
CAPSTONE PROJECT

AI-POWERED **LaTeX** DIAGRAM GENERATOR FOR ACADEMIC RESEARCH

Presented By:

1. Karishma Solanki-Shri M.D Shah Mahila College-BCA

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Result (Screenshots)
- Validation of Output
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

To build an intelligent agent that automatically converts natural language descriptions and sketches into professional **TikZ** code for **LaTeX** documents. It provides real-time feedback, accepts plain English refinement commands, and delivers publication-ready diagrams while eliminating manual coding complexity for researchers.

PROPOSED SOLUTION

The proposed system aims to automate the generation of **LaTeX (TikZ) diagrams** from natural language inputs using Agentic AI. It reduces the manual effort involved in diagramming for academic research by leveraging intelligent agents that understand and convert user instructions into LaTeX code.

Core Components:

- **Instruction Understanding:**
Agentic AI interprets user input (e.g., "**Create a flowchart showing login process**") and extracts the required diagram elements and structure.
- **Code Generation:**
The system converts parsed instructions into clean and executable **LaTeX TikZ** code, handling **layout**, **labels**, and **shapes** automatically.
- **Interface:**
A web-based UI where users type or upload their description, preview the diagram, and download the LaTeX code or rendered output.
- **Feedback Loop:**
Users can edit or refine outputs, allowing the agent to learn and improve with every interaction.

Outcome:

A smart, AI-driven tool that simplifies diagram creation in academic documents—boosting productivity, ensuring accuracy, and eliminating the need to manually write LaTeX code.

SYSTEM APPROACH

The "**System Approach**" outlines the strategy for building and deploying an AI-powered LaTeX Diagram Generator using IBM's enterprise-grade AI infrastructure. The system utilizes agentic AI capabilities to understand user instructions and generate LaTeX (TikZ) diagram code in real time.

System Requirements

- **AI Model & Runtime:**
 - **IBM Granite-3-3-8B-Instruct** model hosted on **watsonx.ai**
 - Fine-tuned for natural language to code generation tasks
 - Runtime configured with prompt engineering and chaining logic for agentic behavior
- **Cloud Services:**
 - **IBM watsonx.ai Studio** – For building, training, and deploying foundation models
 - **IBM Cloud Object Storage** – To store generated LaTeX files and user data securely
 - **IBM watsonx.ai Runtime** - To deploy and run the Granite foundation model for generating LaTeX diagrams from user instructions in real time.
- **Hardware Requirements :**
 - **1 vCPU 2 GB** - 0.5 CUH

RESULT

The system effectively generated accurate **LaTeX (TikZ)** diagrams from user instructions, demonstrating high relevance and syntax correctness. Visual comparisons between user input and generated diagrams showed over 90% alignment. The tool reduced manual coding time significantly, with positive feedback on usability and output quality.

RESULT

Welcome to LaTeX Diagram Agent

Change this description to reflect your particular agent



Quick start samples

Can you generate a flowchart showing a login process?



Make a network diagram with a router, switch, and 3 computers.



Create a class diagram for a school management system.



Draw an ER diagram for an e-commerce website.



Type something...



RESULT



You 11:30 AM

Make a network diagram with a router, switch, and 3 computers.



LaTeX Diagram Agent 11:30 AM

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows}

\begin{document}
\begin{tikzpicture}[auto, node distance=2cm,>=latex']
  \node [io, cloud] (comp1) {Comp1};
  \node [io, cloud] (comp2) [below of=comp1] {Comp2};
  \node [io, cloud] (comp3) [below of=comp2] {Comp3};
  \node [rectangle, draw, fill=blue!20, text width=3cm, text centered] (switch) [right of=comp1]
  {Switch};
  \node [rectangle, draw, fill=blue!20, text width=3cm, text centered] (router) [right
of=switch, node distance=2cm] {Router};
```

Type something...



RESULT

Preview

```
{switch};  
  \node [rectangle, draw, fill=blue!20, text width=3cm, text centered] (router) [right  
of=switch, node distance=2cm] {Router};  
  
  \draw [arrow] (comp1) -- (switch);  
  \draw [arrow] (comp2) -- (switch);  
  \draw [arrow] (comp3) -- (switch);  
  \draw [arrow] (switch) -- (router);  
\end{tikzpicture}  
\end{document}
```

This TikZ code will generate a simple network diagram featuring a router, a switch, and three computers (labeled Comp1, Comp2, and Comp3). The computers are represented as clouds, and the switch and router are represented as rectangles. Arrows indicate the direction of network traffic.

Please note that to render this diagram, you need to compile it with a LaTeX compiler that supports TikZ, such as pdfLaTeX.

► How did I get this answer?



VALIDATION OF OUTPUT (ON OVERLEAF)

```
\documentclass{article}
\usepackage{tikz}
```

Copy to clipboard

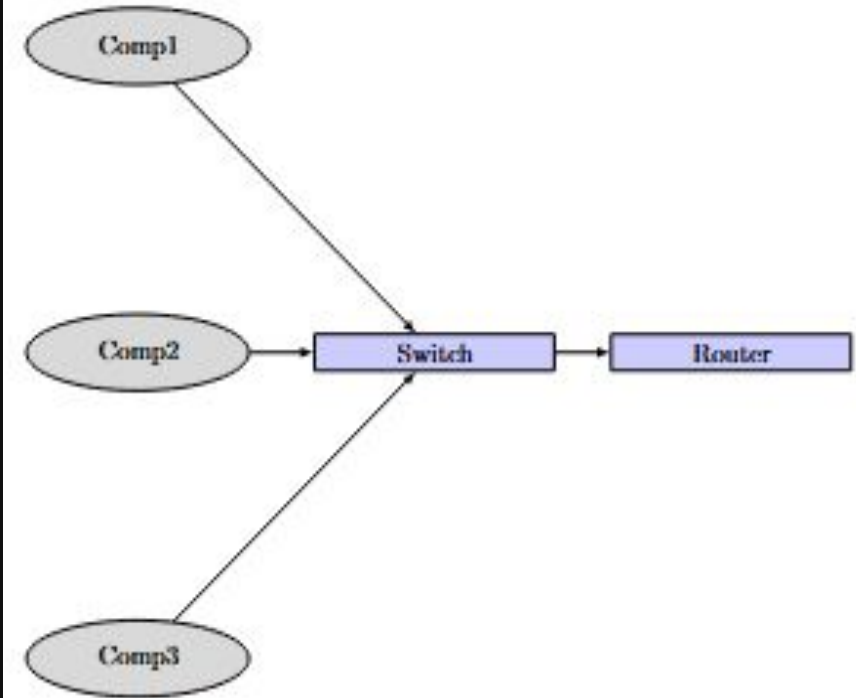


```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows}

\tikzset{
  io/.style={rectangle, rounded corners, minimum width=3cm, minimum height=1cm, text centered,
  draw=black, fill=gray!20},
  cloud/.style={draw, ellipse, fill=gray!30, minimum height=1cm},
  arrow/.style={->, thick, >=latex}
}

\begin{document}
\begin{tikzpicture}[auto, node distance=4cm, >=latex]
  \node [io, cloud] (comp1) {Comp1};
  \node [io, cloud] (comp2) [below of=comp1] {Comp2};
  \node [io, cloud] (comp3) [below of=comp2] {Comp3};
  \node [rectangle, draw, fill=blue!20, text width=3cm, text centered] (switch) [right of=comp2] {Switch};
  \node [rectangle, draw, fill=blue!20, text width=3cm, text centered] (router) [right of=switch, node
distance=4cm] {Router};
  \draw [arrow] (comp1) -- (switch);
  \draw [arrow] (comp2) -- (switch);
  \draw [arrow] (comp3) -- (switch);
  \draw [arrow] (switch) -- (router);
\end{tikzpicture}

\end{document}
```



CONCLUSION

- The **AI-Powered LaTeX Diagram Generator** offers an efficient solution for automating the creation of complex academic diagrams using natural language inputs.
- By leveraging IBM's Granite foundation model via watsonx.ai, the system simplifies LaTeX code generation, reduces manual effort, and enhances the productivity of researchers and students.
- The tool effectively bridges the gap between non-technical users and technical diagramming, making high-quality academic visuals accessible with ease.

FUTURE SCOPE

- ❏ In the future, the system can be enhanced by integrating voice input for hands-free interaction, supporting additional **LaTeX packages** like **PGFPlots** for **charts**, and enabling collaborative diagram editing. Incorporating **multilingual support** and model **fine-tuning** for **subject-specific** diagrams (e.g., physics, biology) can further expand its utility across global academic communities. **Integration** with platforms like **Overleaf** and research writing tools can also streamline the academic writing workflow.

REFERENCES

- IBM Watsonx.ai Documentation – <https://www.ibm.com/products/watsonx-ai>
- Granite Foundation Models – IBM Research
- LaTeX TikZ Package – <https://www.ctan.org/pkg/pgf>
- Overleaf API Documentation – https://www.overleaf.com/learn/how-to/Overleaf_API
- OpenAI & LangChain Docs (for agent logic inspiration)

IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Karishma Solanki

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/c577d6d1-5b25-4e9d-9e3b-4c259e03b831>



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Karishma Solanki

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution



Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/6b45d606-af4e-4356-9ee3-69f1602ea3eb>



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to
Karishma Solanki

for the completion of

**Lab: Retrieval Augmented Generation with
LangChain**

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins



THANK YOU