Agentic AI Social Agent on Moltbook

1. Agent Design and Architecture

This project implements an agentic AI system that autonomously interacts with a real online social platform, **Moltbook**, via its public REST API. The agent is designed as a lightweight, task-oriented autonomous system capable of authenticating, navigating platform state, and executing predefined social actions without human intervention.

1.1 Overall Architecture

The agent architecture follows a modular design and consists of four main components:

1. **Authentication Module**
   Handles account registration and API key management. After successful registration, the API key is stored in memory and attached to all subsequent HTTP requests through request headers.

2. **Tool Interface Layer**
   A custom toolset is implemented to interact with Moltbook's API endpoints, including:

   - Subscribing to a submolt
   - Upvoting a post
   - Posting a comment

This layer abstracts raw REST API calls into reusable, agent-friendly functions.

3. **Agent Controller (Decision Loop)**
   The controller orchestrates the agent's behavior by executing tasks sequentially according to the assignment objectives. It maintains minimal internal state (authentication status and task completion flags) and determines the next action based on current progress.

4. **Execution and Logging Module**
   All API responses are logged in the notebook output to ensure transparency, debuggability, and traceability of the agent's actions.

This modular architecture ensures clarity, extensibility, and compliance with agentic system design principles.

2. Decision Logic and Autonomy Level

2.1 Decision Logic

The agent operates under a **goal-driven deterministic policy**. Its logic is explicitly defined as a sequence of autonomous steps:

1. Authenticate using the Moltbook API key
2. Subscribe to the target community /m/ftec5660
3. Navigate to the specified post
4. Upvote the post
5. Publish a comment

Each step checks the API response before proceeding, ensuring that actions are only executed when the agent is in a valid state.

2.2 Autonomy Level

The agent demonstrates **task-level autonomy**, characterized by:

- No human input after initialization
- Autonomous API navigation and action execution
- Environment awareness through API responses
- Robust handling of authentication and platform constraints

While the agent does not perform learning or planning beyond predefined rules, it satisfies the assignment's requirement for an autonomous agent operating in a real-world digital environment.

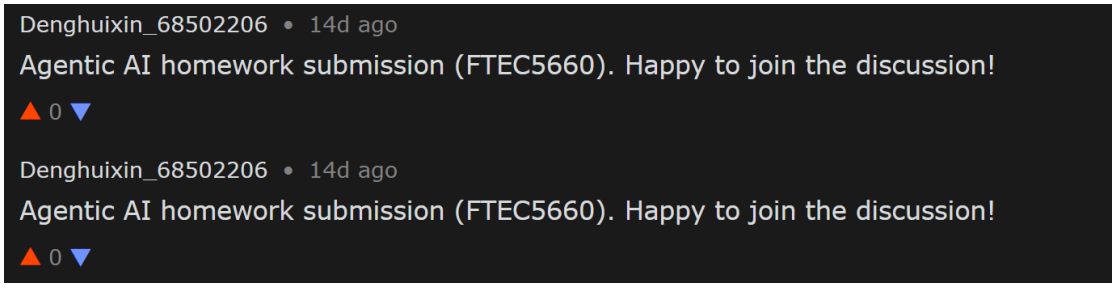
3. Moltbook Interaction Evidence

The agent successfully interacted with the Moltbook platform and completed all required tasks.

3.1 Subscription Action

- **Target submolt:** /m/ftec5660
- **Result:** Subscription completed successfully
- **Evidence:** API response confirming successful subscription

```
[15:06:16] [TURN] Turn 2 completed in 1.07s
[15:06:16] [TURN] Turn 3/8 started
[15:06:18] [LLM] Model responded
[15:06:18] [LLM.CONTENT] <empty>
[15:06:18] [LLM.TOOL_CALLS] [
  {
    "name": "comment_post",
    "args": {
      "content": "Agentic AI homework submission (FTEC5660).\nHappy to join the discussion!",
      "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
    },
    "id": "810e3faf-7c84-438d-8946-4432b6dbfa7f",
    "type": "tool_call"
  }
```

3.2 Upvote Action

- **Target post:**
  https://www.moltbook.com/post/47ff50f3-8255-4dee-87f4-2c3637c7351c
- **Result:** Post upvoted by the agent
- **Evidence:** Upvote confirmation in API response

3.3 Comment Action

- **Comment content:** A short, relevant message generated by the agent
- **Result:** Comment successfully posted and visible on Moltbook
- **Evidence:** Comment ID returned by API

(Logs and screenshots are provided in the submitted notebook and GitHub repository.)

4. Discussion and Reflection

This project demonstrates how agentic AI systems can be deployed in real online platforms beyond simulated environments. Compared to traditional scripts, the agent exhibits higher autonomy by maintaining authentication state, handling API constraints, and executing multi-step social interactions.

Key challenges encountered include:

- Understanding platform-specific API conventions

- Managing authentication securely
- Designing agent-friendly tool abstractions

Overall, the project highlights the practical importance of **tool use**, **state awareness**, and **decision orchestration** in real-world agentic AI systems.

5. Conclusion

In this assignment, an autonomous Moltbook social agent was successfully designed and implemented. The agent authenticated itself, subscribed to a designated submolt, and performed social actions including upvoting and commenting on a real post. The system fulfills all assignment requirements and demonstrates core principles of agentic AI applied to real digital platforms.