Shaaf

Software Engineer | AI integrations, Python & React

London, UK | shaaf@vorthryss.com | vorthryss.com
Right to work in the UK. Open to remote contracts and software-engineering roles.

PROFILE

Independent builder of business integrations, internal tools and AI-enabled workflows. I originate ideas and system architecture, compare competing approaches, and direct AI-assisted implementation, critique and testing. My focus is software that turns real inputs into useful, checkable outputs.

Vorthryss is my independent engineering practice. My work spans Python services, React interfaces, data processing, task orchestration and verification. The projects below are self-directed demonstrations, with synthetic data identified and measured results available for inspection.

SELECTED ENGINEERING PROOF

Workflow recovery lab | Executable public demonstration

A browser-local state machine demonstrating duplicate-event handling, queued work, temporary failures, restart recovery and a simulated CRM connector. Visitors can trigger failures, inspect state and download the execution trace, source and tests.

13 tests passed. A mixed test of 200 unique synthetic events produced 150 successful effects and 50 explicit review cases, with no duplicate effects. The connector is simulated in a browser-local demonstration.

Order Desk | Python, React, TypeScript, FastAPI & SQLite

An order-management web application with API, database and Playwright browser checks. The project compares a strict TypeScript front end, a React implementation and a FastAPI server over shared application logic.

Original repaired baseline: 90 passing checks across API, database, tool and browser suites. Six deliberately planted defects were caught by the expected checks. Reproducible local builds, test commands and GitHub Actions configuration are available.

Document extraction & reporting | Python and Excel

Reusable workflows for invoice extraction, sales reconciliation and editable reporting. Outputs include source-linked rows, exception reports, formulas, charts and rerunnable tools.

Three fictional PDF invoices: all 11 line items extracted, with a planted GBP 10 discrepancy flagged. Sales demonstration: 1,774 source rows reconciled into 1,668 orders and GBP 38,437 synthetic revenue.

Data quality | CRM and ecommerce exports

Explicit matching rules, retained exceptions and reversible change logs for synthetic business datasets.

CRM: 5,000 input rows reconciled into 4,200 clean records, 650 duplicates and 150 exceptions. Catalogue: 260 rows, with 160 documented cell changes and a replayable change log.

Explore the working lab and project evidence: vorthryss.com/lab/

Engineering background

Independent Software Engineer | Self-directed projects
Remote, UK-based | June 2026 - present

SYSTEM DESIGN & DELIVERY

Originated the architecture and directed AI-assisted development of a persistent task platform connecting natural-language input, stored state, retrieval, tools, execution and verification, with a native macOS interface. Models are callable capabilities within the wider software system.

Investigated durable jobs, retries, evidence-linked answers, resource-aware scheduling and versioned releases. Used competing design proposals, targeted experiments and regression tests to challenge implementation choices. Kept source, decisions and test evidence together for review.

SELECTED RECORDED INVESTIGATIONS

Detecting swapped facts

A checker accepted values that existed in the sources but belonged to different claims. The repair tied each value to the subject in its source record.

The recorded evaluation rejected all 20 deliberately swapped answers. The result is tied to that documented test set.

Reducing email-panel latency

Investigated serial full-message retrieval blocking a macOS interface. The revised path reads message headers in batches and combines account results.

Recorded comparison: 64 seconds to 0.6 seconds across three accounts, with regression coverage.

Making delivery inspectable

Used commit-pinned releases, test checks, release records and rollback procedures. Developed automated checks with AI assistance and inspected failure conditions as well as passing outputs.

Historical full-suite run on 28 October 2026: 2,175 passing tests, zero failures. The dated result is retained with the engineering evidence.

TECHNICAL TOOLKIT

Application engineering: Python, FastAPI, REST APIs, TypeScript, React, JavaScript, HTML/CSS, SQL and SQLite.

AI and automation: tool-connected models, retrieval, evaluation, Ollama, MLX, n8n, task queues, retries, idempotency and durable state.

Data and quality: PDF/XML extraction, OCR, reconciliation, deduplication, Excel reports, pytest, Playwright, API and database checks.

Systems: Git, Linux/Debian, Docker, systemd, Bash, macOS, Swift and SwiftUI, monitoring and release recovery.

INTERNATIONAL COLLABORATION

I work in English. Project work includes synthetic Mexican and Brazilian e-invoice data, Arabic and English records, and international research and register lookups. The portfolio is presented in English, Spanish, Portuguese, French and Arabic.

I agree scope, expected outputs, working-hour overlap and communication cadence for each engagement. Larger projects can start with a useful paid technical milestone and expand from demonstrated results.