Frazer Mills

+4407484867219 • psyfm5@nottingham.ac.uk • linkedin/FrazerMills • github/frazermills

PROFESSIONAL SUMMARY

A final-year Computer Science student and Sir Andrew Witty Entrepreneurial Scholar specialising in high performance, secure, and resilient systems engineering. Proven ability to solve deep technical challenges, demonstrated by engineering an ultra-low latency, secure event-processing system at Microsoft that achieved <50µs P99 latency and a throughput of >500,000 messages/sec. Combines expertise in low-level systems programming (Rust, C), security hardening (fuzz testing, memory safety), and scalable cloud architecture (Terraform, AWS, Azure). A passionate and collaborative leader, ready to apply foundational engineering principles to solve critical problems.

PROFESSIONAL EXPERIENCE

Microsoft | Cyber Security Intern | *Reading, United Kingdom* June 2025 - August 2025

- Engineered an ultra-low latency, secure event processing system in Rust, achieving <50µs P99 latency and a throughput of >500,000 messages/sec
- Designed and implemented a thread-per-core architecture with a non-blocking I/O event loop to minimise context switching and maximize CPU cache efficiency
- Systematically hardened the system using fuzz testing and rigorous input validation, resulting in zero crashes and proving the robustness of the memory-safe design
- **Tech Stack:** Rust, Python, GitHub, Prometheus

Intelligent Plant | Lead Infrastructure & Integration Engineer | *Nottingham, United Kingdom* September 2024 - May 2025

- Architected and deployed the project's entire serverless backend on Azure using Terraform (IaC), creating a scalable, modular, and event-driven microservice architecture
- Led the end-to-end integration of all system components (UI, Optimisation, Compression), designing and implementing bespoke APIs where necessary to ensure seamless data flow and system modularity
- Mentored teammates on cloud infrastructure concepts and advanced Git workflows, improving overall team knowledge and productivity
- **Tech Stack:** Azure, Terraform (IaC), C#, .NET, Python, GitLab, Jira, React

Gemba Advantage | Summer Intern | *London, United Kingdom* June 2024 - August 2024

- Developed a suite of security testing scripts performing Static & Dynamic Application Security Testing, providing a suggested improvements report
- Provisioned secure AWS infrastructure (S3, DynamoDB) using Terraform (IaC) in line with best practices for state management
- **Tech Stack:** Python, Terraform (IaC), AWS (S3, DynamoDB, Lambda, IoT Core, MQTT), React

EDUCATION

University of Nottingham | September 2023 - June 2026

- **Dissertation:** Developing a Secure by Design Infrastructure as Code Domain-Specific Language in Haskell, supervised by Prof Graham Hutton
- **Awards & Leadership:** Witty Entrepreneurial Scholarship Recipient; President of the Computing Society; Senior Student Mentor of a cohort of ~400 first year students; Founded 2 student groups, grew membership to 70+ in 6 months
- **Key Modules:** Systems and Architecture (91%), Software Engineering Group Project (84%), Programming and Algorithms (82%)
- **Certifications:** AWS Certified Cloud Practitioner; Microsoft Certified Security, Compliance, and Identity Fundamentals, AWS Certified Solutions Architect Associate (in progress)

TECHNICAL PROJECTS

Product Specification & Design Portfolio: LockIN App | *Mark 79%* April 2025 - July 2025

- Managed the end-to-end requirements engineering lifecycle for a wellness app, from initial vision & scope definition to final verification, using industry-standard tools including Jira, Confluence, and Figma
- Authored a comprehensive Software Requirements Specification (SRS) praised as "exceptionally clear and unambiguous", supported by "professional standard" Use Case analyses for "highly complex" user scenarios

Monadic Compiler in Haskell | *Top Mark in Cohort: 100%* February 2025 - March 2025

- Developed a complete compiler and virtual machine in Haskell for a custom imperative language into a target stack-based instruction set
- Designed and implemented the compiler's monadic architecture from first principles, including a custom WriterT monad transformer to elegantly manage control flow state and code generation

Operating System Simulator in C | *Mark: 88%* October 2024 - December 2024

- Engineered a multi-threaded OS simulator in C featuring a preemptive round-robin scheduler to manage the lifecycle of concurrent processes using Posix Threads (pthreads)
- Implemented core concurrent data structures, including thread-safe blocking and non-blocking queues, ensuring correctness and preventing race conditions and deadlocks using mutexes and semaphores