WORKABLE AI ASSIGNMENT FOR HIRING

Assignment: LLM-Based Pipeline for Structured Question Extraction from RD Sharma (Class 12)

Objective

Design and implement a Retrieval-Augmented Generation (RAG) pipeline (preferably using OpenAI or other LLMs) that extracts mathematics questions from the RD Sharma Class 12 book. The extracted questions must be in LaTeX format, accurately preserving all mathematical expressions and structures.

Context

The RD Sharma Class 12 textbook is organized by chapters and subtopics. For example:

Chapter 30: Probability

- 30.1 Introduction
- 30.2 Recapitulation
- 30.3 Conditional Probability
- …
- 30.9 Bayes' Theorem

Each topic contains:

- Illustrations (worked examples)
- Practice Exercises
- Theory snippets

Your task is to automatically extract the questions (only) from a given chapter and topic, and format them in LaTeX.

Book link:

WORKABLE AI ASSIGNMENT FOR HIRING

https://drive.google.com/file/d/1BQllRXh5_ID08uPTVfEe0DgmxPUm867F/view?usp=sharing

---

Scope of the Task

You are expected to:

1. Input: Accept the chapter number and topic name as input.
2. Extract: From the corresponding content in the RD Sharma PDF or scanned document:
    - Extract questions only (ignore theory and solutions).
    - Include both practice questions and illustrations if they are question-like.
3. Format: Convert each extracted question into LaTeX, preserving:
    - Mathematical symbols
    - Equations
    - Proper formatting (fractions, square roots, summations, etc.)
4. Output: Return a structured list of LaTeX-formatted questions.

Implementation Guidelines

Core Requirements

- Use LLM-based methods (OpenAI GPT-4, Claude, Mistral, etc.) for understanding and formatting the content.
- Use RAG architecture or hybrid ML+LLM approaches for better accuracy.
- Ensure high fidelity in mathematical expression extraction (consider OCR post-processing if using scanned PDFs).

Bonus Points

- Use LangChain or LlamaIndex for document chunking, retrieval, and context-aware prompting.
- Design a prompting strategy that clearly instructs the LLM to extract only questions and return them in LaTeX.

- Handle OCR noise and ambiguous formatting robustly.
- Provide a simple UI or CLI to run the pipeline with a chapter/topic input.
- Add unit tests to validate LaTeX formatting correctness.

Deliverables

1. Codebase (GitHub repo or zip):
    - Modular Python code (well-documented)
    - Clear setup instructions
    - Requirements.txt or environment.yml
2. Demo Notebook or CLI:
    - Accepts chapter & topic
    - Displays extracted LaTeX questions
3. Sample Output:
    - For at least 1 full topic (e.g., *30.3 Conditional Probability*)
    - A .tex file or Markdown with rendered LaTeX output
4. ReadMe:
    - Approach overview (RAG pipeline, tools used, LLM prompting strategy)
    - Challenges faced and how you addressed them
    - Any assumptions or limitations

Evaluation Criteria

Criterion	**Weight**
Accuracy of extracted questions	60%
Correctness of LaTeX formatting	25%
Use of RAG / LLM techniques	10%
Code structure and modularity	5%

Tools & Resources

WORKABLE AI ASSIGNMENT FOR HIRING

You may use any of the following (or similar):

- LLMs: OpenAI GPT-4, Claude, Mistral, LLaMA
- OCR: Tesseract, EasyOCR, LayoutParser (for scanned PDFs)
- RAG Frameworks: LangChain, LlamaIndex, Haystack
- PDF Parsing: PyMuPDF, pdfplumber
- Prompting: Custom prompt engineering or prompt templates

Notes & Tips

- Focus on precision over quantity — it's better to extract 90% of the questions correctly than 100% with formatting errors.
- Use chunking and context windowing to avoid LLM hallucinations.
- Ensure consistent LaTeX formatting — test by rendering the output.
- Use prompt chaining or output validation to refine results.