



Graduated Thesis

A middleware framework to support scientific generation using an AI approach

Presenter:

BA11-007 Nguyễn Tuấn Anh

Supervisor:

MSc. Lê Như Chu Hiệp

July, 2025



Table of Content

01. INTRODUCTION

- Problem Statement
- Proposal System
- Objectives and Scope

02. REQUIREMENT ANALYSIS

- Requirement
- System Functionalities

03. SYSTEM DESIGN

- Overview Pipeline
- System Architecture
- System Implementation
- Backend Processing

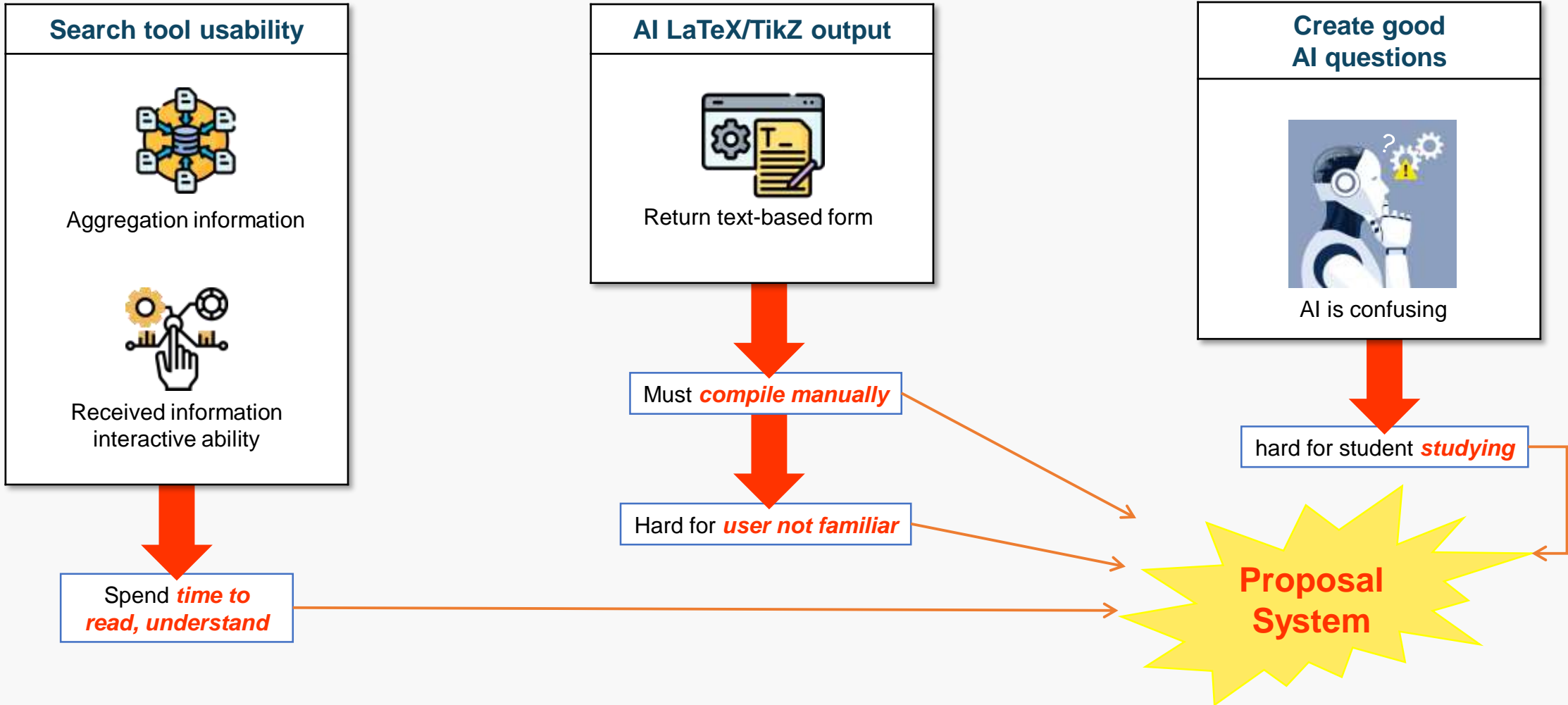
04. RESULTS & DISCUSSIONS

05. CONCLUSIONS & FUTURE WORKS

01. INTRODUCTION

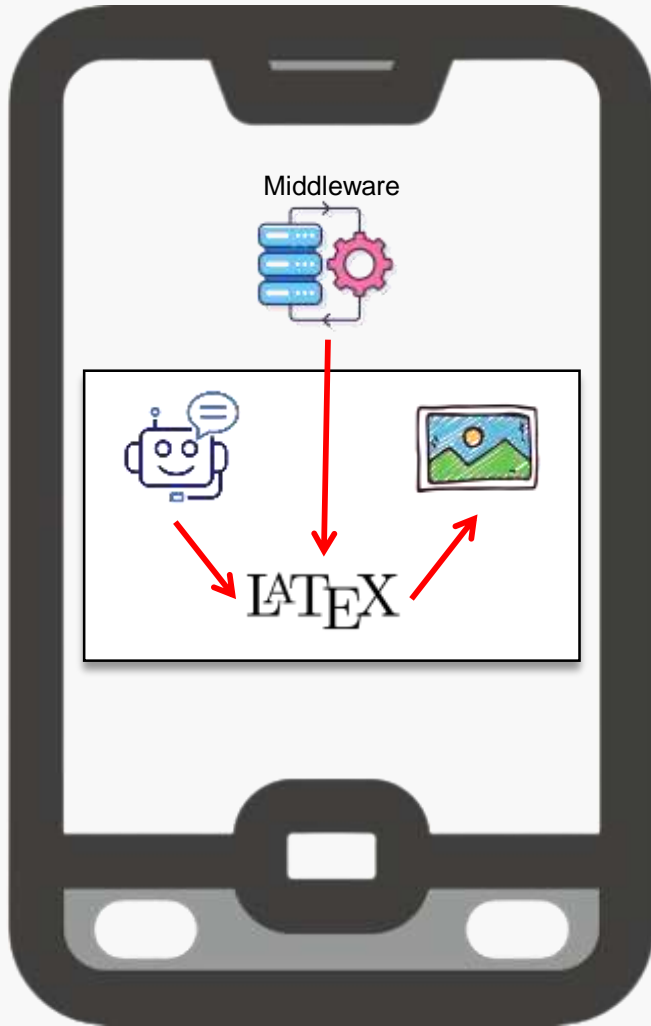
■ Problem Statement:

*What problems exist when users use AI to **learn**, **search** and **visualize** scientific content?*



01. INTRODUCTION

■ **Proposal System:** **AI PELaX** = Artificial Intelligence for Processing and Exploring LaTeX



What it can do?

- ✓ Allow users send questions to AI
- ✓ Show AI response with diagrams/formulas directly.

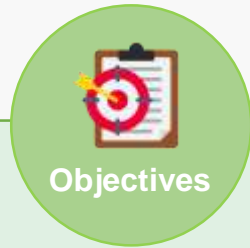
What its benefits?

- ✓ Supports learning, writing scientific documents...
faster & easier

01. INTRODUCTION

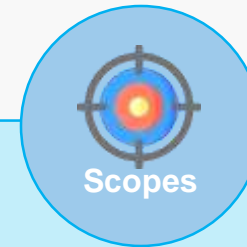
■ Objectives and Scope:

Create a mobile application & middleware with scientific content generated by AI



Objectives

- Help users create better questions
- Support different AI models
- Make it easier for students to use AI in learning



Scopes

- Mobile devices (Android) using Flutter framework
- LaTeX/TikZ diagrams/maths content generated by AI
- UI design for ease of use (especially TikZ/LaTeX)

02. REQUIREMENT ANALYSIS

■ Requirement:

01

System Requirement

Core Function:

- Flutter-based app displays results directly on Android devices.
- Automatically identifying & extracting LaTeX/TikZ code
- Displaying results directly inside application
- Rendering LaTeX/TikZ code into images through middleware Flask backend

02

Non-Functional Requirement

- **Performance:** System respond & render images quickly.
- **Scalability:** System work with different AI models via APIs.
- **User-friendliness:** UI is simple & easy use (people no LaTeX background).
- **Offline support:** App allows access saved content even when no internet connection.
- **Privacy and security:** User data are safe & not shared without permission.

02. REQUIREMENT ANALYSIS

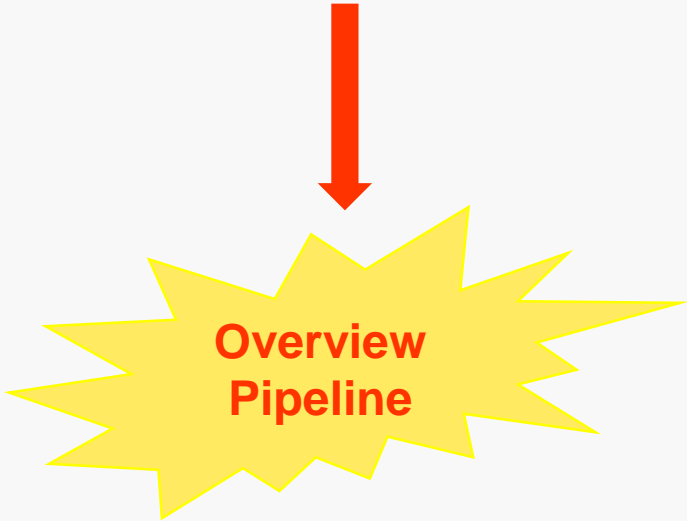
■ System Functionalities:

Table: Overview of System Functionalities

No	Feature	Description	Input	Output
1	Send Question	System allows the user to input a question and send it to the AI system	Raw user input (text prompt)	–
2	Return AI Response	AI Model receives the question and generates an appropriate LaTeX or TikZ code-based answer	User question	AI-generated LaTeX or TikZ text
3	Render LaTeX Code as Image	Middleware converts LaTeX and TikZ content into an image and returns it in Base64 format	Content returned from AI contains LaTeX and TikZ code	Base64 rendered image

What do we **get** from the table?

- 3 main features
- **Send Question** ---> Flutter UI
 - **Return AI Response** ---> AI Model
 - **Render LaTeX Code as Image** ---> Flask Middleware



03. SYSTEM DESIGN

■ Overview Pipeline:

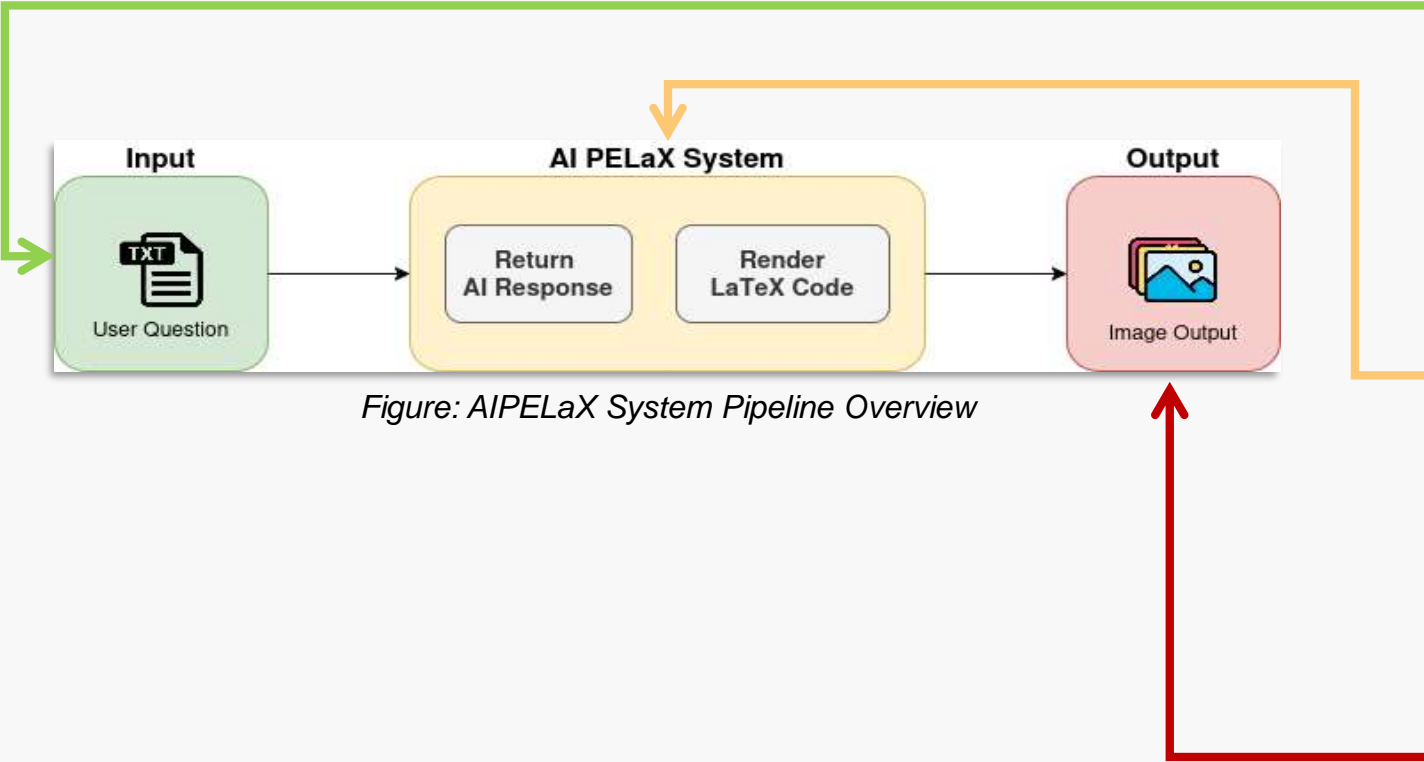


Figure: AIPELaX System Pipeline Overview

Table: Overview of System Functionalities

Stage	Component	Role	Actor	Input Format	Output Format
Input	User Question	The user submits a text-based question, possibly requesting a diagram, formula, or image.	User	Text	Text
AI Processing	Return AI Response	The system sends the question to an AI model and receives a response that may include LaTeX or TikZ code.	AI Model	Text	Text (with LaTeX/TikZ)
Middle-ware	Render LaTeX Code	The middleware extracts LaTeX/TikZ code from the AI response and compiles it into an image.	Middle-ware	LaTeX/TikZ Code	PNG / Base64 Image
Output	Image Output	The frontend decodes the image and displays it to the user as part of the final answer.	Flutter App (UI)	Base64 Image (in JSON)	Displayed Image

03. SYSTEM DESIGN

■ System Architecture:

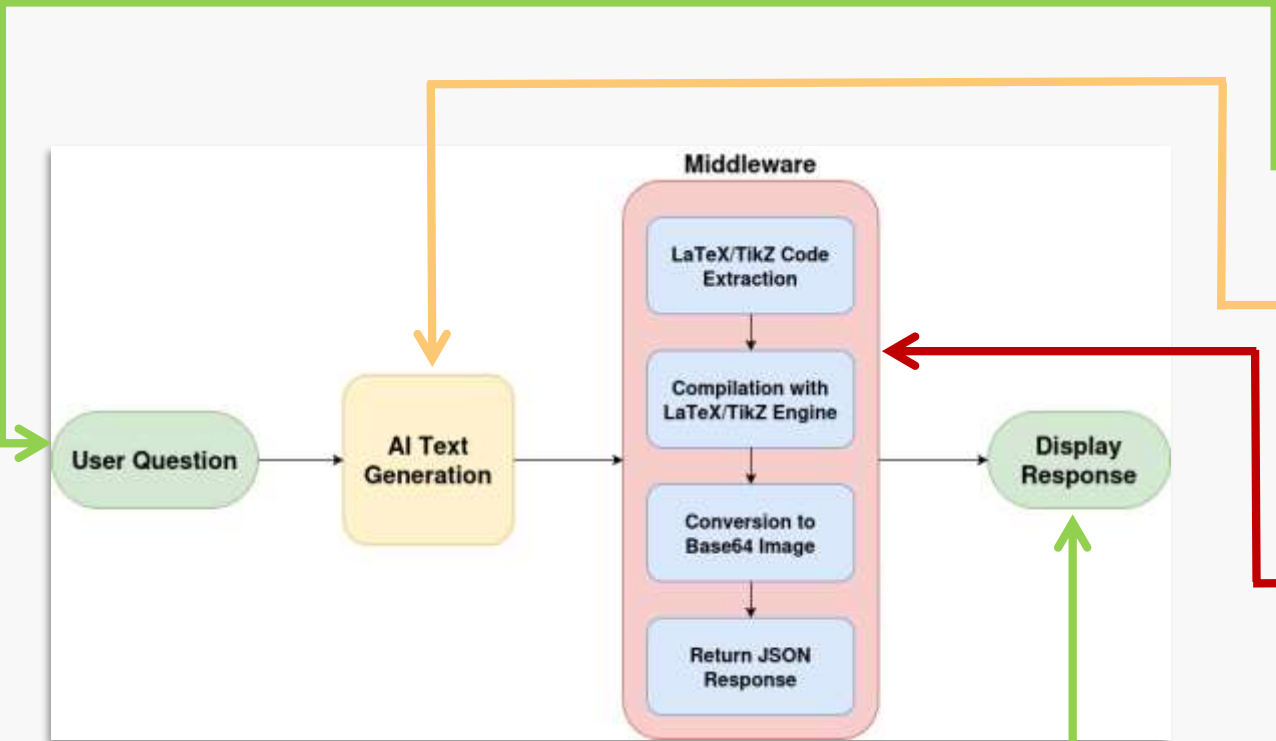


Figure: AIPELaX System Architecture

Table: AI PElax Pipeline: Step-by-Step Description

Steps	Descriptions
User Question	The user asks a question. The question may include a request to create a diagram, formula, or image using LaTeX/TikZ code.
AI Text Generation	The AI reads the question and generates an answer. If needed, the answer includes LaTeX or TikZ code to create the requested image.
LaTeX/TikZ Code Extraction	The middleware finds and extracts the LaTeX/TikZ code from the AI's response.
Compilation with LaTeX/TikZ Engine	The middleware compiles the LaTeX/TikZ code using a LaTeX engine to generate an image (such as PNG or PDF).
Conversion to Base64 Image	The generated image is converted to a Base64 string. This makes it easier to send the image as part of the response.
Return JSON Response	The middleware creates a JSON response that includes the Base64 image. This response is sent back to the user interface.
Display Response	The user interface reads the JSON response, decodes the image from Base64, and shows it to the user as part of the answer.

03. SYSTEM DESIGN

■ System Implementation:

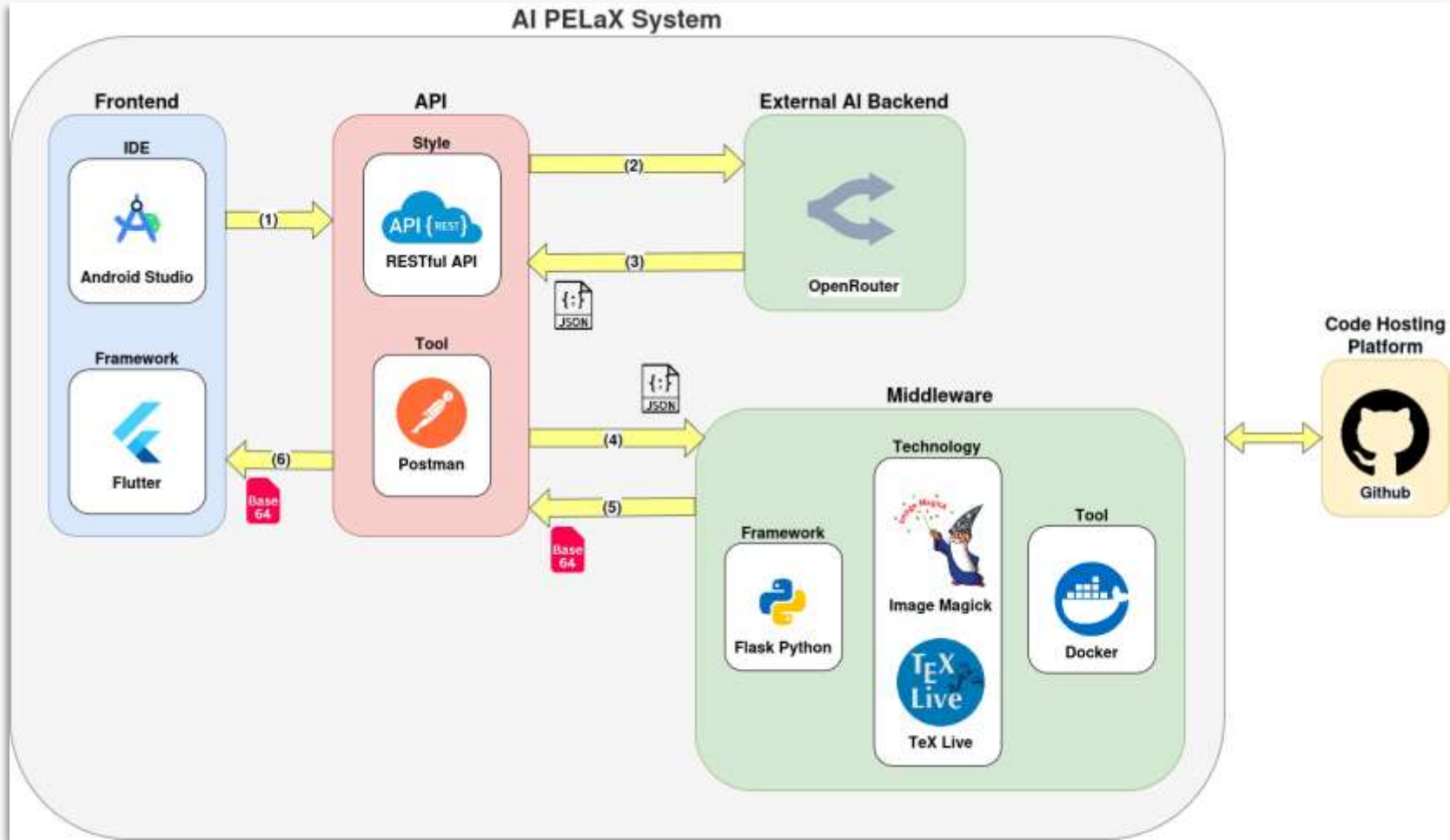
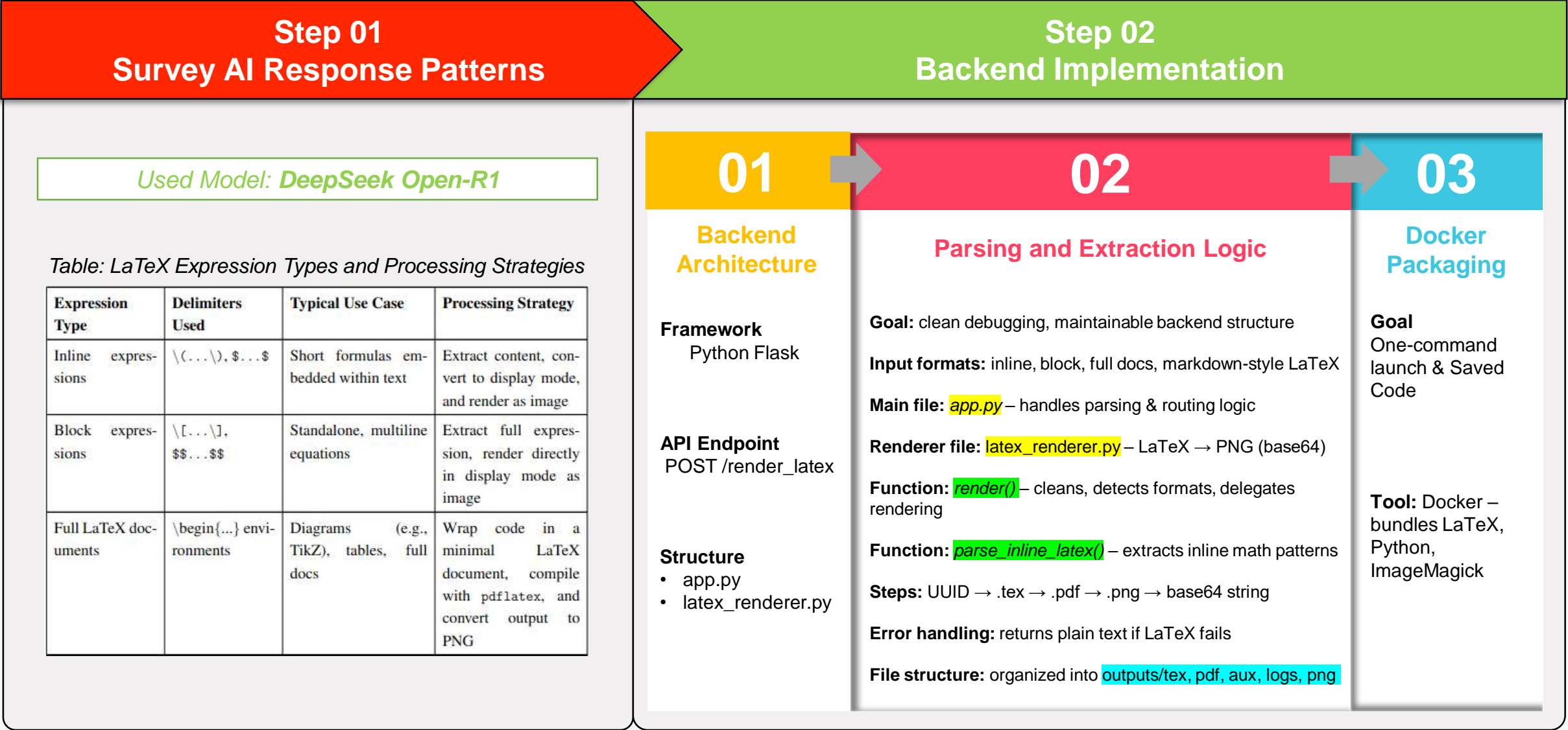


Figure: AIPELaX System Implementation

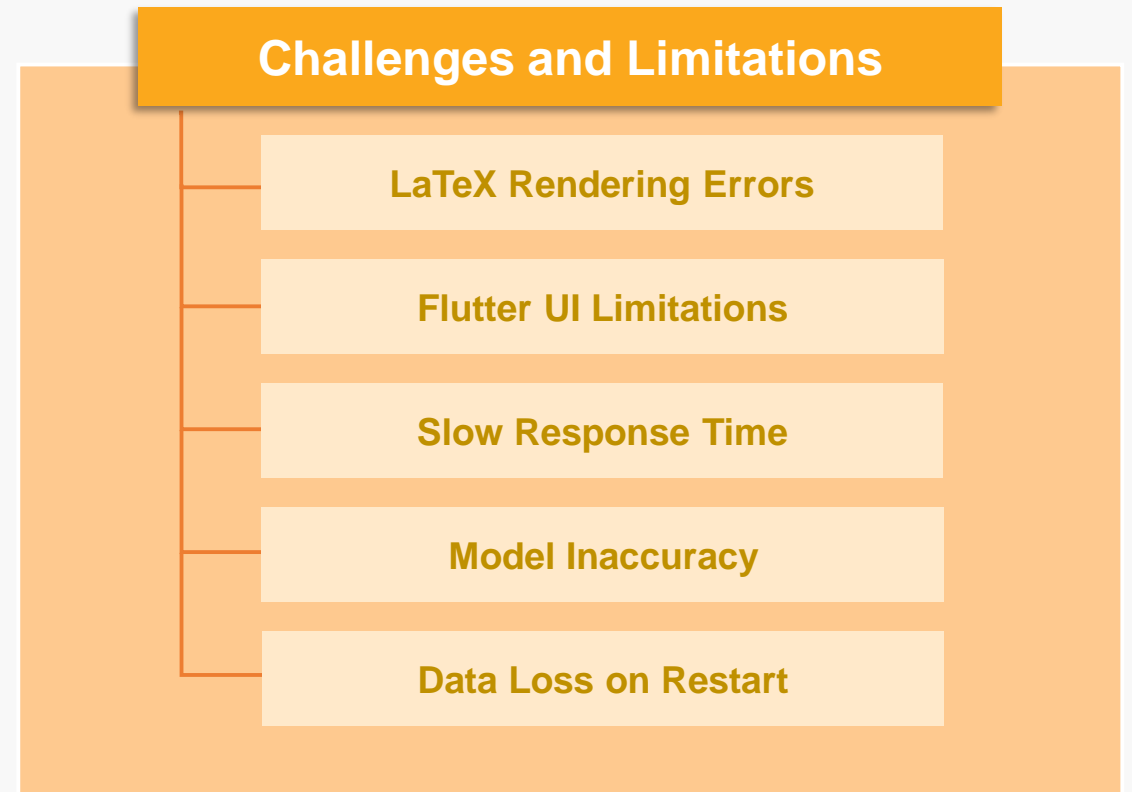
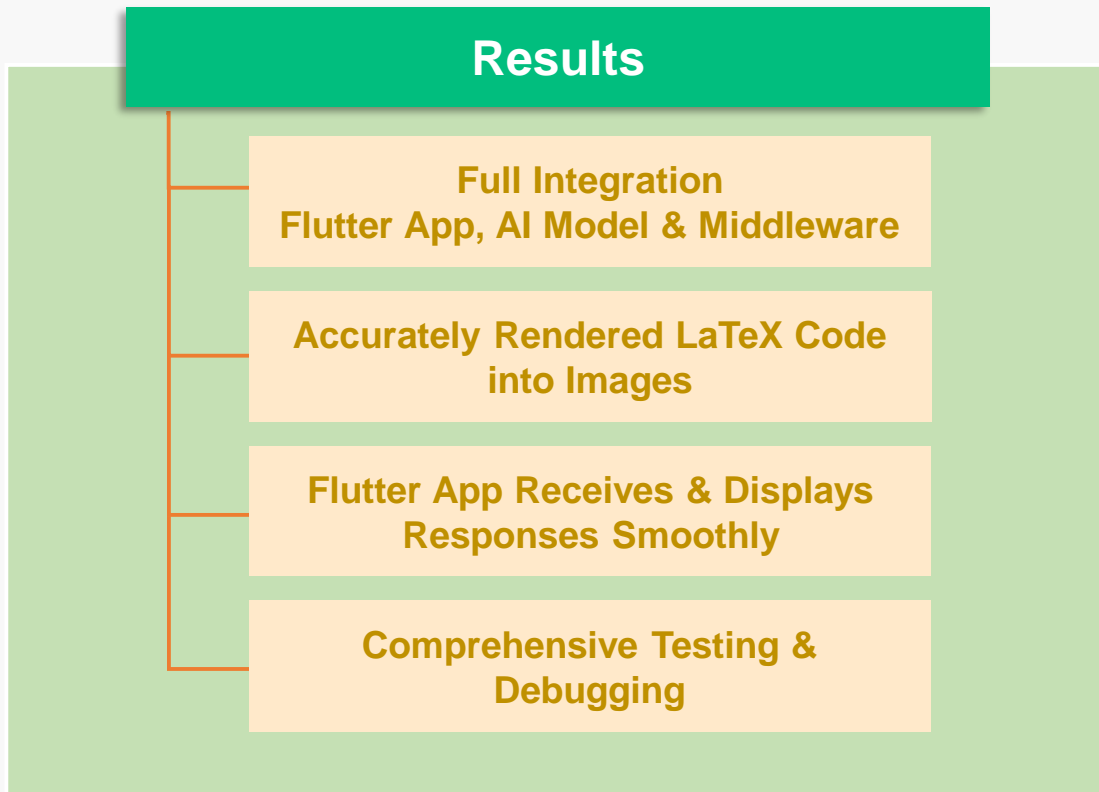
03. SYSTEM DESIGN

Backend Processing:



04. RESULTS & DISCUSSIONS

■ Results:



04.RESULTS & DISCUSSIONS

■ Discussions:

Table: Comparision with other 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

What can our system *do*?

What criteria our system *better*?

05. CONCLUSIONS & FUTURE WORKS

■ Conclusions:



1

Practical Problem

*Based-text format AI generates scientific content (LaTeX, TikZ)
→ Difficult viewing without technical expertise.*

2

Proposal System

*Develop middleware system between AI & App
→ Automatically extract & visualize TikZ content.*

3

Advantages

- *User-friendly interface*
- *Fast & secure processing*
- *Can run directly without 3rd platforms*

4

Practical Significance

- *Education, Learning, Modern Scientific Researching*
- *Serves as foundation for future deeper integration AI & scientific visualization*

05. CONCLUSIONS & FUTURE WORKS

■ Future Works:

01

Cross-
platform &
multilingual
support



02

User
questions &
response
history



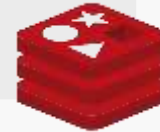
03

Interactive
LaTeX/TikZ
editing



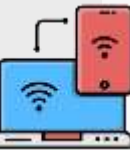
04

Smart caching
mechanism



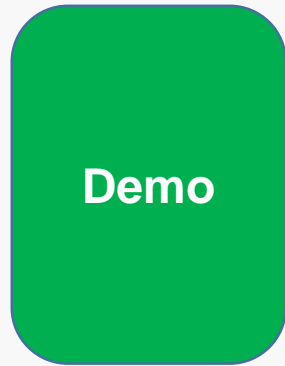
05

Toward
smarter
scientific tools



Demo

Figure: Docker Backend Deploy Running



```
anhnt02@anhnt02: ~/Desktop
anhnt02@anhnt02:~/Desktop$ docker start my-flask-api
my-flask-api
anhnt02@anhnt02:~/Desktop$ docker logs -f my-flask-api
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 123-487-038
192.168.92.3 - - [20/May/2025 08:50:48] "POST /render_latex HTTP/1.1" 200 -
192.168.92.3 - - [20/May/2025 08:52:41] "POST /render_latex HTTP/1.1" 200 -
192.168.92.3 - - [20/May/2025 08:53:29] "POST /render_latex HTTP/1.1" 200 -
192.168.92.3 - - [20/May/2025 08:55:01] "POST /render_latex HTTP/1.1" 200 -
192.168.92.3 - - [20/May/2025 08:55:53] "POST /render_latex HTTP/1.1" 200 -
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
```



Thank you for listening!



Do you have any questions?