SHIFAT ISLAM SANTO

945-527-4974 | shifatislamsanto764@gmail.com | linkedin.com/in/shifatislam-santo | github.com/oneKn8 | Dallas, TX

EDUCATION

The University of Texas at Dallas Dallas, TX
B.S. in Computer Science Expected May 2027

Completed coursework: Data Structures & Algorithms, Computer Architecture, Systems Programming (Unix), Programming Language Paradigms, Computer Science I & II, Discrete Math for Computing, Calculus I & II, Mechanics, Electromagnetism.
In progress (Spring 2026): Advanced Algorithm Design & Analysis, Artificial Intelligence, Digital Logic & Computer Design, Probability & Statistics for CS.

TECHNICAL SKILLS

Languages: Python, TypeScript, Go, Rust, C++, SQL, Bash
AI/ML & Agents: LangGraph, MCP protocol, HuggingFace Transformers, SentencePiece, vLLM, ChromaDB, PyTorch, RAG
Systems & Backend: SQLite (WAL, FTS5), Unix-socket IPC, async I/O, OAuth, age/X25519 cryptography, OpenTimestamps, FastAPI, Hono, Next.js, Node.js, Docker, GitHub Actions CI; pytest, Vitest, Git, Linux

PROJECTS

Smart MCPs — *Production Model Context Protocol server monorepo. Private repo, demo on request*

- Built an 8-package TypeScript monorepo of MCP servers (email, Vercel, Runpod, weather, plus a shared core) exposing 60+ LLM tools on the official @modelcontextprotocol/sdk with stdio transport, custom Zod-to-JSON-schema conversion, and confirm-gated destructive operations.
- Engineered the shared core for OAuth credential storage, typed HTTP error coercion with 429/5xx retry, and Levenshtein fuzzy-match recovery reused by every package; covered by 1,900+ Vitest unit and integration tests (11.5K LOC), all suites green.

Kotha-1 — Bengali LLM + Tokenizer Research — *From-scratch 306M language model and open evaluation framework* | github.com/oneKn8/bengali-tokenizer-eval

- Trained a 306M-parameter LLaMA-style Bengali language model from scratch, building the full pipeline end to end: Bengali data collection/dedup/language-ID, a 32K SentencePiece BPE tokenizer, and a custom Accelerate/bf16 training loop (cosine schedule, gradient accumulation, checkpointing) run to completion.
- Released an open tokenizer-evaluation framework with 13 trained SentencePiece tokenizers measuring up to 5x context-window inflation of multilingual tokenizers versus Bengali-dedicated models, isolating a Unicode normalization failure (U+09BC Nukta) that inflated byte-fallback rates from 2% to 20%.

Research-Agent — *Autonomous research pipeline on LangGraph* | github.com/oneKn8/Research-Agent

- Built a LangGraph state machine (plan → search → analyze → refine-loop → synthesize → write → review) over an OpenAI-compatible LLM client supporting self-hosted DeepSeek-R1 via vLLM or Groq, with ArXiv/web retrieval and circuit-breaker resilience; 269 passing unit and integration tests.
- Hardened the LaTeX/BibTeX output layer against shell-escape injection (\write18, \input, LuaTeX exec), stripping malicious directives, covered by 35 dedicated security tests.

ProfGraph — *Professor-intelligence MCP server* | github.com/oneKn8/profgraph

- Built a Python MCP server exposing 10 LLM-callable tools that integrate the RateMyProfessors GraphQL API and UT Dallas's Nebula grade API, with a regex NLP teaching-style classifier and GPA-conditioned grade prediction over a Starlette REST/OpenAPI bridge.
- Verified end to end against live APIs (real ratings plus 22-semester grade distributions) with 54 unit tests and a 13/13 MCP-over-HTTP suite; designed graceful degradation when an upstream API field changed.

Blackbox — *Local-first encrypted capture and tamper-evident evidence vault. Private repo, demo on request*

- Built an always-on, offline, CPU-only audio capture and transcription system (English + Bangla ASR with diarization) with an X25519/age keystore (scrypt KDF, atomic disk writes with directory fsync) and a SLIP-0039 Shamir M-of-N key-recovery flow.
- Hardened it with a role-separated Unix-socket unlock agent (token-bucket rate limiting, SO_PEERCRED UID auth, idle auto-lock) and OpenTimestamps hash-chain tamper-evidence; 500+ tests with CI across Python 3.11 and 3.12.

Granum — *Evolutionary prompt-optimization engine (Google hackathon, 2026)* | github.com/oneKn8/granum

- Built an evolutionary optimization engine for medical-appeal generation modeling immune-system dynamics — structured mutation operators, an LLM-as-judge tournament with median-of-3 sampling, citation-based negative selection, clonal champion memory, and antigen-drift detection; 3K LOC Python with 159 passing tests.
- Architected it around a pluggable Gemini/Vertex LLM-judge backend with a Next.js audit dashboard, structured for reproducible head-to-head prompt evaluation.