Gandhar's next logical task should transition the Agent Simulation Dashboard from internal tool to platform-integrated infrastructure — turning the simulation system into a live, embeddable agent orchestration + analytics layer for both Gurukul and UniGuru.

Here's the next task:

Gandhar – Task 3: Unified Agent Orchestration Layer + Live Analytics API

Objective:

Create a scalable, production-ready Agent Orchestration & Analytics Service that powers real-time agent behavior across the Gurukul and UniGuru platforms. This includes syncing agent simulation data (from v2 dashboard), enabling live updates to the frontend, and exposing insights via API.

Checklist:

1. Agent Orchestration Core (Live Mode)

- Convert simulation agents into live orchestrators that:
  - Accept real user input (via API or WebSocket)
  - Route input to correct agent (EduMentor, WellnessBot, etc.)
  - Execute LLM-powered decision + response generation
  - Log all actions (including model used, confidence, metadata)

2. Real-Time Insights API

- Expose endpoints for:
  - /agent-insight/live – returns current agent's status, mood, model, confidence
  - /agent-log/timeline – returns full user interaction log
  - /agent-snapshot/{session_id} – returns session state at specific timestamp
- Optional: Add WebSocket or long-polling for real-time frontend updates

3. Streamlit → Modular API Microservice

- Refactor simulation dashboard:

    - Extract core logic into FastAPI or lightweight Node service
    - Keep Streamlit dashboard for internal review/testing only
    - Ensure all simulation logic can run headless (API-only)

4. Session Syncing with Supabase Auth

- Ensure agent sessions are user-linked via Supabase JWT
- Store session IDs, metadata in MongoDB/Supabase
- Prepare hooks for Gurukul's gamified UI and UniGuru's lesson planner

5. Simulation → Production Bridge

- Add a toggle to run agent in:
    - Simulation Mode (test inputs, fake time)
    - Live Mode (actual user data + responses)
- Use shared database/log structure to unify both views

6. Frontend/Node Hooks for Dev 5

- Deliver React-consumable JSON payloads:
    - Agent name, avatar, color theme
    - Current decision + response
    - Confidence score, goal, emotion (if available)
- Add /replay-session/{session_id} API trigger for session playback

7. Deliverables

- GitHub repo:
    - Agent orchestration service (FastAPI/Node)
    - JSON API docs

    - Example React fetch for real-time hooks
- Refactored dashboard (minimal Streamlit UI for dev/internal)
- Integration guide for Gurukul (React) + UniGuru (Flutter)
- Optional: Exportable agent performance reports as CSV/JSON

Summary Output:

You now turn the agent simulation into a modular, production-ready backend service with:

- Live user interaction handling
- Frontend-ready insight APIs
- Realtime metadata + analytics stream
- Clean hooks for EduMentor, FinancialCrew, and other agents