

GenAi assignment 3



Samyan Ahmed 22i-2639

Hassan Ilyas 22i-2414

Table of Contents

Chunking & Ingestion.....	0
Retrieval & Answering.....	1
Models & Tools.....	1
Screenshots.....	2
GenAi assignment 3.....	0

FAST-NUCES FYP Handbook Assistant – Implementation Note

Chunking & Ingestion

- Source: 3. FYP-Handbook-2023.pdf (FAST-NUCES BS Final Year Project Handbook 2023).
- Parser: PyMuPDF via PyMuPDFLoader, which preserves page numbers in metadata.
- Chunking: RecursiveCharacterTextSplitter with chunk_size = 1000 characters (≈300 words) and chunk_overlap = 300. Each chunk stores:
 - page: zero-based loader page
 - section_hint: first heading/few meaningful words from the chunk
 - chunk_id: sequential label (chunk_0, chunk_1, ...)
- Indexing: Embeddings from sentence-transformers/all-MiniLM-L6-v2 (Hugging Face) → FAISS local index (faiss_index). Ingestion script: ingest.py.

Retrieval & Answering

- Query handling: User input is passed through expand_query() to append domain-specific synonyms (chapters, margins, spacing, endnotes, etc.).
- Retrieval settings:
 - k = 5 (top-5 cosine similarity hits from FAISS)
 - Similarity threshold = 0.22. If no chunk exceeds the threshold, the app replies “I don’t have that information in the handbook.”
 - Formatting-focused questions (margins/spacing/fonts) drop the threshold guard but display a warning when similarity is low.
- Grounding: The UI assembles the retrieved chunks (with page metadata) into an answer context, then a deterministic extraction layer builds the final response while citing (p. X) references.
- Precautions:
 - Out-of-scope questions inherit the “no information” guard.
 - Every answer includes page references and optional quotes pulled directly from retrieved chunks.
- Users can inspect all retrieved chunks in the “Sources (Page References)” expander (includes page numbers, section hints, similarity scores, and chunk previews).

Models & Tools

- Embeddings: sentence-transformers/all-MiniLM-L6-v2
- Vector store: FAISS (local)
- UI: Streamlit (app.py) with a single textbox + “Ask” button + answer panel + collapsible sources list
- Prompt template embedded directly in app.py (no external LLM calls; responses come from local extraction logic)
- Prompt log: prompt_log.txt

Screenshots

