Gandhar – Task 2: AI Simulation Dashboard + Multimodal Insights Integration

Checklist:

- Expand the existing agent simulation interface to accept processed data (text/audio/image/video) from other modules.
- Design agent logic to simulate multi-domain interactions (finance, education, wellness).
- Build an AI-aware Streamlit dashboard for agent insight tracking and simulation review.
- Integrate Uniguru/LLM responses as agent dialogue.
- Create real-time visual elements: goal progress, confidence scores, sentiment, tags.
- Add time-series tracking for user interactions and decisions.
- Modularize for integration into the Gurukul + UniGuru system.

Objective:

Develop the Agent Intelligence Simulation Interface (v2) — a smart dashboard where users (and team) can visualize how the agents interpret, respond, and evolve across tasks like:

- Reading multimodal input
- Making decisions (e.g. financial/wellness)
- Simulating thought chains and predictions
- Generating human-style feedback loops

Steps:

1. Agent + Data Pipeline Expansion:

- Connect with Dev 4's output (processed OCR/docs + AI response).
- Ingest into the agent framework you previously built.
- Log each interaction with metadata (timestamp, type, agent confidence).

2. Streamlit Dashboard:

- Create a live dashboard with:
    - User interaction timeline
    - Agent decision explanations (LangChain-style)
    - Confidence scores, mood/tone tracking (optional sentiment model)
    - LLM model used per query
    - Response + audio playback (if TTS used)

3. Simulated Decision Logic:

- Add logic for multiple decision nodes:
    - “What financial advice would this input trigger?”
    - “What educational resource should this user be shown next?”
    - “Should this trigger a wellness check-in?”
- Output as agent-style response cards.

4. Dynamic Agent Identity Rendering:

- Visually distinguish between agents (FinancialCrew, EduMentor, WellnessBot).
- Each agent should have color themes, icons, and thought-trail indicators.

5. Real-Time Component Hooks:

- Allow React frontend or Node backend to fetch:
    - Live agent data (via API or socket)
    - Visual metadata (e.g. time spent, emotional tone)
    - Simulation snapshots or session replays

6. Modular Packaging:

- Structure this dashboard and engine to work standalone (as a testing tool), but easy to integrate into:
    - Gurukul's AI learning flow
    - UniGuru's Flutter interface (via JSON APIs or iframe/embedding)

Deliverables:

- GitHub repo with:
    - Streamlit dashboard + backend logic
    - README for setup + integration notes
- Demo video of an agent simulation (input to response with visual trail)
- Exportable agent logs as JSON
- Integration-ready APIs or modules for Gurukul + UniGuru