BRIAN LAURENCE E. MILLONTE

General Trias, Cavite, Philippines
+63 981 758 2216
millontebry@gmail.com
GitHub: https://github.com/Briuwu
Portfolio: https://brianmillonte.vercel.app
LinkedIn: https://www.linkedin.com/in/brian-laurence-millonte

TECHNICAL SKILL

AI / LLM Systems:
OpenAI API, Local AI, Agentic AI, Prompt Engineering, Tool Calling, Langfuse, Langchain, Langgraph, Vercel AI SDK, OpenTelemetry, Human-in-the-Loop, Token & Context Budgeting

Backend:
Python, FastAPI, Node.js, Express, REST APIs

Frontend:
React, Next.js, TypeScript, TailwindCSS, Electron, Vite, Zustand

Databases & Infra:
PostgreSQL, MongoDB, Firebase, Supabase, Redis, AWS

Tools:
Docker, Git, Postman, Zod, Figma

EDUCATION

Bachelor of Science in Information Technology
Cavite State University – Main Campus
2021 – 2025

WORK EXPERIENCE

Applied AI Engineer

June 2025 - Present

Prosperna, Philippines

- Designed and deployed a production-grade AI copilot for a commerce SaaS platform, enabling automated generation of product content, SEO metadata, landing pages, and business insights via a conversational interface
- Architected a multi-agent LLM system using Python and FastAPI, enabling task routing across specialized agents for content generation, analytics, and workflow automation
- Implemented real-time LLM streaming using Server-Sent Events (SSE), enabling responsive chat interactions and dynamic UI updates
- Built human-in-the-loop workflows with approval systems, guided selections, and safe publish flows to ensure reliability and control over AI-generated outputs
- Developed persistent conversation memory and session recovery, allowing continuity across long-running workflows and user sessions
- Integrated AI with live business data, enabling natural-language queries for sales, traffic, product performance, and customer insights
- Added reliability systems including concurrency controls, fallback handling, and usage safeguards for production stability
- Worked on frontend integration using React, supporting streaming UI, state synchronization, and AI interaction workflows

PERSONAL PROJECT

AIRI - Local-first AI Desktop Assistant

- Built a local-first Windows desktop AI assistant in Electron, React, and TypeScript with hotkey overlay, guided setup, and a managed llama-server runtime.
- Implemented SSE streaming, OpenAI-style tool calling, and a bounded multi-step agent loop with trace and usage feedback in the chat UI.
- Shipped filesystem discovery, host open flows, spreadsheet preview and approval before write, PDF workflows, opt-in memory, and desktop automation tools behind a centralized registry and system prompt contract.
- Added token budgeting and conversation summarization to keep long threads within the local model context window.
- Wired typed IPC between renderer and main for runs, approvals, and setup; Langfuse tracing with payload limits.
- Covered critical paths with Vitest and per-file coverage gates where the project enables them.