Gandhar – Task Set 2 (Data & AI Agent Logic, Streamlit Dashboard)

Objective: Lay the groundwork for the AI-backed learner insights dashboard and prepare a plug-and-play data agent pipeline using Streamlit + Python + Supabase.

Task 1: Supabase Integration with Python Backend

- Set up Python-side access to Supabase (use supabase-py SDK).
- Connect to:
  - User profile table
  - Lesson completion records
- Build a utility module (supabase_utils.py) to:
  - Pull user learning data
  - Log user actions (mock for now)
  - Push feedback from AI agents later

Task 2: AI Agent Scaffold – Learning Companion (v1)

- Create a Python script for an AI Learning Agent that:
  - Reads user lesson history (from Supabase)
  - Outputs a basic feedback summary (stub logic for now)
  - Example: “You’ve completed 3 lessons. Want to continue or revise?”
- Keep it modular — later we’ll plug in LoRA/Ollama model responses.

Task 3: Streamlit Dashboard MVP – Gurukul Analytics

- Build a basic dashboard in Streamlit:
  - Show user’s lesson history
  - Display graphs (e.g., pie chart of progress, time spent, completion %)
  - Include placeholders for future insights from AI agents
- Use Plotly or Altair for visuals.

Task 4: Anomaly Detector Agent – Behavior Analysis (v0.1)

- Create a Python script to:
  - Simulate detecting a drop in learning consistency (e.g., 3-day inactivity)
  - Flag it and generate a small advice output ("Try revising lesson 2 or explore new topic")
- Save this flag in Supabase for dashboard display.

Task 5: Modularization + Repo Clean-Up

- Split code into modules:
  - dashboard.py, agent_trend.py, agent_anomaly.py, supabase_utils.p
- Add a README.md with how to run each component locally.

Deliverables

- GitHub repo with clean structure
- Streamlit dashboard running locally
- Sample data agent responses
- README with setup instructions
- Optional: Loom walkthrough (2–3 min)