AYAAN A. KHAN

(773)-937-2448 . ayaanahmedkhan12@gmail.com . in/ayaanahmedkhan/ . github.com/ayaan47-1

EDUCATION

ILLINOIS INSTITUTE OF TECHNOLOGY | Chicago, Illinois **August 2022 - May 2026**
Bachelor of Science in Artificial Intelligence; **Minor** in Architecture
Relevant Coursework: DSA, AI, ML, NLP, DBMS, Data Mining, Discrete Math, Linear Algebra, Probability, Statistics, OOP

SUMMARY

AI engineer with two years at a construction tech firm, shipping agentic workflows and production RAG across real projects. Built and deployed retrieval pipelines, async backends, and agentic BIM tools used by project teams. Comfortable owning backend to deployment.

WORK EXPERIENCE

D.A. SYNTEC LTD. | Chicago, Illinois

- AI & Digital Developer (Returning Intern) **January 2026 - May 2026**

- Built and deployed the production RAG retrieval layer over firm documents (chunking, embedding, and indexing with ChromaDB) that became the shared knowledge backbone for the SyntecAI agent suite — now used by internal teams to answer project queries with cited sources instead of digging through file shares.
- Built the CCN (Coding, Classification & Naming) Agent — now part of the VeritasLayer agent suite — translating natural language into structured CCN codes with alias mapping across
- Industry/BUSA/Company/Revit naming levels, a /validate threshold calibrated on a labeled golden set (no hardcoded cutoffs), atomic SQLite→ChromaDB sync, and confirmation gates before any destructive write. In production with project teams.

- AI & Digital Developer (Intern) **May 2025 - December 2025**

- Curated and published weekly posts using Canva across LinkedIn and Mailchimp; drove over 60% growth in impressions and a notable increase in audience engagement following a content strategy restructure.
- Built a custom RAG pipeline using OpenAI and ChromaDB -- chunking, embedding, and indexing firm documents to enable natural language retrieval -- serving as the technical foundation for later production chatbot deployment.

PROJECTS

SYNTEC CCN AGENT | Professional Project **February 2026 - May 2026**

- Shipped the CCN Standards Enforcement Agent (A1 of the SyntecAI suite): a /classify endpoint mapping free text → CCN code + alias levels + confidence, validating each element against STW/CCN naming standards and flagging non-conformers with the suggested correct code.
- Built the storage model as a relational join in Postgres for structured codes and standard names, with pgvector/Chroma reserved for fuzzy alias retrieval only; calibrated the validation threshold from a precision/recall sweep instead of a magic number.
- Containerized the full stack with Docker, deployed the frontend to Vercel with HTTPS and server side API rewrites, and added Redis caching with a DeepSeek fallback LLM for cost optimization.

VERITASLAYER | Founder and Solo Developer **March 2026 - Present**

- Architected VeritasLayer, the evidence-grounded platform powering the SyntecAI agent suite — including the GC / Sub-Contractor Contract Review agent, which uses a locate-then-judge split (embeddings find the clause, typed-field rules judge materiality on payment terms, indemnity caps, and insurance limits) so cosine similarity never decides whether a contract conforms.
- Solo-built and deployed the 13-stage extraction pipeline (FastAPI, Inngest, PostgreSQL, LiteLLM) behind it — every obligation, risk, and deadline backed by a verbatim quote, page number, and confidence score, with a human admin gate on all material flags.
- Built a production Next.js frontend with Clerk auth, human review workflows, real-time status polling, and a Jaccard precision/recall eval harness that doubles as the suite's zero-delta reconciliation metric.

CLUTCH | Co-Founder and Developer **January 2026 - Present**

- Built Clutch: a course generation SaaS with a 5 stage agent pipeline, persisted job state, retry policies, and SSE streamed progress. Cut inference cost with LiteLLM routing and shared source deduplication.
- Implemented an asynchronous FastAPI backend with PostgreSQL plus pgvector and Redis caching, and streamed real time job progress to clients using SSE backed by Redis to keep generation workflows responsive.
- Cut inference cost and improved output quality using LiteLLM routing and shared source deduplication; enforced structured outputs with PydanticAI validation and added observability via Sentry and PostHog.

PATCH TST TECHNICAL IMPLEMENTATION | Academic Project **February 2026 - April 2026**

- Implemented PatchTST (Transformer-based time series forecasting model) from scratch in PyTorch, reducing attention complexity from $O(L^2)$ to $O((L/S)^2)$ through patch based tokenization and channel independent processing; validated on ETTh1, ETTh2, and Weather datasets with ablation studies on patch length and stride parameters
- Designed and optimized sliding window data pipeline with per-feature standard scaling across 3 datasets (52K+ time steps, 21 features), achieving performance comparisons against DLinear baseline across 4 prediction horizons (96-720 steps) with MSE/MAE metrics.

SKILLS

Languages: Python, SQL, Java, JavaScript, TypeScript, HTML/CSS
Frameworks & Tools: FastAPI, Flask, React, Svelte, Vector Databases, Redis, LiteLLM, PydanticAI, Inngest
ML and LLM: RAG, embeddings, APIs, LangChain, LangGraph, scikit-learn, PyTorch, Hugging Face
Infra: Docker, Git, CI/CD, Sentry, PostHog, Vercel