Machine Learning Intern – Assignment Project

Company: ServiceHive
Product: Inflx
Project Title: Social-to-Lead Agentic Workflow

1. Background

ServiceHive is building **Inflx**, an AI-powered platform that converts **social media conversations into qualified business leads**.

Unlike simple chatbots, Inflx agents are designed to:

1. Understand **user intent**
2. Answer **product questions accurately**
3. **Identify high-intent users**
4. Trigger **backend actions** like lead capture

This assignment tests your ability to build a **real-world GenAI agent**, not just a conversational bot.

2. Problem Statement

You are required to build a **Conversational AI Agent** for a fictional SaaS company named:

AutoStream

A SaaS product that provides **automated video editing tools for content creators**.

3. Agent Capabilities (Must-Have)

Your agent **must be able to do ALL of the following**:

3.1 Intent Identification

The agent should correctly classify user intent into:

1. Casual greeting
2. Product or pricing inquiry
3. High-intent lead (ready to sign up)

3.2 RAG-Powered Knowledge Retrieval

The agent must answer questions using a **local knowledge base** (RAG).

The knowledge base must include:

AutoStream Pricing & Features

- **Basic Plan**
    - $29/month
    - 10 videos/month
    - 720p resolution
- **Pro Plan**
    - $79/month
    - Unlimited videos
    - 4K resolution
    - AI captions

Company Policies

- No refunds after 7 days
- 24/7 support available **only on Pro plan**

📌 Store this data in a **local JSON or Markdown file**.

3.3 Tool Execution – Lead Capture

When the user shows **high intent**, the agent must:

1. Ask for:
    - Name
    - Email
    - Creator Platform (YouTube, Instagram, etc.)
2. Call a **mock API function** only after collecting all three values.

Required Function

```
def mock_lead_capture(name, email, platform):
    print(f"Lead captured successfully: {name}, {email}, {platform}")
```

⚠️ The tool must **not be triggered prematurely**.

4. Expected Conversation Flow

Step-by-Step Workflow

1. **Greeting**
    - User: *"Hi, tell me about your pricing."*
2. **Knowledge Retrieval (RAG)**
    - Agent retrieves pricing from the knowledge base
    - Agent responds accurately
3. **Intent Shift**
    - User: *"That sounds good, I want to try the Pro plan for my YouTube channel."*
4. **Lead Qualification**
    - Agent detects **high-intent**
    - Agent asks for:
        - Name
        - Email
5. **Tool Execution**
    - Once all details are collected

- Agent calls `mock_lead_capture()`

5. Technical Requirements

Mandatory Stack

1. **Language:** Python 3.9+
2. **Framework:**
   - LangChain (**LangGraph preferred**)
   - OR AutoGen
3. **LLM:** Any of the following:
   - GPT-4o-mini
   - Gemini 1.5 Flash
   - Claude 3 Haiku
4. **State Management:**
   - Must retain memory across **5–6 conversation turns**
   - Use LangGraph state, memory buffers, or equivalent

6. Deliverables

You must submit a **GitHub Repository** containing:

6.1 Core Code

- Agent logic
- RAG pipeline
- Intent detection
- Tool execution

6.2 requirements.txt

- List **all dependencies** required to run the project

6.3 README.md

Must include:

1. **How to run the project locally**

2. **Architecture Explanation (≈200 words):**
    - Why you chose LangGraph / AutoGen
    - How state is managed
3. **WhatsApp Deployment Question:**
    - Explain how you would integrate this agent with WhatsApp using **Webhooks**

6.4 Demo Video (2–3 Minutes)

Screen recording showing:

1. Agent answering a pricing question
2. Agent detecting high-intent
3. Agent collecting user details
4. Successful lead capture using the mock tool

7. Evaluation Criteria

You will be evaluated on:

1. Agent reasoning & intent detection
2. Correct use of RAG
3. Clean state management
4. Proper tool calling logic
5. Code clarity & structure
6. Real-world deployability