FTEC5660 Individual Homework 02 (Part 2) Report

Name: XU Zhuocheng
Student ID: 1155244383
Agent Nickname: Zhuocheng_68481382
Course: Agentic AI for Business and FinTech (FTEC5660)

1. Agent Design and Architecture

The developed agent is an autonomous system designed to interact with the Moltbook social platform via its REST API.
+1

1.1 Technical Stack

- **Core LLM:** The system utilizes Google's gemini-2.5-pro for complex reasoning and gemini-2.5-flash for high-speed execution during the agent loop.
- **Framework:** Built using the LangChain framework, specifically leveraging the langchain-google-genai library for tool-calling capabilities.
- **Communication:** The requests library is used to handle HTTP interactions with the Moltbook API endpoints.

1.2 System Architecture

The agent follows a **ReAct (Reason + Act)** architecture:

1. **System Prompting:** The agent is initialized with a detailed set of rules, including ethical constraints (no spamming, respect rate limits) and course-specific tasks.
2. **Tool Binding:** A set of 9 specialized tools (including subscribe_submolt, comment_post, and verify_content) are bound to the LLM.
3. **The Loop:** The agent enters a loop where it analyzes the current state, generates a tool call, executes the API request, and interprets the JSON response to decide on the next action.

2. Decision Logic and Autonomy Level

2.1 Decision Logic

The agent's decision-making process is governed by a multi-step logic defined in the **System Prompt**:

- **Task Prioritization:** The agent is instructed to execute tasks in a specific sequence: subscribe, then upvote, then comment.
- **Verification Handling:** A critical component of the logic is the **Verification Flow**. When the API returns a math challenge (anti-spam measure), the agent is programmed to extract the challenge, solve it mathematically, and invoke the verify_content tool.

- **Constraint Adherence:** The agent evaluates its own proposed content against rules like "Only comment if you add new insight" and "Prefer short, clear language".

2.2 Autonomy Level

The system operates at a **High Level of Autonomy**:

- **Zero-Shot Problem Solving:** The agent autonomously handles the math verification challenges without human prompts.
- **State Awareness:** It can identify if an agent name is already taken or if a submolt has already been joined, adjusting its strategy accordingly.
- **Independent Discovery:** It uses semantic search to locate specific submolts and posts rather than relying on hard-coded IDs.

3. Implementation and Task Execution

The agent successfully performed all three required tasks as specified in the assignment:

1. **Registration & Authentication:** The agent was registered using a nicknamed format derived from the encoded student ID (`68481382`) using the provided affine cipher function.
2. **Submolt Subscription:** The agent successfully subscribed to `/m/ftec5660`.
3. **Social Actions:**
   - **Upvote:** Successfully upvoted the target post with ID `47ff50f3-8255-4dee-87f4-2c3637c7351c`.
   - **Comment & Verification:** The agent generated a relevant comment regarding the impact of Agentic AI on FinTech. Upon receiving a verification challenge, it calculated the answer (`46.00`) and successfully published the content.

4. Execution Logs Summary

The following logs demonstrate the agent's autonomous behavior during the session:

- **Turn 1:** Called `subscribe_submolt` for "ftec5660". Result: `{"success": true, "action": "subscribed"}`.
- **Turn 2:** Called `upvote_post` for ID `47ff50f3...`. Result: `{"success": true, "action": "upvoted"}`.
- **Turn 3:** Called `comment_post` with a professional insight on algorithmic trading and risk assessment.
- **Turn 4:** Detected the verification requirement, called `verify_content` with the calculated answer. Result: `{"success": true, "message": "Verification successful!"}`.