📚 Metatron Tutor Backend System: Engine-Oriented Design Blueprint

# ✅ Overview

This document captures the final, detailed system blueprint for the Metatron Tutor Brain Backend. It incorporates architecture, module interactions, lifecycle flows, and database logic based on extended discussions, diagrams, and prompting strategies.

# 🧠 Core Backend Engine Breakdown

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 🔧 Component | 🧠 Function | 🔌 API Role | 🗃️ Database Tables | 💬 Used During |
| Tutor Brain Backend Engine | Orchestrates tutor sessions, coordinates all engines, routes logic | None (internal orchestration layer) | None | Global - every step |
| Prompt Compiler | Builds prompt from master, goal, trackers, insights | /prompt/compiled/{session\_id} | None (reads only) | Session Start |
| Session Starter Engine | Triggers tutoring session | /session/start | sessions, users, goals | Session Start |
| Goal Plan Tracker | Stores structured content plan | /goal-plan/{goal\_id} | goal\_plans, milestones | Prompting, Progress Comparison |
| Progress Summary Engine | Analyzes chat to update learner progress | /progress/summarize | chat\_logs, progress\_snapshots | Mid & End Session |
| Tutor Cycle Tracker | Tracks progression through life cycle | /tracker/{goal\_id} | cycle\_stages, cycle\_snapshots | During Session |
| Dynamic Placeholder Engine | Injects dynamic runtime values into prompts | (Internal only) | None | During Prompt Compilation |
| Report Generator | Generates session summaries | /report/generate | reports, summaries | Session End |

# 🔁 Backend Lifecycle Flow

1. User uploads file and defines goal  
 → System generates Goal Plan (topics + milestones) into a central table  
 → Trackers initialized:  
 • Goal Plan Tracker History (user-level progress snapshots)  
 • Tutor Cycle Tracker (stage status across content structure)  
 • Content Plan Snapshot (static content + structure plan per upload)  
  
2. User starts tutoring session  
 → Session Starter → Prompt Compiler → Compiled session prompt using:  
 • Master Prompt  
 • Goal Prompt  
 • Last recorded snapshots (progress + cycle + content tiering)  
 • Dynamic placeholders (user data, session ID, context)  
  
3. Tutor session active  
 → Chat logs stream to DB  
 → Progress Summary Engine runs every N messages or idle timeout:  
 • Updates all trackers based on learning insights  
  
4. Tutor adapts mid-session using tracker data  
5. On session end: snapshot stored, report generated  
6. Next session: last tracker states injected into session prompt