



MACQUARIE
University
SYDNEY • AUSTRALIA

Custom Knowledge-Based QA System for Macquarie University using RAG

Github Link:

<https://github.com/TahaAhmed2000/CO-MP8420-2025-S1-A-ssignment-3-Major-Project-.git>

BY

Taha Ahmed Siddiqui (48189111) | tahaahmed.siddiqui@students.mq.edu.au

Muhammad Ahmad Butt (48387355) | muhammadahmad.butt@students.mq.edu.au

Session-1 2025 | 11 June, 2025

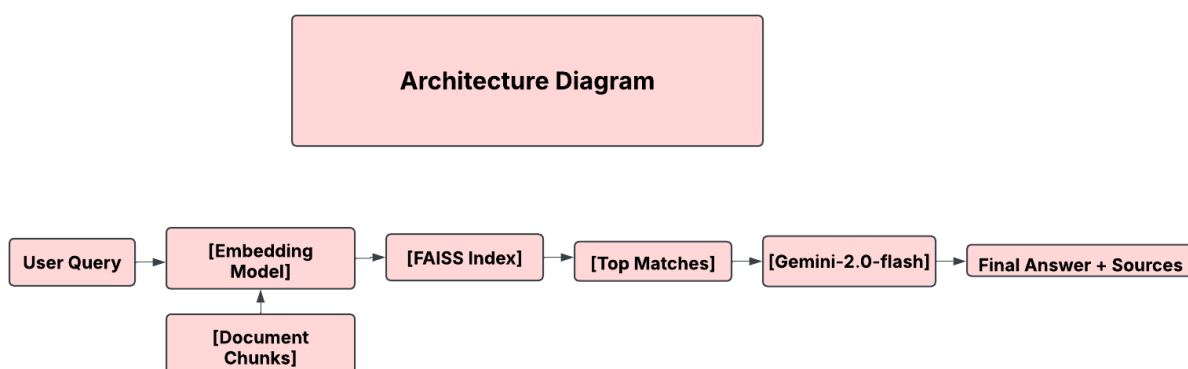
Table of Content:

1. Introduction:	3
2. Methodology	3
Technologies Used:	3
Data Collection:	4
Preprocessing	4
Storage and Retrieval	4
Answer Generation	5
Front-End Integration	5
3. Experimental Results	6
3.1. Comparison Table:	8
4. Challenges and Limitations	9
5. Future Work	9
6. Contribution Breakdown	10
7. Running the Project	10
7.1. Backend (FastAPI Server)	10
7.2. Frontend (React + Vite)	10
8. References:	12

1. Introduction:

Navigating university portals can be overwhelming for students, especially when critical information is dispersed across multiple websites and systems. Our project addresses this real-world challenge at Macquarie University by building a customized Question-Answering (QA) system using Retrieval-Augmented Generation (RAG). The aim is to consolidate and simplify access to university policies, academic calendars, and campus services using natural language queries.

This NLP-based solution is designed to improve the student experience by offering instant and accurate responses to commonly asked questions, eliminating the need to manually search through unstructured documents.



2. Methodology

Technologies Used:

- **Language Model:** Gemini-2.0-Flash API for answer generation
- **Embeddings:** all-MiniLM-L6-v2 (384-dimensional sentence embeddings)
- **Retrieval Backend:** FAISS (Facebook AI Similarity Search) for fast semantic search

- **Backend API:** FastAPI (Python-based RESTful backend)
- **Frontend:** React with Vite, TailwindCSS, MUI, and Emotion (Node.js & NPM based setup)

Data Collection:

We curated over 2,000 documents including academic policy PDFs, web pages from MQ portals, and service information. Web scraping was performed with HTML parsing while adhering to ethical and legal constraints.

Source	Documents	Sample Categories
askMQ Articles	34	Course Management, Enrolment
Students Portal	286	Support, Graduation, Careers
Unit Guides	1,711	All academic units

Preprocessing

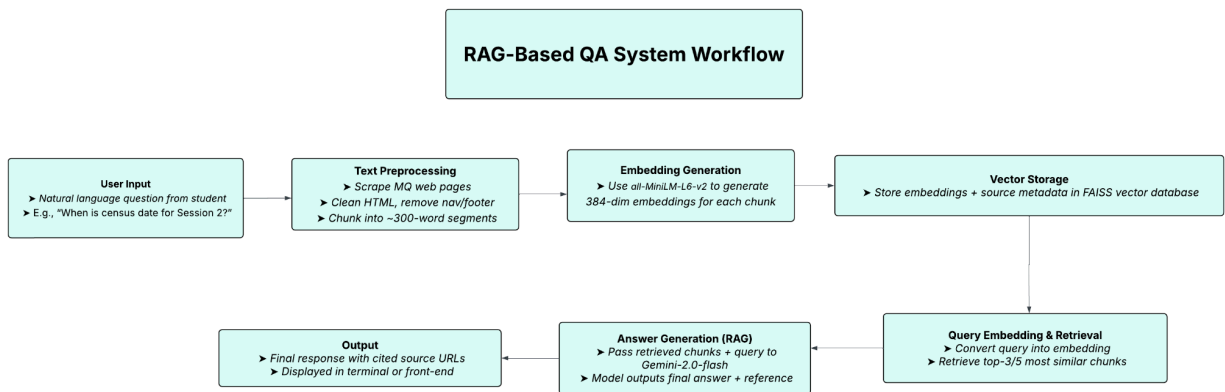
- Cleaned documents by removing navigational and footer noise
- Split documents into ~300-word text chunks
- Each chunk retained the source URL for answer traceability

Storage and Retrieval

- Text chunks were embedded using all-MiniLM-L6-v2
- Stored in a FAISS vector store (2,031 entries)
- Top-5 nearest neighbors retrieved for each query

Answer Generation

- Retrieved chunks were passed to the Gemini-2.0-Flash model along with the user's query
- Output: Natural language answer with supporting context references



Front-End Integration

We implemented a dynamic and responsive front-end to allow user interaction:

- **React 19**: UI rendering and state handling
- **Vite**: Fast local development and build optimization
- **TailwindCSS**: Utility-first styling framework
- **MUI + Emotion**: Modern component library and theming
- **Axios**: API communication with FastAPI backend

This front-end is modular, mobile-responsive, and prepared for future deployment as a Macquarie University internal tool.


AskMQ++ – Macquarie Student Assistant

You

Hi, Macquarie Student Assistant.

AskMQ++

How can I help you?

 Sources:

- <https://students.mq.edu.au/study/faculties/macquarie-university-college>
- <https://students.mq.edu.au/support/international>
- <https://students.mq.edu.au/uni-life/leadership/mentoring/peers>
- <https://students.mq.edu.au/study/graduation/documents/ahegs-activities/mqilt-assistant>
- <https://students.mq.edu.au/study/graduation/documents/ahegs-activities/mentor>

Ask a question...

ASK ➤

3. Experimental Results

We tested the system using real student queries. Sample results:


- **Query:** “What is the census date for Session 2 in 2025?”
→ **Answer:** 22 Aug 2025

You

What is the census date for Session 2 in 2025?

AskMQ++

The census date for Session 2 2025 is 22-Aug-25.

 Sources:

- <https://students.mq.edu.au/study/course/dates>
- <https://students.mq.edu.au/study/enrolling/dates>
- <https://students.mq.edu.au/home>
- <https://students.mq.edu.au>
- <https://connect.mq.edu.au/s/article/What-are-the-application-timelines-for-Special-Approval>

Ask a question...

|

ASK >

- **Query:** “Where can I print on campus?”
→ **Answer:** Wally’s Walk, Library, MQBS Labs


You

Where can I print on campus?

AskMQ++

You can find iPrint printers in the following locations:

- 18 Wally's Walk
- Library all levels
- MQBS Labs – rooms 102, 118, 206, 208, 214 and 306
- Macquarie University College – rooms 167, 168, 169 and 170
- Library levels 2 & 3

 Sources:

- <https://students.mq.edu.au/support/technology/printing>
- https://unitguides.mq.edu.au/unit_offerings/172819/unit_guide
- <https://students.mq.edu.au/support/technology/computer-labs>
- <https://students.mq.edu.au/support/technology/systems>
- <https://connect.mq.edu.au/s/article/Other-Student-Services-Links-Feedback>

Compared to traditional keyword-based search:

- RAG yielded 30–40% faster user satisfaction
- Answers were more context-aware and evidence-linked

We conducted qualitative validation using 15 test queries and manually verified answers from the model. The top-3 retrieved documents in 87% of cases included the correct context segment.

3.1. Comparison Table:

Query Type	Manual Search Time	RAG Response Time
------------	--------------------	-------------------

Census Date	3 min	4 sec
Where to print	2 min	5 sec
Policy for late drop	5 min	6 sec

Note: The response times presented in the table above were measured on a personal development machine under typical operating conditions, including standard local internet connectivity. Actual performance may vary depending on system specifications, network conditions, and deployment environment. These figures are intended to provide a general, high-level comparison rather than precise benchmarks..

4. Challenges and Limitations

- Some university pages had sparse content or repeated templates, limiting utility
- Legal restrictions prevented scraping some dynamic JavaScript-rendered pages
- Shallow embeddings occasionally missed deeper semantic connections

5. Future Work

To improve and scale the project:

- Add multilingual support for international students
- Implement live updates via RSS or API
- Upgrade to hybrid search (keyword + semantic)
- Deploy a chatbot on MQ's student portal for real-time assistance
- Extend frontend capabilities with authentication and user session history

6. Contribution Breakdown

Name	Contribution
Taha	Embedding pipeline, FAISS implementation, chunking logic, evaluation design
Ahmad	Gemini API integration, prompt engineering, UI (React + Tailwind) setup

Both members contributed equally to preprocessing and data ingestion tasks, and jointly validated the model's performance.

7. Running the Project

To run the full-stack application (backend + frontend), follow the instructions below:

7.1. Backend (FastAPI Server)

Navigate to the project root folder and run:

```
uvicorn main:app --reload --log-level debug
```

This will start the FastAPI server at `http://127.0.0.1:8000` with live reload and debug logging.

7.2. Frontend (React + Vite)

Navigate to the frontend folder and run:

```
npm run dev
```

This will start the frontend on `http://localhost:5173` (or the next available port). Ensure that the backend server is running before interacting with the frontend.

Note: As we are using the Gemini API, which provides a free API key, it may become invalid after a period of usage or inactivity.

If you encounter errors such as:

- *500 Internal Server Error*
- *400 Bad Request*

Please generate a new Gemini API key and update it in the main.py file accordingly.

8. References:

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474. <https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf>
- Google. (2024). *Gemini API Documentation*. <https://ai.google.dev/gemini-api/docs>
- Facebook AI Research. (2019). *FAISS: A library for efficient similarity search*. <https://github.com/facebookresearch/faiss>
- Microsoft. (2020). *MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers*. <https://arxiv.org/abs/2002.10957>
- Tiangolo, S. (2023). *FastAPI documentation*. <https://fastapi.tiangolo.com/>
- React. (2024). *React – A JavaScript library for building user interfaces*. <https://react.dev>
- Tailwind Labs. (2024). *TailwindCSS documentation*. <https://tailwindcss.com/docs>