# Shreyas Hegde

Github: github.com/shreyas-omkar
LinkedIn: linkedin.com/shreyas-omkar/

Email: shreyashegde@acm.org
Portfolio: shreyasomkar.vercel.app

## Education

- **Dayananda Sagar College Of Engineering** — Bengaluru, India
  *Bachelor of Engineering - Information Science and Engineering; GPA: 9.8/10* — *July 2024 - June 2028*

## Skills Summary

- **Programming Languages**: C, C++, Zig, Julia, Bash, TypeScript
- **Compilers & Toolchains**: LLVM, GCC, Clang, GDB, CMake, Make
- **Tooling & Platforms**: Git, GitHub, Linux

## Open Source Contributions

**GCC Rust (gccrs)**

- Contributed to compiler correctness and robustness improvements.
- Fixed internal compiler errors related to null pointer handling and canonical path resolution.
- Ensured semantic alignment with other GCC frontends.

**Julia & JuliaGPU Ecosystem**

- Resolved correctness and performance issues in array indexing and GPU dispatch.
- Implemented GPU native functionality to eliminate scalar fallbacks.
- Addressed method ambiguities involving views and reshaped arrays.
- Working on cuTile.jl regarding parellelism using alias aware token threading.

## Experience

- **JuliaGPU** — Remote
  *GSoC'26 Mentee / Maintainer* — *May 2026 – Present*
  - Maintainer of JuliaGPU, an organisation which implements all the vendor specific and vendor agnostic GPU and GPGPU kernels in Julia.
  - Contributing as Google Summer of Code Mentee in adding vendor agnostic kernels in AcceleratedKernels.jl.
  - Working on optimising KernelAbstractions.jl and making Vendor Agnostic kernels nearly match the performance of Vendor Specific Implementations.

- **Indian Institute of Technology, Kanpur** — Hybrid
  *Research Intern* — *May 2025 – Present*
  - Working on Heterogeneous dispatch systems for CPU/GPU/QPU.
  - Working on Cache warming and prediction based dispatch.

## Projects

- **SecureWipe**:
  (Kernel Level Secure Data Wiping Tool) **Open-source secure data wiping framework** (ISO based on Tiny Core Linux) focused on forensic-grade sanitization. Implements device level wipe workflows (ATA Secure Erase, NVMe Sanitize, SCSI SANITIZE, ADB/Fastboot), along with verification mechanisms to ensure irrecoverable data removal. **Actively maintained** with a stable build and ongoing contributions.

- **Compromyler**:
  (Compiler Integrity Verification Framework) **Tri-layer binary analysis system** to detect compromised compilers inspired by Ken Thompson's "Reflections on Trusting Trust". Implements **Diverse Double Compilation (DDC)** with a sandboxed trusted toolchain to generate reference binaries and compare against suspect outputs.

  Performs **CFG similarity analysis** (Ghidra + BinDiff), **dynamic syscall tracing** (ptrace-based), and **static ELF profiling** (sections, entropy, symbols) to detect injected behavior. Demonstrated detection of backdoored compiler introducing network-accessible payloads (socket injection) into compiled binaries.

## Participation & Achievements

- Got selected into *Google Summer of Code 2026*, in The Julia Language.
- Won **1st Place** in *Aegis Sandbox 2.0*, designing a compiler that injects vulnerabilities in every binary it compiles and a tool to test the integrity of the compiler.
- Won **Runner Up** in *HackNockturne 2.0*, for designing a solution to safely process DICOM files generated by medical imaging devices and use SiMG (cutom binary) to compare and process them.