



HACKEREARTH RAG APPLICATION – SPRINT 6

Ethan Villalovoz, Molly Iverson, Adam Shtrikman, Chandler Juego

Introduction

Develop a RAG (Retrieval-Augmented Generation) application for HackerEarth that will utilize vector search, knowledge graphs, and a LLM to answer questions and generate content from a knowledge base of more than 10,000 Wikipedia articles.

Sprint Objectives

Presentation Poster: comprehensive visual summary of our capstone project

Project Video and Pitch Video: Two videos created to communicate our project at a high-level summary and its broader impact

Client Presentation: presentation delivered to our project's sponsor, demonstrating progress, key deliverables, and challenges

Refinement of Project Report: final refinement of project report, incorporating feedback from advisors and clients

Documentation and Reporting: Continue realigning priorities and goals with client

Feature Implementation

Team Member	Feature	Impact
Ethan	Presentation Poster	Communicates our project's core contributions in a visually engaging format
Chandler	Project Videos	Increases accessibility and outreach by showing the project's purpose and value to a broader audience, including stakeholders.
Molly	Project Report Refinement, Client Presentation	Ensured alignment with stakeholder goals and validated the practical relevance of our RAG solution. Refining the project report enhanced the clarity and professionalism of our final deliverable, making it a strong reference for future work.
Adam	Project Videos	Increases accessibility and outreach by showing the project's purpose and value to a broader audience, including stakeholders.

Feature Demos

Deployment and Feedback

Deployment: Our client is not using our project, but we've successfully deployed using Docker

Final Deployment Steps: CI/CD pipeline

Feedback: The client is happy with our progress

Post-Deployment Testing: Functional requirements met. Performance is efficient (~8 sec). Users can easily pull the Docker images to run locally

Deployment and Feedback

Packages 2

- acme10-he-ragapp-backend
- acme10-he-ragapp-frontend



CI/CD pipeline
using docker-
publish.yml
workflow and
GitHub actions

README MIT license

Running the Application with Docker

If you prefer to use the pre-built Docker images instead of setting up the project manually, follow these steps.

Prerequisites

- Install Docker Desktop: Download and install [Docker Desktop](#).
- Install Docker Compose: Ensure you have `docker-compose` installed. You can install it via Python:

```
pip install docker-compose
```

- LLM API Key: Ensure you have an OpenAI key saved in your environment

```
# For Linux  
export OPEN_API_KEY=value
```

Running the Application

1. Pull the Frontend Image:

```
docker pull ghcr.io/mollyiverson/acme10-he-ragapp-frontend:latest
```

2. Pull the Backend Image:

```
docker pull ghcr.io/mollyiverson/acme10-he-ragapp-backend:latest
```

3. Download the `docker-compose.yml` File:

```
curl -O https://raw.githubusercontent.com/mollyiverson/ACME10-HE-RAGApp/main/docker-compose.yml
```

Versions

1 tagged 13 untagged

latest

Published about 15 hours ago · Digest ...

sha256:37614bd5baead41b005c7fb9c7b127319f3c684

Published 6 days ago

sha256:4b64d21ff8085e05f2070f829e37944a4e88000e

Published 6 days ago

sha256:9f80ff497fd385fe58f397b114cb59b5be798bf5c

Published 7 days ago

Kanban

Overview &

Contributions

Molly

- Created the final client presentation
- Edited and finalized project report

Ethan

- Created the project poster

Chandler and Adam

- Created the 2-3-minute LinkedIn video
- Created the 30-second pitch video

Finalize and edit Project Report #98

Open

mollyiverson/ACME10-HE-RAGApp Private

Edit



mollyiverson opened 2 weeks ago · edited by mollyiverson

Edits · ...

As a client and instructor
I need a revised project report
So that the project information is detailed and complete

Details and Assumptions

- From PK:
 - Add two more paragraphs in introduction about scope, motivation, what we've done
 - Fix use case diagram (blurry)
 - More some titles to the next page if most of the content is cut off
 - Add more to UI design section
 - Testing section should be in past tense
 - Use bullets instead of 1, 2, 3
 - Add more tests
 - Change languages used in project table to be like tools/frameworks table
 - update project delivery status
 - Add a paragraph to conclusion
 - Add 10 references
- More revisions
 - Update text in all sections
 - Update images
 - Update figure #s and ToC
 - Add more unit tests to code

Assignees

mollyiverson

Labels

documentation

Projects

Capstone Agile Planning

Status In review

Priority Choose an option

Size Choose an option

Estimate Enter number...

Start date No date

End date No date

Sprint Sprint 6 · Mar 11 - Apr 11
Current

Milestone

No milestone

Relationships

Done 85 / 75 Estimate: 171

This has been completed

ACME10-HE-RAGApp #80

Compile Custom Dataset and Prep for Embeddings

Sprint 5

ACME10-HE-RAGApp #88

Embed full Wikipedia dataset w/ retrieval instructions

3 Sprint 5

ACME10-HE-RAGApp #79

System Testing by Timing Components

Sprint 4

ACME10-HE-RAGApp #77

Implement system performance testing

2 Sprint 4

ACME10-HE-RAGApp #81

Chunking Dataset During Embedding

Sprint 4

Evidence of Client Meetings

We didn't have regular meetings with our client this sprint per his wishes. We've emailed current progress and demo videos throughout the sprint to keep him updated.

Project Report Refinements

Final Sprint Achievements & Challenges

Key Achievements:

- Delivered fully integrated RAG Application combining KG and VS
- Obtained client approval for all major functionalities
- Completed detailed project report, professional videos, and promotional poster

Challenges:

- NLP struggled with complex multi-entity queries, reducing accuracy
- Vector search errors caused stability issues
- Needed to develop stronger LLM guidelines to prevent harmful outputs

Solutions Implemented:

- Optimized NLP for key scenarios, openly communicated existing limitations
- Integrated thorough automated and manual testing for response accuracy
- Fixed vector search indexing errors and improved logging for better monitoring
- Enhanced LLM instructions to prevent harmful or inappropriate content generation

Retrospective & Final Reflection

Retrospective:

- *Worked Well:* Clear sprint planning, regular mentor engagement.
- *Could Improve:* Early NLP and vector search performance evaluations, proactive integration tests to detect issues earlier.

Lessons Learned:

- Importance of clear and continuous communication within team and with stakeholders.
- Early and frequent integration tests improve stability and reduce late-stage complications.
- Agile practices significantly benefited our adaptability to evolving requirements and technical challenges.

Client Feedback Actioned:

- Switched LLM model to OpenAI for improved response speed as requested by client.
- Enhanced user interface clarity based on client feedback, improving user interaction and experience.

Next Steps:

- Client Presentation on 4/11
- Poster Presentation on 4/22

Conclusion & Handover

Project Recap:

- Successfully developed a RAG application leveraging Knowledge Graph and Vector Search to enhance response accuracy and contextual relevance
- Fully functional and meets all and desirable requirements set forth by HackerEarth
- Demonstrates strong real-world readiness for applications in conversational agents, research assistance, and custom dataset integrations

Handover Status:

- Project fully documented and available on GitHub with detailed setup instructions
- All final promotional materials including videos and posters are completed and available for client use

Thank You!

Deep gratitude to our client,
HackerEarth CEO Vikas Aditya,
and mentor Dr. Parteek Kumar.