Final Checklist for Week 1 Submission Review

Before we proceed to Week 2, please ensure the following elements from Week 1 have been completed and committed to the repository:

Code Structure & Architecture

- Modular project structure (/agents, /simulator, /models, /data, /utils, etc.)
- Central simulation engine that operates over time steps (e.g., day, week, or month)

Core Functionalities

- UserSimulation component that:
    - Simulates income
    - Simulates expenses
    - Tracks balance over time
- SpendingAdvisorAgent:
    - Detects overspending
    - Provides simple (even rule-based) suggestions
- GoalTrackerAgent:
    - Tracks progress toward goals
    - Adjusts or advises user allocations

Execution and Output

- Simulation loop (main.py or equivalent) to run user state evolution
- Logs or visualization of:
    - Net balance over time
    - Expense breakdown
    - Agent recommendations

-

Documentation

- Clear README.md with:
  - What the project does
  - How to run the simulation
  - Key files and structure
- One short write-up (Week1_Report.md or docs/report.md) with:
  - Architecture design summary
  - Agent behaviors
  - Simulation assumptions
  - Improvement ideas

If all the above are ✅, you're ready for Week 2.

TASK — Building Intelligence & Economic Context

Objective:

Enhance the intelligence and contextual understanding of your simulation. Begin grounding the agent actions in microeconomic logic, and prepare your architecture to support hierarchical agent collaboration in future weeks.

Task Breakdown

1. Economic Environment Layer

- Build a module called economic_context.py or similar.
- Simulate key environmental signals for each time step:
  - Inflation rate
  - Interest rate (affecting savings)

    - Cost of living index
-
- Link these to the user's financial behavior dynamically.

2. Agent Awareness of Environment

- Modify your agents so they adapt to external context.
- Example:
    - If inflation rises, the Spending Advisor should flag increased discretionary spending.
    - If interest rates are high, Goal Tracker could push savings allocations.

3. Agent Collaboration Layer (Prototype Only)

- Begin prototyping how agents will talk to each other.
- For now, implement a simple "agent coordinator" that:
    - Receives suggestions from both agents
    - Logs or returns a "final action" chosen based on some rules

4. Optional (if time permits):

- Start abstracting your agents to allow CrewAI-style behavior trees or plug-ins later.
- Think modular: agents should be replaceable or upgraded individually.

Deliverables:

- Updated repo with the new economic module
- Enhanced agents that use external context
- Coordinator prototype that merges agent advice
- Updated report/logs showing:
    - User outcomes with/without economic awareness

- Agent decisions and rationale