Generative AI Engineer - Multi-Agentic Coding Framework Exercise

Problem Statement

Develop a Multi-Agentic Framework using AutoGen where agents perform the following tasks in a structured manner:

• Requirement Analysis Agent - Takes input in natural language and refines it into a structured software requirement.

• Coding Agent - Converts the refined requirements into functional Python code.

• Code Review Agent - Reviews the generated code for correctness, efficiency, and security. If improvements are needed, it provides feedback for re-iteration.

• Documentation Agent - Generates proper documentation for the developed code.

• Test Case Generation Agent - Creates unit tests and integration test cases for the developed code.

• Deployment Configuration Agent - Generates a deployment script to deploy the developed code.

• Streamlit UI Agent - Creates a basic Streamlit-based user interface to interact with the multi-agent system.

The agents should work collaboratively, meaning each agent's output feeds into the next agent in the pipeline.

Technical Expectations

Tech Stack:

• Python

• AutoGen for multi-agent coordination

• LLMs (OpenAI GPT-4, Claude, or Llama models via API)

• Streamlit for UI development

Key Functionalities:

• The requirement should be processed iteratively.

• If the code fails review, it must be sent back to the Coding Agent for improvements.

• Documentation should be clear and structured, including code explanation, function definitions, and usage.

• The framework should generate at least one functional test case per module.

• The deployment configuration should include a basic script for setting up and running the project.

Deliverables

• Python Codebase implementing the multi-agent system.

• Documentation (Markdown or PDF) covering:
  - Overview of the implemented agents.
  - Workflow and architecture.
  - How to set up and run the system.

• Test Cases with execution results.

• Streamlit UI Implementation for interacting with the system.

• README File with step-by-step instructions to run the project.

Evaluation Criteria

• Correctness: Does the framework generate working code based on requirements?

• Modularity: Are agents properly separated and interacting effectively?

• Code Quality: Is the generated code efficient, readable, and maintainable?

• Error Handling: Does the system gracefully handle incorrect inputs and failures?

• Documentation Quality: Is the generated documentation structured and comprehensive?

• Testing Coverage: Are appropriate test cases generated and executed?

• UI Implementation: Is the Streamlit interface functional and user-friendly?

Time Estimate

Candidates are expected to complete the exercise within one week.

Submission Instructions

• Submit the completed project as a GitHub repository or a zip file.

• Ensure all required artifacts are included in the submission.