Assignment: Agent-Powered Business Assistant

Objective:

Build a **smart agent** that represents a **fictitious business**. It should be able to:

- Answer questions about the business,
- Collect leads (customer name, email, notes),
- Record customer feedback or unanswered questions (tool calls),
- Run as a **chatbot** using Gemini API or OpenAI (your choice),
- Deploy via Gradio.

Project Structure

```
business_bot/
├──  C3 - Assignment.pdf        #Assignment description, guidelines and requireme
├── me/
│   ├── about_business.pdf       # Simulated business profile (PDF)
│   └── business_summary.txt     # Short summary (TXT)
│
├── business_agent.ipynb         # Your main chatbot code (Colab or Jupyter)
├── app.py (optional)           # For deployment
├── .env                       # Contains API keys
└── requirements.txt            # Libraries used
```

Tasks & Guidelines

1. Create Your Business Identity

Invent a fictional business with:

- A name (e.g., “GreenLeaf Tech”),
- A mission (e.g., sustainable AI solutions),
- Services offered (e.g., consulting, APIs),
- Team (e.g., profiles of founders),
- Unique value proposition.

Write this in two formats:

- A **PDF** file named about_business.pdf
- A **summary.txt** describing your business in a few paragraphs

2. Create Tool Functions

You must implement **at least two tool-calling functions**:

- record_customer_interest(email, name, message)
- record_feedback(question) → called when chatbot doesn't know the answer

Both tools should log via a push or print mechanism (e.g., print or file logging).

3. System Prompt & Chat Setup

Create a system_prompt where the agent:

- Stays in character as the business
- Uses summary and PDF content for answering questions
- Logs unknown questions via tool
- Encourages leads to leave contact info

4. Agent Interaction

Use google.generativeai or openai.ChatCompletion:

- Pass user input + chat history + tools to the model
- If model calls a tool, handle it and return response

- Else return generated text

5. Gradio Interface

Add a Gradio ChatInterface so you can demo your bot.

6. (Optional Bonus) Deployment to HuggingFace Spaces

Example Use Cases

- A bakery taking orders and collecting feedback
- A tutoring service helping students pick classes
- A travel agency answering trip questions

- A SaaS product collecting bug reports

Tips

- Reuse structure from Lab 4 but change the context
- Be creative with your business
- Ensure tools are actually called by the model when needed
- Keep tool descriptions short and helpful

Submission Checklist

Submit a Github repo with a video showcasing the functionality, containing the following:

- Business_summary.txt
- About_business.pdf
- Business_agent.ipynb
- Two working tools
- API Key in .env (not uploaded, of course)
- Runs successfully
- (Bonus) Deployed on HuggingFace Spaces