# Data Scientist / AI Specialist Technical Assessment

## Assignment Overview

As part of our hiring process for the Data Scientist / AI Specialist role, we ask you to complete a technical assignment designed to assess your ability to build and productize GenAI solutions. This assignment simulates a real-world scenario where you will be required to automate a document-driven workflow by utilizing agentic frameworks.

---

## Background

The Court Orders department receives scanned court orders containing instructions that must be processed. Assume each court order document contains two main parameters:

- **National ID**: Unique identifier for a person
- **Action**: The requested action (e.g., freeze account, release funds)

The current manual process is as follows:

1. Scan the court order document.
2. Extract the National ID and Action.
3. Check if the person (by National ID) is a customer of the bank.
4. If the person is a customer, take the necessary action.
5. If not, discard the court order.

---

## Assignment Task

Design and implement an **agentic AI solution** that automates this process. Your solution should:

1. **Accept a scanned court order document as input** (PDF).
2. **Extract the necessary fields** (National ID and Action) from the document.
3. **Check if the National ID is in the provided list of bank customers.**
    - The provided list will contain National IDs alongside their corresponding customer IDs.
    - You should design a function that takes a National ID as input and, if present, returns the corresponding customer ID. This function should be used to determine whether the person is a customer at the bank.
4. **If the National ID is not found in the list, discard the court order and present the outcome.**
5. **If the National ID is found, process the extracted action**:
    - Actions should be implemented as dummy functions that take in a customer ID and process the extracted action.
    - A list of hypothetical actions will be provided.
    - **If the extracted action is not in the provided list of actions, stop processing and present an appropriate message to the user.**
6. **Present the outcome to the user**.

---

## Technical Requirements

- **Agentic Framework**: You may use any agentic framework of your choice; **LangGraph is preferred**.
- **Backend Service**: Implement a minimal backend using Flask, FastAPI, or a similar framework of your choice.
    - Expose at least one endpoint:
        - `process_doc` : Accepts document uploads and processes them.
- **Frontend**: Build a simple web page for:
    - Uploading documents
    - Viewing the output
- **Sample Data**: We will provide sample documents, a list of hypothetical customer National IDs, and a list of possible actions.

---

## Deliverables

1. **Flowchart**: Clearly illustrating your solution's architecture and workflow.
2. **Code Repository**: Well-structured, documented code (GitHub or similar).
3. **Demo Interview**: Be prepared to demonstrate your solution and explain your approach in a follow-up interview.

---

## Assignment Timeline

- **You have one week to complete this assignment.**
- Please submit your deliverables as a link to your code repository and attach your flowchart.
- Prepare to demo your solution and discuss your approach during the interview.

---

## Additional Notes

- You may use any open-source or commercially available AI tools, libraries, or frameworks.
- The use of ChatGPT or similar AI tools for code generation, problem-solving, or workflow design is allowed and **strongly encouraged**; however, you are expected to fully understand and be able to explain all parts of your solution.

---

We look forward to seeing your solution!