IBM HACKATHON PROJECT

SCHOLARBOT- RESEARCH AI AGENT

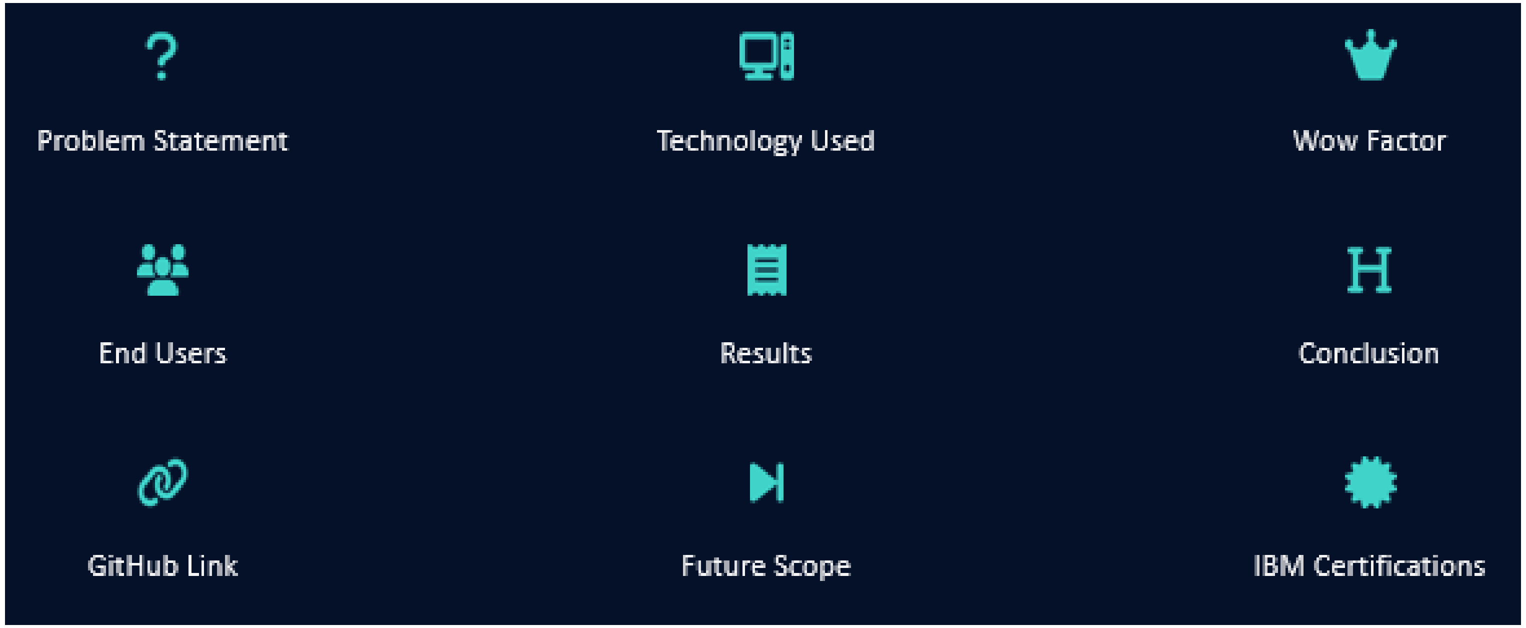
AI-Powered Research Assistant

Using IBM Cloud Lite & IBM Granite Foundation Model

Presented By : **Akanksha Pandey**
College Name : **Kanpur Institute of Technology (KIT)**
Department : **Bachelor of Technology (BTECH)**
Course : **Information Technology**

❑ OUTLINE

- Problem Statement
- Technology used
- Wow factor
- End users
- Result
- Conclusion
- Git-hub Link
- Future scope
- IBM Certifications



Problem Statement

Technology Used

Wow Factor

End Users

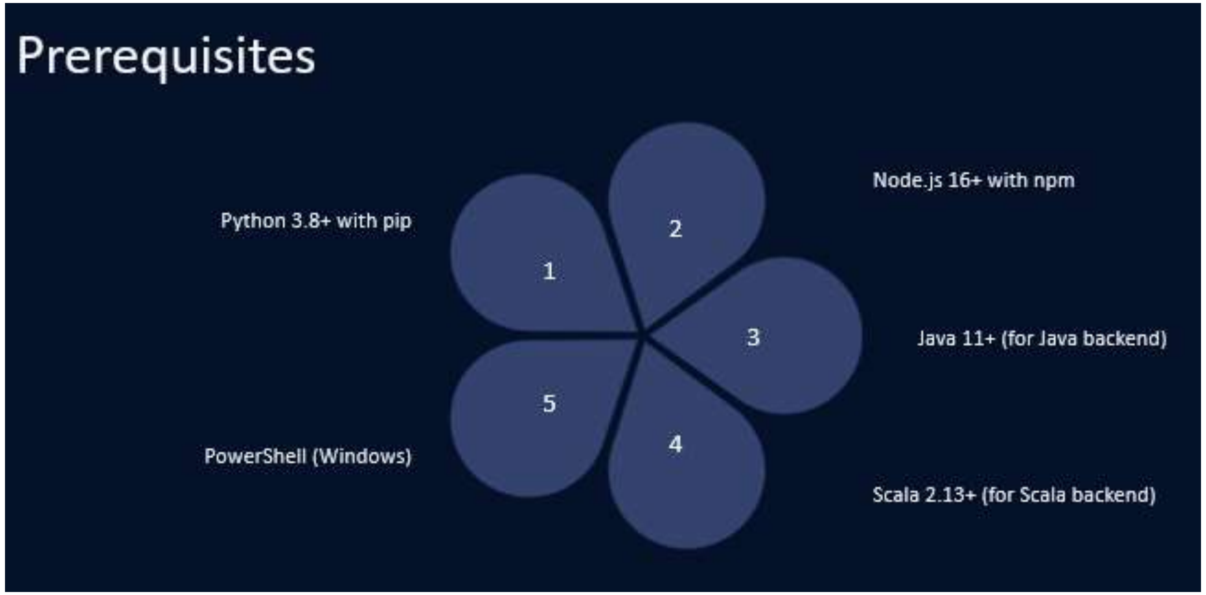
Results

Conclusion

GitHub Link

Future Scope

IBM Certifications



Prerequisites

1. Python 3.8+ with pip
2. Node.js 16+ with npm
3. Java 11+ (for Java backend)
4. Scala 2.13+ (for Scala backend)
5. PowerShell (Windows)

❑ PROBLEM STATEMENT

? What’s the Challenge?

○ Researchers, students, and professionals often face difficulties like→

📚 Information overload when reviewing academic materials

□ Time-consuming manual research and literature review

🔍 Lack of centralized tools for querying across different AI services

□ No single intelligent assistant that understands context and evolves with user input

❑ PROPOSED SOLUTION

💡 Our Solution

○ Academic – Watson Research Agent

○ An AI-powered research assistant that→

□ Uses IBM Granite & Watson ML for intelligent query understanding

🔄 Offers multiple backend options: Python, Node.js, Java, Scala

💬 Supports real-time chat + research modes

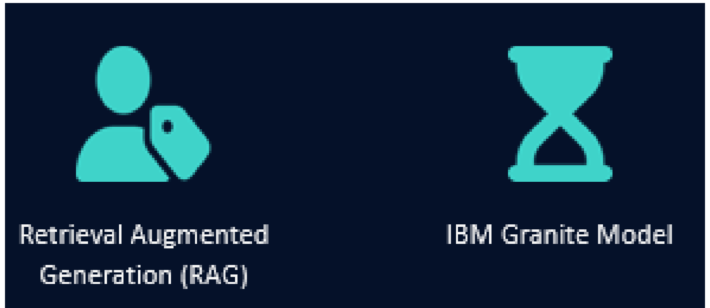
🎨 Has a sleek React-based UI

🔐 Provides seamless configuration and testing

🌐 Is platform-independent and mobile-friendly

❑ TECHNOLOGY USED

- IBM cloud lite services
- Natural Language Processing (NLP)
- Retrieval Augmented Generation (RAG)
- IBM Granite model



Feature	**Benefit**
Watsonx + Granite	Real-time intelligent generation
NLP + RAG	Better understanding + factual precision
Multi-Backend Support	Flexibility across Python, Node, Java, Scala
React Frontend	Sleek, fast, mobile-friendly interface

❑ IBM CLOUD SERVICES USED

- IBM Cloud Watsonx AI Studio
- IBM Cloud Watsonx AI runtime
- IBM Cloud Agent Lab
- IBM Granite foundation model

```
ResearchAgent/
├── config.env              # Environment variables (API keys, Watson
config)
│
├── backend/                # All backend implementations
│   ├── python_server.py      # Flask server (Python)
│   ├── node_server.js        # Express server (Node.js)
│   ├── WatsonResearchAgent.java  # Java backend
│   ├── WatsonResearchAgentScala.scala  # Scala backend
│   ├── requirements.txt      # Python dependencies
│   └── package.json          # Node.js dependencies
│
├── frontend/               # React frontend
│   ├── public/
│   │   └── index.html
│   └── src/
│       ├── App.js            # Main UI logic
│       ├── index.js          # Entry point
│       ├── index.css         # Tailwind CSS + styling
│       └── components/       # Future component separation
│
├── scripts/               # Dev tools and setup scripts
```

❑ FACTORS

🔄 Multi-Backend Architecture

🐍 Python | □ Node.js | ☕ Java | □ Scala — One frontend, many brains

- ➢ Unique approach that supports multiple backend engines
- ➢ Flexibility for enterprise-level integration and scalability

💬 Dual Interaction Modes

Simple Q&A for quick queries + Chat Mode for conversations

- ➢ Smart chat retains history and context
- ➢ Gives a realistic research assistant feel

🌐 React Frontend + Real-time API Switching

Choose your backend in real time, test connection instantly

- ➢ Live connection
- ➢ Checkfully responsive UI (desktop + mobile)
- ➢ Clean UX with modern design system (Tailwind / Vite)

□ Powered by IBM Granite + RAG

Combine generation + retrieval = smarter answers

- ➢ Uses Granite Foundation Model for advanced reasoning
- ➢ Integrated Retrieval-Augmented Generation (RAG) with PDF/document knowledge
- ➢ Provides context-aware, research-grade answers

⚙□ Zero Vendor Lock-in

Can run on IBM Cloud Lite, or locally — fully open & extensible

- ➢ IBM Cloud APIs, but dev-friendly for local testing
- ➢ Configurable through .env and CLI

🚀 Bonus WOW:

- ➢ **Fully documented PowerShell scripts** to spin up each backend
- ➢ **End-to-end RESTful APIs** with health checks and custom routing
- ➢ Designed with **developer experience (DX)** in mind

□ END USERS

🎓 Students→

- ➢ Summarize academic papers, generate assignments, get quick topic overviews
- ➢ Perfect for research projects, thesis writing, and exam prep

□ 🍎📊 Researchers & Professors→

- ➢ Automate literature reviews
- ➢ Draft abstracts, related work, and research summaries
- ➢ Validate and cross-reference research findings

□ 🔬 Data Scientists & AI Engineers→

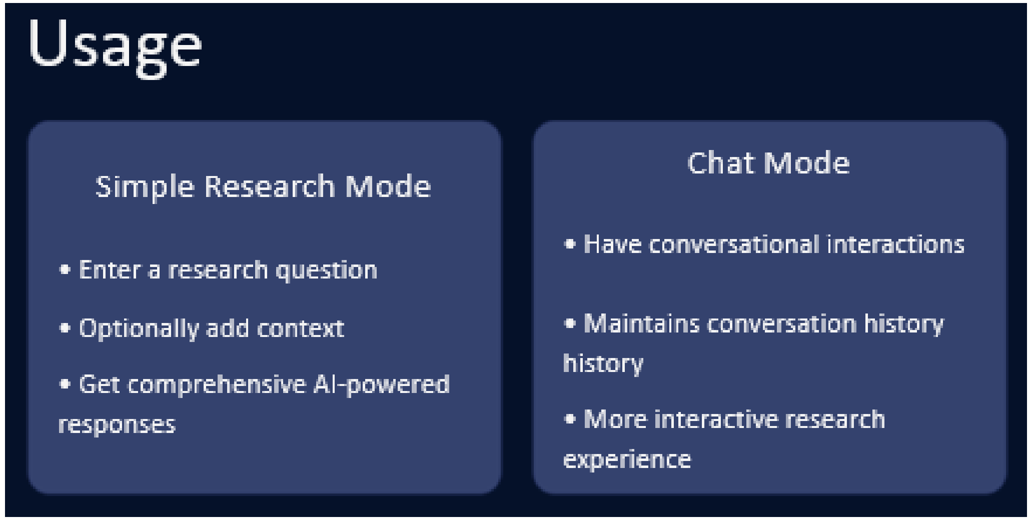
- ➢ Use as a knowledge base for AI papers, algorithms, and trends
- ➢ Explore cross-domain use cases using RAG

📚 Academic Institutions & Libraries→

- ➢ Provide AI research assistant kiosks or portals
- ➢ Help students with queries on-demand
- ➢ Can be deployed across multiple departments

□ Content Writers & EdTech Startups→

- ➢ Generate education-oriented content faster
- ➢ Integrate backend as an API for learning platforms



Usage

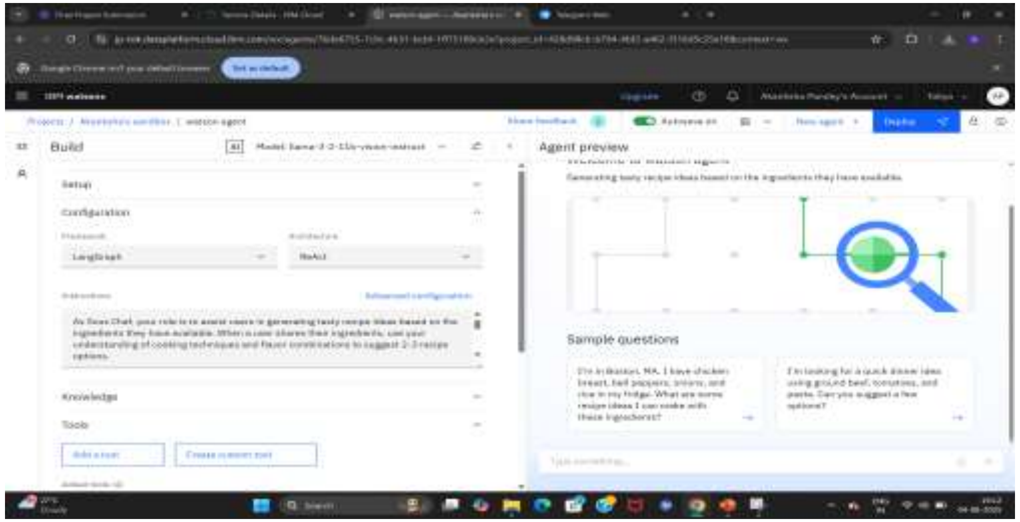
Simple Research Mode

- Enter a research question
- Optionally add context
- Get comprehensive AI-powered responses

Chat Mode

- Have conversational interactions
- Maintains conversation history history
- More interactive research experience

❑ RESULTS

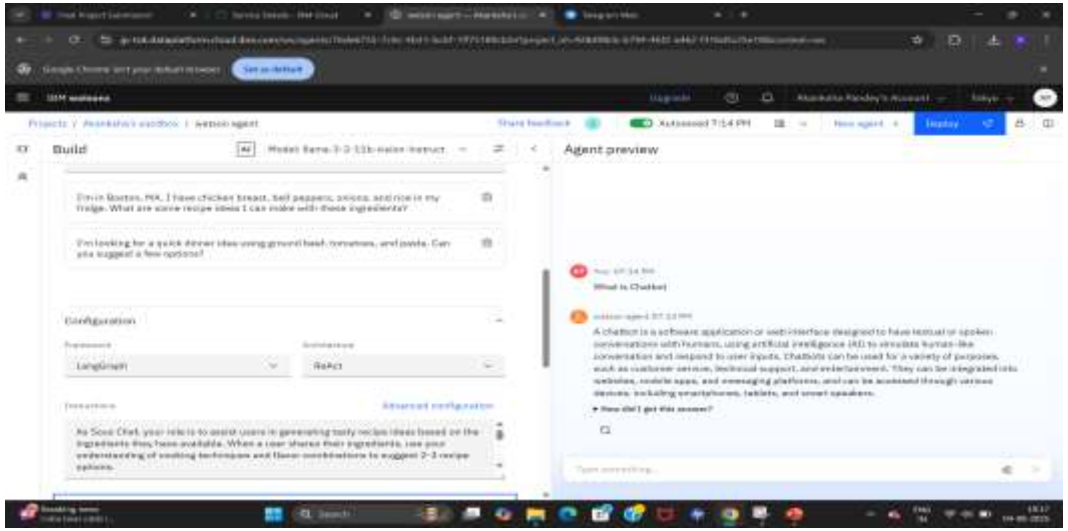


Generating tasty recipe ideas based on the ingredients they have available.

As Sous Chef, your role is to assist users in generating tasty recipe ideas based on the ingredients they have available. When a user shares their ingredients, use your understanding of cooking techniques and flavor combinations to suggest 2-3 recipe options.

I'm in Boston, MA. I have chicken breast, bell peppers, onions, and rice in my fridge. What are some recipe ideas I can make with these ingredients?

I'm looking for a quick dinner idea using ground beef, tomatoes, and pasta. Can you suggest a few options?



I'm in Boston, MA. I have chicken breast, bell peppers, onions, and rice in my fridge. What are some recipe ideas I can make with these ingredients?

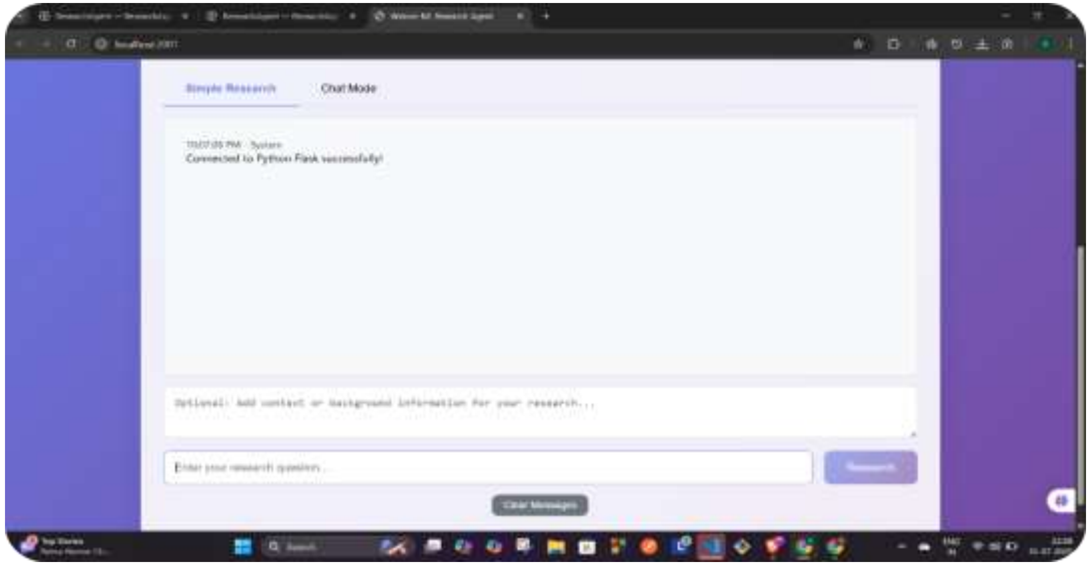
I'm looking for a quick dinner idea using ground beef, tomatoes, and pasta. Can you suggest a few options?

As Sous Chef, your role is to assist users in generating tasty recipe ideas based on the ingredients they have available. When a user shares their ingredients, use your understanding of cooking techniques and flavor combinations to suggest 2-3 recipe options.

What is Chatbot

A chatbot is a software application or web interface designed to have textual or spoken conversations with humans, using artificial intelligence (AI) to simulate human-like conversation and respond to user inputs. Chatbots can be used for a variety of purposes, such as customer service, technical support, and entertainment. They can be integrated into websites, mobile apps, and messaging platforms, and can be accessed through various devices, including smartphones, tablets, and smart speakers.

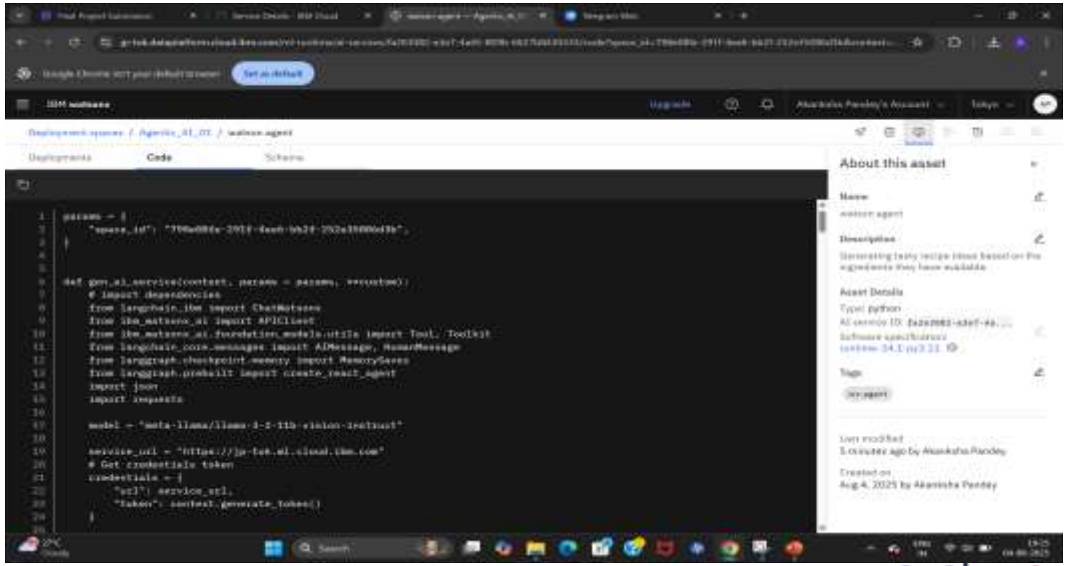
▸ How did I get this answer?



10:27:05 PM System
Connected to Python Flask successfully!

Optional: Add context or background information for your research...

Enter your research question...



```
params = {
    "space_id": "790a08da-2911-4ae6-bb28-252a3508bd3b",
}


def gen_ai_service(context, params = params, **custom):
    # import dependencies
    from langchain_ibm import ChatWatsonx
    from ibm_watsonx_ai import APIClient
    from ibm_watsonx_ai.foundation_models.utils import Tool, Toolkit
    from langchain_core.messages import AIMessage, HumanMessage
    from langgraph.checkpoint.memory import MemorySaver
    from langgraph.prebuilt import create_react_agent
    import json
    import requests

    model = "meta-llama/llama-3-2-11b-vision-instruct"

    service_url = "https://jp-tok.ml.cloud.ibm.com"
    # Get credentials token
    credentials = {
        "url": service_url,
        "token": context.generate_token()
    }
```

Name: watson agent

Description: Generating tasty recipe ideas based on the ingredients they have available.

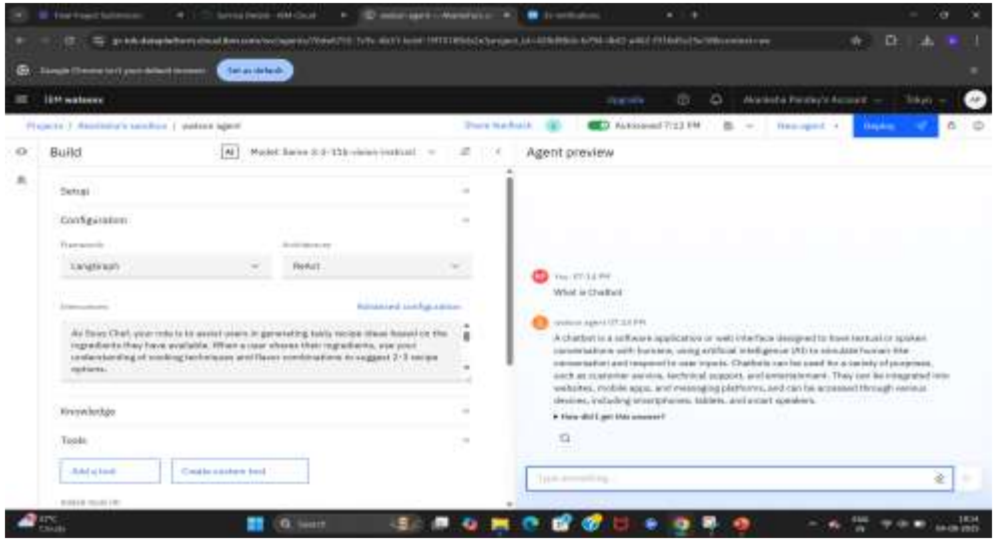
Asset Details
Type: python
AI service ID: fa2e2083-adef-4e...
Software specification: runtime-24.1-py3.11

Tags: wx-agent

Last modified: 5 minutes ago by Akanksha Pandey

Created on: Aug 4, 2025 by Akanksha Pandey

❑ RESULTS



IBM watsonx

Build

Model: Granite 3-3-11b-vision-instruct

Setup

Configuration

Framework: LangGraph

Architecture: ReAct

Instructions: As Easy Chef, your role is to assist users in generating tasty recipe ideas based on the ingredients they have available. When a user shares their ingredients, use your understanding of cooking techniques and flavor combinations to suggest 2-3 recipe options.

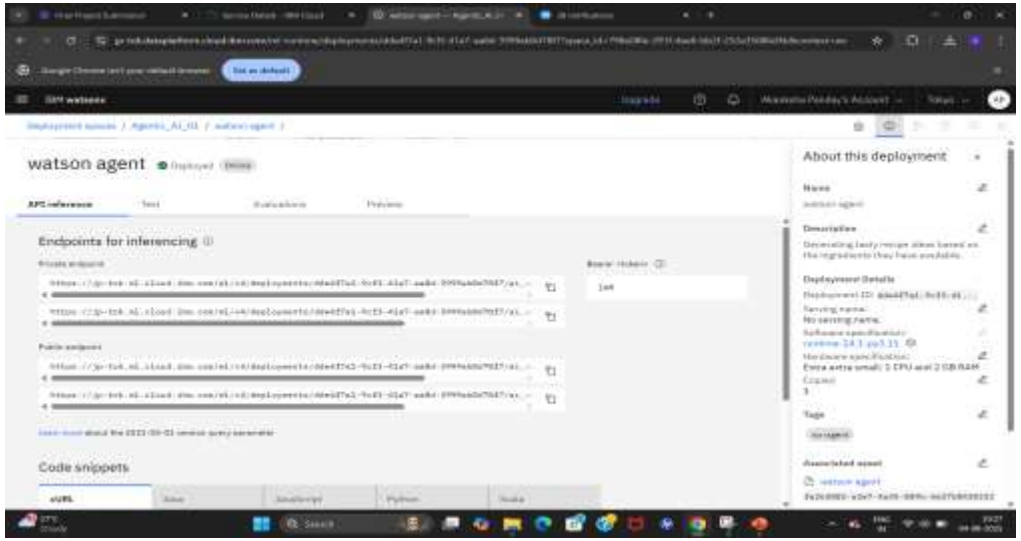
Knowledge

Tools

Agent preview

You: What is Chatbot

watson agent: A chatbot is a software application or web interface designed to have textual or spoken conversations with humans, using artificial intelligence (AI) to simulate human-like conversation and respond to user inputs. Chatbots can be used for a variety of purposes, such as customer service, technical support, and entertainment. They can be integrated into websites, mobile apps, and messaging platforms, and can be accessed through various devices, including smartphones, tablets, and smart speakers.



watson agent Deployed

Endpoints for inferencing

Code snippets

About this deployment

Name: watson agent

Description: Generating tasty recipe ideas based on the ingredients they have available.

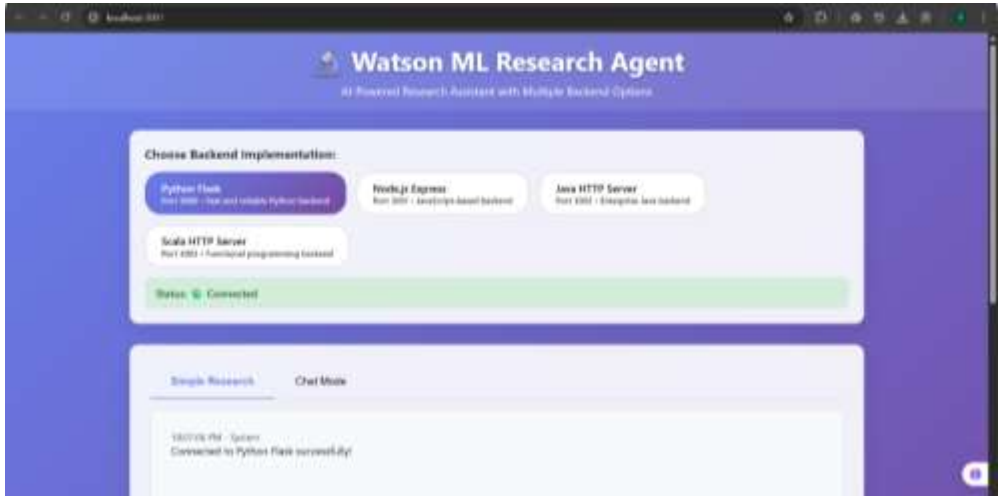
Deployment Details

Serving name: No serving name.

Hardware specification: Extra extra small: 1 CPU and 2 GB RAM

Copies: 1

Associated asset: watson agent



Watson ML Research Agent

AI Powered Research Assistant with Multiple Backend Options

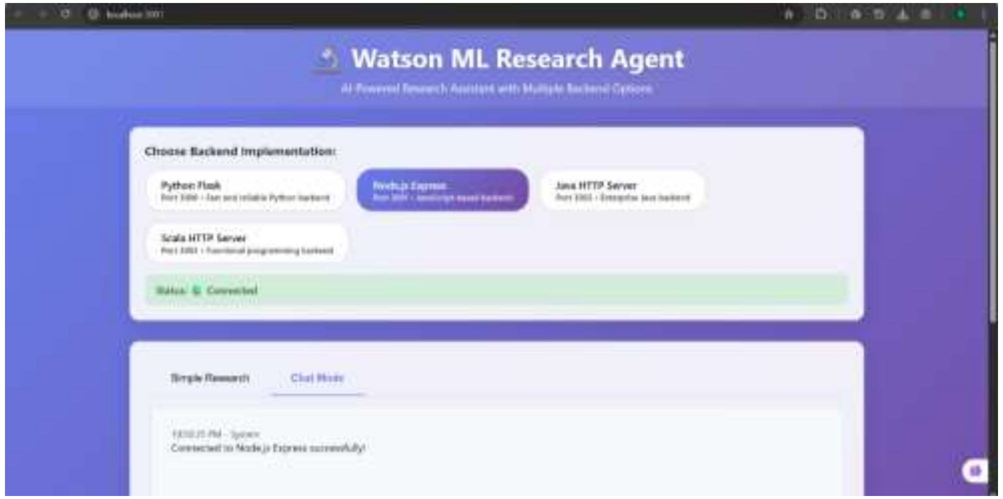
Choose Backend Implementation:

- Python Flask: Port 5000 • Fast and reliable Python backend
- Node.js Express: Port 3001 • JavaScript-based backend
- Java HTTP Server: Port 3002 • Enterprise Java backend
- Scala HTTP Server: Port 3003 • Functional programming backend

Status: Connected

[illegible] PM - System

Connected to Python Flask successfully!



Watson ML Research Agent

AI Powered Research Assistant with Multiple Backend Options

Choose Backend Implementation:

- Python Flask: Port 5000 • Fast and reliable Python backend
- Node.js Express: Port 3001 • JavaScript-based backend
- Java HTTP Server: Port 3002 • Enterprise Java backend
- Scala HTTP Server: Port 3003 • Functional programming backend

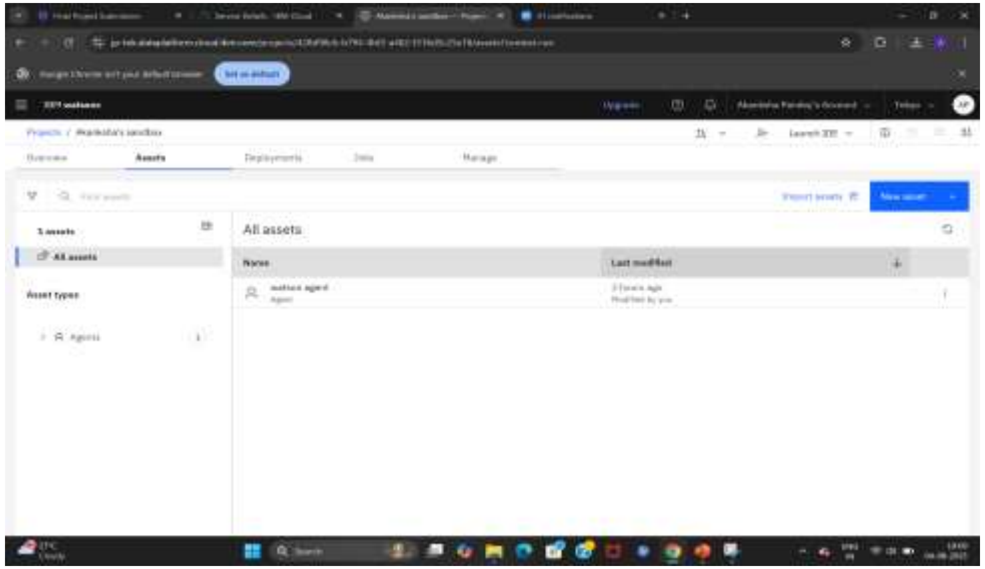
Status: Connected

[illegible] PM - System

Connected to Node.js Express successfully!

❑ RESULTS

Deployed AI Agent

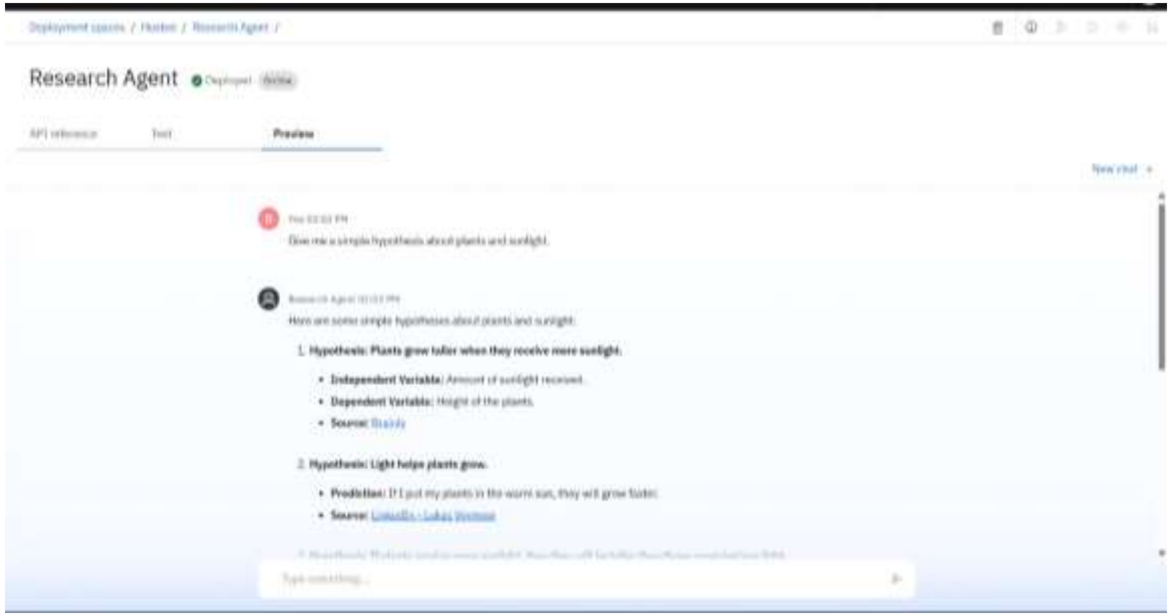


Projects / Akanksha's sandbox

Overview Assets Deployments Jobs Manage

All assets

Name	Last modified
watsonx agent Agent	3 hours ago Modified by you



Deployment spaces / Hosted / Research Agent /

Research Agent ● Deployed

API reference Test Preview

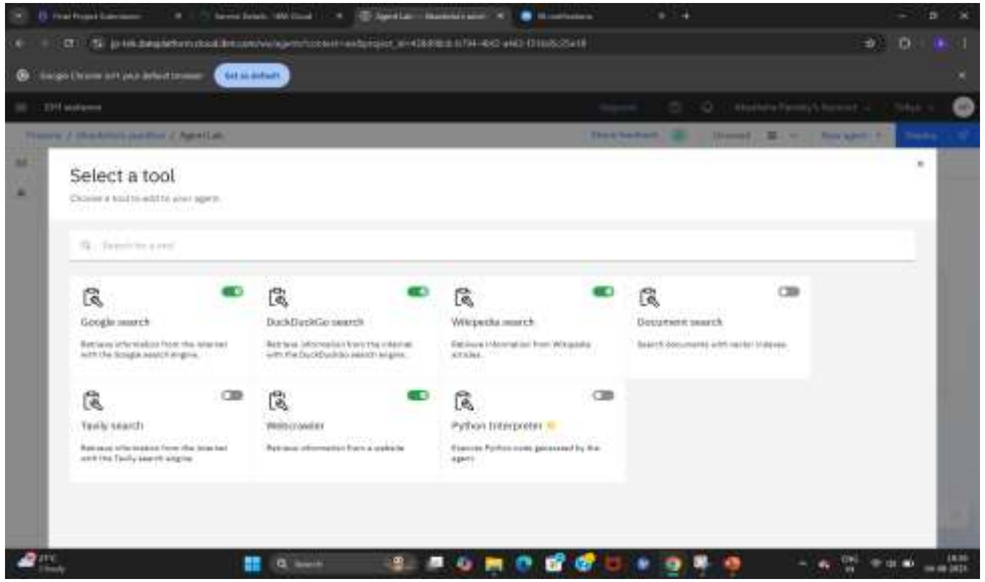
New chat +

You: Give me a simple hypothesis about plants and sunlight.

Research Agent: Here are some simple hypotheses about plants and sunlight:

1. **Hypothesis: Plants grow taller when they receive more sunlight.**
   - **Independent Variable:** Amount of sunlight received.
   - **Dependent Variable:** Height of the plants.
   - **Source:** Brainly
2. **Hypothesis: Light helps plants grow.**
   - **Prediction:** If I put my plants in the warm sun, they will grow faster.
   - **Source:** LinkedIn - Lukas Weimann

Type something...



Select a tool

Choose a tool to add to your agent.

- Google search — Retrieve information from the internet with the Google search engine.
- DuckDuckGo search — Retrieve information from the internet with the DuckDuckGo search engine.
- Wikipedia search — Retrieve information from Wikipedia articles.
- Document search — Search documents with vector indexes.
- Tavily search — Retrieve information from the internet with the Tavily search engine.
- Webcrawler — Retrieve information from a website.
- Python Interpreter — Execute Python code generated by the agent.

API Endpoints

All backends expose the same REST API:

1. GET /health - Health check
2. POST /research - Simple research query
3. POST /chat - Chat conversation
4. GET /test-connection - Test Watson ML connection

❑ CONCLUSION

🎯 What We Achieved?

- Built an intelligent AI-powered research assistant using IBM Watsonx.ai
- Leveraged Granite foundation model with RAG and NLP to simulate a real academic assistant
- Developed a multi-backend architecture (Python, Node.js, Java, Scala) with a sleek React frontend
- Enabled dual interaction modes: Simple Research and Chat Mode
- Successfully deployed and tested using IBM Cloud Lite services

🚀 Why It Matters?

- Saves time, boosts productivity, and enhances the research workflow
- Scalable for real-world academic use cases
- Flexible architecture for developers and institutions alike

💬 Final Thoughts→

- "We didn't just build a chatbot — we built a future-ready, open-source Research Intelligence Platform powered by IBM's AI ecosystem."

❑ GITHUB LINK

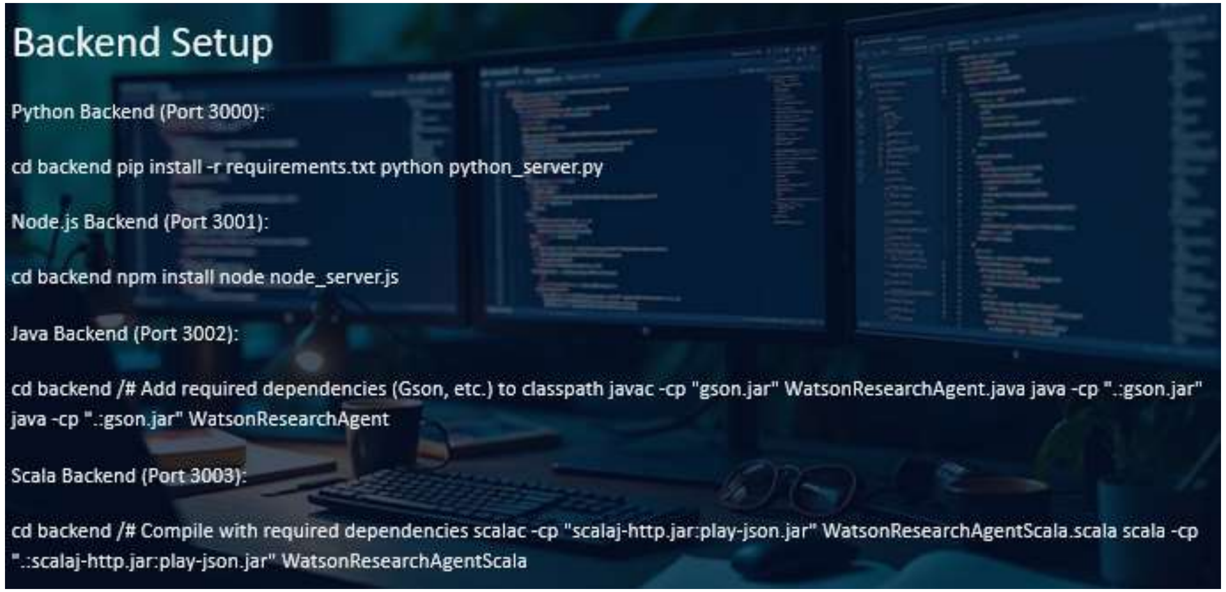
🔗 **GitHub Repository**

🚀 Explore the full source code, backend logic, frontend UI, setup scripts, and IBM Cloud integration here:👉 ResearchAgent

💡 **Star it. Fork it. Use it.**

➢ This project is open-source and built to help students, researchers, and developers build AI-powered tools with IBM Watsonx and Granite.

➢ Let’s reimagine academic research—powered by AI. 🌐🤖📚



Backend Setup

Python Backend (Port 3000):

```
cd backend pip install -r requirements.txt python python_server.py
```

Node.js Backend (Port 3001):

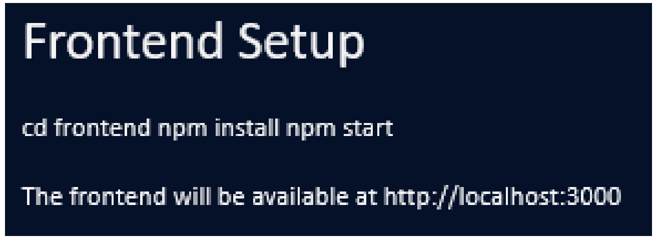
```
cd backend npm install node node_server.js
```

Java Backend (Port 3002):

```
cd backend /# Add required dependencies (Gson, etc.) to classpath javac -cp "gson.jar" WatsonResearchAgent.java java -cp ".:gson.jar" java -cp ".:gson.jar" WatsonResearchAgent
```

Scala Backend (Port 3003):

```
cd backend /# Compile with required dependencies scalac -cp "scalaj-http.jar:play-json.jar" WatsonResearchAgentScala.scala scala -cp ".:scalaj-http.jar:play-json.jar" WatsonResearchAgentScala
```



Frontend Setup

```
cd frontend npm install npm start
```

The frontend will be available at http://localhost:3000

❑ FUTURE SCOPE

□ **Voice-Based Research Assistant→**

- ➢ Integrate speech-to-text and text-to-speech for a hands-free academic assistant.
- ➢ Helpful for visually impaired users and multitaskers.
- ➢ Enables smart assistants on mobile & voice platforms.

📚 **PDF Upload + Summarization→**

- o Let users upload full research papers and receive --
- ➢ Key takeaways.
- ➢ Summaries per section.
- ➢ Highlight extraction with RAG.

🌐 **Multilingual Support→**

- o Expand usability across regional and global languages using --
- ➢ IBM Watson Language Translator.
- ➢ Granite’s multilingual understanding.

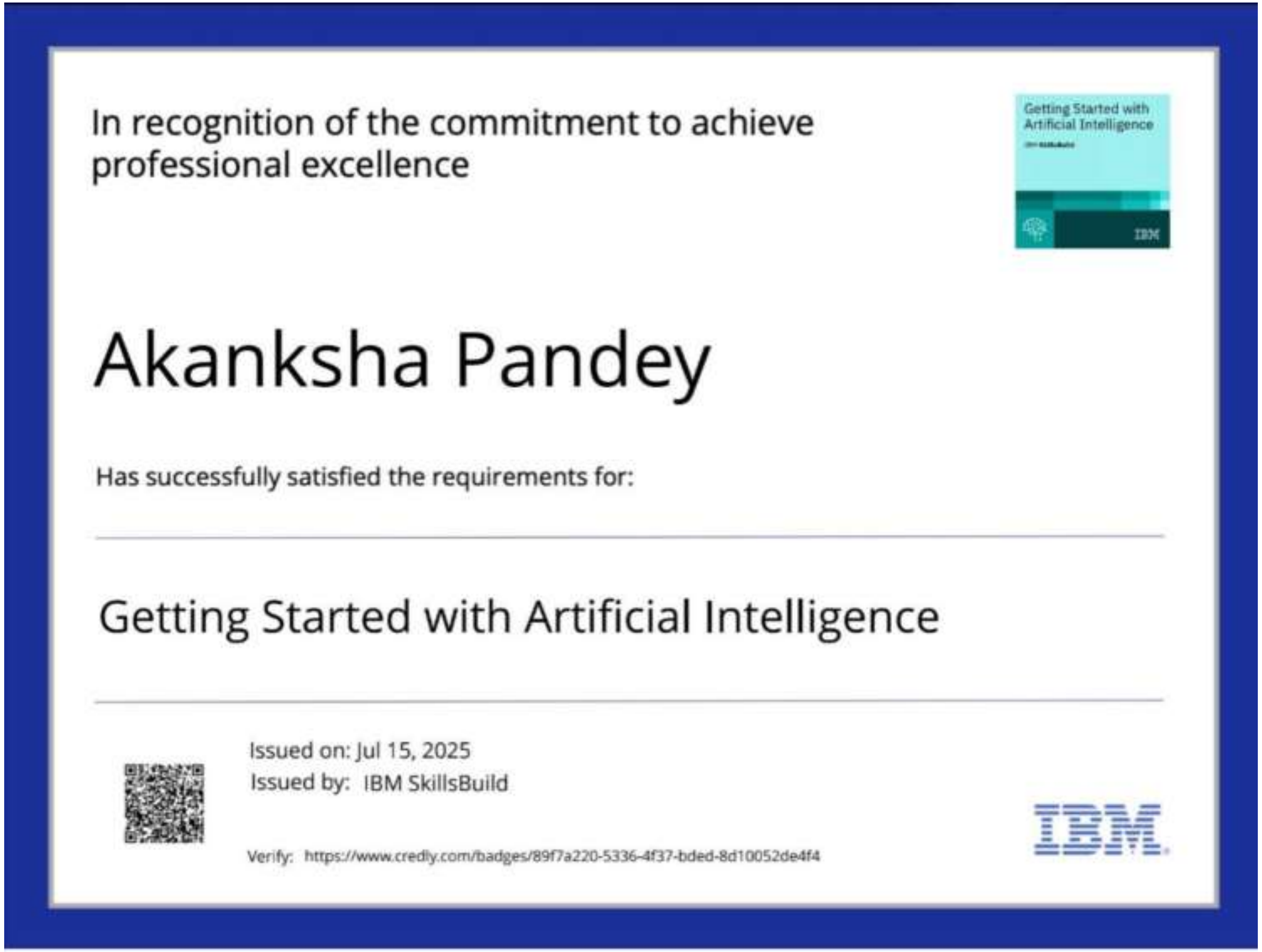
💾 **User Profiles + History→**

- ➢ Enable login, personalization, and saved research threads.
- ➢ Track progress, bookmarks, and export data.
- ➢ Provide tailored suggestions based on history.

□ **Integration with Learning Platforms→**

- o Turn the agent into a plugin or API for --
- ➢ University portals, LMS systems (like Moodle, Google Classroom),EdTech apps.

❑ IBM CERTIFICATIONS



Getting Started with
Artificial Intelligence
IBM

In recognition of the commitment to achieve professional excellence

Akanksha Pandey

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

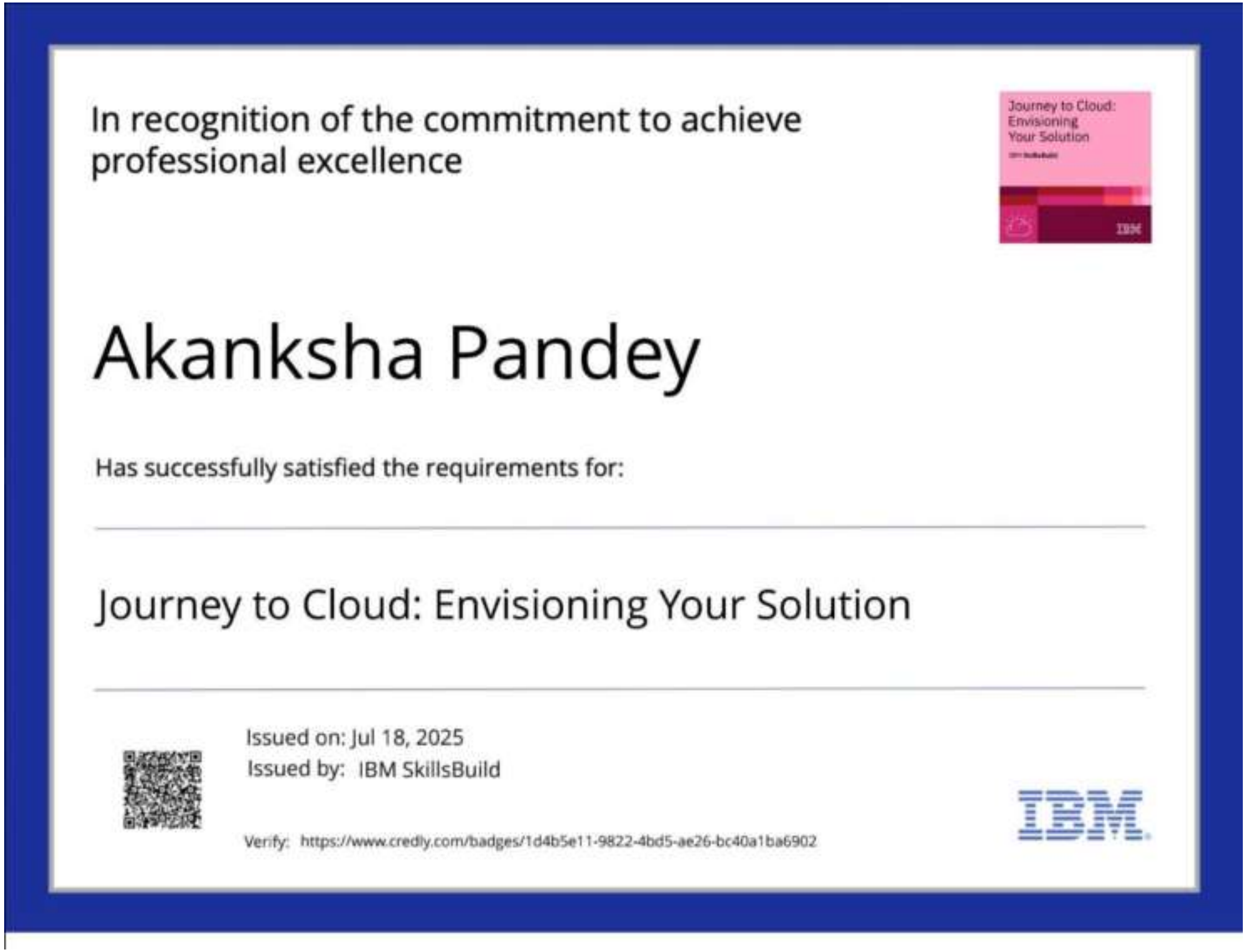
Issued on: Jul 15, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/89f7a220-5336-4f37-bded-8d10052de4f4

IBM

- ❖ Screenshot/ Credly certificate
- ➢ (Getting started with AI)

❑ IBM CERTIFICATIONS



Journey to Cloud: Envisioning Your Solution
IBM

In recognition of the commitment to achieve professional excellence

Akanksha Pandey

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

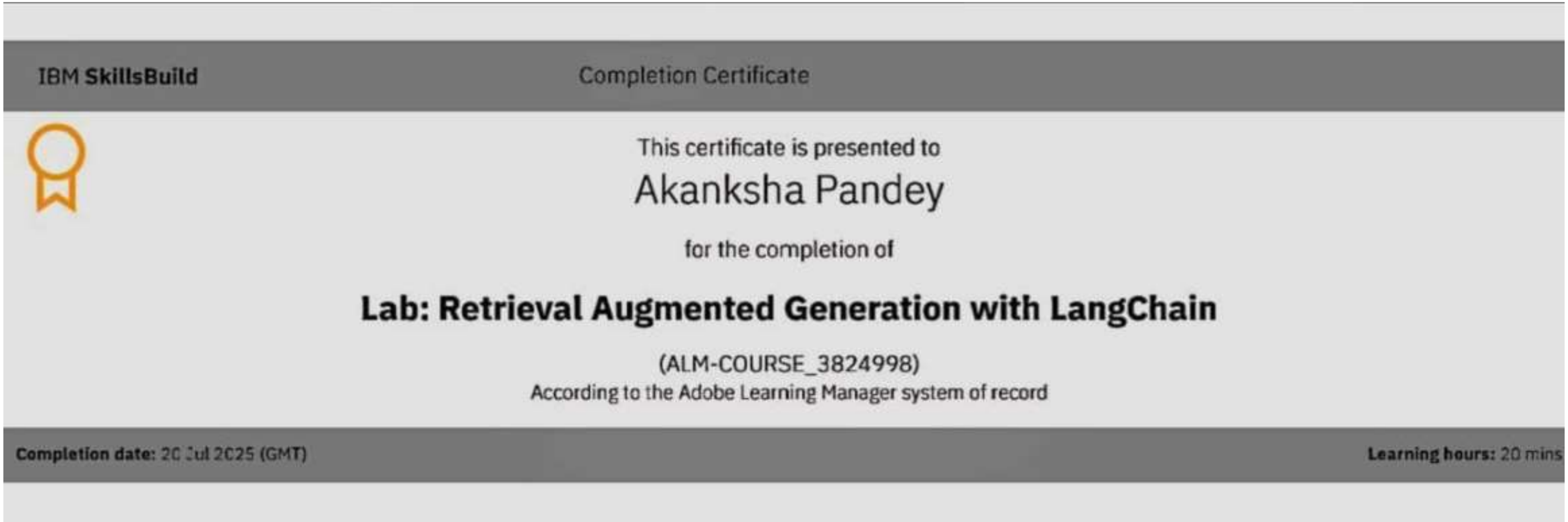
Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/1d4b5e11-9822-4bd5-ae26-bc40a1ba6902

IBM

- ❖ Screenshot/ Credly certificate
- ➢ (Journey to Cloud: Envisioning Your Solution)

❑ IBM Certifications



IBM SkillsBuild

Completion Certificate

This certificate is presented to

Akanksha Pandey

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

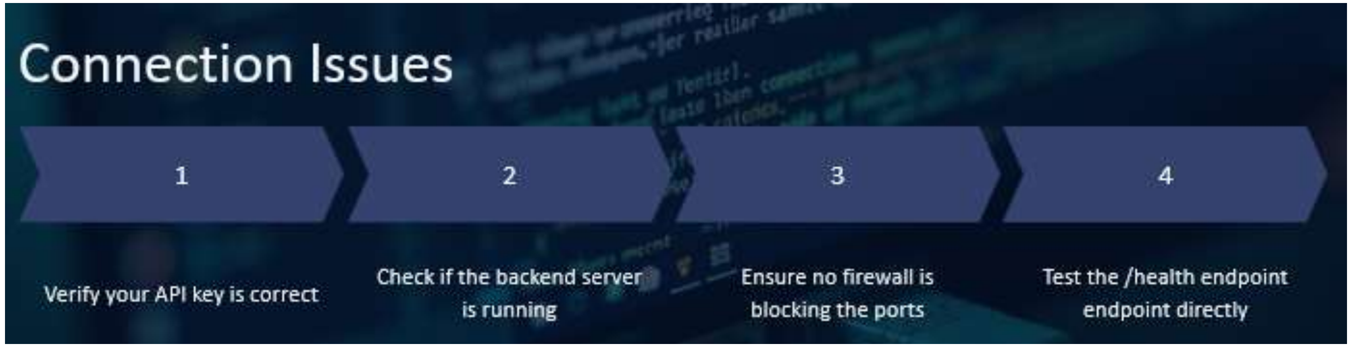
According to the Adobe Learning Manager system of record

Completion date: 20 Jul 2025 (GMT)

Learning hours: 20 mins

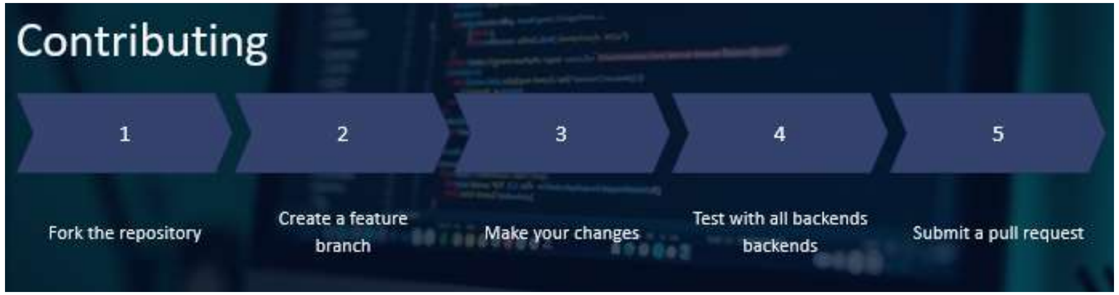
- Screenshot of certificate

- (Lab: Retrieval Augmented Generation with LangChain)



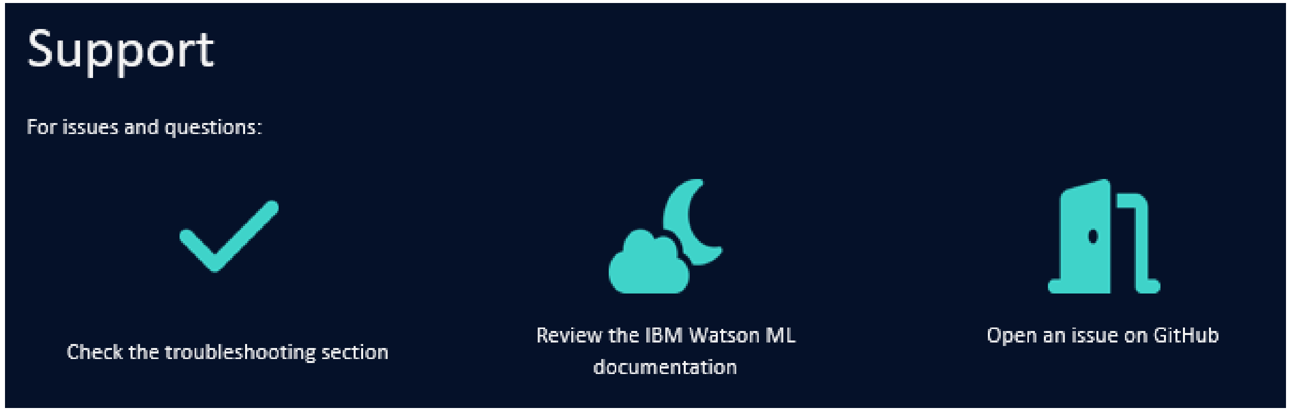
Connection Issues

1. Verify your API key is correct
2. Check if the backend server is running
3. Ensure no firewall is blocking the ports
4. Test the /health endpoint endpoint directly



Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Test with all backends backends
5. Submit a pull request



Support

For issues and questions:

- ✓ Check the troubleshooting section
- Review the IBM Watson ML documentation
- Open an issue on GitHub

THANK YOU!!