Shashwat Agrawal

+91 9420153261 | shashwatagrawal473@gmail.com | Linkedin | Github

Education

Vishwakarma Institute of Information Technology (VIIT) — Pune, Maharashtra
B.Tech in Information Technology, CGPA: 8.36 — *Sep 2024 – July 2027*

SiddhiVinayak Technical Campus — Khamgaon, Maharashtra
Diploma in Computer Science and Engineering, Percentage: 89.83% — *Sep 2021 – July 2024*

Experience

Open Source Contribution

Pandas

- Refactored **format_is_iso** by replacing an explicit loop-based ISO datetime check with a compiled regex, reducing Python iteration overhead and improving performance. [PR]
- Fixed incorrect ISO week 53 handling in **to_datetime**, preventing silent rollover for non-existent ISO weeks, ensuring standards-compliant behavior for ISO calendar years with only 52 weeks, and adding regression tests [PR]
- Aligned datetime parsing documentation with the PDEP-4 deprecation of **infer_datetime_format**, reflecting updated behavior and preventing user reliance on deprecated APIs. [PR]

Collaborative Project

Text Content Summarizer — *Python, Flask*

- Led the development of a flexible Content Summarization API, utilizing Flask, the sumy library, and mediawikiapi to enable URL, text, and keyword-based summarization with the Latent Semantic Analysis(LSA).
- Leveraged mediawikiapi to retrieve Wikipedia content based on user-provided keywords.
- Reduced content summarization time by 40% through the utilization of the LSA and efficient API design.

Projects

tec.h | *C/C++*

- Built a header-only, zero-dependency unit testing framework for C/C++ using macro meta-programming for automatic test discovery, registration, and rich assertions.
- Added test suites, fixtures (setup/teardown), expected failures (XFAIL), skipping, and advanced filtering (positive/negative, file-level).
- Implemented rich assertions (equality, floats, strings, comparisons), colored output, and C++ exception testing.
- Actively maintaining with recent additions like exclusion filters and CI improvements.

Byte Machine | *Rust*

- Designed and implemented an 8-bit virtual machine with a custom ISA and 16-bit address space, complete with CPU emulation and memory management system.
- Engineered a custom two pass assembler that translates assembly code to bytecode, featuring support for label resolution, variables, direct memory addressing, and basic control flow.
- Implemented key components such as a stack, registers, and memory management, mimicking real-world processor functionality.
- Designed the VM and assembler for performance and extensibility, enabling easy addition of new opcodes and architectural features.

HTTP Server | *Rust, Tokio, Multi-threading*

- Developed a high-performance **multi-threaded and asynchronous HTTP server** leveraging **Tokio** for concurrency.
- Achieved a **100% success rate** handling **100,000 requests** at 50 concurrent requests/sec.
- Optimized response times ranging from **400µs to 32.8ms**, with an average of **5.9ms** and peak throughput of **8,423 requests/sec**.

Technical Skills

Languages: C, Rust, Python, JavaScript, SQL (Postgres)
Frameworks: Flask, Django, Node.js
Developer Tools: Git, Docker, Podman, NeoVim, Arch Linux
Libraries: Pandas, Matplotlib, Numpy, Tokio, Axum