Ali Husain

Software engineer and product designer

ali@gallowforge.dev · gallowforge.github.io · github.com/gallowforge · Remote, UTC+3 · Open to remote work

Summary

I build complete systems, from the interface to the infrastructure.

- 11 public repositories, each one specified, tested, documented and released
- 4,503 automated tests across them, run on every change
- 15 self-hosted services in production on a platform I designed and operate

Projects

Holdfast Command-line tool, open source · gallowforge.github.io/holdfast · github.com/gallowforge/holdfast
A memory for coding agents in which every fact carries its own proof. Each fact is stored with a check, run again before the agent is told. What no longer holds comes back as broken.
1,651 automated tests, on Linux and macOS. 10 kinds of check, from a file to a live endpoint.

Seaward Command-line tool, open source · gallowforge.github.io/seaward · github.com/gallowforge/seaward
Shows which ports on a server can really be reached from outside, and the rule that decides each one. Follows a packet through the rules the kernel has loaded, so a port Docker published past the firewall's list is found.
1,212 automated tests, run on every change. 2,111 real packets checked against the kernel.

Ballast Live app, open source · gallowforge.github.io/ballast · github.com/gallowforge/ballast
Personal finance that turns savings into a salary that lasts. Designed and built end to end, offline-first. Says what is safe to spend today and reads bank statements on the device.
599 automated tests. 40 currencies.

Ethereal Live app, open source · gallowforge.github.io/Ethereal · github.com/gallowforge/Ethereal
A calm coach for training, food and the shape of the day. An installable app that plans the day, guides workouts and tracks nutrition. Designed, built and delivered end to end.
397 automated tests. 1,474 foods, offline.

Evalbench Self-hosted app, open source · gallowforge.github.io/evalbench · github.com/gallowforge/evalbench
Names the language model to use for your own job, and shows the proof. Finds the cheapest model that clears your bar. Answers are graded by automatic checks, then rated blind in pairs.
291 automated tests, plus sandbox checks in CI. 7 check types, up to sandboxed code tests.

Vesper Self-hosted app, open source · gallowforge.github.io/vesper · github.com/gallowforge/vesper
A read-only operations dashboard that answers one question: does anything need you? One Go binary, read-only by construction. When it cannot confirm something it says so instead of showing an all-clear.
7.4 MB: one Go binary, about 27 MB of memory. 191 tests, 34 end to end under real systemd.

Homelab Platform, open source · github.com/gallowforge/homelab
Fifteen services on one small machine, run as code, with backups that are proven every week. Zero port forwards. Every backup set is verified nightly, copied to three machines and restored in a weekly drill.
15 containers, 21 timers, 43 automated tests. 77 alert conditions and 13 uptime probes.

Smaller tools

agent-guardrails (github.com/gallowforge/agent-guardrails): Agents that judge work they did not write, and a warning before a destructive command. 36 tests.

aih (github.com/gallowforge/aih): Every AI conversation from every machine in one local full-text index. 31 tests.

bwtidy (github.com/gallowforge/bwtidy): Audits and tidies a password vault, with a plan you approve before anything changes. 12 tests.

filmforge (github.com/gallowforge/filmforge): A tested production method for painted films made with AI models. 40 tests.

Capabilities

Product design. Interfaces designed for phone and desktop together and accessible by default, each with its own design system, light and dark themes, data charts and motion. Seen in Ballast, Ethereal, Vesper: Ethereal scores Lighthouse 100 on key pages, swept against WCAG 2.1 AA.

Front end. Installable apps that work offline with no account or server, statements read on the device, Go engines in the browser as WebAssembly, and a live dashboard streamed to it. Seen in Ballast, Ethereal, Vesper, Seaward: 599 and 397 automated tests on the two apps.

Back end. Services and command-line tools in Go, Python, TypeScript, Bash, SQL, C and C++, cross-compiled to single binaries and shipped with their documentation. Seen in Holdfast, Seaward, Vesper, Evalbench: three Go binaries, each with no third-party Go dependencies.

Data and statistics. SQLite schema design with full-text search, SQL triggers, statement parsing and reconciliation, exact money arithmetic, and the statistics behind a fair comparison. Seen in Ballast, Evalbench, aih: 40 currencies in exact integer arithmetic; BM25 full-text search.

AI evaluation. Rubrics, automatic checks and sandboxed code tests, then blind pairwise rating scored with Bradley-Terry ratings and confidence intervals, and preference pairs for training. Seen in Evalbench: finds the cheapest model that clears your bar; 7 check types.

Agent engineering. Agent memory re-checked before every use, an agent tool server, hooks and skills, reviewer and verifier agents that judge work they did not write, and an always-on operations agent. Seen in Holdfast, agent-guardrails, Homelab: over 400 agent runs; coordinated teams of 9, 13 and 98 agents.

Infrastructure and operations. Linux, containers, systemd, CI/CD and infrastructure as code with drift checks, monitoring with a dead-man switch, and backups restored in a drill every week. Seen in Homelab, Vesper: 450 commits, 21 systemd timers, about 4,000 lines of runbooks.

Networking. DNS for a whole house over HTTPS with DNSSEC, TLS at one reverse proxy, a mesh VPN through carrier-grade NAT with zero port forwards, and Linux firewall internals. Seen in Seaward, Homelab: 2,111 real packets checked against the kernel; 7-node mesh VPN.

Security practice. Threat models, privilege separation, systemd and container sandboxing, password hashing, web hardening, secret scanning and vault audits; zero port forwards. Seen in Seaward, Homelab, Vesper, Evalbench: systemd exposure scores 0.5 and 2.4; code tests run with no network.

Testing and releases. Unit, end-to-end, fuzz and golden tests, differential tests against the real Linux kernel, accessibility and performance gates in CI, and checksummed, versioned releases. Seen in all 11 public repositories: 4,503 tests, 36 releases, 21 CI workflows, 12 fuzz targets.

Video and audio. Painted films made with AI image and video models and finished in code: a beat tracker written from scratch, broadcast loudness and 4K mastering on the GPU. Seen in filmforge: 9 skills and 7 tested tools; every encode verified.

Smart home and media. Home automation across lights, locks, TV and audio in one home app, lighting that follows the TV picture, self-hosted media streaming, and room acoustics tuned by measurement. Seen in Homelab: 15 self-hosted services; ambient TV lighting write-up.

How I work

Specified before it is built, tested before it ships, documented so it runs without me.

Prove it works. A backup isn't done until it's restored; a fix isn't done until it's tested from the user's side.

Write it down. Every system ships with a runbook someone else could follow; every failure gets a post-mortem.

Direct, then verify. Agents build to my specification, an independent reviewer checks the work, and I verify it before it ships.

Learn by shipping. Linux, Go, AI agents, post-production: each learned by building something real, with continuous self-study.

Fewer things, done properly. Simple setups that stay up beat clever ones that break.

Automated tests by repository. holdfast 1,651, seaward 1,212, ballast 599, Ethereal 397, evalbench 291, vesper 191, homelab 43, filmforge 40, agent-guardrails 36, aih 31, bwtidy 12.
Total 4,503 across 11 public repositories.

Computer science

Programming. C, C++ with object-oriented design, templates and the STL, Python, functional programming in Standard ML and logic programming in Prolog; recursion, memory and pointers.

Algorithms. Lists, stacks, queues, search trees, heaps, hashing and graphs; divide and conquer, greedy methods, dynamic programming, graph algorithms, network flow and NP-completeness.

Theory. Discrete mathematics, logic and proof, automata and formal languages, Turing machines, decidability and complexity, and compiler front ends: lexing and parsing.

Computer systems. Data representation, x86-64 and MIPS assembly, digital logic design, datapaths and pipelining, caches and the memory hierarchy, and program optimisation.

Operating systems. Linking, processes and signals, virtual memory and allocators, Unix I/O, sockets, threads and synchronisation, scheduling, deadlock and file systems.

Networks. HTTP, DNS, TCP and UDP, congestion control, IP routing and NAT, Ethernet and wireless, network security, and distributed systems: clocks, replication and consistency.

Databases. The relational model, SQL and PL/SQL, ER design, normal forms, transactions, concurrency control and recovery.

Software engineering. Requirements and specification, architecture and design, testing and validation, configuration and release management, Agile delivery, ethics and usability.

AI and security. Search, logic, probabilistic reasoning and learning; cryptography from RSA and Diffie-Hellman to hashes, MACs and TLS; access control and web security.

Mathematics and science. Calculus, linear algebra, probability and statistics in R, numerical methods, physics from mechanics to electricity and magnetism, and engineering graphics.

Toolkit

Languages. Go, Python, TypeScript and JavaScript, Bash, SQL, C, C++, HTML and CSS, Standard ML, Prolog, R, x86-64 and MIPS assembly.

Front end. Astro, Preact, React, Tailwind CSS and Vite; installable offline web apps, IndexedDB, Server-Sent Events, WebAssembly, pdf.js, Canvas and WebGL.

Back end and data. The Go standard library, FastAPI and Pydantic, SQLite with write-ahead logging and full-text search, JSON APIs and JSON Schema.

Testing and quality. Vitest, Playwright, axe, Lighthouse CI, Go tests with the race detector and fuzzing, pytest, ShellCheck and gitleaks.

Infrastructure. Linux (Debian, Arch, Ubuntu), Docker and Compose, systemd, Caddy, Tailscale, AdGuard Home, Syncthing, Home Assistant, GitHub Actions and Raspberry Pi.

AI and agents. Coding agents with hooks, skills and subagents, MCP tool servers, model APIs and local inference, and image, video, voice and music generation models.

Media. ffmpeg, GPU hardware encoding, EBU R128 loudness, numpy signal processing, colour and audio calibration.

Expanding into. Kubernetes, Ansible, Terraform and cloud GPU rendering.

Education

Computer Science and Engineering

Contact

Open to remote work. Remote, on UTC+3: the full European working day overlaps.
ali@gallowforge.dev · gallowforge.github.io · github.com/gallowforge