

CAPSTONE PROJECT

Library AI Agent

Presented By:

1. **Student Name : Saketh Ram Sandepudi**
2. **Collage Name : Amrita Vishwa Vidyapeetam , Amaravati**
3. **Branch : CSE**

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Students often face the challenge of sifting through vast library collections to find learning materials that are most relevant to their specific academic needs. The traditional search process can be time-consuming and inefficient, making it difficult to pinpoint the right books and resources aligned with their current study topics or course syllabi. There is a need for an intelligent system that can streamline this search process and provide personalized, timely recommendations to enhance resource utilization and student engagement in educational environments.

PROPOSED SOLUTION

- The proposed system is an intelligent Library AI Agent designed to assist students in finding the right learning materials efficiently. The system will analyze user profiles and academic requirements to suggest relevant books and resources.
- **Data Collection:**
 - Gather data from student profiles, including their courses and study topics.
 - Analyze course syllabi to understand academic requirements.
 - Utilize the library's database for books, resources, and real-time availability.

Data Preprocessing:

- Clean and structure the collected data from the library database and course syllabi to handle any inconsistencies.
- Perform feature engineering to extract key topics and concepts that are relevant to academic queries.

Machine Learning Algorithm:

- Implement a natural language processing model using **IBM Granite** to understand and interpret student queries.
- Use a recommendation algorithm to match student needs with the most suitable books and resources in the library database.

Deployment:

- Develop a user-friendly interface, such as a chatbot, that provides real-time assistance to students
- Deploy the agent on a scalable and reliable platform using **IBM Cloud Lite services**.

PROPOSED SOLUTION

■ Evaluation:

- Assess the agent's performance by measuring user satisfaction, the accuracy of recommendations, and the reduction in search time for students.
- Continuously fine-tune the model based on user feedback and interaction data.

■ Result:

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a user profile 'Saketh Ram Sandepudi's A...', location 'Dallas', and a session ID 'SS'. Below the navigation bar, the breadcrumb trail shows 'Projects / Library_AI_Agent / Agent Lab'. The main interface is divided into two panels. The left panel, titled 'Build', contains sections for 'Setup', 'Configuration', 'Instructions', 'Knowledge', and 'Tools'. Under 'Configuration', the 'Framework' is set to 'LangGraph' and the 'Architecture' is 'ReAct'. The 'Instructions' section contains a text box with the role definition: 'You are "LibrarianAI", an intelligent and dedicated AI library assistant. Your primary purpose is to connect students and researchers with the perfect learning materials from the library's collection.' Below this, there are buttons for 'Add a tool' and 'Create custom tool'. The 'Tools' section shows 'Added tools (1)' with a 'Google search' tool listed. The right panel, titled 'Agent preview', shows a chat log. The user's message is 'I want to learn deep learning , suggest me some resources'. The agent's response is a list of seven resources for learning deep learning, including books, GitHub repositories, AI resource centers, online courses, and textbooks. At the bottom of the chat log, there is a text input field with the placeholder 'Type something...' and a 'Send' button.

SYSTEM APPROACH

- **System Requirements:** A cloud-based platform to host the AI agent and a web-based interface for students to interact with the system.
- **Library required to build the model :**
 - **Cloud Platform:** IBM Cloud Lite services
 - **AI / Language Model:** IBM Granite
 - **Programming Language:** Python
 - **Frameworks:** A web framework such as Flask or Django for the user interface.
 - **Database:** A database system (e.g., PostgreSQL, MongoDB) to manage library records, user profiles, and course data.

ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting bike counts. Here's an example structure for this section:
- **Algorithm Selection:**
 - The core of the agent will be **IBM Granite**, chosen for its powerful Natural Language Processing (NLP) capabilities. This allows the system to accurately understand student queries in natural language.
 - A **Content-Based Filtering** algorithm will be used for recommendations. This approach will analyze the content of library resources (like topics, keywords, and summaries) and match them to the keywords extracted from a student's query and course syllabus.
- **Data Input:**
 - The algorithm will use the student's typed query, their academic profile (enrolled courses), course syllabi, and the library's resource database as input.
- **Training Process:**
 - The IBM Granite model will be leveraged to parse student queries. The recommendation algorithm will be trained to map keywords and topics from these queries to the indexed content of the library's database
- **Prediction Process:**
 - The agent will take a student's real-time query, understand its intent using IBM Granite, and then use the recommendation algorithm to search the database for relevant materials. It will also check for real-time book availability and assist with reservation or waitlist actions.

RESULT

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a help icon, a notification bell, the user's name 'Saketh Ram Sandepudi's A...', the location 'Dallas', and a profile icon. Below the navigation bar, the breadcrumb trail shows 'Projects / Library_AI_Agent / Agent Lab'. The main interface is divided into two panels: 'Build' on the left and 'Agent preview' on the right.

Build Panel:

- Model:** llama-3-3-70b-instruct
- Framework:** LangGraph
- Architecture:** ReAct
- Instructions:** You are 'LibrarianAI', an intelligent and dedicated AI library assistant. Your primary purpose is to connect students and researchers with the perfect learning materials from the library's collection. When a user submits a query for resources, you MUST ask for the following details if they are...
[Advanced configuration](#)
- Knowledge:**
- Tools:**
 - [Add a tool](#)
 - [Create custom tool](#)
- Added tools (1):**
 - Google search**
Retrieve information from the Internet with the Google search engine.

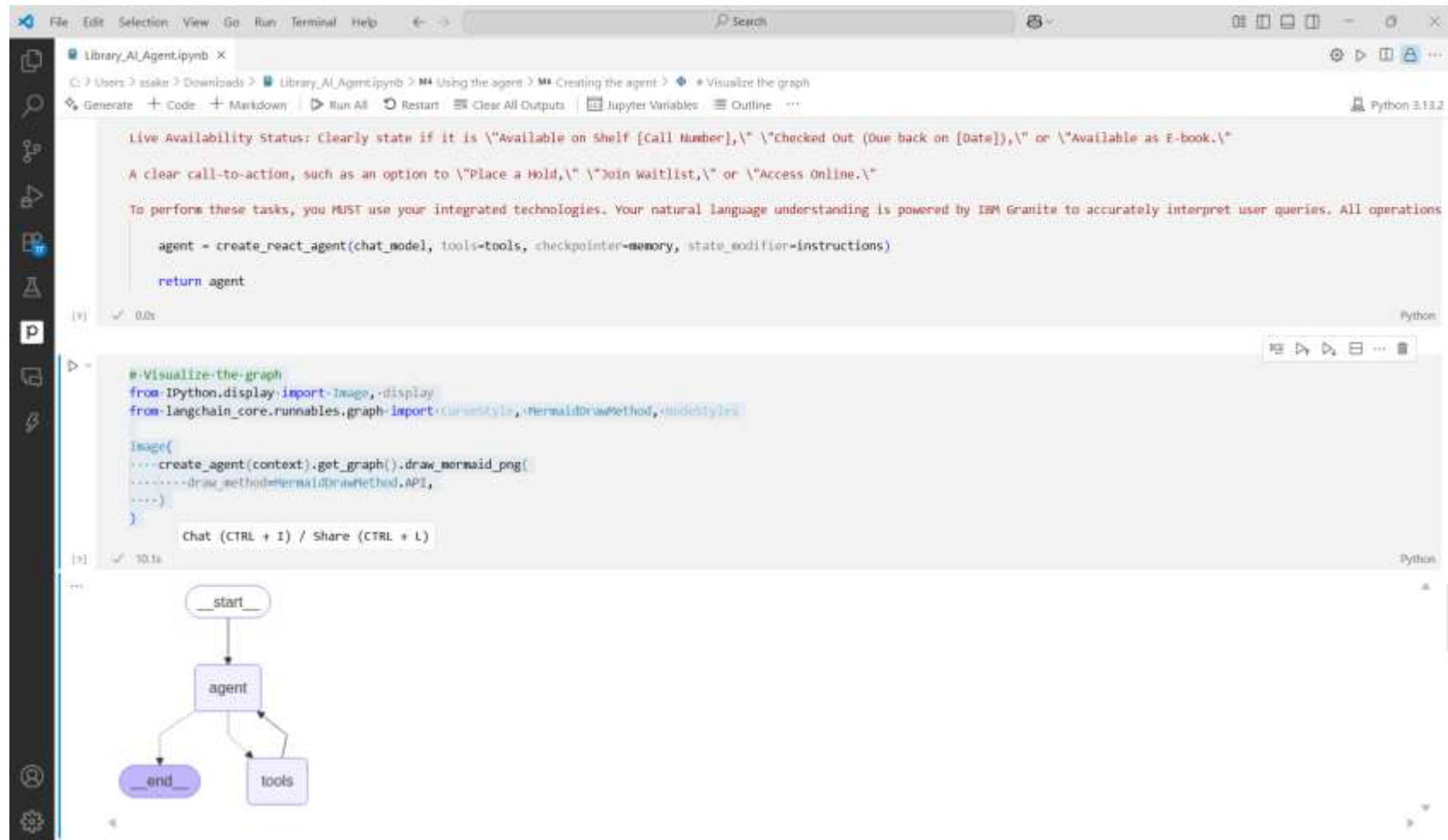
Agent preview Panel:

- User Query:** I want to learn deep learning , suggest me some resources
- Agent Response:** Based on the search results, here are some resources to learn deep learning:
 1. The DeepLearning book (<https://www.deeplearningbook.org/>) - a general overview of deep learning
 2. Awesome Deep Learning (<https://github.com/ChristosChristofidis/awesome-deep-learning>) - a curated list of deep learning resources
 3. DeepLearning.AI (<https://www.deeplearning.ai/resources/>) - a resource center for AI practitioners
 4. Coursera, Udacity, MIT, Google AI, Distill (<https://www.quora.com/What-are-the-best-resources-to-learn-about-deep-learning>) - online courses and tutorials
 5. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (<https://www.deeplearningbook.org/>) - a textbook on deep learning
 6. AI Hub from Google (<https://medium.com/data-science/top-7-machine-learning-resources-i-wish-i-knew-earlier-89f4b8d3364c>) - a platform for datasets, pretrained models, and more
 7. Machine Learning Crash Course (<https://developers.google.com/machine-learning/crash-course>) - a self-contained course on machine learning

These resources provide a good starting point for learning deep learning, including textbooks, online courses, and platforms for datasets and models.

► How did I get this answer?

RESULT



The screenshot displays a Jupyter Notebook window titled 'Library_AI_Agent.ipynb'. The interface includes a top menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help), a search bar, and a toolbar with icons for file operations and execution. The notebook content is divided into two cells. The first cell contains a block of text providing instructions for the agent's behavior, such as 'Live Availability Status: Clearly state if it is "Available on Shelf [Call Number], " "Checked Out (Due back on [Date]), " or "Available as E-book."' and a code snippet for creating the agent:

```
agent = create_react_agent(chat_model, tools=tools, checkpoint=memory, state_modifier=instructions)
return agent
```

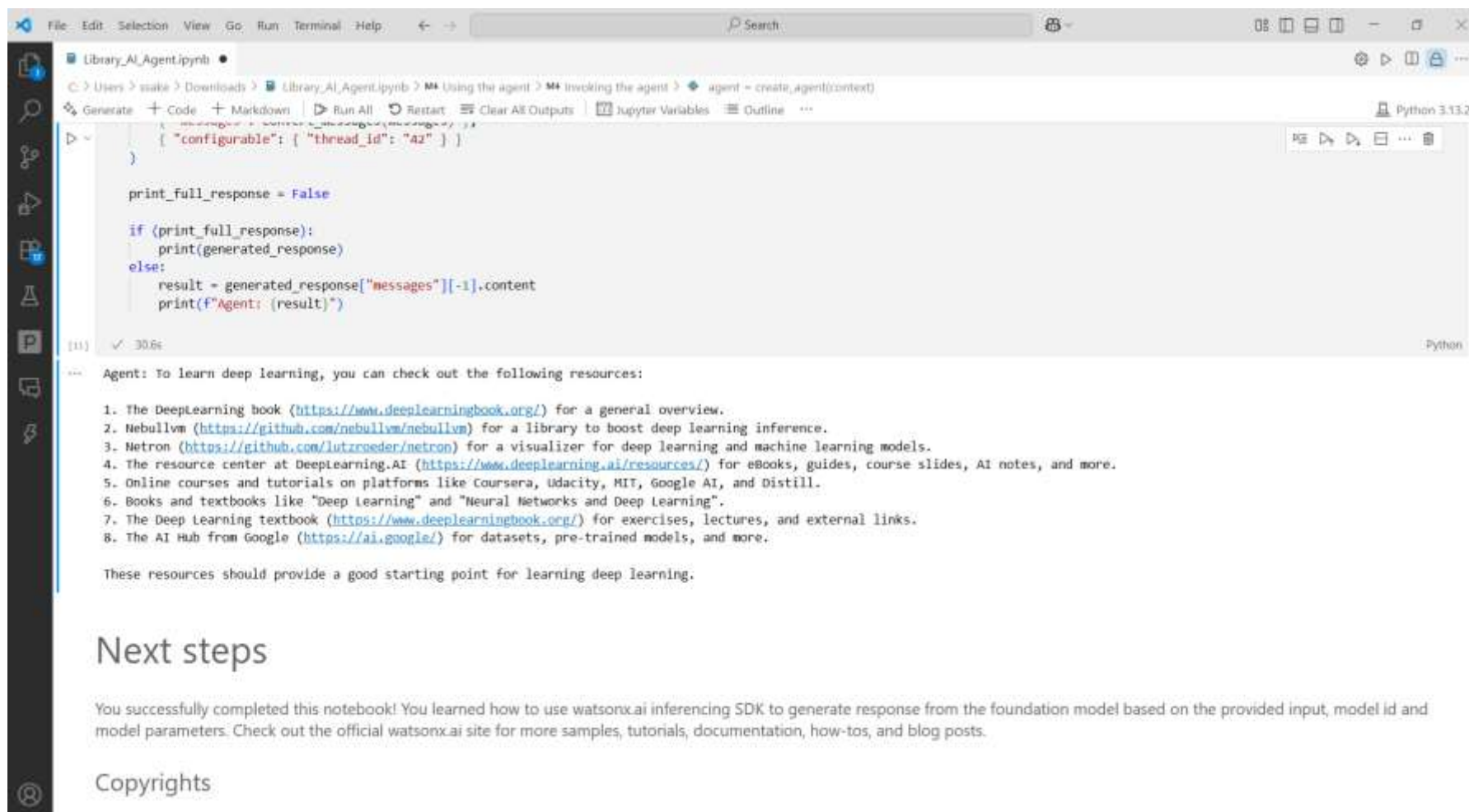
. The second cell is titled '# Visualize the graph' and contains code for importing necessary modules and drawing the agent's graph:

```
from IPython.display import Image, display
from langchain_core.runnables.graph import CurvedStyle, MermaidDrawMethod, NodeStyle
Image(
    create_agent(context).get_graph().draw_mermaid_png(
        draw_method=MermaidDrawMethod.API,
    )
)
```

. Below the code, a chat input field is visible with the text 'Chat (CTRL + I) / Share (CTRL + L)'. At the bottom of the notebook, a graph visualization is shown, starting with a 'start' node, leading to an 'agent' node, which then branches into an 'end' node and a 'tools' node.

```
graph TD
    start([start]) --> agent[agent]
    agent --> end([end])
    agent --> tools[tools]
```

RESULT



```
File Edit Selection View Go Run Terminal Help
Library_AI_Agent.ipynb
C:\Users> ssake > Downloads > Library_AI_Agent.ipynb > M4 Using the agent > M4 Invoking the agent > agent = create_agent(context)
Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline Python 3.13.2
{ "configurable": { "thread_id": "A2" } }

print_full_response = False

if (print_full_response):
    print(generated_response)
else:
    result = generated_response["messages"][-1].content
    print(f"Agent: {result}")

[11] ✓ 30.6s Python

Agent: To learn deep learning, you can check out the following resources:

1. The DeepLearning book (https://www.deeplearningbook.org/) for a general overview.
2. Nebullvm (https://github.com/nebullvm/nebullvm) for a library to boost deep learning inference.
3. Netron (https://github.com/lutzroeder/netron) for a visualizer for deep learning and machine learning models.
4. The resource center at DeepLearning.AI (https://www.deeplearning.ai/resources/) for eBooks, guides, course slides, AI notes, and more.
5. Online courses and tutorials on platforms like Coursera, Udacity, MIT, Google AI, and Distill.
6. Books and textbooks like "Deep learning" and "Neural Networks and Deep Learning".
7. The Deep Learning textbook (https://www.deeplearningbook.org/) for exercises, lectures, and external links.
8. The AI Hub from Google (https://ai.google/) for datasets, pre-trained models, and more.

These resources should provide a good starting point for learning deep learning.
```

Next steps

You successfully completed this notebook! You learned how to use watsonx.ai inferencing SDK to generate response from the foundation model based on the provided input, model id and model parameters. Check out the official watsonx.ai site for more samples, tutorials, documentation, how-tos, and blog posts.

Copyrights

CONCLUSION

- The Library AI Agent successfully addresses the challenge of inefficient resource discovery in an academic library. By using IBM Granite, the agent offers a personalized and streamlined search process, saving students valuable time. The system enhances resource utilization and student engagement by providing recommendations that are directly aligned with their current academic work. The implementation demonstrates the significant value of AI in modernizing educational environments.
- GitHub link : https://github.com/Sakethram14/Library_AI_Agent

REFERENCES

- **Chen, L., Chen, G., & Wang, F. (2020).** "A review of the research and applications of artificial intelligence in library and information science." *Journal of Academic Librarianship*, 46(5), 102213.
- **Ricci, F., Rokach, L., & Shapira, B. (2022).** *Recommender Systems Handbook*. Springer.
- **IBM. (2023).** "Granite: A Family of Decoder-Only Foundation Models for Business." *IBM Research Blog*.

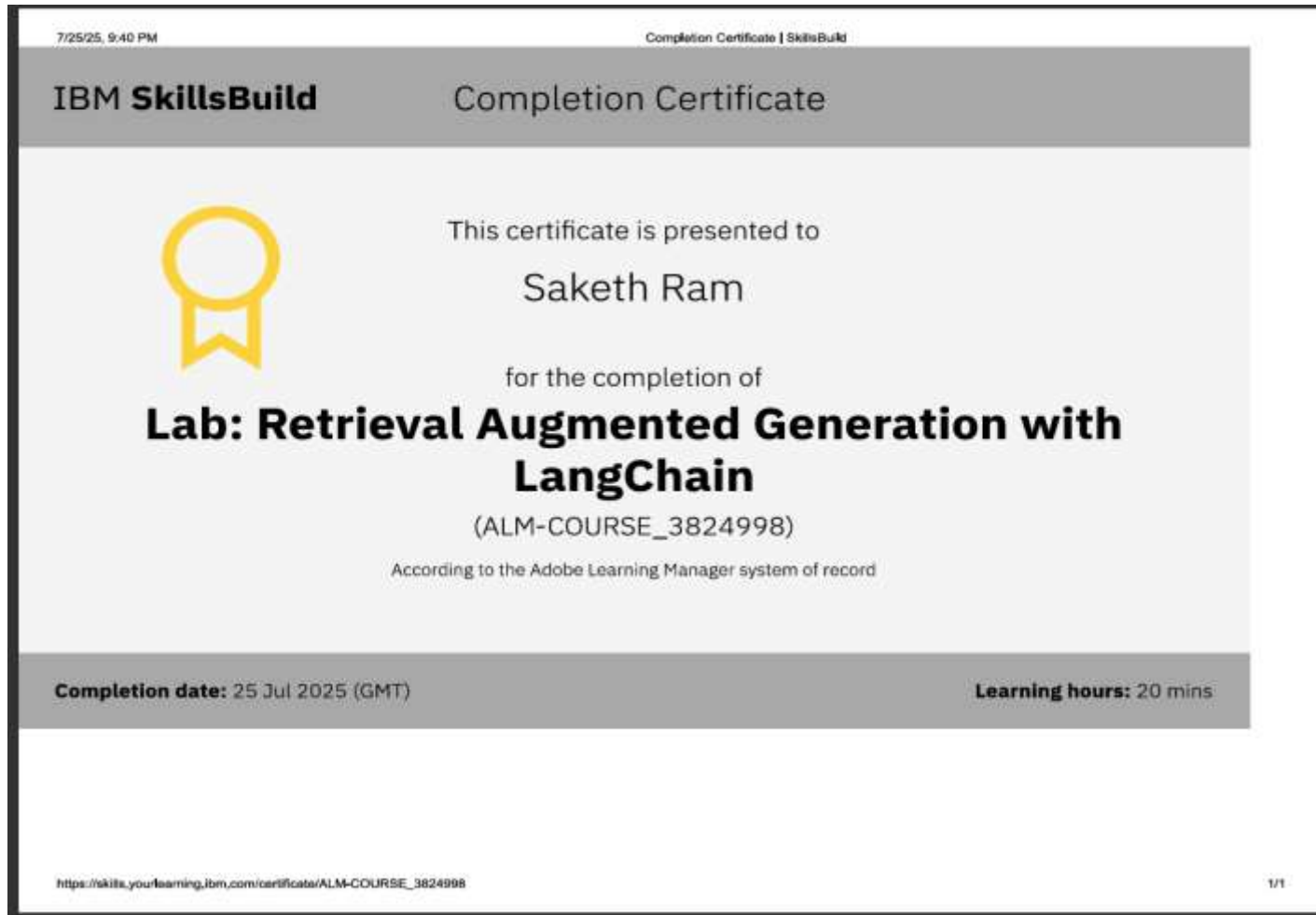
IBM CERTIFICATIONS



IBM CERTIFICATIONS



IBM CERTIFICATIONS





THANK YOU