Viktor Sharha

Computer Science student at the University of Edinburgh focused on AI automation, NLP, and full-stack systems.

vsharha07@gmail.com | linkedin.com/in/vsharha | github.com/vsharha | viktorsharha.com

Education

University of Edinburgh — Edinburgh, Scotland
BEng Computer Science — *Expected Graduation: May 2028*

Technical Skills

Languages: Python, JavaScript, Java, C++, Bash, SQL, HTML/CSS
Frameworks: React, Next.js, Redux, Tailwind CSS, FastAPI
Tools & Platforms: Git, Linux, Docker, OpenAI API, Cloudflare

Experience

VDrive Chip, Key & Diagnostics — June 2025 – Present
Technical Assistant — *Edinburgh, Scotland*

- Reduced manual data processing time by 10+ hours monthly by building a Python automation tool that parsed 6+ months of inconsistent transaction records
- Designed brand identity and built a customer-facing booking website with an integrated booking system.

Standard Life — July 2024
Internship — *Edinburgh, Scotland*

- Designed and analyzed customer survey with 100+ parent and children respondents to identify key financial planning concerns for university-bound students, informing new FAQ content for Standard Life's mobile app
- Recommended 5 priority FAQ topics based on survey insights, presented findings to product and design teams

Awards & Honors

First Place, OpenEuler Challenge — October 2025
Samantha OS Assistant — *Edinburgh, Scotland*

Personal Projects

Samantha OS Assistant | *Python, OpenAI API, openEuler Linux, Natural Language Processing* — October 2025
A conversational AI terminal assistant that translates natural language into system commands

- Engineered a command-line interface (CLI) assistant that maps conversational requests to file, search, and diagnostic workflows via Python agents and the OpenAI API, streamlining day-to-day system maintenance.
- Built a modular tool framework with extensible base classes for web fetches, media handling, and directory analysis, enabling rapid addition of new capabilities.
- Enabled self-correction with conversation memory that logs failures, retries alternative strategies, and preserves context across multi-turn tasks to maintain reliability.

Wordle International | *React, Redux Toolkit, Tailwind CSS, Vite* — September 2025 – October 2025
Multilingual Wordle with configurable modes and API-backed words

- Shipped configurable game modes with React and Redux Toolkit so players can adjust word length and attempts for unlimited practice.
- Built a FastAPI random-word service supporting 15 languages; implemented an auto-adjusting on-screen keyboard with locale-aware input.
- Designed a responsive Tailwind UI for mobile and desktop with an adaptive grid and touch-friendly controls.
- Managed guess validation and keyboard state with Redux Toolkit, adding an offline fallback when the API is unavailable; added URL parameters using React Router.

Words-FrequencyPOS | *Python, Bash, Data Pipelines* — September 2025
Cross-lingual frequency + POS dataset pipeline powering language-learning and NLP research

- Delivered an automated extraction-transformation-load (ETL) pipeline that merges multilingual word-frequency and part-of-speech (POS) datasets with Python and Bash to support cross-lingual analysis.

- Linked frequency data with POS and morphology metadata to answer POS-specific vocabulary queries for linguistics and natural language processing (NLP) studies.
- Automated end-to-end runs with command-line scripts and environment-based configuration, enforcing idempotent downloads and resume-safe processing.
- Produced reproducible, versioned artifacts ready for batch processing and programmatic integration.

AI Expense Pipeline | *Python, AI provider APIs* July 2025 – August 2025

An automated invoice and receipt processing pipeline with configurable AI-driven extraction

- Constructed a modular pipeline that ingests PDF invoices and receipts, normalizes inputs with optical character recognition (OCR), and extracts structured expense data through AI-generated prompts to speed bookkeeping.
- Designed YAML-driven processing passes with carry-forward fields, skip logic, and post-processing rules (joins, length limits, date bounds) to keep outputs consistent.
- Orchestrated multiple AI providers with structured logging and tests, mapping responses into typed fields for reliable auditing.