TASK: Smart Assistant for Research Summarization
Objective:

EZ has designed this task to evaluate candidates on their ability to build AI-powered tools that go beyond basic automation and demonstrate contextual understanding and reasoning.

The objective is to develop an **AI assistant** that not only reads content from documents but can also understand and reason through it.

You are required to create a document-aware assistant capable of handling both **free-form question answering** and **logic-based question generation** using user-uploaded documents (in PDF or TXT format).

Problem Statement:

Reading through large documents like research papers, legal files, or technical manuals is time-consuming. Traditional summarizers or keyword search tools fall short when it comes to **deep comprehension** and **logical reasoning**.

Build a GenAI assistant that reads user-uploaded documents and can:

- Answer questions that require *comprehension* and *inference*
- Pose logic-based questions to users and evaluate their responses
- Justify every answer with a reference from the document

Functional Requirements:

1. Document Upload (PDF/TXT)

- Users must be able to upload a document in either PDF or TXT format.

- Assume the document is a structured English report, research paper, or similar.

2. Interaction Modes

The assistant should provide two modes after a document is uploaded:

a. Ask Anything

- Users can ask free-form questions based on the document.
- The assistant must answer with contextual understanding, drawing directly from the document's content.

b. Challenge Me

- The system should generate three logic-based or comprehension-focused questions derived from the document.
- Users attempt to answer these questions.
- The assistant evaluates each response and provides feedback with justification based on the document.

3. Contextual Understanding

- All answers must be grounded in the actual uploaded content.
- The assistant must not hallucinate or fabricate responses.
- Each response must include a brief justification (e.g., "This is supported by paragraph 3 of section 1...").

4. Auto Summary (≤ 150 Words)

- Immediately after uploading, a concise summary (no more than 150 words) of the document should be displayed.

5. Application Architecture

- The application should provide a clean, intuitive web-based interface that runs locally.

- You may use any frontend framework (e.g., Streamlit, Gradio, React, etc.) to build the interface.

- You are free to use any Python backend framework (e.g., FastAPI, Flask, Django) to implement the core logic and APIs.

- The focus should be on delivering a seamless and responsive user experience.

Bonus Features (Optional but Encouraged)

- **Memory Handling**: Support follow-up questions that refer to prior interactions to maintain context.

- **Answer Highlighting**: Display the specific snippet from the source document that supports each answer.

Submission Instructions (GitHub):

Your repository should include:

1. `README.md` with:

    - Setup instructions
    - Architecture / reasoning flow
2. Organized source code folder
3. Optional: 2–3 min Loom/YouTube demo walkthrough

Evaluation Criteria:

- **Response Quality (Accuracy + Justification) – 30%**

  Measures how accurately the assistant answers questions and whether it provides clear, document-based justifications for each response.

- **Reasoning Mode Functionality – 20%**

  Assesses the quality of the “Challenge Me” mode — how well the assistant generates logical questions and evaluates user answers with reasoning.

- **UI/UX and Smooth Flow – 15%**

  Evaluates the interface design, ease of navigation, and the overall user experience during document upload and interaction.
- **Code Structure & Documentation – 15%**

  Focuses on how well the code is organized, readable, and documented, including setup instructions and architectural clarity in the README.

- **Creativity / Bonus Features – 10%**

Rewards implementation of innovative features such as memory, snippet highlighting, or enhanced reasoning capabilities beyond the base requirements.

- **Minimal Hallucination & Good Context Use – 10%**

  Checks whether responses are grounded in the document without fabrications, and how well context is maintained across interactions.