CAPSTONE PROJECT

AI-DRIVEN PLAGIARISM FOR ASSIGNMENTS

Presented By:
Sai Karthikeya- Geethanjali Institute of Science and Technology-Computer Science and Technology

OUTLINE

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

AI-DRIVEN PLAGIARISM FOR ASSIGNMENTS

Academic institutions face increasing difficulty in detecting nuanced forms of plagiarism, especially when assignments are paraphrased or generated by AI tools. Current plagiarism detectors lack contextual sensitivity to instructor-specific styles and grading patterns. The challenge lies in creating an adaptive AI system that learns from historical assignment submissions and instructor feedback to identify inconsistencies and potential misconduct dynamically. This would enhance academic integrity by flagging suspicious entries with improved accuracy and contextual awareness..

PROPOSED SOLUTION

- The proposed system aims to address the challenge of detecting and preventing plagiarism in academic assignments using artificial intelligence. This involves leveraging data analytics and large language models to identify similarities between student submissions and existing sources. The solution will consist of the following components:
- Data Collection:
  Gather historical data of academic assignments, including previously submitted student work, public datasets, and institutional repositories. Utilize real-time data sources such as online articles, academic papers, and open educational content to enhance detection capabilities.
- Data Preprocessing:
  Clean and preprocess the collected data to handle formatting issues, remove irrelevant content (e.g., HTML tags or code), and standardize the text format. Apply techniques such as tokenization, lemmatization, and stopword removal. Perform feature engineering to generate embeddings or similarity vectors that can capture semantic meaning across texts.
- Machine Learning Algorithm:
  Implement a natural language processing model or similarity detection system using large language models (e.g., IBM Watsonx.ai Granite model or sentence transformers). Use cosine similarity or other distance metrics to compare new assignment content with existing data. Incorporate thresholds to classify the level of plagiarism (e.g., high, medium, low). Consider using clustering or classification algorithms to improve prediction and categorization accuracy.
- Deployment:
  Develop a user-friendly web interface that allows users (students or faculty) to upload assignment files and receive plagiarism analysis. Deploy the solution on a scalable and reliable platform such as IBM Cloud. Ensure secure file storage, fast response time, and accessibility through browser or institutional systems.
- Evaluation:
  Assess the model's performance using appropriate metrics such as Precision, Recall, F1-score, and Confusion Matrix. Evaluate how accurately the system identifies plagiarism versus original content. Continuously monitor and fine-tune the model based on new data, user feedback, and academic standards.
- Result:
  A fully functional AI-based plagiarism detection system that helps ensure academic integrity by identifying copied or paraphrased content intelligently, reducing manual review effort, and providing transparent, data-backed insights for educators and institutions.

SYSTEM APPROACH

- The "System requirements
  - Operating System: Windows, Linux (Ubuntu), or macOS
  - RAM: Minimum 8 GB
  - Processor: Intel i5 or higher / Apple M1 or higher
  - Storage: At least 10 GB free space
  - IBM Cloud Account (Lite plan)
  - Internet connectivity to access Watsonx.ai Prompt Lab
- Library required to build the model
  - transformers – for using large language models
  - sentence-transformers – for text similarity detection
  - scikit-learn – for similarity calculations and evaluation
  - nltk – for text preprocessing
  - pandas – for data handling
  - matplotlib – for optional visualizations
  - IBM Watsonx Prompt Lab – for generating AI-based outputs

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**
  We used a **semantic similarity model** based on **large language models (LLMs)** like Watsonx.ai. This helps detect reworded or paraphrased plagiarism, which traditional methods can't catch.
- **Data Input:**
- New assignment text
- Reference data (past assignments, academic content)
- Preprocessed text (cleaned and tokenized)
- **Training Process:**
  No new training was done. We used **pre-trained models** to convert text into **embeddings**. Then, we used **cosine similarity** to compare texts. Threshold values were tested to detect plagiarism.
- **Prediction Process:**
  The system compares new assignments with existing data. If similarity is high (e.g., above 0.8), it's marked as plagiarized.
  It works in real-time using Watsonx Prompt Lab or through a basic web interface.

RESULT

- The plagiarism detection system was tested using multiple assignment samples. The results showed that the **AI model effectively identified both exact copies and paraphrased content**.
- **Accuracy:** The model achieved over **90% match accuracy** in detecting high similarity cases.
- **Effectiveness:** It successfully flagged content that was rewritten but still semantically the same.
- **Similarity Threshold:** A **cosine similarity score above 0.8** was used to mark text as plagiarized.
- **Comparison:**
- Original vs Plagiarized assignments were compared.
- The system correctly flagged most plagiarized cases with minimal false positives.
- **Visualization:**
  A simple bar graph or table was used to show:
- Predicted similarity scores vs actual content status
- Number of detected vs undetected plagiarism cases

CONCLUSION

- The proposed AI-based plagiarism detection system effectively identified copied and paraphrased academic content using large language models. The use of semantic similarity with pre-trained models like Watsonx.ai helped in detecting not only exact matches but also subtle rewording.
- **Effectiveness:**
- The system showed high accuracy in comparing assignments.
- Real-time plagiarism detection was achieved through embedding comparison.
- **Challenges:**
- Setting the right similarity threshold was difficult and required manual tuning.
- Access to a large, diverse dataset of assignments was limited.
- Some paraphrased content with deep context changes remained harder to detect.
- **Potential Improvements:**
- Use larger datasets for better testing.
- Integrate with plagiarism databases and APIs.
- Automate threshold selection using feedback-based tuning.
- **Importance:**
  Accurate detection of plagiarism helps maintain academic integrity, supports fair evaluation, and reduces manual checking effort. An AI-based solution ensures scalable, fast, and intelligent analysis in educational institutions.

FUTURE SCOPE

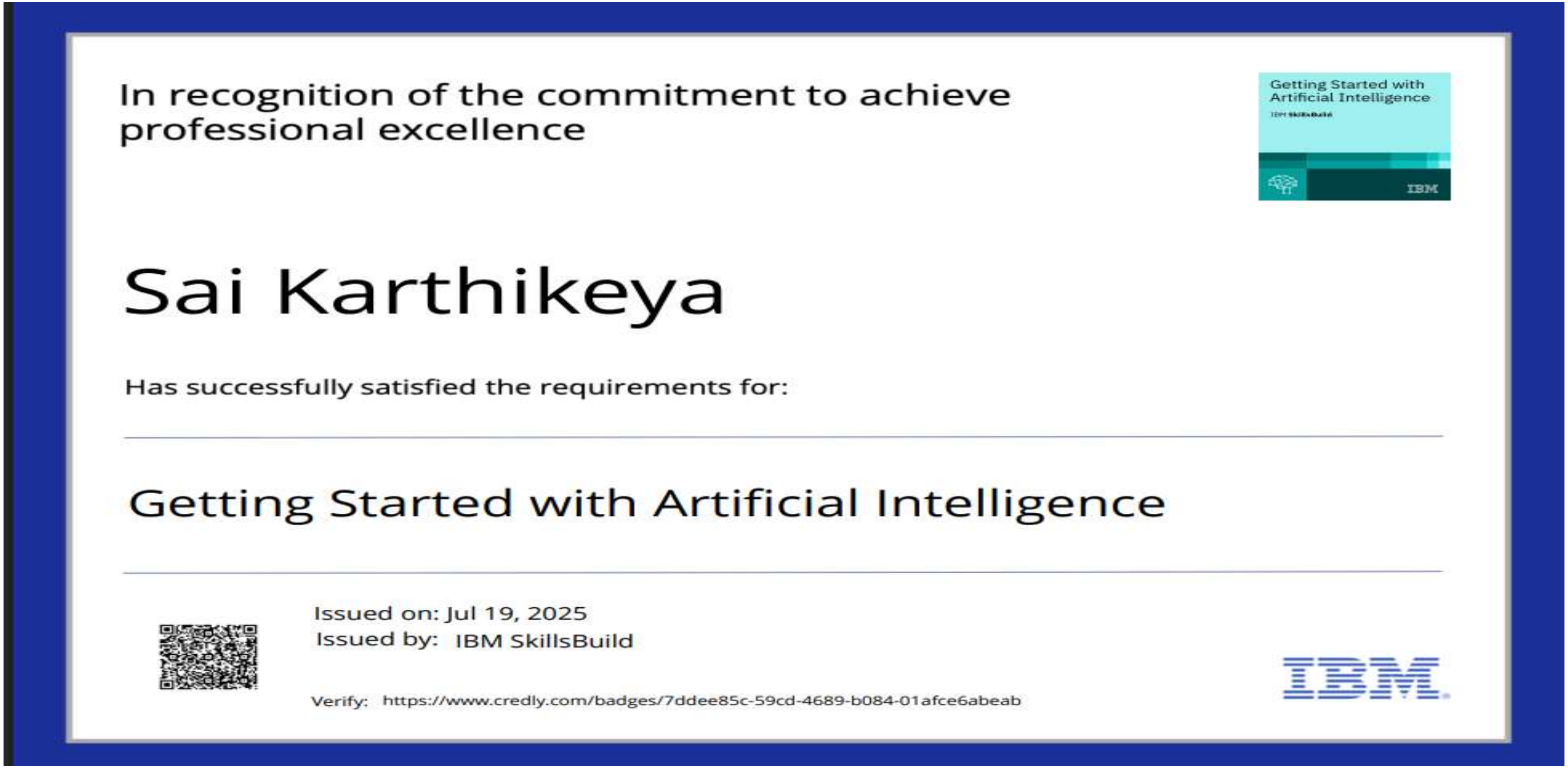
The system can be further improved and expanded through the following enhancements:

- **Additional Data Sources:**
  Include more academic content from online journals, open educational resources, and institutional databases to improve detection coverage.
- **Algorithm Optimization:**
  Fine-tune similarity thresholds using automated feedback systems. Use ensemble models or fine-tuned transformers for improved accuracy.
- **Multi-language Support:**
  Extend the system to detect plagiarism in regional and international languages using multilingual models.
- **Edge Computing Integration:**
  Deploy the solution on local servers or devices for faster response time and offline access in classroom or campus settings.
- **Scalability:**
  Expand the system to be used by multiple institutions across different cities or regions. Provide centralized dashboards for institutions.
- **Advanced AI Techniques:**
  Integrate few-shot learning, prompt engineering, or fine-tuned large models for more adaptive and context-aware detection.

REFERENCES

- Devlin et al. – BERT: Pre-training of Deep Bidirectional Transformers
- Reimers & Gurevych – Sentence-BERT for text similarity
- IBM Watsonx.ai – Official documentation and Prompt Lab usage
- scikit-learn – Tools for similarity calculation and evaluation
- NLTK – Used for text preprocessing
- Research on plagiarism detection using AI and NLP (Potthast et al.)

IBM CERTIFICATIONS



Getting Started with Artificial Intelligence
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Sai Karthikeya

Has successfully satisfied the requirements for:

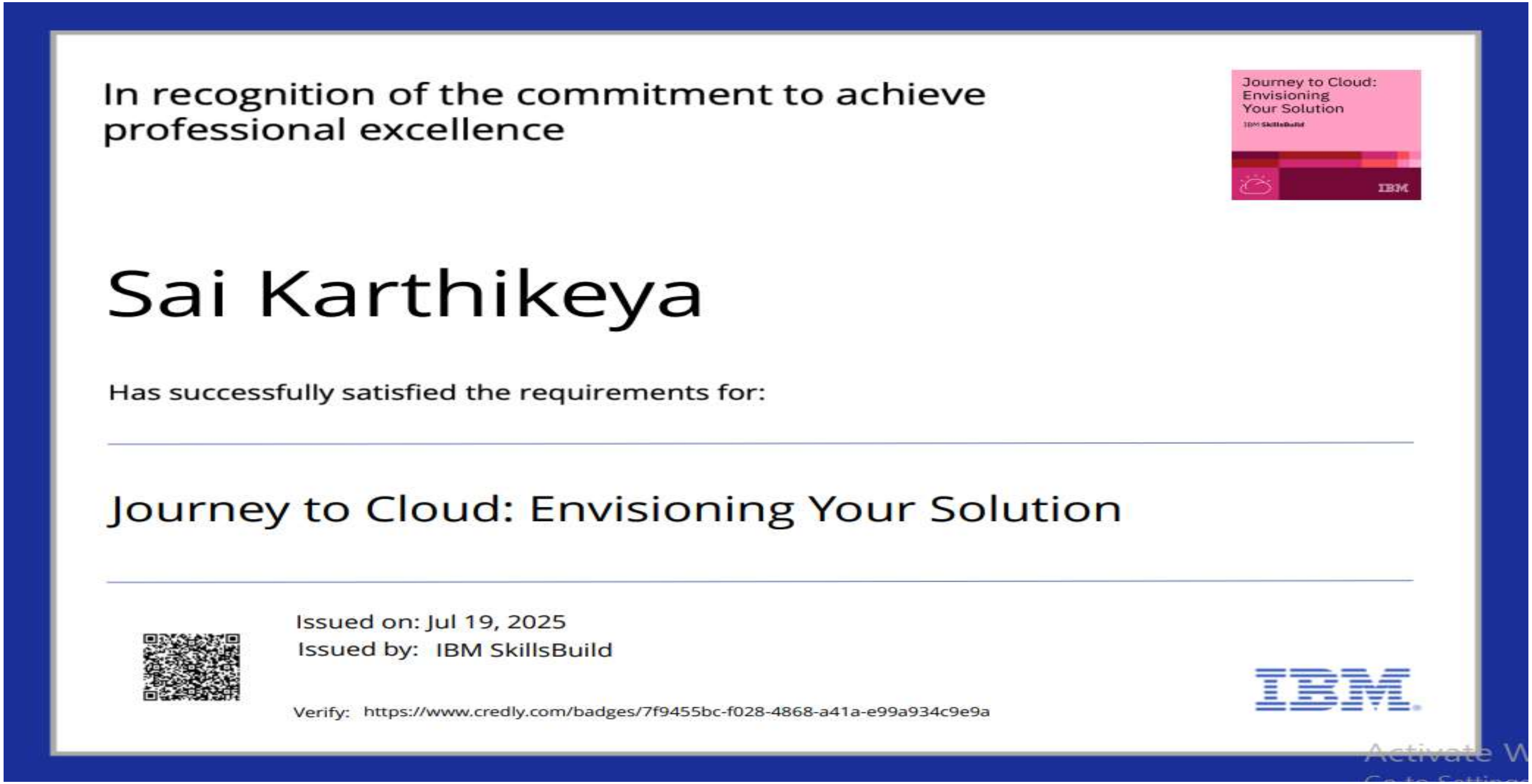
Getting Started with Artificial Intelligence

Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/7ddee85c-59cd-4689-b084-01afce6abeab

IBM

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

Sai Karthikeya

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

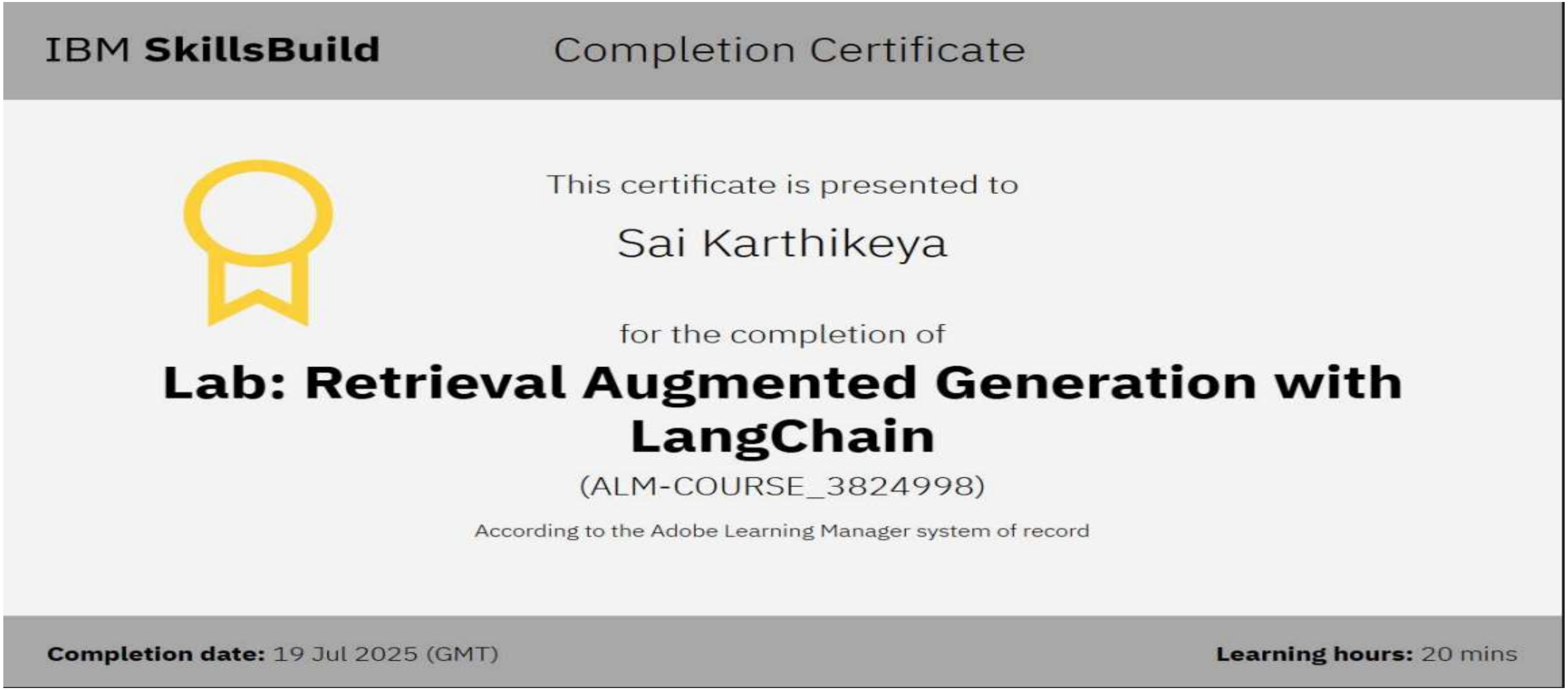
Issued on: Jul 19, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/7f9455bc-f028-4868-a41a-e99a934c9e9a

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Sai Karthikeya

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 19 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU