



Digital Back Office Ltd.

Gen-AI Mini Project Assignment

Enterprise Knowledge Assistant — Using Technical Documentation

1. Dataset & Context

You will build a knowledge assistant using public technical documentation, treating it as internal company knowledge.

Choose ONE source:

- FastAPI documentation
- Django documentation
- Kubernetes documentation
- React documentation
- Any official API/library documentation

Download or scrape multiple pages/sections and treat them as the company's knowledge base. Your assistant must understand, summarize, and answer questions based on these docs — just like internal developer support tools used at companies.

2. Functional Requirements (Prototype Focus)

Your system must demonstrate:

✓ Document Ingestion

- Load text or HTML docs
- Clean and extract readable content
- Split into sections/chunks

✓ Summarization

- Generate short, accurate summaries of sections
- Produce a higher-level executive summary for the entire documentation source

✓ FAQ Generation

- Identify 3–6 frequently useful developer questions
- Generate concise answers based strictly on the documentation

✓ Retrieval + Answering (RAG Behaviour)

- Accept a user question
- Retrieve relevant documentation content
- Generate a grounded answer using retrieved text
- If information is missing, respond:
"The knowledge base does not contain this information."

UI is NOT required.

Console or notebook output is acceptable — functionality matters.

3. Deliverables

A GitHub repo containing:

- ✓ Source code
- ✓ Document ingestion scripts
- ✓ Prompt logic / pipeline code
- ✓ README explaining:

Live Working Prototype Link

✓ Output Evidence

Show:

1. Sample document summaries
2. Generated FAQs and answers
3. A few query → response examples with retrieved source references

Example:

Q: How do I create a FastAPI route?

A: You define a function, decorate it with `@app.get("/path")`, and return data.

Source: `fastapi/tutorial/first-steps/`

✓ Short Write-Up (1)

- What documentation you used
- Architecture overview
- Limitations & improvements

4. Evaluation Criteria

Students are graded on:

- ✓ End-to-end working pipeline
- ✓ Meaningful summaries

- ✓ FAQ accuracy (grounded in docs)
- ✓ Retrieval + reasoning quality
- ✓ Error-handling (missing info responses)
- ✓ Structural clarity — NOT UI polish

Innovation is appreciated but small bugs or minimal UI do not affect marks as long as the prototype works.