Safi Ullah

Software Engineering Student & Full-Stack Developer

Sydney, Australia | sssafiullahhh@gmail.com | Portfolio: safiullah.app | GitHub: https://github.com/FeymanMCSQ | LinkedIn: www.linkedin.com/in/safi-ullah-80a4912ba

PROFILE

Software engineering student and full-stack developer focused on building full-stack web apps, interactive systems, AI-assisted tools, and technically polished browser experiences. Strong project work across event-driven architecture, adaptive learning systems, Chrome extensions, procedural game systems, and systems-level CS problem solving.

TECHNICAL SKILLS

Frontend	React, Next.js, TypeScript, JavaScript, CSS, HTML, Canvas API, responsive UI, component architecture
Backend	Node.js, Express, Python, Ruby, Rust, REST APIs, authentication, validation, PostgreSQL, Prisma, Zod
AI / Data	Prompt engineering, machine learning, data mining, fine-tuning, scikit-learn, XGBoost, model evaluation, feature engineering
Systems / CS	C, data structures, algorithms, graph search, A* / Dijkstra, BFS / DFS, MIPS assembly, memory management, bitwise operations, file I/O, complexity analysis
Tools	Git, GitHub, Docker, Linux, VS Code, npm/pnpm, Browser DevTools, CLI workflows, deployment, AI-assisted development

EDUCATION

University of New South Wales (UNSW) — Bachelor of Science/Computer Science, Third-year student, Sydney, Australia
Relevant coursework: algorithms, artificial intelligence, machine learning, databases, systems programming, computer architecture. Expected graduation: May 2028

SELECTED PROJECTS

Atlas Resonance Engine | TypeScript, Node.js, Next.js, Vercel AI SDK, Deepgram, Prisma, Zod, node-cron

- Built a distributed content intelligence pipeline that ingests technical signals, transcribes voice memos, synthesizes narrative drafts with AI, and publishes approved content to social platforms.
- Designed a five-service event-driven architecture with ingestion, transcription, synthesis, publishing, and orchestration isolated so failures do not cascade across layers.
- Used Zod-validated structured AI outputs and a fault-isolated publisher to separate intelligence logic from external API delivery, retries, and rate-limit handling.

Archetype-Based Learning Engine | TypeScript, Next.js, React 19, Prisma, Zod, Vitest

- Built a skill calibration engine that serves targeted micro-skill problems based on a live performance rating derived from append-only attempt history.
- Enforced Clean Architecture boundaries across UI, API, domain, and data layers, keeping domain logic independent from Prisma and environment variables.
- Implemented strict TypeScript and Zod validation to make illegal API states difficult to represent and support testable adaptive difficulty flows.

Instagram Sanitizer | JavaScript, Chrome Extension MV3, Declarative Net Request, MutationObserver, Service Workers, chrome.storage

- Built a Chrome extension that strips Instagram algorithmic content and enforces a chronological, followed-only feed by blocking Reels, Explore, Suggested Posts, and Sponsored Content.
- Implemented three interception layers: GraphQL payload scrubbing, DOM mutation heuristics, and History API navigation control for resilient filtering in a React SPA.
- Designed secure cross-context messaging between isolated content scripts, injected page context, extension popup, and persisted user settings.

ADDITIONAL PROJECT WORK

Runtime Rush — Browser-based 2D momentum skating game built with a custom Canvas loop, procedural risk/reward routes, power-ups, mobile landscape controls, audio, and Techno-Oasis visual polish.