Homework 4 – Building an OpenAI Plugin with Azure

Objective

The goal of this homework is to design, implement, and deploy a **simple OpenAI-compatible plugin** using **Azure OpenAI** and **Azure cloud services**.

You will expose a web API that:

- Accepts a user prompt
- Calls Azure OpenAI to generate a response
- Returns the result to the caller

The focus is on **integration, deployment, and API design**, not on building a complex application.

What Is an OpenAI Plugin (In This Homework)

For this assignment, a *plugin* is:

- A **web API** (HTTP-based)
- That accepts structured requests (prompt + parameters)
- Uses **Azure OpenAI** internally
- Returns AI-generated results

You do **not** need to integrate with ChatGPT UI directly.
You are building the **backend service** that *could* be used as a plugin.

Plugin Ideas (Choose One)

You may choose **one** of the following ideas or propose a similar one of comparable complexity.

Suggested Plugin Ideas

- **Word Inventing Plugin**
  Input: description of a concept
  Output: a new word to name that concept
- **Text Summarizer Plugin**
  Input: long text
  Output: short summary
- **Grammar & Style Checker**
  Input: text
  Output: corrected / improved text
- **Code Explanation Plugin**
  Input: code snippet
  Output: explanation of what the code does

All plugins must use **Azure OpenAI**, not public OpenAI APIs.

Assignment Requirements

1. API Implementation

Your application must:

- Expose the following **HTTP endpoints:**
  - GET /info
    - Returns a description of what your plugin is supposed to do
    - e.g. “Summarizes the text received through the prompt”
  - `POST /prompt`
    - Accept a request body containing the user prompt (text)
    - Calls **Azure OpenAI** with this prompt
    - Returns the model’s response to the client

You may use any backend technology you feel comfortable with.

2. Azure OpenAI Integration

Your application must:

- Use an **Azure OpenAI deployment**
- Authenticate using **Azure-supported authentication** (API key or managed identity)
- Use a deployed model suitable for text generation

The model name and deployment details must be configurable (Application Settings / Environment Variables)

3. Error Handling

Your API must include explicit error handling:

- Invalid or empty prompt
- Azure OpenAI request failure
- Internal server error

Errors must:

- Return appropriate HTTP status codes
- Return a clear error message in the response body

You must document these cases in your README.

4. Deployment to Azure

Your application must be **deployed to Azure** and publicly accessible. You can use Azure App Service as you did for HW2.

No database is required.

5. Configuration & Secrets

Your application must:

- Not hardcode secrets in source code
- Use Azure-supported configuration mechanisms
- Be configurable through environment variables or app settings

Deliverables

You must submit:

1. Source Code

- Submit your application source code to your Cloud Computing Github Repository in a dedicated HW4 folder
    - Add **StefanNedelcu** as contributor if you didn't already

2. Deployment Script

This should be a bash / shell script using Azure CLI commands. It must contain:

- Commands that create your resources
- Deployment commands
- Configuration commands

3. README.md

Your README must include:

- Public API URL
- Description of the plugin functionality
- Example request and response
- Azure OpenAI model and deployment used
- How to trigger an error
- Short explanation of how and where the app is deployed

Submission Instructions

Send me a private message on Teams with:

- GitHub repository link
- ZIP archive of the repository

📅 **Deadline:** SUNDAY, January 18th, 23:59