

Patrick Semler

patricksemler@tamu.edu | linkedin.com/in/patricksemler | github.com/patricksemler | patricksemler.dev

EDUCATION

Texas A&M University

Bachelor of Science in Computer Science

College Station, TX

Expected May 2028

- **GPA:** 3.79/4.00
- **Honors:** National Merit Scholar; President's Endowed Scholar
- **Relevant Coursework:** Program Design and Concepts, Data Structures and Algorithms, Discrete Math
- **Organizations:** Aggie Coding Club, Aggies Create, TAMUhack, TIDAL

EXPERIENCE

Machine Learning Research Assistant

Texas A&M University

June 2026 – Present

College Station, TX

- Reduced production classifier peak memory by 83% while maintaining 91% accuracy across 144 veterinary conditions by optimizing random-forest tree count and leaf constraints
- Eliminated hallucinated care plans in evaluation by grounding a Gemini retrieval-augmented generation pipeline in complete case context—symptoms, vitals, and ranked differential diagnoses
- Expanded diagnostic coverage to swine and equine (64 new condition mappings) by integrating both species into the filtered prediction pipeline end-to-end, verified with automated endpoint tests

Full-Stack Developer

Apply Finch

Jan. 2026 – Present

College Station, TX

- Built and deployed an AI job-application platform on a 5-developer team that scrapes postings, ranks them via a multi-factor priority score, and generates tailored resumes and cover letters (React, TypeScript, Flask, PostgreSQL, AWS)
- Cut generation cost to under \$0.01 per application by routing pipeline stages across cost-optimized LLMs, enforcing one-page output on every document through an automated quality gate with progressive content shortening
- Hardened the asynchronous pipeline against 4 failure modes—lost state, duplicate applications, transient database failures, and orphaned scraper runs—via persisted run snapshots, deduplication, retries, and cross-worker cancellation

Quantitative Developer

Maroon Fund

Feb. 2026 – May 2026

College Station, TX

- Extended risk-adjusted strategy evaluation with 3 benchmark metrics—annualized Jensen's alpha, OLS beta, and tracking error—using SPY data from a 3-source fallback pipeline
- Consolidated 6 standalone strategy charts into a single interactive Plotly report covering equity, drawdown, trade P&L, and rolling risk metrics
- Increased backtest fidelity by modeling slippage, commissions, and forced exits; validated the engine, metrics, and reports with an automated pytest suite

PROJECTS

LeetMind | *TypeScript, Python, React, Fastify, PostgreSQL, Docker*

- Created an adaptive coding-practice platform that generates progressively harder problems tailored to each user's per-concept strengths and weaknesses, using Claude for content generation
- Validated every generated problem through 6 stages of solution, edge-case, and hidden-test verification before release
- Sustained 75 submissions/min with 0 dropped jobs under load by coordinating 6 judge slots through a PostgreSQL job queue and isolated Python/C++20 Docker runners
- Recovered unfinished jobs in under 10 seconds after simulated judge crashes by automatically reassigning work and blocking duplicate progress updates, preventing lost or double-counted submissions

Phobos | *TypeScript, React, Node.js, PostgreSQL, Supabase, MCP*

- Developed a personal AI assistant exposing 63 natural-language tools across 13 capability modules through a Telegram bot and a Realtime React dashboard backed by Supabase PostgreSQL
- Engineered an agent runtime powered by Codex and Claude CLIs that translates natural-language requests into schema-validated tool calls, protecting high-risk actions with approval gates and secret-redacted audit logs
- Implemented a self-extending plugin “forge” that plans and generates TypeScript modules in isolated Git worktrees, then runs type checks, unit tests, linting, and independent cross-model review

TECHNICAL SKILLS

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

Frameworks/Libraries: React, Fastify, Flask, Node.js, scikit-learn, Pandas, Plotly, pytest

Data/Infrastructure: PostgreSQL, Supabase, Firebase, Docker, AWS

Tools/Protocols: Git, GitHub, REST APIs, MCP