ code is found, it is sent to a dedicated rendering server developed using Flask Python and a LaTeX engine (TeX Live). The server compiles the TikZ code into PDF format using pdflatex, then converts it to a PNG image using the ImageMagick library. The resulting image is returned to the app, where it is displayed inline within the response. This process is fully automated and requires no manual steps from the user.

This architecture below allows users—especially students and researchers with limited LaTeX/TikZ experience—to visualize AI-generated scientific diagrams instantly on mobile devices, enhancing the usability of LLMs in education and research contexts.

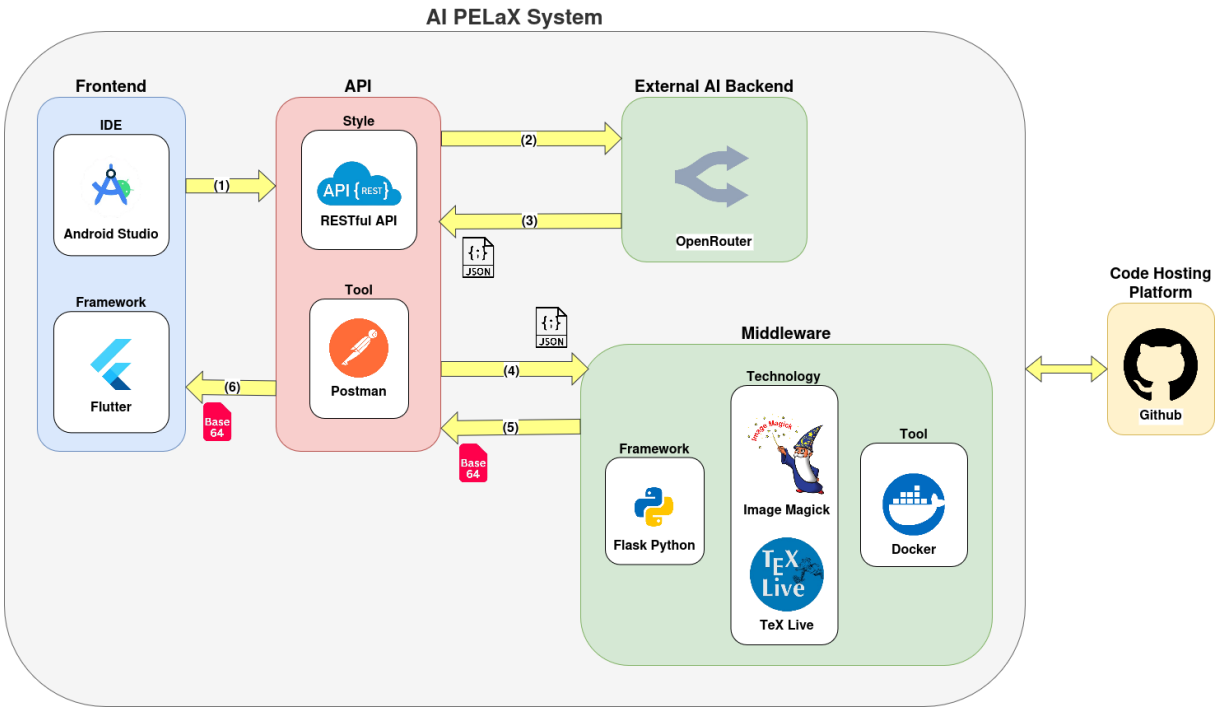

Figure 3.1 AI PELaX System Implementation

Chapter 4: Results and Discussions

4.1 Results

The project successfully built a mobile app that connects with AI models (like DeepSeek) to generate and display scientific content. Users can input questions in natural language, receive AI responses with LaTeX TikZ code, and view the generated diagrams directly in the app. The system includes three parts:

- A Flutter-based frontend with a chat interface.
- AI model integration via API to process user prompts.
- A Flask middleware that detects LaTeX, compiles it and returns images in Base64.

4.2 Discussions

This section reviews the system's outcomes, compares it with similar tools, and reflects on development challenges.

4.2.1 Comparison with other systems

To evaluate the system's performance, it is helpful to compare it with existing AI tools that support scientific or LaTeX content, such as ChatGPT, Claude, and Gemini.

Table 4.1 Comparison between AI PELaX and other AI-based systems

Criteria	AI PELaX	ChatGPT	Claude	Gemini
Platform Type	Mobile App	Web-based, Mobile App	Web-based, Mobile App	Web-based, Mobile App
Prompt Support	General, scientific prompts	General, scientific prompts	General, scientific prompts	General, scientific prompts
LaTeX Output	**Yes**	**Yes**	**Yes**	**Yes**
TikZ Support	**Full rendering**	Text only	Text only	Text only
Automatic LaTeX Image Rendering	**Yes**	**Yes**	No	**Yes**
Suitable for Education	Highly suitable	Partial	Partial	Partial

4.2.2 Challenges and Limitations

Despite the system working well, some issues remain:

- Some AI-generated LaTeX code is invalid, causing rendering errors.
- Image rendering may lag on low-end devices.
- The app does not save user history after closing.
- LaTeX compilation introduces delays in the response.

These limitations suggest areas for future improvement, such as error correction, performance optimization, and data persistence.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

AI PELaX addresses a key limitation of large language models by providing an automatic solution to render LaTeX TikZ code into images. Through a middleware system integrated with a Flutter app, users can easily view AI-generated scientific diagrams without technical requirements.

The project balances technical functionality with user-friendly design and demonstrates strong potential for use in education and research. It represents a meaningful step toward combining AI and scientific visualization.

5.2 Future Work

Although AI PELaX has met its initial goals, several directions can enhance its usability and impact. These include expanding to web or desktop platforms, adding multilingual support (e.g., Vietnamese), and implementing a prompt-response history feature for better user experience. Future updates could also focus on interactive TikZ diagram editing, enabling users to modify visual elements directly. Lastly, introducing caching mechanisms would optimize performance by avoiding repeated rendering. These improvements aim to make AI PELaX more scalable, user-friendly, and suitable for broader educational and scientific use.

References

[1] R. OpenAI, "Gpt-4 technical report. arxiv 2303.08774," *View in Article*, vol. 2, no. 5, p. 1, 2023.

[2] Anthropic, "Introducing claude," 2023. [Online]. Available: https://www.anthropic.com/index/introducing-claude

[3] G. DeepMind, "Gemini: Our most capable and general ai model yet," 2023. [Online]. Available: https://deepmind.google/technologies/gemini/

[4] F. R. Jensenius, M. Htun, D. J. Samuels, D. A. Singer, A. Lawrence, and M. Chwe, "The benefits and pitfalls of google scholar," *PS: Political Science & Politics*, vol. 51, no. 4, pp. 820–824, 2018.

[5] J. Belouadi, S. P. Ponzetto, and S. Eger, "Detikzify: Synthesizing graphics programs for scientific figures and sketches with tikz, 2024b," *URL https://arxiv.org/abs/2405.15306*, 2024.

[6] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, "Application development using flutter," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, no. 8, pp. 1262–1266, 2020.