

# CAPSTONE PROJECT

IBM SkillsBuild Internship 2025

## TikZ Agent

Natural Language to TikZ code - powered by AI

**Presented By-**

**Student name : Shraavani Salunkhe**

**College Name : A. P. Shah Institute of Technology Mumbai University**

**Department : Computer Science and Engineering(Artificial Intelligence and Machine Learning)**

# Outline

- **Problem Statement**
- **IBM cloud services and technology used**
- **Wow factors**
- **End users**
- **Results**
- **Conclusion**
- **Future scope**
- **IBM Certifications**
- **Git-hub Link**

# Problem Statement

Researchers often struggle to create accurate, publication-quality diagrams in LaTeX due to TikZ's complexity. Our solution provides an AI-powered assistant that converts natural language prompts and refinement commands into valid TikZ code—removing the need for manual coding and streamlining the academic publishing process.

Proposed Solution:

An AI Agent hosted on IBM watsonx.ai Agentic Lab that:

- Converts natural language descriptions into LaTeX TikZ code
- Accepts plain-English refinement commands for editing diagrams
- Generates publication-ready TikZ code for Overleaf
- Diagrams can be rendered and exported from Overleaf directly

Note: Frontend-based live preview and export features are currently under development.

# IBM cloud services and technology used

- **IBM Watsonx.ai Agentic Lab:** For agent orchestration
- **IBM Granite Models (Command):** For generating/editing LaTeX code
- **IBM Cloud:** For hosting API services
- **IBM Cloud Watsonx AI Studio**
- **IBM Cloud Watsonx AI runtime**
- **Watsonx.ai Inference API:** For agent deployment
- **Frontend:** HTML5, JavaScript, Tailwind CSS, Fetch API, TikZJax
- **Backend:** Node.js, Express.js, Fetch API(Native)

# Wow factors

1. Converts **plain English** into accurate TikZ LaTeX diagram code
2. Powered by **IBM Watsonx.ai Agentic Lab + Granite Command models**
3. Real-time agent generates code that's compatible with **Overleaf**
4. Users can refine diagrams using **plain-English editing instructions**
5. Eliminates complex LaTeX learning for researchers
6. Frontend (under development) aims to render live previews + downloads
7. Backend API successfully deployed using **Watsonx Inference API**

# End users

- **Researchers / Academicians**  
To speed up technical diagram creation for LaTeX-based publications
- **Students / Educators**  
To visually explain concepts using trees, graphs, block diagrams in lectures and assignments
- **Technical Writers / Engineers**  
To integrate system designs, process flows, and architecture diagrams in formal documents
- **Anyone using Overleaf**  
To generate and export TikZ-based diagrams without LaTeX expertise

# Results

IBM watsonx

Upgrade ⓘ 🔔 Shraavani Salunkhe's Acco... Frankfurt SS

Projects / TikZ Agentic Project / TikZAgent

Share feedback ⓘ Autosaved 7:52 PM New agent + Deploy 🔒 ⓘ

Build

AI Model: granite-3-3-8b-instruct 🔗 <

Setup

Configuration

Framework Architecture

LangGraph ReAct

Instructions

You are a LaTeX expert whose only job is to create TikZ diagrams. When a user describes a diagram, you will generate the full, correct LaTeX document with the TikZ code.

You must follow these rules strictly:

Knowledge

Tools

Add a tool Create custom tool

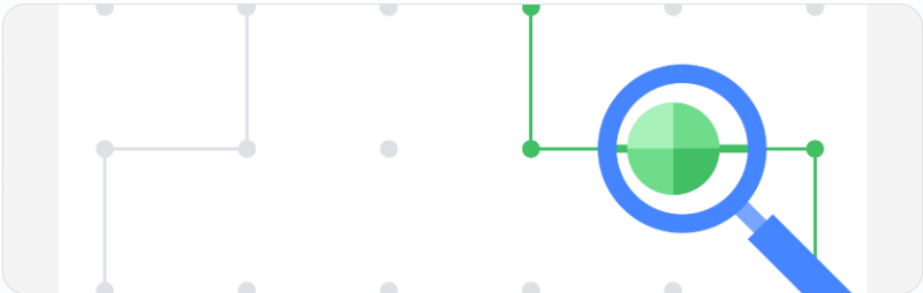
Advanced configuration

Agent preview

TikZAgent 07:52 PM

Welcome to TikZAgent

Generates and edits LaTeX TikZ diagrams from natural language prompts.



Type something... 🔍 ➤

# Binary Tree

## Agent preview

SS You: 07:55 PM

Draw a binary tree with A as root and B, C as children.

TikZAgent 07:55 PM

```
\documentclass{article}
\usepackage{forest}

\begin{document}

\begin{forest}
for tree={circle, draw, minimum size=1cm, edge={->}}
[A
  [B]
  [C]
]
\end{forest}

\end{document}
```

► How did I get this answer?

Type something...

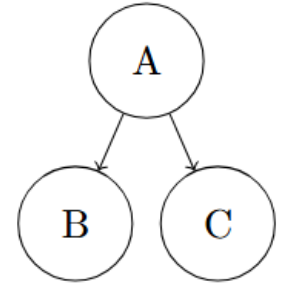
```
\documentclass{article}
\usepackage{forest}

\begin{document}

\begin{forest}
for tree={circle, draw, minimum size=1cm, edge={->}}
[A
  [B]
  [C]
]
\end{forest}

\end{document}
```

Editing ▾



## Prompt-to-TikZ Conversion Efficiency

- Converts natural language prompts into valid TikZ LaTeX code in <10 seconds
- Handles diverse academic diagram types: tree structures, flowcharts, graphs, neural nets



# Binary Tree

## Agent preview



You: 08:04 PM

Add D and E as children of B.



TikZAgent 08:04 PM

```
\documentclass{article}
\usepackage{forest}

\begin{document}

\begin{forest}
for tree={circle, draw, minimum size=1cm, edge={->}}
[A
  [B
    [D]
    [E]
  ]
  [C]
]
\end{forest}

\end{document}
```

Type something...

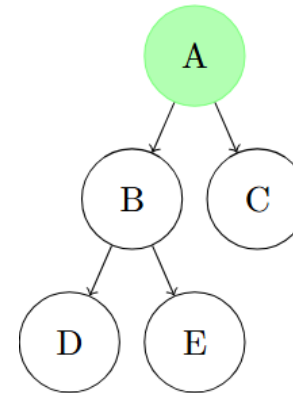
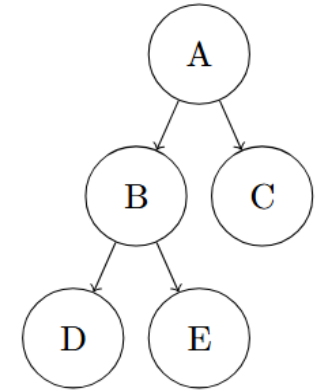
```
\documentclass{article}
\usepackage{forest}

\begin{document}

\begin{forest}
for tree={circle, draw, minimum size=1cm, edge={->}}
[A
  [B
    [D]
    [E]
  ]
  [C]
]
\end{forest}

\end{document}
```

Editing



## Supports Dynamic Editing

- Accepts plain-English refinement commands(e.g., “Add node”, “Change shape”, “Make green”)
- Regenerates complete updated code with applied changes

# Flowchart

## Agent preview



You: 08:13 PM

Create a flowchart with "Start", "Input", "Process", "Output", and "End" boxes.



TikZAgent 08:13 PM

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{arrows.meta, positioning, shapes.geometric}

\tikzstyle{startstop} = [circle, minimum width=1cm, minimum height=1cm,
text centered, draw=black, fill=blue!30]
\tikzstyle{process} = [rectangle, rounded corners, minimum width=2.5cm,
minimum height=1cm, text centered, draw=black, fill=orange!30]
\tikzstyle{decision} = [diamond, aspect=2, text centered, draw=black,
fill=green!30]

\begin{document}

\begin{tikzpicture}[node distance=2cm]
\node (start) [startstop] {Start};
\node (input) [process, below of=start] {Input};
\node (process) [process, below of=input] {Process};
\node (output) [process, below of=process] {Output};
\node (end) [startstop, below of=output] {End};

\draw[->] (start) -- (input);
\draw[->] (input) -- (process);
\draw[->] (process) -- (output);
\draw[->] (output) -- (end);
\end{tikzpicture}
\end{document}
```

Type something...



```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{arrows.meta, positioning,
shapes.geometric}

\tikzstyle{startstop} = [circle, minimum width=1cm,
minimum height=1cm, text centered, draw=black,
fill=blue!30]
\tikzstyle{process} = [rectangle, rounded corners,
minimum width=2.5cm, minimum height=1cm, text centered,
draw=black, fill=orange!30]
\tikzstyle{decision} = [diamond, aspect=2, text
centered, draw=black, fill=green!30]

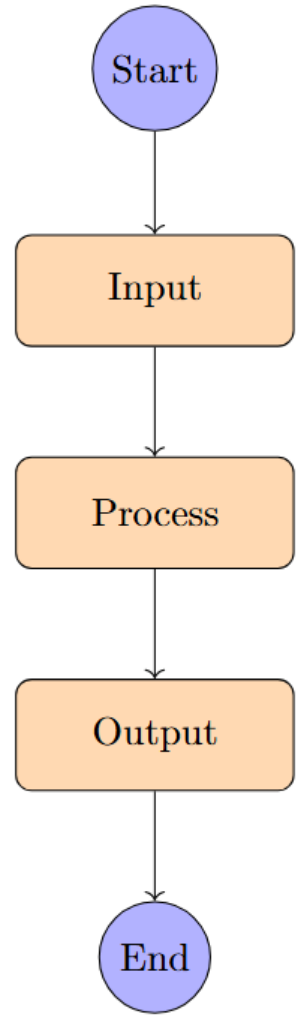
\begin{document}

\begin{tikzpicture}[node distance=2cm]
\node (start) [startstop] {Start};
\node (input) [process, below of=start] {Input};
\node (process) [process, below of=input] {Process};
\node (output) [process, below of=process] {Output};
\node (end) [startstop, below of=output] {End};

\draw[->] (start) -- (input);
\draw[->] (input) -- (process);
\draw[->] (process) -- (output);
\draw[->] (output) -- (end);
\end{tikzpicture}

\end{document}
```

Editing



# Neural Network

## Agent preview



You: 08:16 PM

Generate a neural network with 3 input neurons, 4 hidden neurons, and 1 output neuron.



TikZAgent 08:16 PM

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{arrows.meta, positioning, shapes.geometric}

\tikzstyle{input} = [circle, draw, minimum size=1cm, text centered,
fill=blue!30]
\tikzstyle{hidden} = [rectangle, draw, minimum size=1cm, text centered,
fill=orange!30]
\tikzstyle{output} = [circle, draw, minimum size=1cm, text centered,
fill=red!30]

\begin{document}

\begin{tikzpicture}[auto, thick, node distance=2cm]
\node[input] (input1) {I1};
\node[input] (input2) [below=of input1] {I2};
\node[input] (input3) [below=of input2] {I3};

\node[hidden] (hidden1) [right=of input1] {H1};
\node[hidden] (hidden2) [right=of hidden1] {H2};
\node[hidden] (hidden3) [right=of hidden2] {H3};
\node[hidden] (hidden4) [right=of hidden3] {H4};

\node[output] (output1) [right=of hidden4] {O1};

input1 --> hidden1;
input2 --> hidden2;
input3 --> hidden3;

hidden1 --> hidden2;
hidden2 --> hidden3;
hidden3 --> hidden4;
hidden4 --> output1;
\end{tikzpicture}
\end{document}
```

Type something...

```
fill=blue!30]
len) = [rectangle, draw, minimum size=1cm,
fill=orange!30]
ut} = [circle, draw, minimum size=1cm,
fill=red!30]

}

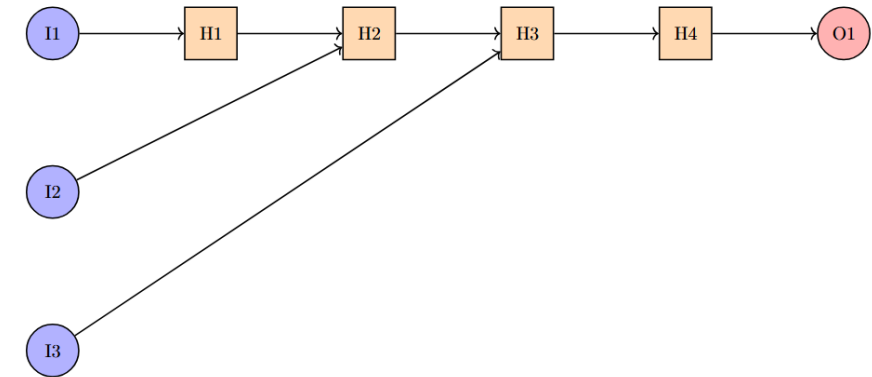
\begin{tikzpicture}[auto, thick, node distance=2cm]
\node[input] (input1) {I1};
\node[input] (input2) [below=of input1] {I2};
\node[input] (input3) [below=of input2] {I3};

\node[hidden] (hidden1) [right=of input1] {H1};
\node[hidden] (hidden2) [right=of hidden1] {H2};
\node[hidden] (hidden3) [right=of hidden2] {H3};
\node[hidden] (hidden4) [right=of hidden3] {H4};

\node[output] (output1) [right=of hidden4] {O1};

input1 --> hidden1;
input2 --> hidden2;
input3 --> hidden3;

hidden1 --> hidden2;
hidden2 --> hidden3;
hidden3 --> hidden4;
hidden4 --> output1;
\end{tikzpicture}
```



## Output Compatibility

- Generated code is plug-and-play compatible with Overleaf and LaTeX editors
- Preview and export to PDF possible through Overleaf even if frontend fails temporarily

# Entity Relationship Diagram

## Agent preview

SS

You: 08:20 PM

Draw an ER diagram with entities "Student", "Course", and "Enrollment".

TikZAgent

08:20 PM

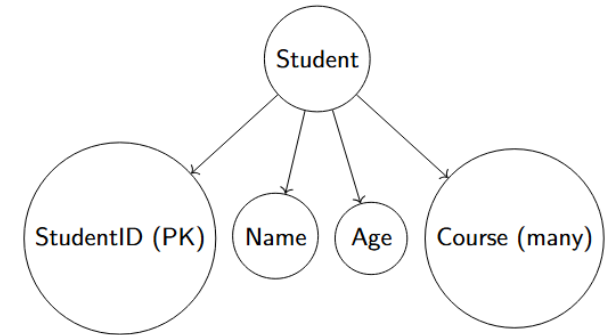
```
\documentclass{article}
\usepackage{forest}
\usepackage{tikz}
\usetikzlibrary{arrows.meta, positioning, shapes.geometric}

\begin{document}

\begin{forest}
for tree={
  circle,
  draw,
  minimum size=1cm,
  edge={->},
  text centered,
  font=\sffamily,
},
[Student
[StudentID (PK)]
[Name]
[Age]
[Course (many)]
]
```

Type something...

```
\usetikzlibrary{arrows.meta, positioning,
shapes.geometric}
\begin{document}
\begin{forest}
for tree={
  circle,
  draw,
  minimum size=1cm,
  edge={->},
  text centered,
  font=\sffamily,
},
[Student
[StudentID (PK)]
[Name]
[Age]
[Course (many)]
]
[Course
[CourseID (PK)]
[Name]
[Credit]
[Student (many)]
]
[Enrollment
[EnrollmentID (PK)]
[StudentID (many)]
[CourseID (many)]
]
]\end{forest}
```



# Deployed AI Agent

IBM watsonx

Upgrade ? Bell


Shraavani Salunkhe's Acco... Frankfurt SS

Deployment spaces / TikZAgentDeployment / TikZAgent /

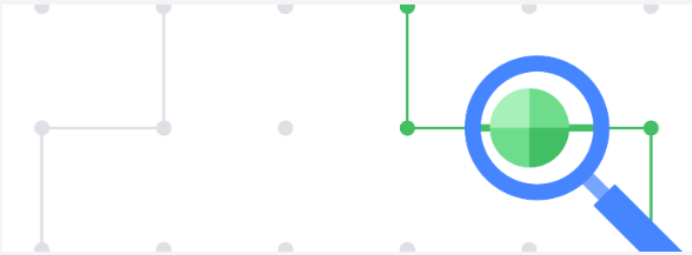
TikZAgent ✓ Deployed Online


API reference Test Evaluations **Preview**

New chat +


 TikZAgent 08:26 PM


**Welcome to TikZAgent**  
Generates and edits LaTeX TikZ diagrams from natural language prompts.











Type something... 

About this deployment ×

**Name**   
TikZAgent

**Description**   
Generates and edits LaTeX TikZ diagrams from natural language prompts.

**Deployment Details**  
Deployment ID: 108ea519-098c-41...  
Serving name:   
No serving name.  
Software specification:   
[runtime-24.1-py3.11](#)   
Hardware specification:   
Extra extra small: 1 CPU and 2 GB RAM  
Copies:   
1  
**Tags**   
wx-agent  
**Associated asset**   
 [TikZAgent](#)

# Frontend

→ ↺ 127.0.0.1:5500/index.html

Google Gmail YouTube Stack Overflow - W... Computer Science C... Practice Problems ChatGPT WhatsApp Artificial Intelligenc... TikZAgentDeploym...

All Bookmarks

# TikZ Diagram Generator

Describe a diagram in natural language, and let the AI generate the TikZ code for you.

### Diagram Description

e.g., Draw a flowchart with a 'Start' box, a 'Process' box, and an 'End' box, with arrows connecting them.

Generate Diagram

### Edit Diagram (Natural Language)

e.g., Change the 'Process' box to a diamond shape.

Apply Edit

### Generated TikZ Code

Copy to Clipboard

### Diagram Preview

Your diagram will appear here.

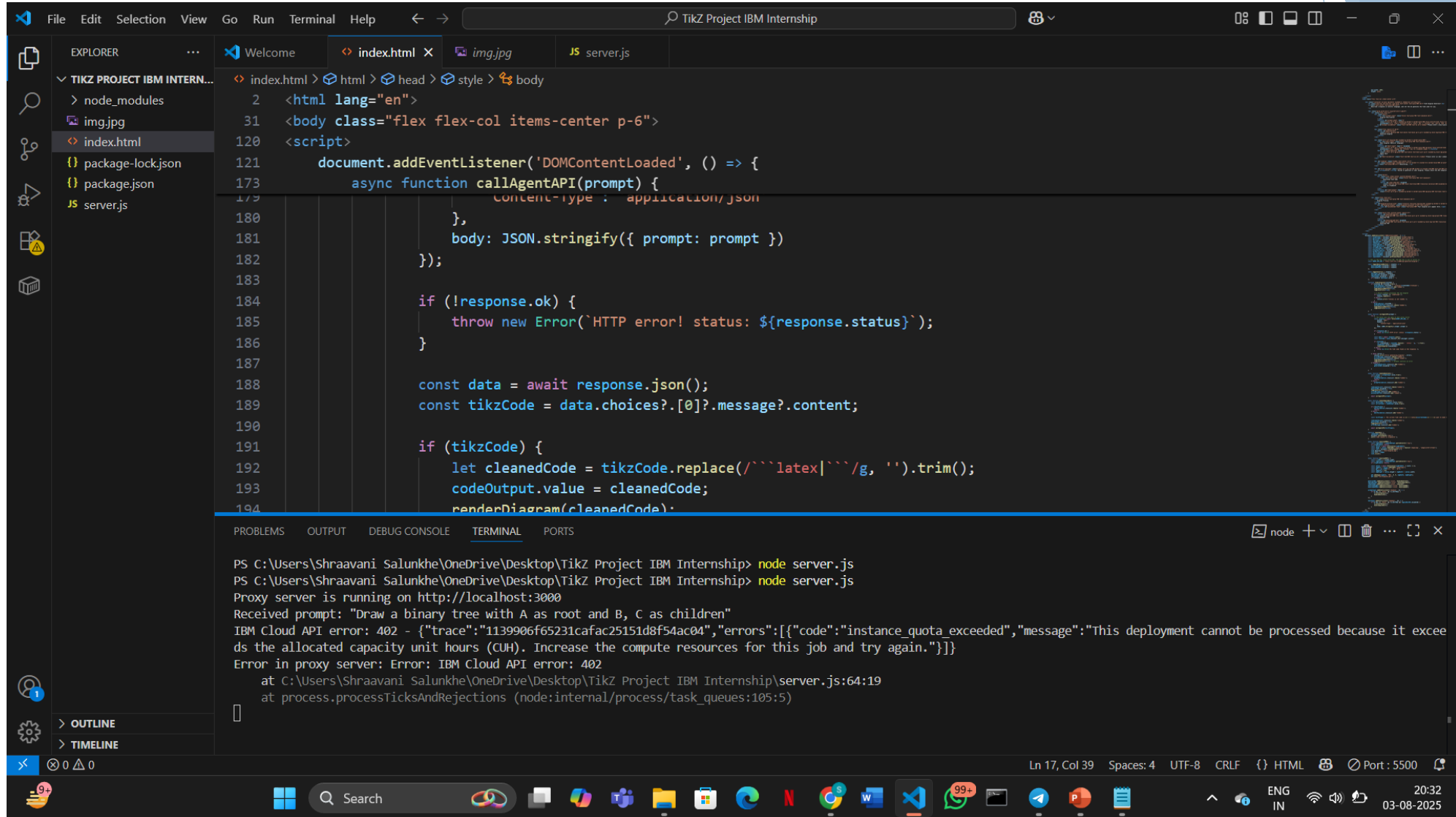
Download PNG

Download PDF

edunet

foundation

# Why Frontend is getting delayed: THE ISSUE



The screenshot shows a Visual Studio Code editor window for a project named "TikZ Project IBM Internship". The Explorer sidebar on the left shows the file structure with files like `index.html`, `img.jpg`, `package-lock.json`, `package.json`, and `server.js`. The main editor area displays the `index.html` file, which includes a `<script>` block with a `document.addEventListener('DOMContentLoaded', ...)` and an `async function callAgentAPI(prompt)`. The terminal at the bottom shows the command `node server.js` being executed, followed by a series of log messages and an error: `IBM Cloud API error: 402 - {"trace": "1139906f65231caf25151d8f54ac04", "errors": [{"code": "instance_quota_exceeded", "message": "This deployment cannot be processed because it exceeds the allocated capacity unit hours (CUH). Increase the compute resources for this job and try again."}]}`. The error message indicates that the deployment is failing due to a quota issue, which is likely causing the frontend to be delayed.

```
2 <html lang="en">
31 <body class="flex flex-col items-center p-6">
120 <script>
121     document.addEventListener('DOMContentLoaded', () => {
173         async function callAgentAPI(prompt) {
179             content-type : 'application/json'
180         },
181         body: JSON.stringify({ prompt: prompt })
182     });
183
184     if (!response.ok) {
185         throw new Error(`HTTP error! status: ${response.status}`);
186     }
187
188     const data = await response.json();
189     const tikzCode = data.choices?.[0]?.message?.content;
190
191     if (tikzCode) {
192         let cleanedCode = tikzCode.replace(/```latex|```/g, '').trim();
193         codeOutput.value = cleanedCode;
194         renderDiagram(cleanedCode);
    }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Shraavani Salunkhe\OneDrive\Desktop\TikZ Project IBM Internship> node server.js
PS C:\Users\Shraavani Salunkhe\OneDrive\Desktop\TikZ Project IBM Internship> node server.js
Proxy server is running on http://localhost:3000
Received prompt: "Draw a binary tree with A as root and B, C as children"
IBM Cloud API error: 402 - {"trace": "1139906f65231caf25151d8f54ac04", "errors": [{"code": "instance_quota_exceeded", "message": "This deployment cannot be processed because it exceeds the allocated capacity unit hours (CUH). Increase the compute resources for this job and try again."}]}
Error in proxy server: Error: IBM Cloud API error: 402
    at C:\Users\Shraavani Salunkhe\OneDrive\Desktop\TikZ Project IBM Internship\server.js:64:19
    at process.processTicksAndRejections (node:internal/process/task_queues:105:5)
```

Ln 17, Col 39 Spaces: 4 UTF-8 CRLF {} HTML Port: 5500

# Conclusion

- This project removes the manual complexity of TikZ diagram creation in LaTeX.
- Using **IBM Watsonx.ai** and **Granite foundation models**, we built an AI agent that enables researchers and students to describe diagrams in English and receive Overleaf-compatible code instantly.
- Even with the frontend temporarily limited, the backend agent pipeline is fully functional and integrated.



# Future Scope

- **Frontend Fix:** Resolve IBM Cloud resource quota issue and restore live preview using TikZJax
- Enhance editing support (multi-turn, error correction, advanced styling)
- Add support for **hand-drawn sketch input** → **TikZ conversion** (by using OCR)
- Save user prompts, diagrams, and edit history
- Expand prompt understanding to cover math, electronics, control systems, etc

# IBM Certifications



## Getting Started with Artificial Intelligence

**Issued on: Jul 15, 2025**

**Issued by: IBM SkillsBuild**

**Verify:**

**<https://www.credly.com/badges/ca84dfb8-5124-47cc-9bd2-62850449bdfa>**

# IBM Certifications



## Journey to Cloud: Envisioning Your Solution

Issued on: Jul 19, 2025

Issued by: IBM SkillsBuild

Verify:

<https://www.credly.com/badges/5d2467de-98f7-4819-b3de-0f9853d68b65>

# IBM Certifications

IBM SkillsBuild

Completion Certificate



This certificate is presented to

Shraavani Salunkhe

for the completion of

**Lab: Retrieval Augmented Generation with  
LangChain**

(ALM-COURSE\_3824998)

According to the Adobe Learning Manager system of record

**Completion date:** 22 Jul 2025 (GMT)

**Learning hours:** 20 mins

**Lab: Retrieval Augmented  
Generation with LangChain**

**(ALM-COURSE\_3824998)**

**According to the Adobe  
Learning Manager system of  
record**

# GitHub Link

**<https://github.com/shraavani30/TikZ-Agent>**

# THANK YOU