

# Mustafa Asghari

Brisbane, Australia | asghari.mustafa99@gmail.com | github.com/mustafa-asghari

AI systems-focused software developer building LLM-powered products, source-grounded tutoring systems, high-performance APIs, and production-style backend platforms.

Learning software development since mid 2024 and actively building end-to-end projects since late 2024, with a strong interest in AI engineering, backend architecture, retrieval workflows, tool-using assistants, and system design.

## EDUCATION

---

### TAFE Queensland, South Bank

Expected 2026

*Diploma of Advanced Programming*

- Relevant study: software development, C#, Java, JavaScript, databases, REST APIs, algorithms, object-oriented design, and practical project delivery.

## TECHNICAL SKILLS

---

**Languages:** Python, TypeScript/JavaScript, SQL, C#, Java, HTML, CSS

**Frameworks & Libraries:** FastAPI, Next.js, React, Prisma, SQLAlchemy, Celery, Tailwind CSS, OpenAI SDK

**Data & Infrastructure:** PostgreSQL, Redis, ClickHouse, Typesense, Docker, Docker Compose, Alembic, S3-compatible storage, WebSockets, Stripe

**Developer Tools:** Git, GitHub, Vercel, Railway, API design, debugging, modular backend architecture, system design

**AI & LLM Engineering:** OpenAI APIs, Claude, Codex, Gemini, local LLM integration, agent/tool planning, RAG workflows, embeddings, OCR-assisted pipelines, prompt engineering, source-grounded tutoring

## SELECTED AI AND SYSTEMS PROJECTS

---

### Nexus

*Python, OpenAI SDK, faster-whisper, Redis, PostgreSQL, ClickHouse, gRPC, Playwright, Apple MCP*

- Built a local-first macOS AI assistant that converts voice or text requests into structured action plans and executes them through explicit tools rather than free-form model side effects.
- Implemented deterministic routing, app alias normalization, URL handling, message routing, plan validation, JSON extraction, provider switching, and retry or fallback behavior inside the planning layer.
- Designed trust-level safety controls for CLI access, filesystem scope, code execution, browser automation, and HTTP requests, with different capability boundaries for beginner, advanced, and expert modes.
- Integrated memory and orchestration components spanning Redis, PostgreSQL, ClickHouse, gRPC services, local fallback storage, workspace tools, and Apple ecosystem integrations for notes, mail, messages, calendar, and reminders.

### GrindUp

*Next.js, TypeScript, FastAPI, PostgreSQL, Prisma, ClickHouse, Redis, Docker, OpenAI*

- Built an AI-native learning platform for subject generation, tutoring, spaced repetition, homework workflows, analytics, and LeetCode-style coding practice.
- Developed a separate sandboxed runner service for Python, JavaScript, Java, and C++ code execution using Docker-based isolation, test-case driven evaluation, timeout handling, and structured result parsing.
- Modeled a large relational learning domain in Prisma covering users, subjects, topics, review cards, homework, submissions, contests, tutor sessions, leaderboards, and social systems.
- Applied source-grounded AI workflows for curriculum creation, tutoring, lesson generation, quiz generation, semantic retrieval, and learning support from imported PDFs, notes, and other study material.

## Kabul Sweets

*FastAPI, Next.js, PostgreSQL, Redis, Celery, Stripe, S3, WebSockets, Docker*

- Developed a production-style commerce platform with a customer storefront, admin dashboard, ML-assisted pricing interface, backend API, background workers, and realtime operational updates.
- Structured backend services across authentication, users, products, orders, carts, payments, analytics, business settings, ML endpoints, media or image flows, Telegram integrations, realtime, and support modules.
- Worked with asynchronous infrastructure using Redis and Celery for background jobs and scheduled tasks, alongside health checks, logging, compression, CORS, and multi-container local orchestration.
- Built around real business workflows including catalog operations, order handling, payments, media processing, and applied AI or ML support for pricing and operational tooling.

## SyntaX API

*FastAPI, Redis, Typesense, ClickHouse, Docker*

- Built a high-performance social-data API focused on low-latency retrieval and analytics-ready response paths.
- Implemented pre-warmed TLS session pooling, proxy-aware clients, token-pool based request handling, response timing metadata, and cache bypass controls for live versus cached reads.
- Designed multi-layer caching with Redis, Typesense, and ClickHouse, including stale-while-revalidate behavior, request coalescing, asynchronous write-through flows, and degraded fallback handling.
- Exposed user, tweet, search, followers, and following endpoints with structured API responses and cache-layer visibility for performance monitoring.

## ADDITIONAL PROJECT WORK

---

**AI Chatbot** - *Python, OpenAI API*. Built a web-based chatbot that sends requests to an LLM and returns generated responses through a Python backend.

**Personal Portfolio Website** - *Next.js, JavaScript, Vercel*. Created and deployed a personal portfolio site to showcase projects, technical skills, and ongoing software development work.

**Sports Team Data Manager** - *C#*. Built a desktop-style application for storing and managing sports team information using object-oriented design principles.

**Python CLI Expense Tracker** - *Python, pandas*. Implemented data validation, file persistence, and object-oriented structure in a command-line expense tracking application.

**Repurposing Ideas Application** - *Java*. Designed a Java application for adding, saving, and categorizing repurposing ideas using OOP-based application structure.

## INDEPENDENT DEVELOPMENT EXPERIENCE

---

- Self-directed learning path focused on software engineering, AI systems, backend development, and full-stack product building since mid 2024.
- Consistently building end-to-end projects since late 2024 across APIs, databases, LLM integrations, async workers, retrieval workflows, Dockerized services, and production-style project structure.
- Comfortable working across application layers from frontend to backend, data modeling, infrastructure setup, debugging, and system-level architecture decisions.

## SELECTED STRENGTH AREAS

---

- AI agents, structured planning, tool-use systems, and safety-aware execution flows.
- Retrieval-grounded AI products, tutoring workflows, prompt design, embeddings, and imported-content pipelines.
- Backend architecture with FastAPI services, modular API routing, data modeling, async processing, and production-style application structure.
- Data and performance engineering using PostgreSQL, Redis, ClickHouse, Typesense, cache hierarchies, and latency-focused API design.
- Sandboxed code execution, developer tooling, and practical systems work involving Docker, Docker Compose, runners, isolated services, and full-stack product delivery across frontend and backend layers.