



IBM PROJECT



PAPERPILOT - RESEARCH AI AGENT

AI-Powered Research Assistant

Using IBM Cloud Lite & IBM Granite Foundation Model

Presented By : Amit Kumar Thakur

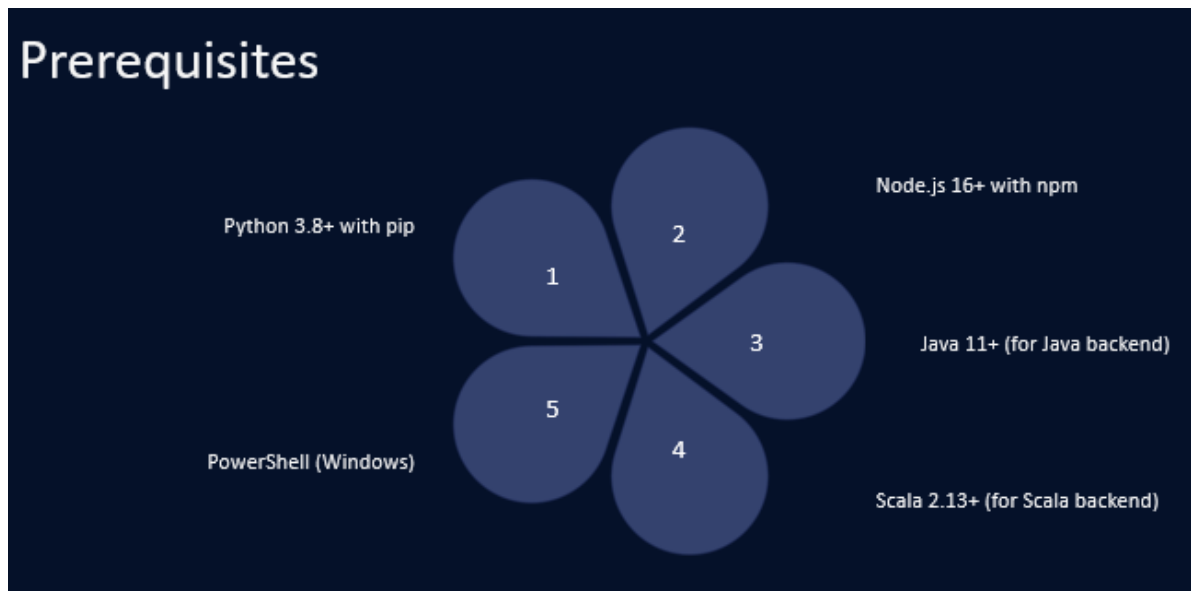
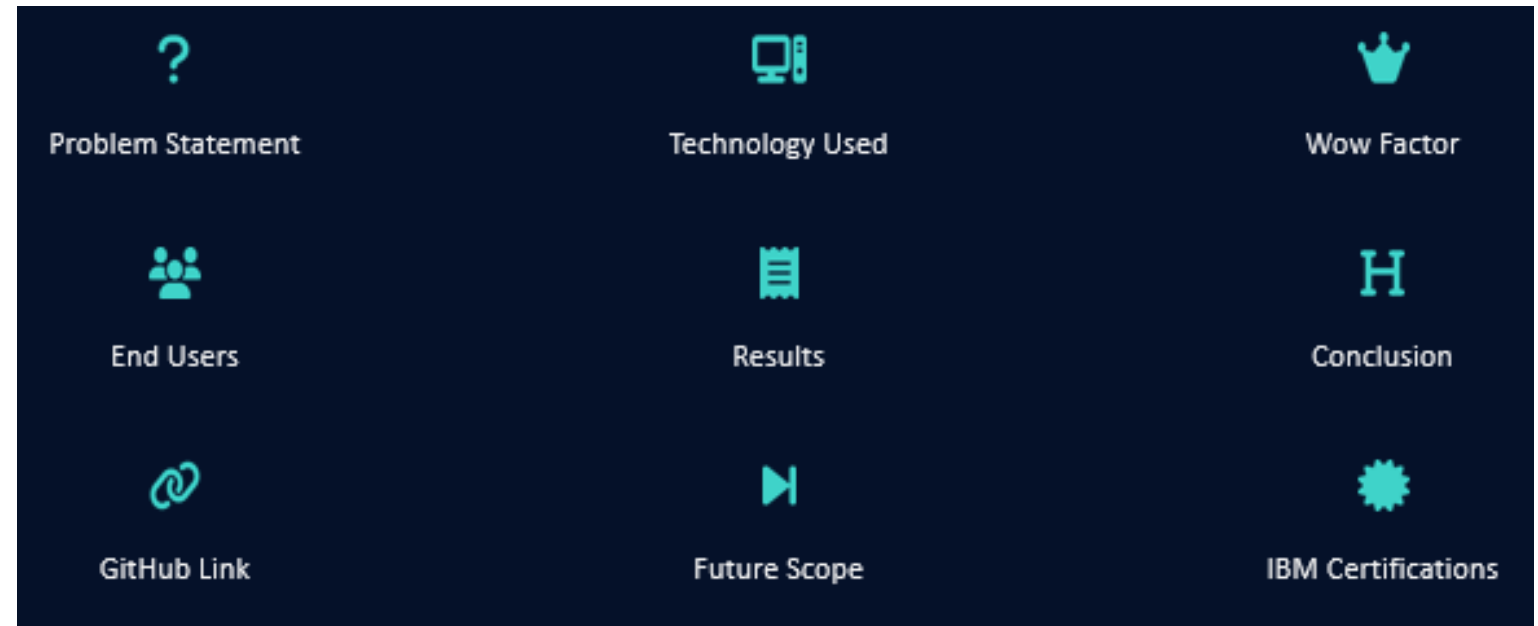
College Name : Vishveshwarya Group Of Institution

Department : Bachelor of Technology (B.TECH)

Course: Computer science and engineering

OUTLINE





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



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

💡 Our Solution

AlCademic – 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)

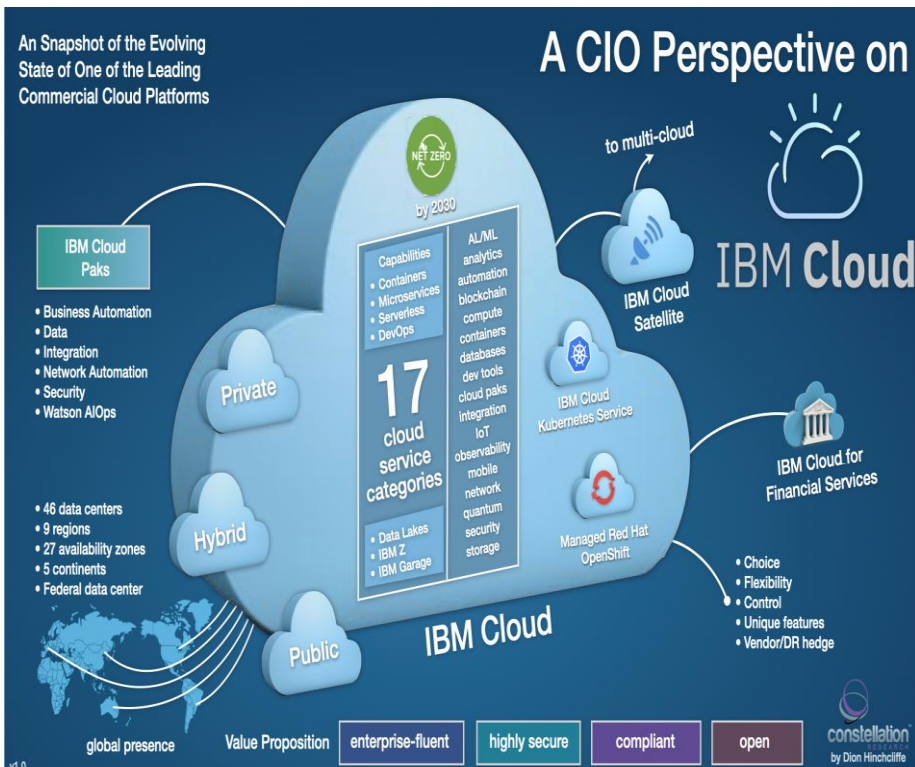
IBM Granite model



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

WOW 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

Academic Institutions & Libraries

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

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

Content Writers & EdTech Startups

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

Usage

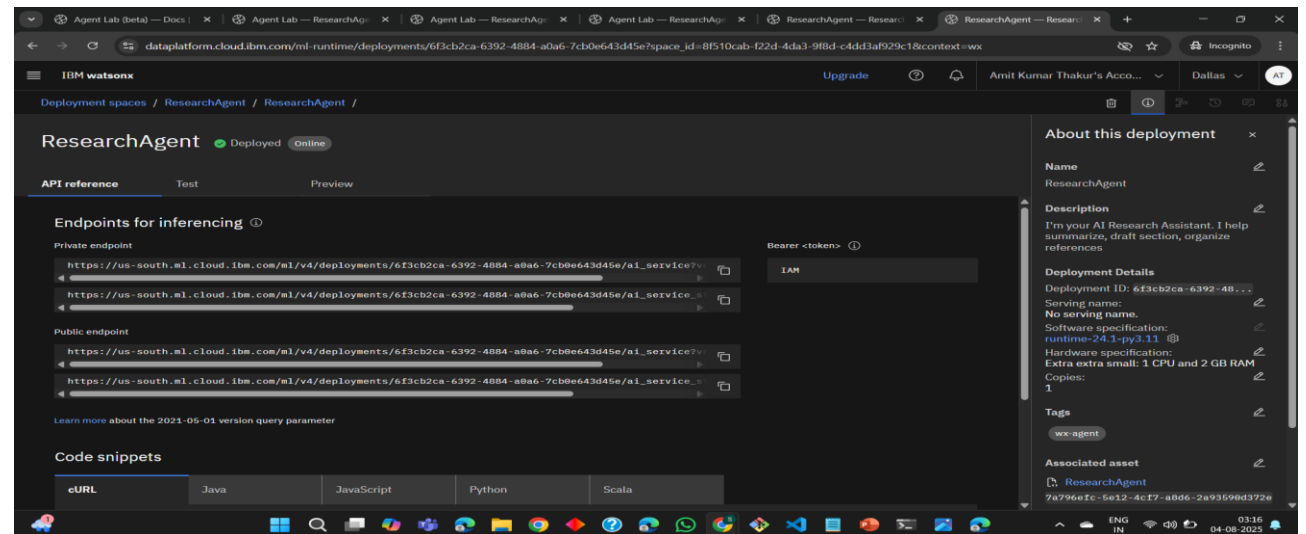
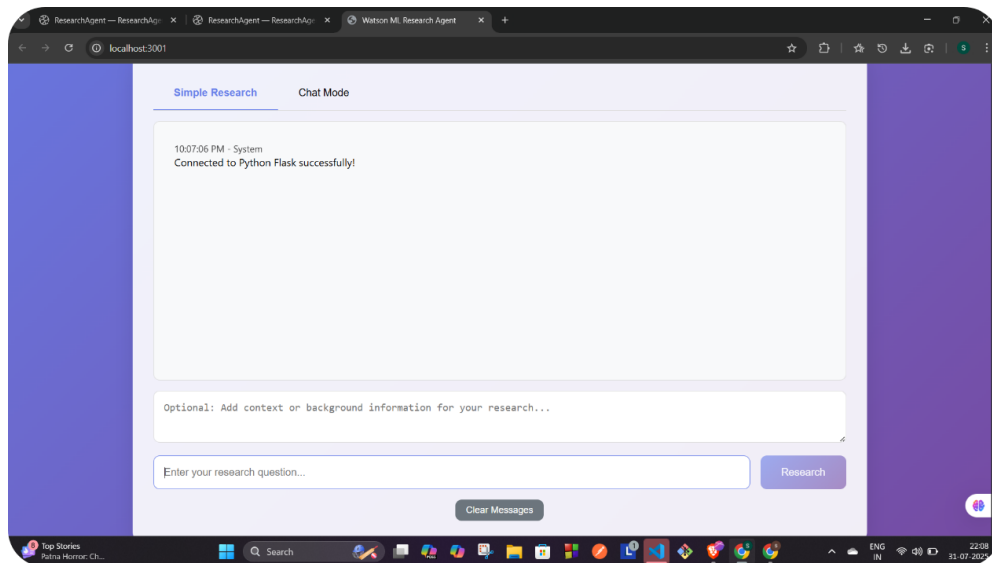
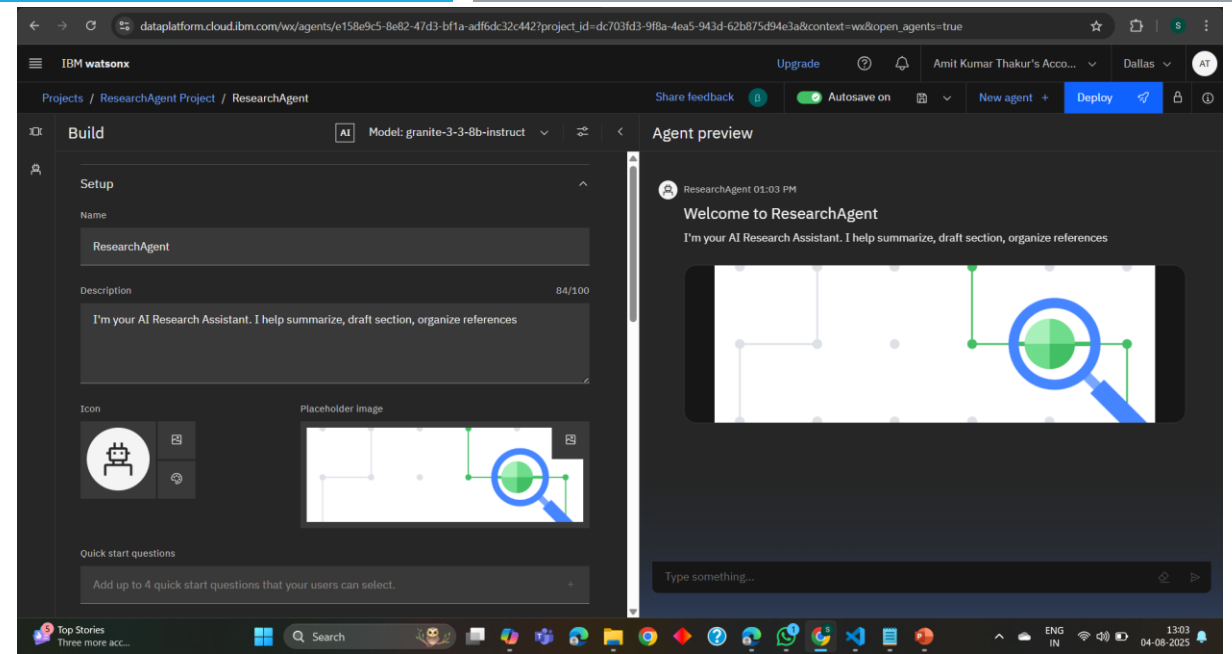
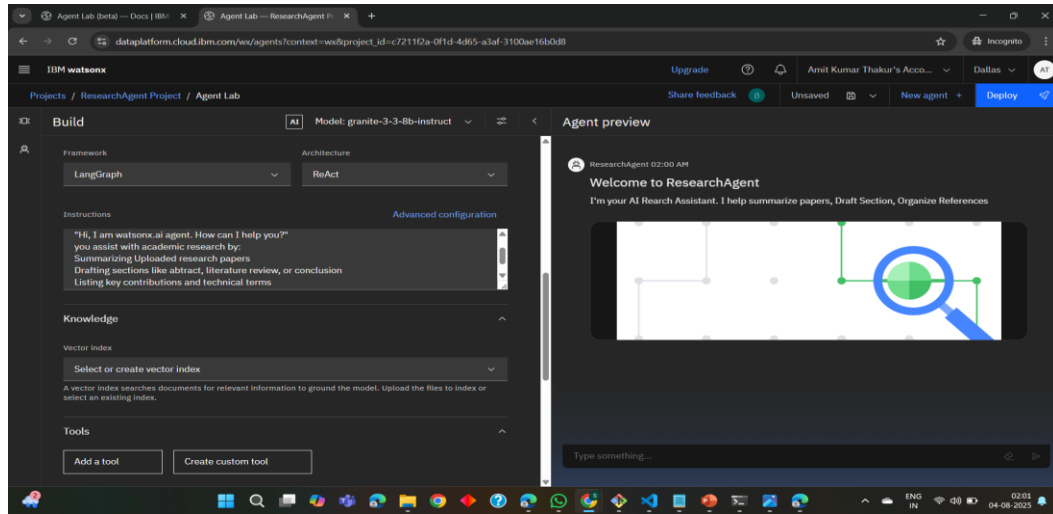
Simple Research Mode

- 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



RESULTS

The screenshot shows the IBM watsonx AI interface. The 'Build' tab is active, displaying instructions for the agent: "Hi, I am watsonx.ai agent. How can I help you?" and "you assist with academic research by: Summarizing uploaded research papers, Drafting sections like abstract, literature review, or conclusion, Listing key contributions and technical terms". The 'Knowledge' section shows a vector index. The 'Tools' section includes 'Google search'. The 'Agent preview' tab shows a chat history with a user query "List references related to LSTM in NLP." and a response from the ResearchAgent listing 10 references related to LSTM in NLP.

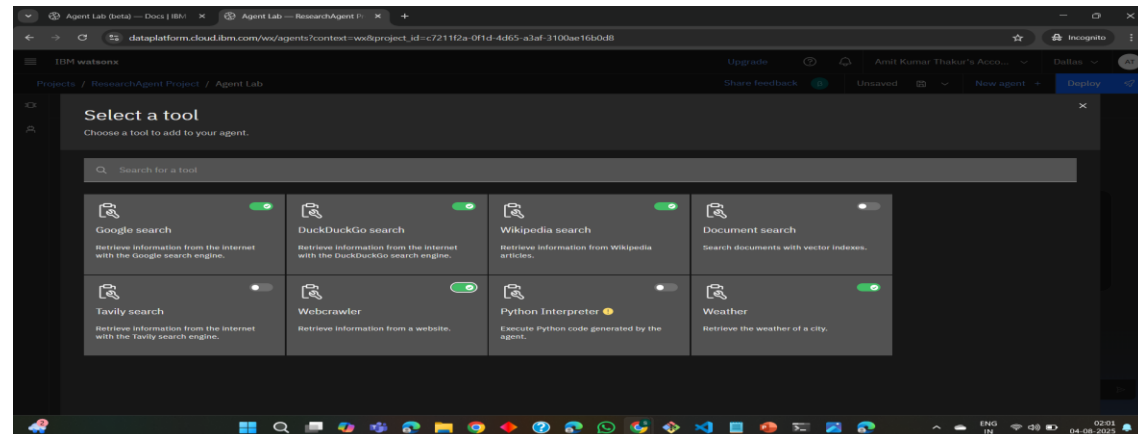
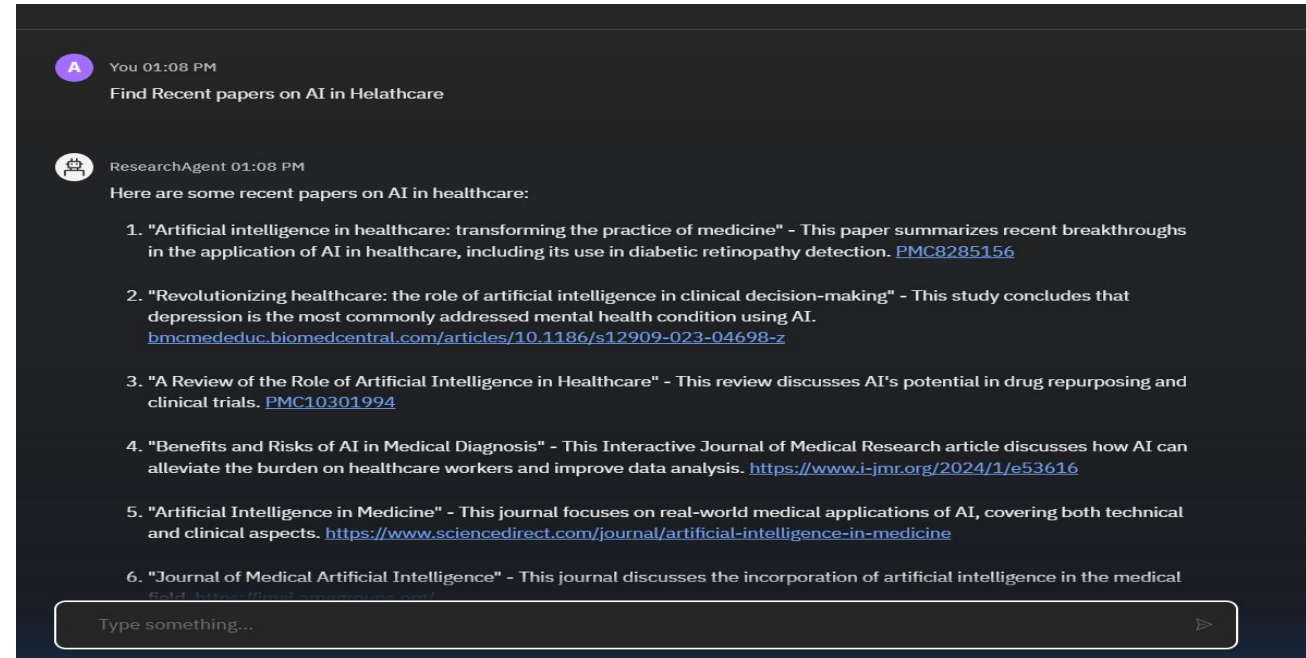
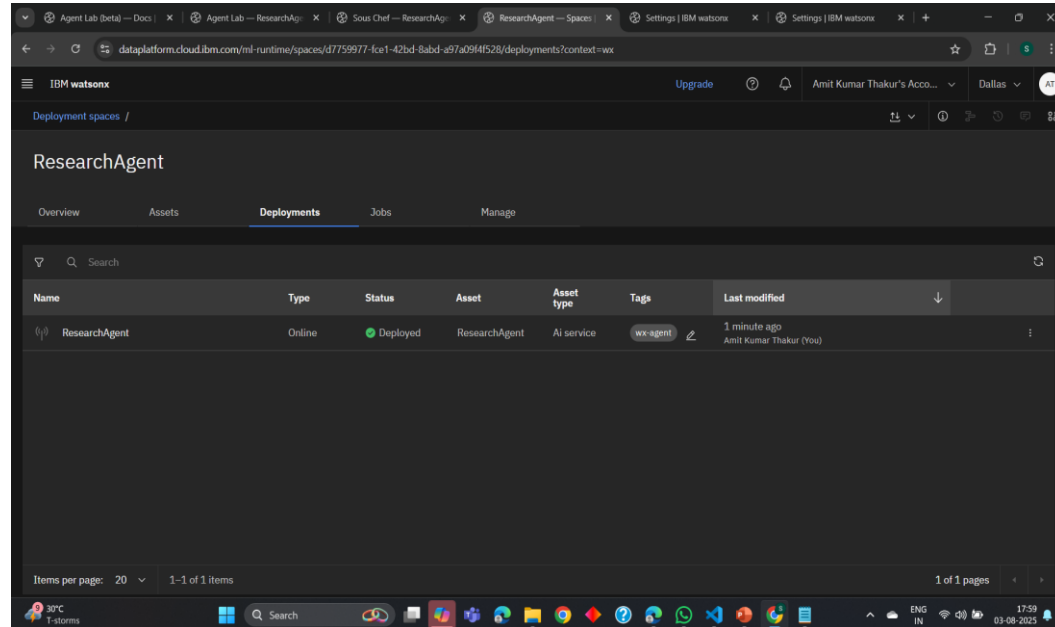
The screenshot shows the IBM watsonx AI interface. The 'Preview' tab is active, displaying a chat history with a user query "Find recent papers on AI in Healthcare" and a response from the ResearchAgent listing 3 references related to AI in Healthcare. The references include "Artificial Intelligence (AI) in Healthcare & Medical Field", "AI in Healthcare | Coursera", and "Artificial intelligence in healthcare: transforming the practice of medicine".

The screenshot shows the Watson ML Research Agent web application interface. The 'Choose Backend Implementation' section displays four options: Python Flask (Port 3000 - Fast and reliable Python backend), Node.js Express (Port 3001 - JavaScript-based backend), Java HTTP Server (Port 3002 - Enterprise Java backend), and Scala HTTP Server (Port 3003 - Functional programming backend). The 'Status' is 'Connected'. The 'Simple Research' tab is active, showing a message: "10:07:06 PM - System Connected to Python Flask successfully!".

The screenshot shows the Watson ML Research Agent web application interface. The 'Choose Backend Implementation' section displays four options: Python Flask (Port 3000 - Fast and reliable Python backend), Node.js Express (Port 3001 - JavaScript-based backend), Java HTTP Server (Port 3002 - Enterprise Java backend), and Scala HTTP Server (Port 3003 - Functional programming backend). The 'Status' is 'Connected'. The 'Simple Research' tab is active, showing a message: "10:50:35 PM - System Connected to Node.js Express successfully!".

RESULTS

Deployed AI Agent



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:  [AI_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

1. 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

2. PDF Upload + Summarization

Let users upload full research papers and receive:

- Key takeaways
- Summaries per section
- Highlight extraction with RAG

3. Multilingual Support

Expand usability across regional and global languages using:

- IBM Watson Language Translator
- Granite's multilingual understanding

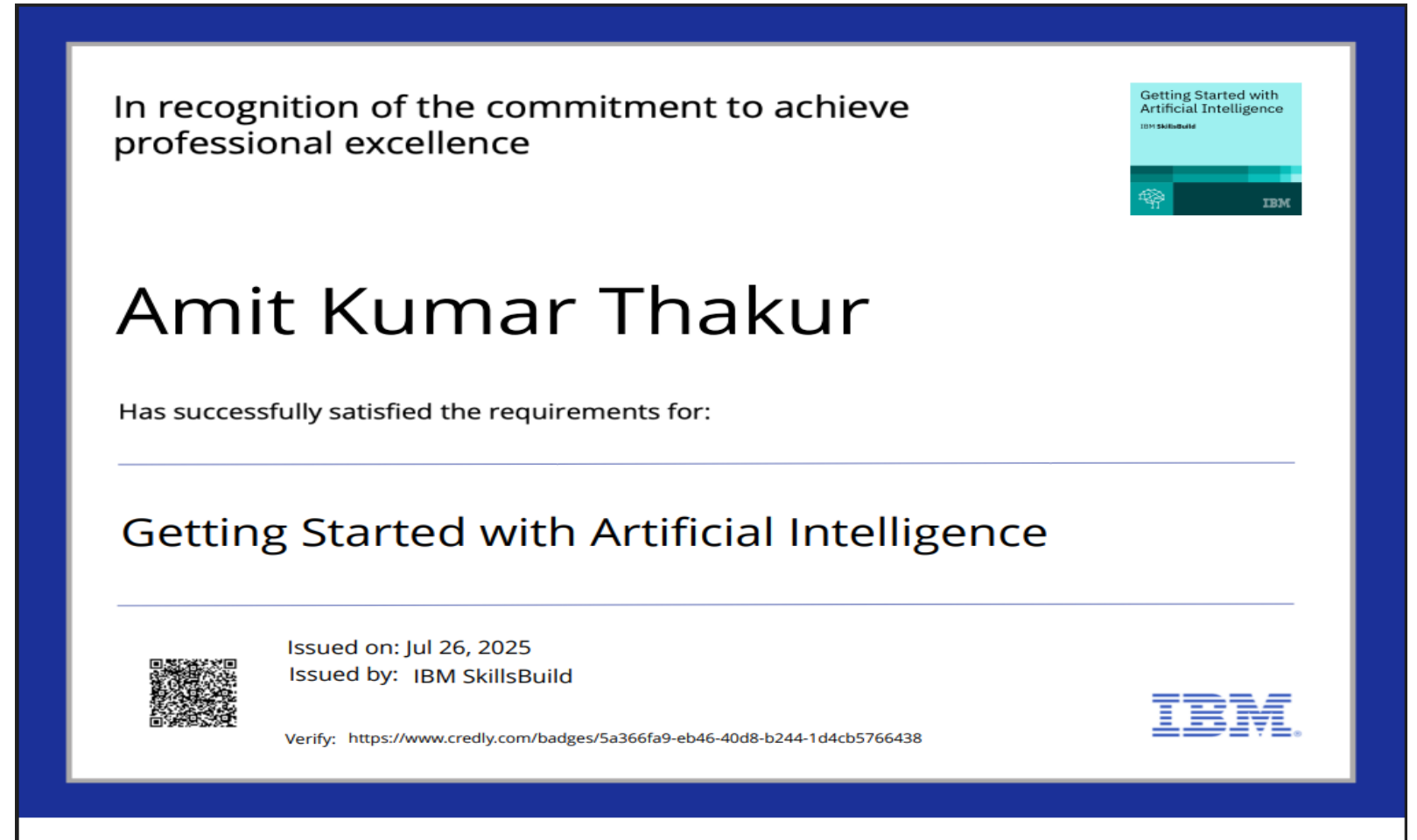
4. User Profiles + History

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

5. Integration with Learning Platforms

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

IBM CERTIFICATIONS



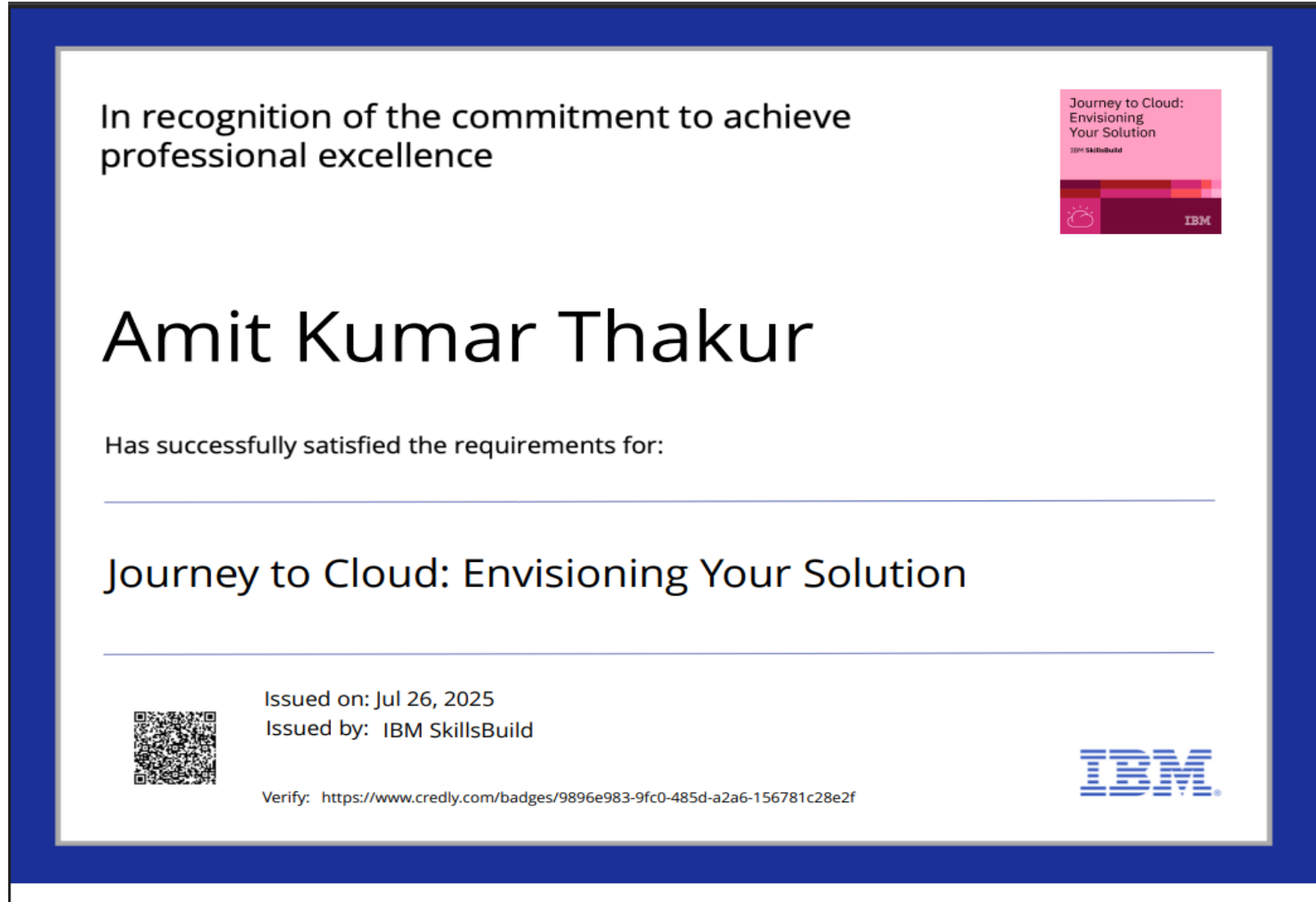
- Screenshot/ credly certificate
(Getting started with AI)



IBM CERTIFICATIONS

- Screenshot/ credly certificate

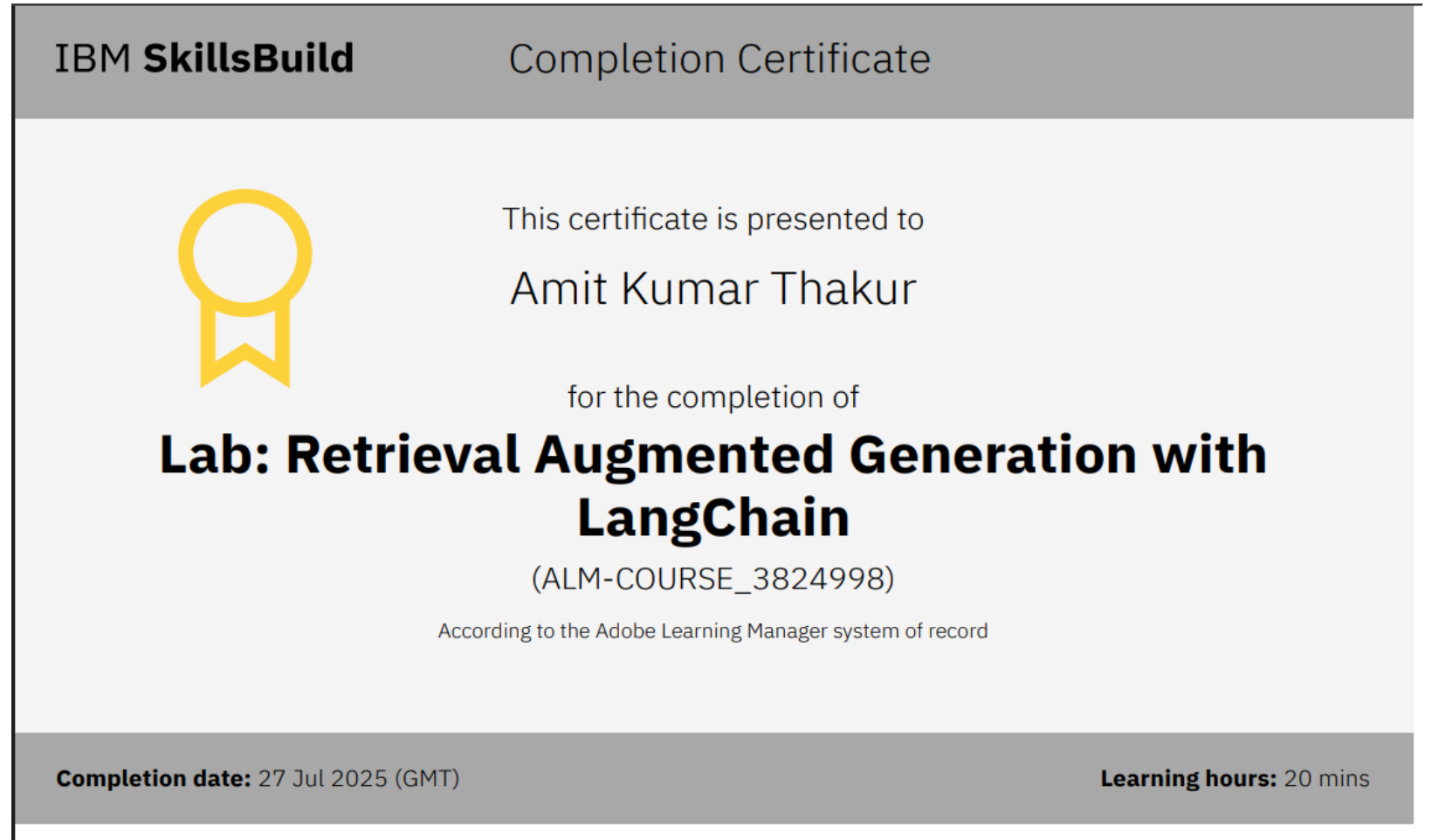
(Journey to Cloud: Envisioning Your Solution)



IBM Certifications

- Screenshot of certificate

(Lab: Retrieval Augmented Generation with LangChain)



Connection Issues

1

Verify your API
key is correct

2

Check if backend
server is running

3

Ensure no firewall
is blocking the
ports

Support

For issues and questions:



Check the troubleshooting
section



Review the IBM Watson ML
documentation



Open an issue on SHL



Open an issue on GitHub

Contributing

1

Fork the repository

2

Create feature
branch

3

Make your
changes

4

Submit a
pull request

THANK YOU