Tyler Emanuel

Houston, TX (open to relocating to New York or San Francisco) • tyleremanuel38@gmail.com • 515-745-1996
linkedin.com/in/tylerdemanuel • github.com/temanuel1

Education

Rice University, Houston, Texas — *May 2027*
Bachelor of Science in Computer Science, Minor in Business — GPA: 3.49
Relevant Coursework: Database Systems, Computer Networks, Algorithms, Artificial Intelligence

Des Moines Area Community College, Ankeny, Iowa — *May 2023*
Associate in Science, Associate in Arts — GPA: 3.92

Work Experience

Software Engineer Intern — May – November 2026
Turnstile (Series A, quote-to-cash for B2B SaaS), Remote

- Built a **billing-configuration service** that compiles natural-language pricing terms into **executable, schema-validated rule sets** for the billing engine, using a **6-turn LLM pipeline** with per-turn retries and a **validation/self-repair loop** that rejects invalid configurations before they can affect an invoice
- Built an **end-to-end correctness harness** over **75+ real-world SaaS pricing models** (Snowflake, Stripe, AWS Lambda, Anthropic) spanning **7+ pricing archetypes**, verifying generated rules compute usage-based charges correctly down to **penny-exact** invoice totals
- Designed **statistical regression detection** for nondeterministic output (complexity-scaled invariants, 5-run consistency scoring, change detection over **Postgres** run history) after naive pass/fail flagged ∼**41%** of healthy batches, separating real regressions from noise
- Owned customer-facing support automation from **design to production**: orchestrated a public **REST API**, a hosted **MCP server** tool, and an internal **AI agent** into one **idempotent pipeline** connected to **Intercom**, including the error-contract and replay layer that makes retries and **failure recovery** safe
- Enforced **tenant data isolation** with **Postgres row-level security** for an internal admin dashboard exposing AI agent sessions to the support team
- Built a **build-time data pipeline** ingesting **Sanity CMS** docs into the agent's knowledge layer, with generated citations making every answer **auditable** to its exact source (GitHub path + line, or doc)

Software Engineer — November 2025 – January 2026
yAI, New York, New York

- Built an **agentic orchestration layer** that **integrated external financial data providers** (market data APIs, document stores) into unified multi-step workflows, removing manual multi-source aggregation for equity research analysts
- Engineered **parallel, fault-tolerant execution** that surfaced partial results when individual providers failed, keeping workflows running during real-time data retrieval
- On a **5-person team**, served as the **liaison between beta customers and engineering**, turning feedback into **shipped features** such as a conversational interface with **RAG-powered iterative refinement**

Software Engineer Intern — May – August 2025
Principal Financial Group, Des Moines, Iowa

- Built **reusable** React/TypeScript components for the **Principal Design System**, a shared UI library adopted by **4,000+ engineers** across dozens of customer-facing financial services applications, including **WCAG 2.2** accessibility compliance updates on sites serving **millions** of customers

Front-End Software Developer Intern — May – August 2024
W. R. Berkley Regional Shared Services, Urbandale, Iowa

- Contributed **15%** of sprint output on a 10-person team building a production React + **GraphQL** insurance quoting application; increased Jest test coverage by **10%**

Projects & Competitions

MyDB Relational Database Engine (Database Systems course, in progress)

- Implementing a relational database system in **C++** on a shared skeleton over a semester, starting with an **LRU buffer manager** for page caching and eviction

NBA Game Recap MCP Server

- Built an **MCP server** exposing live NBA stats and community sentiment to LLM agents through type-safe, **schema-validated** tool calls with fault-tolerant error handling for multi-step agent reasoning; runs plug-and-play in Claude Desktop

Stanford TreeHacks — *Acceptance rate <**5%***

- Built a Next.js app with the **Google Gemini API** to track lecture attention and auto-summarize missed content, achieving **80–90% accuracy** reconstructing missed lecture segments

Skills

Languages: Python, TypeScript/JavaScript, SQL, Go, Java, C, C++
Backend & Infrastructure: FastAPI, Node.js, PostgreSQL, REST APIs, GraphQL, tRPC, Prisma, Docker, AWS (Lambda, DynamoDB)
Frontend: React, Next.js, TypeScript, design systems, WCAG accessibility, Jest
Reliability & Correctness: schema validation, idempotent workflows, retries and failure recovery, fault-tolerant parallel execution, regression detection, multi-tenant access control (row-level security)
AI & Agents: agent orchestration, LLM/agent evaluation (Harbor framework), self-repair/validation loops, MCP servers, RAG, Anthropic API/SDK, Vercel AI SDK, LangChain