Course Companion FTE

Agent Factory Hackathon IV Document

Building a Digital Full-Time Equivalent Educational Tutor

A Dual-Frontend Architecture with Zero-Backend-LLM Default
→ Selective Hybrid Intelligence

Version 1.0 — January 2026

Prepared for Panaversity Agent Factory Development Hackathon

AI Agent Factory Presentation:
https://docs.google.com/presentation/d/1UGvCUk1-O8m5i-aTWQNxzg8EXoKzPa8fgcwfNh8vRjQ/edit?usp=sharing

AI Agent Factory Textbook: https://agentfactory.panaversity.org/

Agent Factory Architecture:
https://docs.google.com/document/d/15GuwZwIOQy_g1XsIJjQsFNHCTQTWoXQhWGVMhiH0swc/

Before starting Hackathon IV, please complete the first three hackathons in order:

- Hackathon III: https://ggl.link/hackathon-3
- Hackathon II: https://ggl.link/hackathon-2
- Hackathon I: https://ggl.link/hackathon-1

1. Executive Summary

1.1 The Challenge

Build a **Course Companion FTE** — a Digital Full-Time Equivalent that serves as a 24/7 educational tutor, working 168 hours per week at 85-90% cost savings compared to human tutors.

1.2 The Innovation

This hackathon applies the **Agent Factory Architecture** to education:

- **General Agents (Claude Code)** manufacture the Custom Agent using Spec-Driven Development
- **Custom Agents (Digital FTE)** deliver tutoring at scale with enterprise-grade reliability
- **Dual Frontends** - ChatGPT App (Phase 1 and 2) + Web (Phase 3) provide maximum reach
- **Zero-Backend-LLM Default** ensures cost efficiency and scalability (Phase 1)

1.3 The Outcome

A production-ready Digital FTE that can:

- Tutor thousands of students simultaneously
- Operate 24/7 without fatigue
- Students can use the app from inside ChatGPT or a stand alone Web App
- Maintain 99%+ consistency in educational delivery
- Scale from 10 to 100,000 users without linear cost increase

The hackathon enforces a modern AI architecture principle: start with a Zero-Backend-LLM architecture (see Appendix I) by default (Phase 1), and introduce Hybrid intelligence (Phase 2) only where it is demonstrably valuable, cost-justified, and premium. In Phase 3 a stand alone end-to-end Web App is required to be built with full features.

2. The Digital FTE Thesis

> *"In the AI era, the most valuable educational platforms won't sell courses — they'll hire Digital Tutors that work 168 hours per week, powered by agents, specs, skills, MCP, and cloud-native technologies."*

2.1 Human Tutor vs Digital FTE Comparison

Feature	Human Tutor	Course Companion FTE
Availability	40 hours/week	168 hours/week (24/7)
Monthly Cost	$2,000 – $5,000	$200 – $500
Students per Tutor	20-50	Unlimited (concurrent)
Consistency	Variable (85–95%)	Predictable (99%+)
Ramp-up Time	Weeks of training	Instant (via SKILL.md)
Personalization	Limited by time	Infinite patience
Cost per Session	$25 – $100	$0.10 – $0.50
Language Support	1-3 languages	50+ languages

The 'Aha!' Moment

A Digital Tutor FTE can conduct **50,000+ tutoring sessions per month** at approximately **$0.25 per session** — compared to $50+ for human tutoring. This represents a **99% cost reduction** while maintaining quality through guardrails and Agent Skills.

3. Project Overview

3.1 What You're Building

An **AI-Native Course Companion** that:

1. **Teaches** — Delivers course content with intelligent navigation
2. **Explains** — Breaks down concepts at the learner's level
3. **Quizzes** — Tests understanding with immediate feedback
4. **Tracks** — Monitors progress and identifies knowledge gaps
5. **Adapts** — Adjusts difficulty and approach (Phase 2)

6. **Web App** — Provides a Comprehensive Stand Alone Web App (Phase 3)

3.2 Dual Frontend Architecture

Teams must build two frontends that share backend features but separate code base:

Component	Technology	Purpose
ChatGPT App Frontend (Phase 1)	OpenAI Apps SDK	Conversational UI, 800M+ user reach
Deterministic Backend (Phase 1)	FastAPI (Python)	Content APIs, Quiz APIs, Progress APIs
Hybrid Backend (Phase 2)	FastAPI (Python) + LLM API Calls	Paid Premium Features
Web Frontend (Phase 3)	Next.js / React	Full LMS dashboard, progress visuals, admin features
Consolidated Backend (Phase 3)	FastAPI (Python) + LLM API Calls	All Features
Content Storage (Phase 1, 2, and 3)	Cloudflare R2	Course content, media assets, quiz banks

3.3 Course Content Options

Teams must choose **ONE** course topic to build their FTE around:

Option	Course Topic	Suggested Content
A	AI Agent Development	Claude Agent SDK concepts, MCP, Agent Skills
B	Cloud-Native Python	FastAPI, Containers, Kubernetes basics
C	Generative AI Fundamentals	LLMs, Prompting, RAG, Fine-tuning
D	Modern Python	Modern Python with Typing

4. Architecture Overview

4.1 Agent Factory Context

This project implements the **Agent Factory Architecture** where:

- **You (the team)** write the **Spec** (requirements, guardrails, skills)
- **General Agent (Claude Code)** manufactures the **Custom Agent** code
- **Custom Agent (Course Companion FTE)** runs in production serving students

4.2 Technical Stack Layers

Following the Agent Factory 8-layer architecture:

Layer	Technology	Purpose	Phase
L0	Agent Sandbox (gVisor)	Secure execution	Phase 2 and 3
L1	Apache Kafka	Event backbone	Phase 2 and 3
L2	Dapr + Workflows	Infrastructure + Durability	Phase 2 and 3
L3	FastAPI	HTTP interface + A2A	**Phase 1, 2, and 3**
L4	OpenAI Agents SDK	High-level orchestration	Phase 2 and 3
L5	Claude Agent SDK	Agentic execution	Phase 2 and 3
L6	Runtime Skills + MCP	Domain knowledge + Tools	**Phase 1, 2, and 3**
L7	A2A Protocol	Multi-FTE collaboration	Phase 2 and 3

Phase 1 Focus: L3 (FastAPI) + L6 (Skills + MCP) — Deterministic backend
Phase 2 Addition: L4 + L5 (Hybrid SDK) for premium features

4.3 Zero-Backend-LLM Architecture Principle

Key Insight: In Phase 1, ChatGPT does ALL the intelligent work. Your backend is purely deterministic — serving content, tracking progress, and enforcing rules. This means **near-zero marginal cost** per user.

Backend Responsibilities	ChatGPT Responsibilities
• Serve content verbatim from R2 • Track progress and streaks • Grade quizzes (rule-based) • Enforce access control • Search content (keyword/embedding)	• Explain concepts at learner's level • Provide analogies and examples • Answer questions from content • Encourage and motivate • Adapt tone to student

5. Phase 1 ChatGPT App: Zero-Backend-LLM

5.1 Phase 1 Goal

Build a **fully functional Course Companion** where:

- **Backend** performs ZERO LLM inference
- **ChatGPT** handles ALL explanation, tutoring, and adaptation
- **System** is production-viable and can scale to 100k+ users cheaply

5.2 Architecture Rules (STRICT)

Zero-Backend-LLM:
User → ChatGPT App → Deterministic Backend

✅ ALLOWED in Backend

Component	Purpose	Example
Content APIs	Serve course material	GET /chapters/{id}
Navigation APIs	Chapter sequencing	GET /chapters/{id}/next
Quiz APIs	Rule-based grading	POST /quizzes/{id}/submit
Progress APIs	Track completion	PUT /progress/{user_id}
Search APIs	Keyword/semantic search	GET /search?q=neural+networks
Access Control	Freemium gating	GET /access/check

❌ NOT ALLOWED in Backend

Forbidden	Why
LLM API calls	Violates Zero-Backend-LLM
RAG summarization	LLM inference
Prompt orchestration	LLM inference
Agent loops	Must be in ChatGPT
Content generation	Pre-generate only

5.3 Required Features (Phase 1)

All teams must implement these 6 features in Phase 1:

#	Feature	Backend Does	ChatGPT Does
1	Content Delivery	Serve content verbatim	Explain at learner's level
2	Navigation	Return next/previous chapters	Suggest optimal path
3	Grounded Q&A	Return relevant sections	Answer using content only
4	Rule-Based Quizzes	Grade with answer key	Present, encourage, explain
5	Progress Tracking	Store completion, streaks	Celebrate, motivate
6	Freemium Gate	Check access rights	Explain premium gracefully

5.4 Disqualification Criteria

Teams are IMMEDIATELY DISQUALIFIED from Phase 1 if the backend contains ANY LLM API calls, summarization logic, or agent loops. Detection method: Code review + API audit.

Distribution: The OpenAI Apps Ecosystem (https://chatgpt.com/apps)

Reaching 800 Million Users via OpenAI Apps. The Global Marketplace for Digital FTEs

Instant Visibility: OpenAI Apps provide a direct pipeline to:

- **800+ Million** individual users.
- **1+ Million** businesses already using the OpenAI ecosystem.

The "App Store" Moment: Just as the App Store created the Mobile Economy, OpenAI Apps is creating the **Agent Economy**.
Low Friction: Enterprises can discover and "hire" your Digital FTE with a single click, bypassing traditional 6-month sales cycles.

"We don't need a sales team of 500 people. We leverage the world's most powerful AI distribution engine. By placing our Custom Agents on the OpenAI platform, we are standing in front of a million businesses on day one."

6. Phase 2 ChatGPT App: Hybrid Intelligence (Paid Premium)

6.1 Phase 2 Goal

Add **selective backend intelligence** that:

- Delivers clear additional educational value
- Is cost-justified as a premium feature
- Is cleanly isolated from Phase 1 logic

Teams must explain why each hybrid feature cannot be implemented using zero-LLM design.

Hybrid ChatGPT App:
User → ChatGPT App → Backend → LLM APIs

6.2 Phase 2 Rules

✅ Hybrid intelligence MUST be:	❌ You may NOT:
• Feature-scoped (limited to specific features) • User-initiated (user requests it) • Premium-gated (paid users only) • Isolated (separate API routes) • Cost-tracked (monitor per-user cost)	• Convert entire app to hybrid • Auto-trigger hybrid features • Make hybrid required for core UX • Hide hybrid costs from analysis

6.3 Allowed Hybrid Features (Choose Up to 2)

Feature	What It Does	Why It Needs LLM
A. Adaptive Learning Path	Analyzes patterns, generates personalized recommendations	Requires reasoning over learning data
B. LLM-Graded Assessments	Evaluates free-form written answers with detailed feedback	Rule-based can't evaluate reasoning
C. Cross-Chapter Synthesis	Connects concepts across chapters, generates "big picture"	Requires multi-document reasoning
D. AI Mentor Agent	Long-running agent for complex tutoring workflows	Multi-turn problem solving

Phase 2 ChatGPT App Architecture Pattern

```
ChatGPT App

  ↓

Backend

  ├─ Deterministic APIs (Phase 1)

  └─ Hybrid Intelligence APIs (Phase 2, gated)

      └─ LLM calls
```

Teams must clearly show:

- Which features are zero-LLM
- Which features are hybrid
- Why hybrid was necessary
- Estimated cost per user

7. Phase 3 Web App

Full end-to-end Web App.

Phase 2 Web App Architecture Pattern

```
Web App (Next.js)

  ↓

Backend

  └─ APIs (All Features) (FastAPIs)

      └─ LLM calls
```

For Web App there will be a single set of APIs

8. Agent Skills Design

Agent Skills teach the Course Companion FTE how to perform educational tasks consistently. Each skill is a SKILL.md file containing procedural knowledge.

8.1 Required Runtime Skills

Skill Name	Purpose	Trigger Keywords
concept-explainer	Explain concepts at various complexity levels	"explain", "what is", "how does"
quiz-master	Guide students through quizzes with encouragement	"quiz", "test me", "practice"
socratic-tutor	Guide learning through questions, not answers	"help me think", "I'm stuck"
progress-motivator	Celebrate achievements, maintain motivation	"my progress", "streak", "how am I doing"

8.2 Skill Structure Template

Each skill SKILL.md file should contain:

1. **Metadata:** Name, description, trigger keywords
2. **Purpose:** What this skill accomplishes
3. **Workflow:** Step-by-step procedure
4. **Response Templates:** Example outputs
5. **Key Principles:** Guidelines and constraints

9. Cost Analysis Framework

9.1 Phase 1 Cost Structure (Zero-Backend-LLM)

Component	Cost Model	Est. Monthly (10K users)
Cloudflare R2	$0.015/GB + $0.36/M reads	~$5
Database (Neon/Supabase)	Free tier → $25/mo	$0 - $25
Compute (Fly.io/Railway)	~$5-20/mo	~$10
Domain + SSL	~$12/year	~$1
TOTAL		**$16 - $41**
Cost per User		**$0.002 - $0.004**

ChatGPT Usage: $0 to developer (users access via their ChatGPT subscription)

9.2 Phase 2 Cost Structure (Hybrid Intelligence)

Feature	LLM Model	Est. Tokens/Request	Cost/Request
Adaptive Path	Claude Sonnet	~2,000	$0.018
LLM Assessment	Claude Sonnet	~1,500	$0.014
Synthesis	Claude Sonnet	~3,000	$0.027
Mentor Session	Claude Sonnet	~10,000	$0.090

9.3 Monetization Model

Tier	Price	Features
Free	$0	First 3 chapters, basic quizzes, ChatGPT tutoring
Premium	$9.99/mo	All chapters, all quizzes, progress tracking
Pro	$19.99/mo	Premium + Adaptive Path + LLM Assessments
Team	$49.99/mo	Pro + Analytics + Multiple seats

10. Judging Rubric

10.1 Phase 1 Scoring (45 points total)

Criteria	Points	Evaluation Method
Architecture Correctness	10	Code review: zero backend LLM calls
Feature Completeness	10	Checklist verification
ChatGPT App Quality	10	UX testing in ChatGPT
Web Frontend Quality	10	UX testing + responsiveness
Cost Efficiency	5	Cost analysis review

10.2 Phase 2 Scoring (20 points total)

Criteria	Points	Evaluation Method
Hybrid Feature Value	5	Demo + justification
Cost Justification	5	Cost analysis document
Architecture Separation	5	Code review
Premium Gating	5	Functional testing

10.3 Phase 3 Scoring (30 points total)

Criteria	Points	Evaluation Method
Architecture Correctness	10	Code review: zero backend LLM calls
Feature Completeness	5	Checklist verification
Web Frontend Quality	10	UX testing + responsiveness
Cost Efficiency	5	Cost analysis review

10.4 Bonus Awards

- **Best Zero-LLM Design** (+3 points)
- **Most Creative ChatGPT App** (+3 points)
- **Best Educational UX** (+2 points)
- **Most Justified Hybrid Feature** (+2 points)
- **Most Creative Web App** (+3 points)

10. Deliverables

10.1 Required Submissions

Deliverable	Format	Description
Source Code	GitHub repo	Complete codebase with README
Architecture Diagram	PNG/PDF	Visual system design
Spec Document	Markdown	Course Companion FTE specification
Cost Analysis	Markdown/PDF	1-page cost breakdown
Demo Video	MP4 (5 min)	Walkthrough of both frontends
API Documentation	OpenAPI/Swagger	Backend API spec
ChatGPT App Manifest	YAML	App definition

10.2 Demo Video Requirements

Segment	Duration	Content
Introduction	30 sec	Team, project overview
Architecture	1 min	Explain zero-LLM + hybrid design
Web Frontend Demo	1.5 min	Full user journey
ChatGPT App Demo	1.5 min	Conversational learning session
Phase 2 Features	30 sec	Premium feature demo

11. Final Checklist

11.1 Phase 1 Checklist

- Backend has ZERO LLM API calls
- All 6 required features implemented
- ChatGPT App works correctly
- Progress tracking persists
- Freemium gate is functional

11.2 Phase 2 Checklist

- Maximum 2 hybrid features
- Features are premium-gated
- Features are user-initiated
- Architecture clearly separated
- Cost tracking implemented

11.3 Phase 3 Checklist

- Backend has LLM API calls
- All 6 required features implemented
- Web frontend is functional and responsive
- Progress tracking persists
- Freemium gate is functional

11.4 Documentation Checklist

- README is complete
- Architecture diagram included
- Cost analysis submitted
- Demo video recorded
- API documentation complete

GOLDEN RULES

Zero-Backend-LLM is the default. Hybrid intelligence must always be selective, justified, and premium.**Your Spec is your Source Code.** If you can describe the excellence you want, AI can build the Digital FTE to deliver it.

Appendix I : Zero-Backend-LLM vs Hybrid Architecture

Below is a **clear, side-by-side, decision-grade comparison** of **Zero-Backend-LLM** vs **Hybrid** architecture, specifically in the context of **ChatGPT Apps**, **AI-native books**, and **agentic education tools**.

I will avoid theory and focus on **engineering, cost, risk, and product impact**.

1️⃣ Definitions (precise)

Zero-Backend-LLM Architecture

- **No LLM calls in your backend**
- ChatGPT does **all reasoning, explanation, tutoring**
- Backend is **deterministic**
- Content is **served verbatim**

Hybrid Architecture

- ChatGPT does **user interaction**
- Backend **also calls LLMs**
- Backend performs:
    - RAG reasoning
    - Summarization
    - Evaluation
    - Agent workflows

2️⃣ High-level architecture diagrams

Zero-Backend-LLM

```
User
 ↓
ChatGPT App (LLM reasoning)
 ↓
Backend (content + rules)
```

Hybrid

```
User
 ↓
ChatGPT App
 ↓
Backend
 ├─ Retrieval
 ├─ LLM calls
 ├─ Agents
 └─ Post-processing
```

3️⃣ Side-by-side comparison table

Dimension	Zero-Backend-LLM	Hybrid

Backend LLM calls	❌ None	✅ Yes
Your LLM cost	**$0**	**High / variable**
Infra complexity	Very low	Medium → High
Latency	Low	Higher
Hallucination risk	Low (source-grounded)	Medium
Personalization depth	Medium	High
Advanced analytics	❌	✅
Agent autonomy	❌	✅
Scalability	Excellent	Cost-bound
Monetization ease	High	Medium
Compliance / auditability	High	Medium

4️⃣ Cost profile (real numbers)

Zero-Backend-LLM

- LLM inference: **$0**
- Storage + bandwidth: cents/user/month
- Flat infra cost

Hybrid

- LLM inference: **scales with usage**
- Embeddings + re-ranking
- Agent loops
- Cost spikes under load

Hybrid requires pricing discipline. Zero-LLM does not.

5️⃣ Engineering complexity

Zero-Backend-LLM

You build:

- Content APIs
- Search
- Entitlement checks
- Progress tracking

You do NOT build:

- Prompt orchestration
- Agent memory
- Token optimization
- Cost controls

Hybrid

You must manage:

- Prompt versions
- Token limits
- Model selection
- Latency budgets
- Cost ceilings
- Agent failures

This is **real operational burden**.

---

6 Quality & correctness

Zero-Backend-LLM

- ChatGPT answers **only from provided text**
- Easy to enforce:

  "If not in content, say 'not covered'"

- Excellent for **education & documentation**

Hybrid

- More flexible
- More creative
- More hallucination vectors
- Requires guardrails

---

7 Personalization & intelligence

Zero-Backend-LLM

Can do:

- Explain differently
- Simplify
- Give analogies
- Adjust tone

Cannot do:

- Deep learner modeling
- Long-term adaptive planning
- Cross-content synthesis beyond provided text

Hybrid

Can do:

- Personalized learning paths
- Skill graphs
- Adaptive curricula
- Automated assessment
- Spec-driven feedback

---

8 Where each architecture wins

Zero-Backend-LLM is best for:

✅ Books
✅ Courses
✅ Reference manuals
✅ LMS companions
✅ Corporate training
✅ Compliance content

Hybrid is best for:

✅ Coding agents
✅ Research agents
✅ Analytics platforms
✅ Personalized tutors
✅ Autonomous workflows

9️⃣ Risk & business impact

Zero-Backend-LLM

- Predictable costs
- Easy pricing
- Fewer outages
- Faster iteration
- Lower team size

Hybrid

- Cost overruns risk
- Model regressions
- Vendor dependency
- Requires MLOps discipline

🔟 A common mistake to avoid

❌ Starting with Hybrid too early

Most teams:

- Overestimate need for backend intelligence
- Underestimate ops complexity
- Struggle with margins

Best practice:

> Start zero-LLM → add hybrid only where value is proven.

1️⃣1️⃣ Recommended evolution path (important)

```
Phase 1: Zero-Backend-LLM
  ↓
Phase 2: Hybrid for premium users
  ↓
Phase 3: Agentic workflows (optional)
```

Example:

- Free users: zero-LLM
- Pro users: hybrid (adaptive tutoring, exams)

1️⃣2️⃣ Decision checklist (use this)

Choose **Zero-Backend-LLM** if:

- You sell content
- You want low cost
- You want predictable scaling
- You care about correctness

Choose **Hybrid** if:

- Intelligence *is* the product
- You need autonomy
- You can charge accordingly
- You have ops maturity

Bottom line

Zero-Backend-LLM

- Cheap
- Simple
- Scalable
- Ideal for education & books

Hybrid

- Powerful
- Expensive

- Complex
- Needed for advanced agents

Given your **AI-native book + LMS direction**,
👉 **Zero-Backend-LLM should be your default architecture**
👉 Hybrid should be **selective and premium**

—— End of Document ——

Document Version 1.0 — January 2026
Prepared for Panaversity Agent Factory Development Hackathon