



Code For Good - Agentic AI Infrastructure & Framework

SARITHA IVATURI CODEFORGOODUS@GMAIL.COM

TABLE OF CONTENTS

• Detail Design	2
• FTN Use Case	2
• Core Intelligence Needed	2
• High-Level Agentic Workflow:	2
• Orchestration Pattern Used	2
• Detailed Agentic AI Workflow	2
• Step 1: INGEST	2
• Step 2: PLAN	3
• Step 3: GROUND	3
• Step 4: SCORE	3
• Step 5: TRIAGE (Human-in-the-loop)	4
• Step 6: REFLECT	4
• Step 7: ACT	4
• Step 8: LEARN	5
• Development Approach & Tech Stack	5
• Architecture	5



- **Detail Design**
- **FTN Use Case**
 - **Goal:**
Enable admins to **continuously understand child progress**, mentor effectiveness, and **proactively trigger support events** using an Agentic AI system, not just dashboards.

-
- **Core Intelligence Needed**
 - Track child-mentor 1:1 journeys
 - Interpret events, activities, badges, and milestones
 - Predict who is on-track, at-risk, or excelling
 - Recommend interventions (support events, mentor nudges)

- **High-Level Agentic Workflow:**

-
- **Orchestration Pattern Used**
 - Ingest → Plan → Ground → Score → Reflect → Act → Learn
 - This loop runs:
 - On every event
 - On weekly admin review
 - On mentor or child inactivity signals

- **Detailed Agentic AI Workflow**

-
- **Step 1: INGEST**
 - What comes in
 - Child profile (age, goals, badge targets)
 - Mentor assignment & interaction logs
 - Event participation (sessions attended, activities completed)
 - Badge rules (targets, thresholds, timelines)
 - Agents Involved
 - Extractor Agent
 - Extracts structured data:
 - Child ID, mentor ID
 - Badge type, target score
 - Event timestamps



- Progress metrics (e.g., arrows hit vs target)
- Memory Updated
 - Short-term: current session context
 - Long-term: historical progress per child
 - Episodic log: “Child X completed Event Y, hit 60% of target”

- **Step 2: PLAN**

- Planner / Controller (Deterministic)
Decides:
 - “Is this a progress update?”
 - “Do we need a scoring pass?”
 - “Is human review required?”
- Example Decisions
 - If new event → Score progress
 - If weekly batch → Evaluate cohorts
 - If score borderline → Trigger reflection

- **Step 3: GROUND**

- **RAG Answerer Agent**
Grounds decisions using:
 - Badge definitions
 - Program rules (time limits, fairness constraints)
 - Child-specific goals
- **Example:** “For Bronze Archery Badge, child must hit 70% targets across 3 events in 30 days.”

- **Step 4: SCORE**

- Scorer Agent
Computes:
 - Progress % toward badge
 - Momentum (improving / stagnant / declining)
 - Risk level (behind schedule or on track)
- Outputs
 - Progress score: 0.0 — 1.0
 - Status label:
 - 🎯 On Track
 - ⚠️ At Risk
 - 🏆 Achieved



- **Step 5: TRIAGE (Human-in-the-loop)**

- Based on defined bands:

Score	Action
> 0.85	Auto-award badge
0.65 – 0.85	Human confirm (mentor/admin)
< 0.65	Escalate → support intervention

- Guardrails Applied
 - Child safety rules
 - No automated negative labeling
 - Mentor + admin review for escalations

- **Step 6: REFLECT**

- Reflection Agent
Analyzes:
 - Why is the child behind?
 - Mentor engagement low?
 - Event access issues?
 - Skill vs opportunity gap?
- Writes to Episodic Memory
 - “Child A missed 2 sessions due to scheduling conflict.”

- **Step 7: ACT**

- Dispatcher Agent
Triggers actions:
 - Update Admin Dashboard
 - Notify mentor (Slack/Email)
 - Recommend support events
 - Suggest badge ceremony or recognition
- Examples
 - “Create beginner practice event”
 - “Nudge mentor to reschedule session”
 - “Recommend peer-group event”



- **Step 8: LEARN**

- Weekly Evaluation Loop
 - Accuracy of predictions
 - Badge fairness review
 - Bias & accessibility checks
 - Mentor effectiveness analysis
- Memory Update
 - Adjust scoring weights
 - Improve future recommendations

•

- **Development Approach & Tech Stack**

- Initial automation via API enabled Flutter / Node.JS/MYSQL stack on AWS
- Understand usage patterns
- Build multiple AI agents for specific workflows
- Test inferences for acceptable accuracy rates / reinforcement training
- Agentic Architecture:
 - Planner/Controller (small prompt, deterministic): decides next step.
 - Workers/Tools (narrow prompts): - Extractor: pulls structured fields from docs. - RAG Answerer: cites policy. - Scorer: computes fit/risk. - Dispatcher: writes to CRM, sends Slack/email.
- Memory: short-term (conversation thread), long-term (vector store of policies/past decisions), episodic log (for reflection).
- Guardrails: PII filter, policy constraints, allowlist of tools, human-in-the-loop for high-risk actions.
- Orchestrator: state machine / graph (e.g., LangGraph, Semantic Kernel planners, or your own FSM).
- Observability: traces, tokens, latency, tool errors, eval scores (logged per step).
- Implement Orchestration pattern: Ingest, Plan, Ground, Score, Reflect, Act, Learn
- Define Triage bands with a human in the loop: >0.85 Auto route; 0.65-0.85 Human confirm; <0.65 Escalate
- Weekly evaluation with safety/Privacy and Fairness checks

- **Architecture**



Agentic AI Architecture

