Assignment

(Kindly note that the timeline for submission is 24hours)

Objective

Build an agentic application that accepts Text, Images, PDFs, or Audio files, extracts content, understands the user's goal, and autonomously performs the correct task.

If the query or goal is not clear, the agent must ask a follow-up question before acting.

All final outputs must be text-only.

Requirements:

1. Inputs Supported

- Text
- Image (JPG/PNG) → OCR
- PDF (text or scanned) → PDF parsing + OCR fallback
- Audio (MP3/WAV/M4A) → Speech-to-Text + cleanup

2. Agent Behavior (Core Requirement)

A. Intent Understanding

The agent must:

- Extract or transcribe content
- Identify the user's goal

Detect constraints (timing, format, instructions)

B. Mandatory Follow-Up Question Rule

If the input does not contain enough information to determine the task, or if multiple tasks are equally plausible, the agent must not guess. It must:

Ask a short, clear follow-up question like:

- “Could you clarify whether you want a summary or sentiment analysis?”
- “What do you want me to do with this extracted text?”
- “Should I explain this code or rewrite it?”

The agent should ONLY proceed after receiving clarity.

3. Tasks the Agent Must Handle (Autonomously)

1. Image/PDF Text Extraction

- Return cleaned transcript + OCR confidence.

2. YouTube Transcript Fetching

- Detect URL anywhere → fetch transcript (or fallback message).

3. Conversational Answering

- Friendly, helpful response for general questions.

4. Summarization

Output must include:

- 1-line summary
- 3 bullets
- 5-sentence summary

5. Sentiment Analysis

- Label + confidence + one-line justification.

6. Code Explanation

- Explain what code does, detect bugs, and mention time complexity.

7. Audio Transcription + Summary

- Convert audio → text → summarize (same 3 formats).

4.UI Requirements

- One text box
- File upload for Image / PDF / Audio
- Clean, minimal UI
- Text-only output
- Show extracted text + final result
- Chat like UI would work the best

5. Deliverables

- Clean codebase
- Architecture diagram
- FASTAPI + simple UI
- Test cases
- README

6.Evaluation Rubric (100 points)

- Correctness (30) — tasks produce correct outputs across inputs.
- Autonomy & Planning (20) — agent plans sensible, minimal-step workflows and uses fallbacks.
- Robustness (15) — error handling, retries, partial results.
- Explainability (10) — readable plan + logs returned for each run.
- Code Quality & Modularity (10) — clean structure, linting, DI, tests.
- UX & Demo (10) — usable UI, clear demo walkthrough, sample inputs.

Minimum passing score: 75/100.

7. Sample Test Cases

- Audio lecture (5 min) → agent must transcribe → 1-line + bullets + 5-sentence summary + duration.
- PDF (3 pages) containing meeting notes + "What are the action items?" → agent extracts text → finds and returns action items.
- Image screenshot containing code snippet + prompt "Explain" → agent OCRs → detects language → explains + warns about any bug.

8. Bonus (extra credit)

- Multi-agent orchestration: split planner & executor as separate services.
- Cost estimator: approximate token/API costs per plan prior to execution.