William Chastain

williamchastain.com · linkedin.williamchastain.com · will.chastain@bucknell.edu

SUMMARY

AI/LLM engineering student with a strong foundation in software engineering. Experienced in retrieval pipelines, vector database design, and LangGraph-based agent workflows, alongside building and deploying production-ready backend systems with Python, Go, and Flask/Gunicorn.

EDUCATION

Bucknell University — B.S. in Computer Science and Engineering — Aug 2023 - May 2027

TECHNICAL SKILLS

Languages: Python, Go, Java, JavaScript (React)
AI / LLMs: LangGraph, LangChain, vector databases (Qdrant), RAG pipelines, embeddings, semantic search, prompt engineering, agent workflows, Ollama & local LLMs, OpenAI API
Web / Backend: Flask, Gunicorn, REST APIs, Docker, Linux, CI/CD (GitHub Actions), cybersecurity fundamentals
Other: Git, Docker, Linux, Bash

EXPERIENCE

Hoover Institution — Data Science Intern — May 2024 – Aug 2024

- Built a modular Python grader (Gradeable class) that integrated GPT-3.5 via OpenAI's API to automatically score thousands of open-response poll answers, enabling free-response questions at scale (graded ~5–10k responses per polling round).
- Designed and validated grading on a manual control set (100–200 human-graded samples) and achieved high agreement (>95%) with human labels through prompt engineering.
- Collaborated closely with researchers and fellow interns to iterate on grading prompts and testing workflows; maintained code in Git with peer review.

YouGov — Data Visualization Intern — May 2023 – Aug 2023

- Created demographic and policy visualizations for political polling using tidyverse and ggplot2; delivered graphs consumed directly by a research lead.
- Analyzed consistency between stated ideology and policy support across demographic segments to support research conclusions.
- Automated repeatable charting processes and produced clear, presentation-ready figures for internal research use.

QOMPLX Inc. — Software Engineering Intern — May 2022 – Aug 2022

- Implemented system information collection for a client-side telemetry "heartbeat" agent in C++ (Windows VM environment), extracting hardware and runtime metrics for monitoring.
- Integrated telemetry collection with TCP socket-based transport and collaborated within a small team to ship the feature.
- Participated in frequent standups and peer review cycles; focused on reliability and maintainability of production telemetry code.

PROJECTS

Cogito — Q&A style agentic philosophy research engine (Python, LangGraph) · github.com/CrazyWillBear/cogito-ai

- Built an agentic research workflow using LangGraph that generates semantic queries with filters, retrieves evidence, extracts relevant text, and synthesizes citation-backed answers from philosophical texts.
- Constructed a 166,480-embedding vector database from Project Gutenberg "Philosophy" and "Philosophy & Ethics" works with rich metadata (author, source, chapter, section) to enable high-precision semantic search.
- Engineered a dual-database architecture combining Qdrant (embeddings) and PostgreSQL (author/source filters), deployed as containerized services via Docker.
- Implemented conversation-aware query planning, parallel retrieval and extraction nodes, an iterative refinement cycle for assured resource sufficiency, and research-informed response generation.

Blogman — Personal blog engine (Python, Flask) · github.com/CrazyWillBear/blogman

- Deployed Flask app serving rendered Markdown posts, designed for simplicity and performance without a database.
- Built a caching layer with json library to serve pre-rendered pages, reducing request-time rendering overhead.
- Implemented a hot-load file system, detecting changes at runtime and startup to update cached blogs.
- Added features including tagging, pinning, search over titles/tags/body text, and enforced code quality with Codacy.

Key — Secure file encryption CLI (Go, crypto) · github.com/CrazyWillBear/key

- Developed a cross-platform command-line utility for encrypting and decrypting files using a two-factor model.
- Implemented AES-GCM encryption with PBKDF2 password-based key derivation using Go's crypto package.
- Built a modular CLI using Cobra and Viper for flexible configuration via file, flags, or environment variables.
- Integrated Codacy and CodeFactor for automated code quality checks.