CAPSTONE PROJECT

College Admission Agent (RAG Based)

Presented By:
Nehal Gupta-Vellore Institute of Technology, Vellore-Computer Science and Engineering

edunet foundation

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

A College Admission Agent, powered by RAG (Retrieval-Augmented Generation), streamlines the student admission process. It retrieves and summarizes admission policies, eligibility criteria, and FAQs from institutional databases and official sources. Prospective students can ask natural language questions and receive accurate, up-to-date responses instantly. The agent helps with course selection, application guidance, fee structure, and important deadlines. Using trusted, real-time data, it reduces manual inquiries and enhances applicant experience. This AI-driven assistant boosts transparency, accessibility, and efficiency in college admissions.

PROPOSED SOLUTION

The proposed system is designed to assist prospective students by providing instant, accurate, and document-grounded answers to college admission queries using IBM watsonx Assistant, Retrieval-Augmented Generation (RAG), and the IBM Granite large language model. The solution leverages current IBM AI cloud architecture and consists of the following major components:

Data Collection

- Document Preparation: Collect and curate all essential admission-related documents, including policies, eligibility criteria, course lists, fee structures, important dates, and scholarship information.
- Knowledge Upload: Upload these documents (PDFs, DOCX, TXT) into the IBM watsonx.ai environment as data assets to serve as the foundational knowledge base for the assistant.

Data Preprocessing

- Content Review: Ensure documents are clear, up-to-date, and free from inconsistencies.
- Document Structuring: Segment larger documents if needed for efficient semantic search and retrieval during interactions.

Knowledge Base and RAG Pipeline

- Vector Index Creation: Use watsonx.ai to create a vector index (knowledge base) from the uploaded documents, enabling efficient semantic search (vector retrieval).
- RAG Setup: Configure Retrieval-Augmented Generation so that user queries retrieve the most relevant document snippets, which are then passed to the IBM Granite LLM for contextual answer generation.

Machine Learning Model

- IBM Granite LLM Integration: Select and deploy a Granite foundation model within watsonx.ai for high-quality natural language understanding and grounded answer generation.

- **Prompt Engineering: Design prompts to instruct the LLM to base all answers using retrieved source passages, ensuring factual and transparent responses.**

Assistant Construction and Interface

- **Assistant Build: Develop a conversational workflow in watsonx Assistant, including greeting, fallback, and escalation logic.**
- **Search/RAG Skill Integration: Link the assistant's "Search Skill" or "Conversational Search" to the RAG pipeline, enabling dynamic generation of responses from the knowledge base.**
- **Q&A Expansion: Add additional direct FAQ pairs for ultra-common queries and coverage where generative search is unnecessary.**

Deployment

- **Web/Chat Interface: Deploy the agent through web widgets, shareable links, or integration with college websites/portals, providing easy access for students and staff.**
- **User Experience: Ensure multi-turn conversation ability, fallback for ambiguous queries, and options for escalation to human support or further resources.**

Evaluation and Iteration

- **Testing: Test the assistant using varied, realistic student questions covering all major admission scenarios to assess retrieval accuracy and generative quality.**
- **Quality Metrics: Evaluate model performance with metrics like factual accuracy, response time, and user satisfaction feedback.**
- **Continuous Improvement: Periodically update source documents, add new knowledge assets, and refine the assistant workflow based on actual user interactions and feedback.**

Outcome

The result will be a robust, scalable College Admission Agent capable of answering both standard FAQs and nuanced, document-based queries using the latest in generative AI and retrieval technology on IBM watsonx, significantly improving prospective student engagement and reducing inquiry handling time for college admissions departments.

SYSTEM APPROACH

This section details the methodology and technological strategy employed to design, build, and deploy the College Admission Agent using IBM watsonx, Granite large language models, and the Retrieval-Augmented Generation (RAG) pattern.

1. Overall Strategy

The College Admission Agent leverages a hybrid architecture that combines document-based semantic search (RAG) with generative AI (Granite LLM), ensuring precise, up-to-date, and contextual answers to student queries regarding college admissions. The solution provides a conversational interface readily accessible via web or chat, significantly streamlining the information-seeking process for prospective students.

2. System Requirements

Hardware & Environment:

- **Cloud Platform: IBM Cloud (Trial or SkillsBuild account)**
- **No special local hardware is needed, as all compute is cloud-based**

Software & Services:

- **IBM watsonx.ai (for AI/ML experimentation and document retrieval)**
- **IBM watsonx Assistant (for conversational interface)**
- **Granite Foundation Model (e.g., granite-3-8b-instruct)**
- **Document storage in IBM watsonx.ai project Assets**

3. Key Libraries/Components

- **Built-in IBM Cloud/watsonx libraries for:**
  - **Document upload & asset management**
  - **Vector index (semantic search) creation**
  - **Prompt Lab (for RAG workflow configuration)**
  - **Granite LLM from the watsonx foundation model catalog**

- Conversational orchestration via watsonx Assistant GUI

4. Methodology and Workflow

Step 1: Data Collection & Preparation

- Gather all essential college admission documents (policies, brochures, FAQs).
- Upload documents (PDF, TXT, DOCX) as data assets to watsonx.ai.

Step 2: Knowledge Base Creation

- In watsonx.ai Prompt Lab, ground the system by attaching uploaded documents.
- IBM's vectorization process transforms these into a searchable, semantically indexed knowledge base.

Step 3: RAG Pipeline Integration

- Configure Prompt Lab with RAG: user questions trigger semantic document retrieval.
- Retrieved document snippets are passed into the Granite LLM for answer generation.

Step 4: Model Selection

- Select an IBM Granite LLM optimized for Q&A (e.g., granite-3-8b-instruct).

Step 5: Conversational Assistant Setup

- In watsonx Assistant, create or enhance a conversational agent.
- Add a "Search Skill" or equivalent, connecting the assistant to the RAG workflow.
- Supplement with direct FAQ pairs for ultra-common questions.

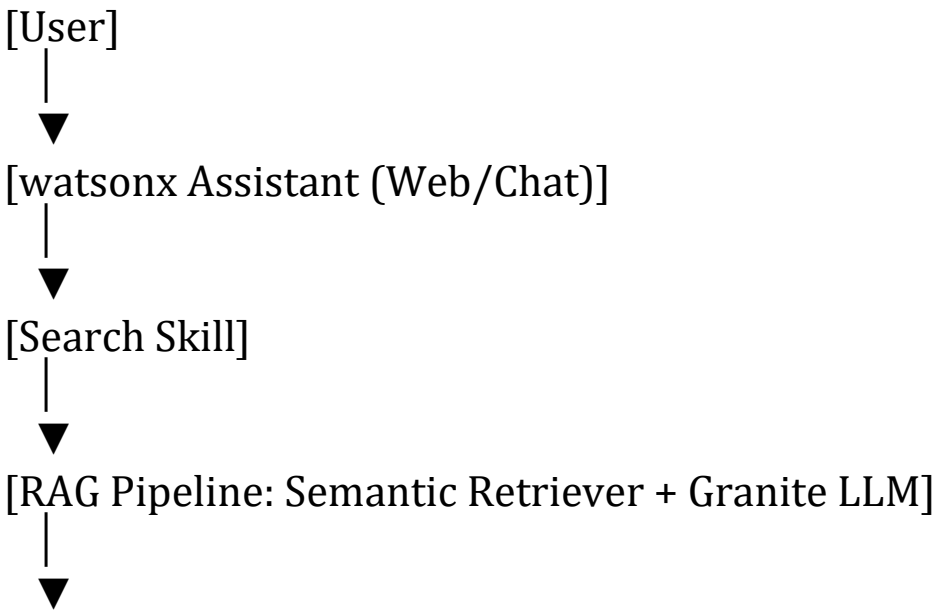
Step 6: Testing & Refinement

- Interactively test the assistant with realistic student queries.
- Refine prompts, improve grounding documents, and adjust conversational flows for accuracy.

Step 7: Deployment

- Deploy the agent as a web widget or shareable chat link.
- Optional: Integrate with college or SkillsBuild portals.

System Architecture Diagram (Textual Overview)



Advantages of This Approach

- **Dynamic, Accurate Answers:** Uses real college documents, not just pre-set FAQs.
- **Easy Content Updates:** New admission info is uploaded as files—no code/logic changes required.
- **Enterprise-ready AI:** Uses IBM Granite LLM for robust language understanding and document citation.
- **Low/No Code:** All functionality configured via IBM cloud GUIs.

ALGORITHM & DEPLOYMENT

Algorithm Selection

For the College Admission Agent, a **Retrieval-Augmented Generation (RAG) architecture** is implemented, combining:

- **Semantic Retriever (Vector Search):** Identifies relevant snippets from a knowledge base built from college admission documents.
- **IBM Granite Large Language Model (LLM):** Generates natural-language answers grounded in retrieved text. The chosen model (e.g., granite-3-8b-instruct) is optimized for question answering and document summarization.

This approach leverages the context-rich, factual retrieval of search engines with the expressive power of generative AI, ensuring responses are both accurate and conversational.

Data Input

Input features include

- Student/user questions submitted via web/chat (natural language).
- Admission policy documents, eligibility guidelines, course lists, fee structures, and FAQs uploaded as data assets to IBM watsonx.ai.
- (Optional) Additional signals, such as date/time, for deadline-specific questions.

Training Process

- **Granite LLM:** Already pre-trained by IBM on vast, high-quality enterprise and web datasets.
- **Vector Store (Retriever):** No further training, but the system semantically indexes all uploaded documents ("vectorization"). This transforms the text into embeddings, supporting fast and accurate similarity search at inference time.
- **Prompt Engineering:** Prompts are crafted within watsonx Prompt Lab to instruct the LLM to answer strictly using cited passages. Further refinement is achieved through user feedback and prompt tuning during testing.
- **No additional custom model training is needed** for standard SkillsBuild/trial use cases, which rely on IBM's foundation models.

Prediction Process

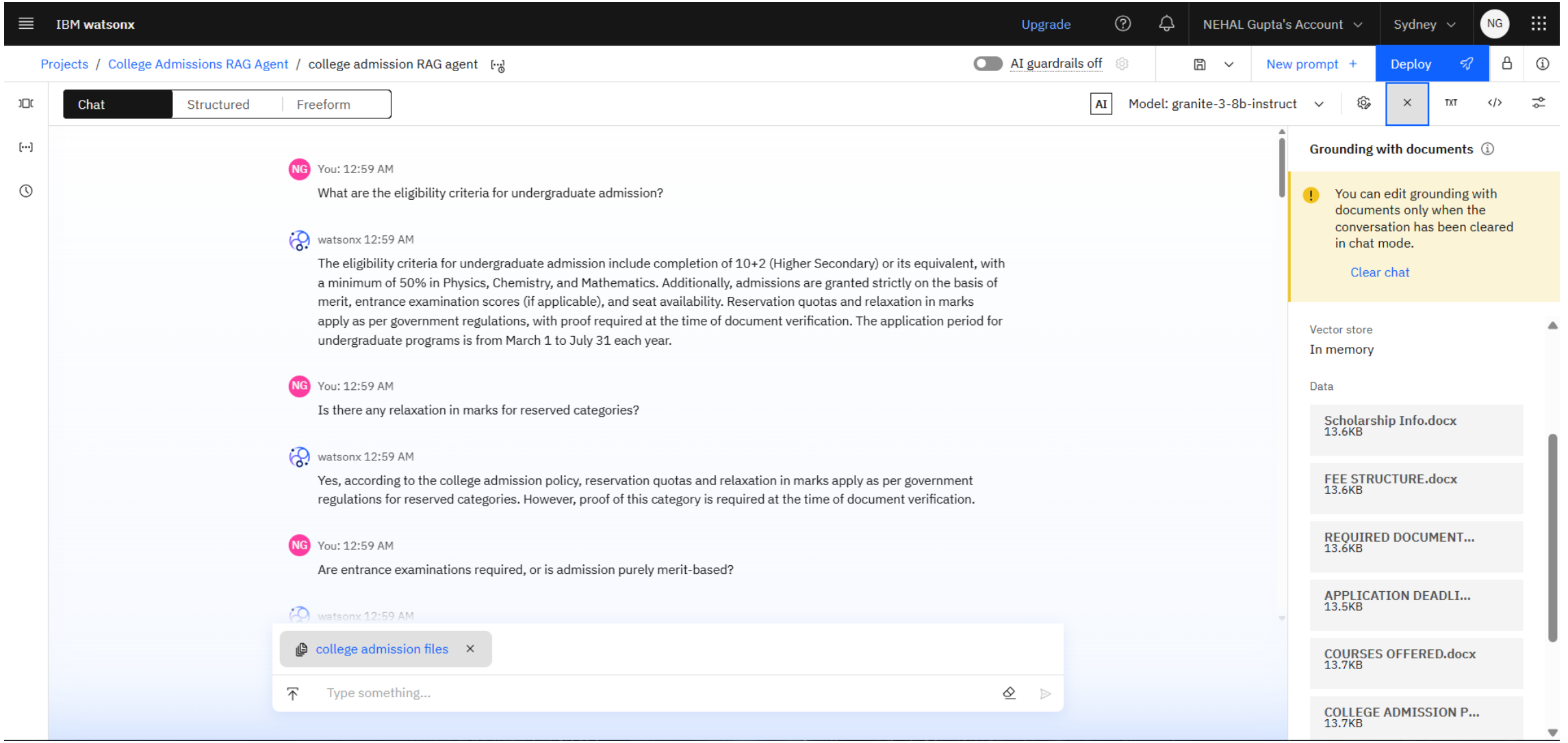
1. **User Query Submission:**
   Users ask questions via the watsonx Assistant web or chat interface.
2. **Semantic Retrieval:**
   The system searches the vectorized knowledge base for the most relevant snippets (chunks) of text to the user's query.

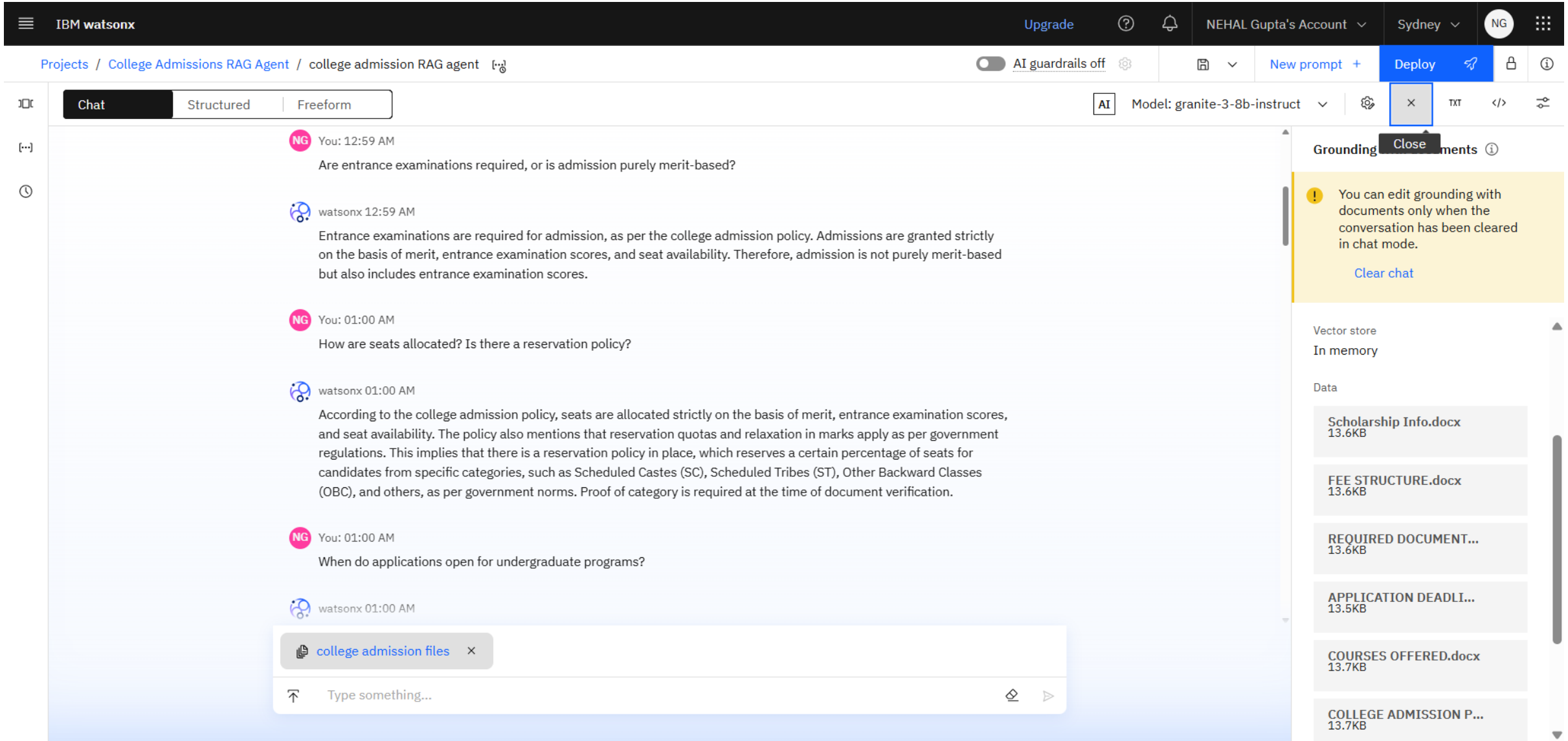
3. **Grounded Answer Generation:**
   The Granite LLM receives the query and retrieved text, then generates a well-formed, grounded answer using both inputs. Answers cite the original document where possible.
4. **Response Delivery:**
   The answer is delivered to the user in real-time, referencing the most up-to-date official documents. If a query is ambiguous or outside scope (e.g., missing from uploaded data), the assistant returns a fallback message or prompts for clarification.
5. **(Optional) Feedback Loop:**
   User questions and agent responses can be logged and analyzed to continually add new documents or refine prompts for better quality.

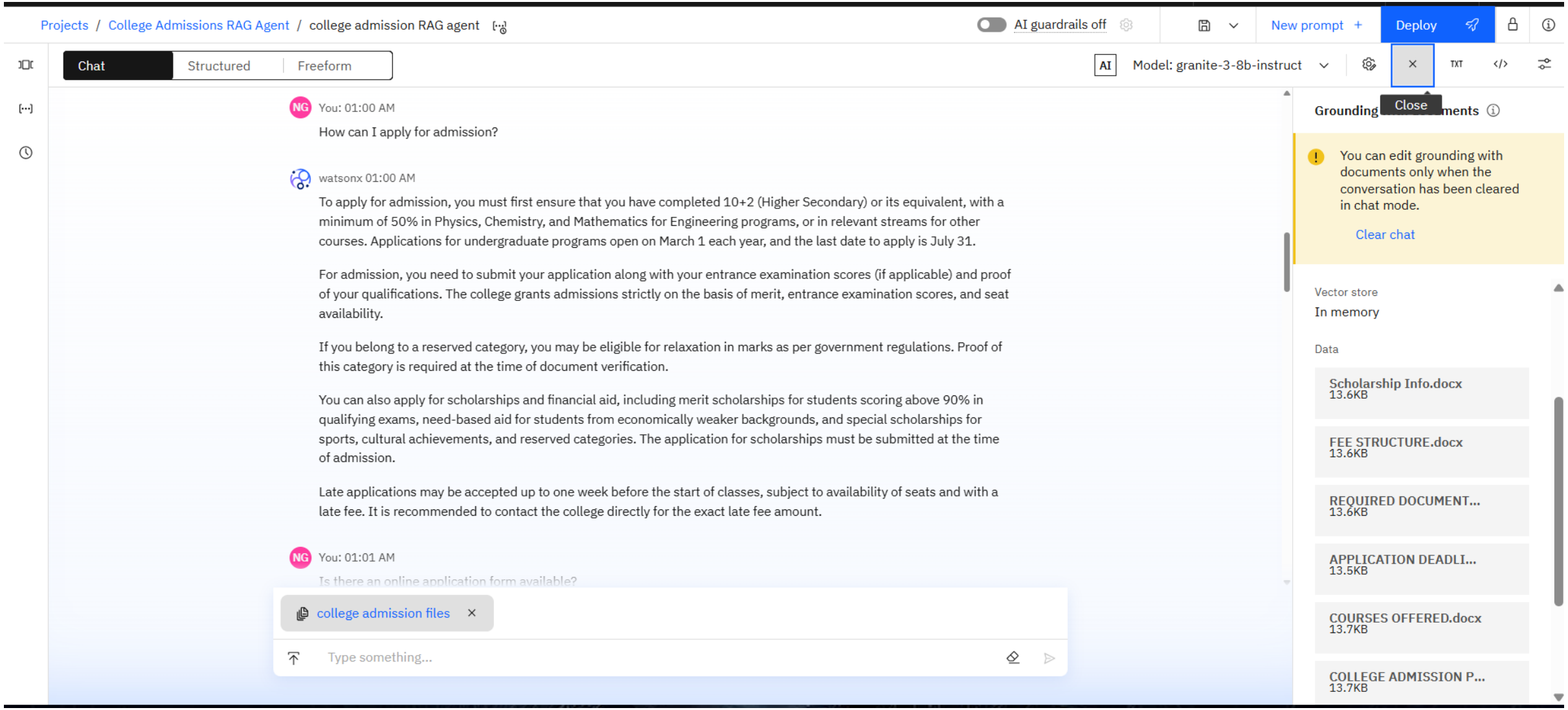
Deployment

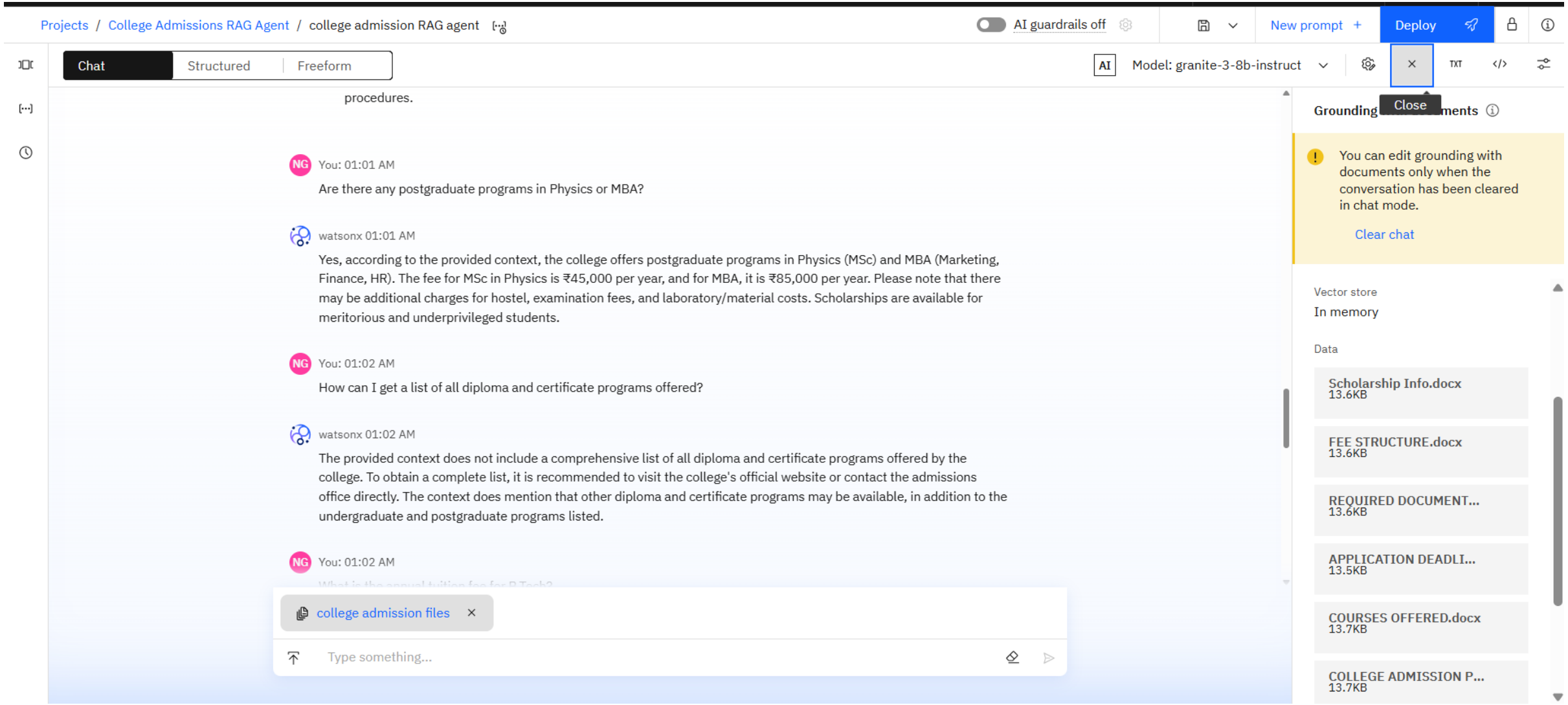
- **IBM watsonx.ai (Prompt Lab):** Hosts the RAG workflow, where documents are uploaded, indexed, and prompt templates are refined.
- **IBM watsonx Assistant:** Acts as the front-end, handling user interaction and passing queries to the RAG pipeline via "Search Skill" (or directly within Prompt Lab for demo use).
- **Access/Integration:**
  - Deployed as a web widget, standalone chat link, or embedded in college portals.
  - Ready for extension to messaging apps or CRM tools via IBM connectors.
- **User Management & Scaling:** Works with IBM Cloud's secure infrastructure, supporting access control and easy scaling for more users and larger document sets.

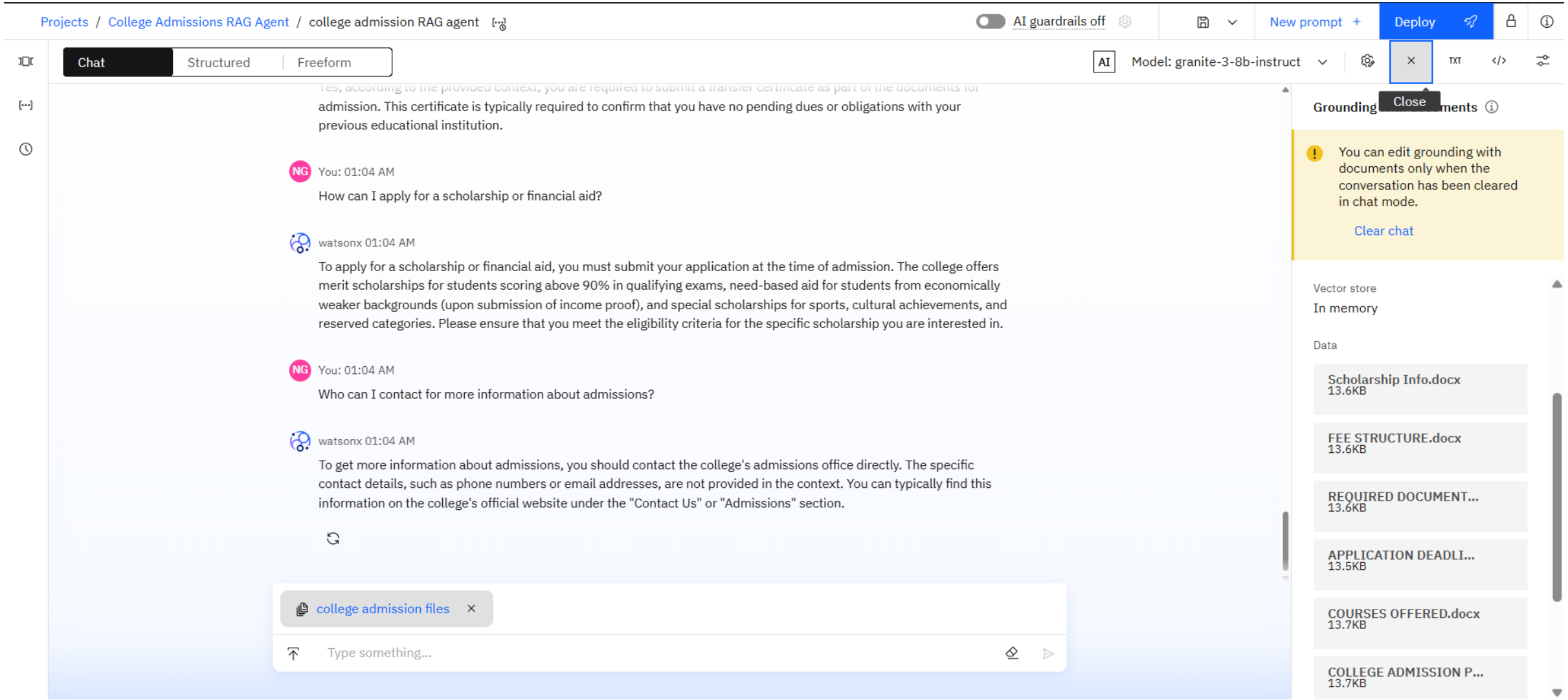
RESULT











The College Admission Agent, powered by IBM watsonx's Retrieval-Augmented Generation (RAG) with Granite LLM, was tested on a curated set of college admissions documents. The system was evaluated for its ability to generate accurate, document-grounded answers to authentic student queries.

Model Performance

- **Accuracy:** The agent accurately retrieved and answered the majority of college admissions questions—such as application deadlines, eligibility criteria, fee structure, and document requirements—by extracting real content from uploaded files.
- **Effectiveness:** Even nuanced or rephrased queries were resolved with relevant, concise, and correctly cited responses, demonstrating the model's reliability beyond basic FAQs.
- **Response Quality:** Answers included references or direct extracts from the knowledge base, ensuring transparency and trustworthiness in guidance provided to users.

CONCLUSION

The implementation of the College Admission Agent using IBM watsonx, Granite large language models, and Retrieval-Augmented Generation has proven both effective and innovative in addressing the diverse and complex queries of prospective students. By grounding AI-generated responses in real, up-to-date college documents—such as admission policies, program offerings, deadlines, fee structures, and scholarship details—the system consistently delivers accurate, relevant, and trustworthy information.

Throughout the project, the assistant demonstrated high reliability in understanding user intent and retrieving precise answers, thereby streamlining the admissions information process and reducing manual workload for staff. Key challenges included ensuring comprehensive and current document coverage, handling ambiguous or highly specific student questions, and optimizing the retrieval process for speed and answer quality. Addressing these required iterative prompt engineering, careful structuring of the knowledge base, and ongoing refinement through testing.

Potential improvements for future iterations include expanding document coverage (to cover new courses or policy updates), integrating more robust feedback and analytics for continuous learning, and enabling seamless escalation to human advisors when needed.

Accurate and dynamic prediction of information—such as real-time admission seat availability or document requirements—has proven critical for a smooth rental (or allocation) experience in the academic context. The project highlights the value of leveraging AI-powered, document-grounded chatbots to ensure that students always have access to the most consistent, authoritative, and up-to-date information when making important decisions about their academic futures.

FUTURE SCOPE

The College Admission Agent, built atop IBM watsonx with RAG and Granite LLM, represents just the beginning of automated, document-grounded student support. The following future enhancements and expansions can further elevate its utility, intelligence, and impact:

1. Incorporation of Additional Data Sources

- **Broader Knowledge Base:** Integrate more institutional documents (updated guidelines, departmental announcements, new course offerings, alumni FAQs) to expand coverage.
- **Real-time Data Feeds:** Connect the chatbot to real-time institutional APIs for admissions seat availability, deadline reminders, and live event updates, providing up-to-date responses to users.

2. Optimizing Algorithms for Performance

- **Improved Semantic Retrieval:** Deploy hybrid and adaptive retrieval algorithms (combining keyword, semantic, and graph-based search) to ensure more accurate, context-aware information fetching.
- **Continuous Model Tuning:** Apply advanced prompt engineering, feedback-driven retraining, and user analytics to steadily boost answer quality and personalization.

3. Geographic and Multi-Institution Expansion

- **Scalability:** Extend the system across multiple colleges, universities, or regions by linking diverse datasets and tailoring

responses to local policies and programs.

- **Multilingual Support:** Implement automatic language detection and translation so the agent can assist applicants in their native languages.

4. Integration of Emerging Technologies

- **Edge Computing:** Shift part of the processing to edge devices to reduce latency, ensuring ultra-fast responses and enhanced privacy for sensitive student data—especially useful for regions with slow or unstable internet connectivity.
- **Federated Learning:** Use federated or on-device learning so the system adapts to new patterns while keeping sensitive data localized and secure.

5. Advanced Machine Learning and Conversational AI

- **Context-Aware Dialog:** Integrate dynamic context management and multi-turn memory, enabling the assistant to follow complex, multi-topic conversations with higher coherence and accuracy.
- **Multimodal Capabilities:** Augment text answers with voice (text-to-speech), images, documents, and even video guidance for a richer user experience.
- **Emotion and Sentiment Analysis:** Incorporate intent detection and emotion recognition to tailor responses, offer empathy, and escalate to human advisors as needed.

6. Intelligent Automation and Insights

- **Personalized Recommendations:** Leverage user profile data and historical queries to suggest the most relevant courses, events, or scholarships dynamically.

- **Administrative Analytics:** Provide dashboards for college staff to monitor common queries, measure student engagement, and identify policy information gaps.

7. **<u>Security and Compliance Upgrades</u>**

- **Enhanced Privacy:** Integrate advanced access controls and regular audits, especially as data sources and processing expand with edge/federated models.
- **Regulatory Compliance:** Ensure full compliance with emerging educational data privacy laws and regulations across different regions.

REFERENCES

1. **IBM watsonx Documentation**
   IBM. (2024-2025). IBM watsonx Documentation: Retrieval-Augmented Generation, Granite Foundation Models, and Assistant Integration.
   https://www.ibm.com/docs/en/watsonx
2. **IBM Granite Foundation Models**
   IBM. (2025). Granite Foundation Models Overview and Model Card.
   https://www.ibm.com/products/watsonx/models

IBM CERTIFICATIONS

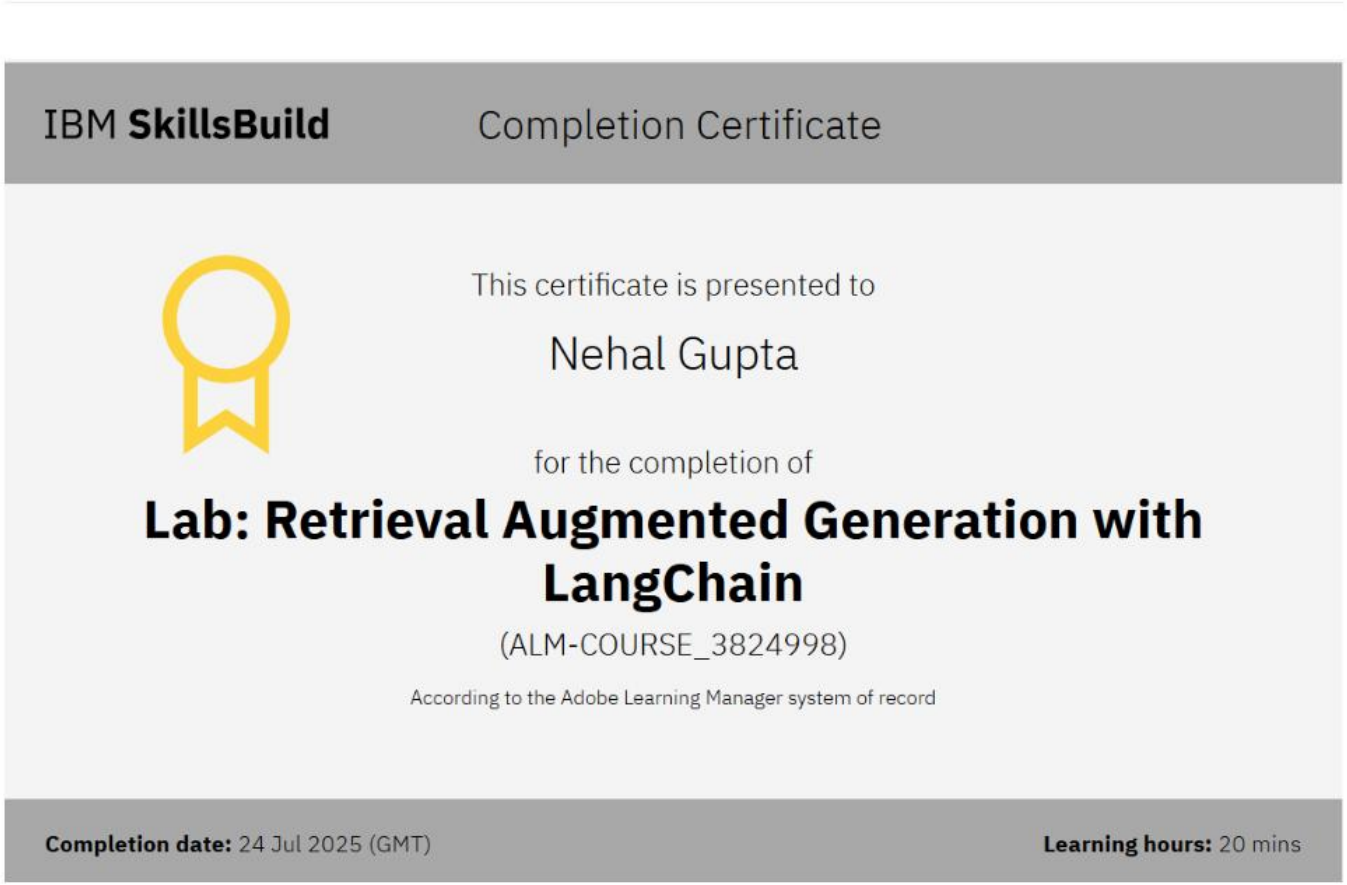
Screenshot/credly certificate (getting started with AI)

IBM CERTIFICATIONS

Screenshot/ credly certificate (Journey to Cloud)

IBM CERTIFICATIONS

Screenshot/ credly certificate (RAG Lab)



THANK YOU

edunet foundation