Prompt-Driven Agent for Generating Work Instructions for New Car Models

1. Objective

To develop an AI-powered assistant that generates detailed, step-by-step work instructions for assembly line workers, technicians, or quality inspectors when a new car model is introduced. The solution should use prompt engineering with a Large Language Model (LLM)—such as Azure ChatOpenAI—without relying on retrieval-augmented generation. The goal is to deliver clear, consistent, and customizable instructions for new or updated automotive manufacturing processes.

2. Problem Statement

When launching new car models, automotive manufacturers must quickly provide accurate and standardized work instructions for various assembly, inspection, or calibration tasks. Traditionally, this process involves significant manual effort and coordination between engineering, documentation, and production teams—often resulting in inconsistent or delayed instructions on the factory floor.
This exercise explores using a prompt-driven LLM agent to automatically draft work instructions from high-level descriptions or engineering notes, accelerating the deployment of new models and reducing human error.

3. Inputs / Shared Artifacts

1. Azure OpenAI Resource:

   API key, endpoint URL, and deployment name (to be set as environment variables).

2. You are provided with the following **mock engineering notes** or high-level task descriptions for a new car model.

Example Input Data:

id	task_description
1	"Install the battery module in the rear compartment, connect to the high-voltage harness, and verify torque on fasteners."
2	"Calibrate the ADAS (Advanced Driver Assistance Systems) radar sensors on the front bumper using factory alignment targets."
3	"Apply anti-corrosion sealant to all exposed welds on the door panels before painting."
4	"Perform leak test on coolant system after radiator installation. Record pressure readings and verify against specifications."
5	"Program the infotainment ECU with the latest software package and validate connectivity with dashboard display."

4. Expected Outcome

For each input task description, the system should output clear**, step-by-step work instructions** suitable for use on the shop floor, including safety reminders, necessary tools or equipment, and acceptance criteria when relevant.

Example Input → Output:

Task Description	Generated Work Instruction
"Install the battery module in the rear compartment, connect to the high-voltage harness, and verify torque on fasteners."	1. Ensure the power is off and wear high-voltage PPE. 2. Place battery module in rear compartment. 3. Connect to high-voltage harness securely. 4. Use a calibrated torque wrench to tighten fasteners to specified values.

Task Description	Generated Work Instruction
	5. Verify all connections; record installation in logbook.
"Apply anti-corrosion sealant to all exposed welds on the door panels before painting."	1. Clean all welding areas on door panels. 2. Apply approved anti-corrosion sealant to each exposed weld. 3. Allow proper curing time. 4. Inspect coverage before sending panels to paint booth.

5. Concepts Covered

- Azure OpenAI API usage
- Prompt Engineering
- Analysis and problem solving
- NLP Task - Text Generation

6. Example: Implementation using Azure ChatOpenAI

```
import os
from openai import AzureOpenAI

# Step 1: Mock Input Data
task_descriptions = [
    "Install the battery module in the rear compartment, connect to the high-voltage harness, and verify torque on fasteners.",
    "Calibrate the ADAS (Advanced Driver Assistance Systems) radar sensors on the front bumper using factory alignment targets.",
    "Apply anti-corrosion sealant to all exposed welds on the door panels before painting.",
    "Perform leak test on coolant system after radiator installation. Record pressure readings and verify against specifications.",
    "Program the infotainment ECU with the latest software package and validate connectivity with dashboard display."
]

# Step 2: OpenAI Azure Client Setup
client = AzureOpenAI(
    api_version="2024-07-01-preview",
    azure_endpoint=os.getenv("AZURE_OPENAI_ENDPOINT"),
    api_key=os.getenv("AZURE_OPENAI_API_KEY"),
)

deployment_name = "gpt-4o-mini"  # Your deployment name
```

```
def generate_instruction(task):
    prompt = f"""
You are an expert automotive manufacturing supervisor. Generate step-by-step
work instructions for the following new model task. Include safety
precautions, required tools (if any), and acceptance checks. Write in clear,
numbered steps suitable for production workers.

Task:
\"\"\"{task}\"\"\"

Work Instructions:
"""
    response = client.chat.completions.create(
        model=deployment_name,
        messages=[{"role": "user", "content": prompt}],
        temperature=0,
    )
    return response.choices[0].message.content.strip()

# Step 3: Example Run
for task in task_descriptions:
    instructions = generate_instruction(task)
    print(f"\nTask: {task}\nWork Instructions:\n{instructions}\n")
```

7. Final Submission

- The complete code (.py or .ipynb) is used to generate work instructions.
- A table or file showing all mock task descriptions and their corresponding AI-generated instructions. (5 samples)
- (Optional) A short reflection on how prompt-driven AI agents can standardize and accelerate work instruction creation in automotive manufacturing.