Hemant Bhandari
https://linkedin.com/in/bhandari-hemant
https://github.com/laughingclouds

zeniththaiga@gmail.com
+91 63981 30461
Indirapuram, Ghaziabad, India

Education

Chandigarh University, Mohali, Punjab

Aug 2021 – (Expected) June 2025

Bachelor of Engineering, Computer Science; CGPA: 7.84

St. Thomas School, Indirapuram, Ghaziabad

2020 - 2021

Class 12, CBSE, PCM with CS; Grade: 84.8%

Experience

Research Intern

Jun 2023 – Jul 2023

MARS Lab, Department of CSE, IIT Guwahati

- Simulator Enhancement (C++): Extended gem5's C++ codebase to support custom routing in the Garnet interconnect.
- Python Automation: Wrote scripts to orchestrate batch simulations, collect metrics, and visualize results.
- Hardware-Trojan Research: Injected Trojan logic into simulated pipeline stages; designed and benchmarked detection/mitigation schemes.
- Performance Quantification: Measured throughput and latency impacts; analyzed trade-offs between security and performance.

Technical Projects

Kumo (PDF Reader)

June 2025

Designed and implemented a high-performance PDF reader with image-based dark mode using Dear ImGui and Poppler.

https://github.com/laughingclouds/Kumo

- Integrated Poppler with a custom Dear ImGui interface to render PDF pages, supporting lazy page loading and optimized resource usage for large documents.
- Developed a pixel-wise dark mode renderer that inverts embedded image colors for low-light readability without degrading text clarity.
- Structured the application into modular components for parsing, rendering, and UI, ensuring testability and maintainability.
- Leveraged smart pointers and RAII principles to manage memory safely during image decoding and rendering operations.
- **Technologies Used**: C++, Dear ImGui, Poppler, Git

Montagne (Markdown Editor)

May 2025

Developed a modern, extensible Markdown editor with live preview capabilities using Rust and the Iced GUI framework.

https://github.com/laughingclouds/montagne

- Engineered a real-time Markdown rendering engine with syntax highlighting and custom theming, enabling distraction-free editing and enhancing user productivity.
- Handled asynchronous file I/O with Tokio, allowing non-blocking handling of large Markdown files (>10 MB) for a responsive user experience.
- Architected the application with a modular component system, paving the way for future features like inline editing and plugin-based widgets.
- **Technologies Used**: Rust, Iced, Tokio, Markdown, Git

MonkeyGo (Programming Language)

March 2024

Built a full interpreter for the Monkey programming language in Go, inspired by Writing an Interpreter in Go by Thorsten Ball.

https://github.com/laughingclouds/monkeygo

- Developed a complete compiler front-end with a recursive descent lexer, parser, and AST evaluator, supporting dynamic typing and first-class functions.
- Implemented closures, arrays, hash maps, and a REPL interface for interactive programming and expression evaluation.
- Designed the runtime environment with support for lexical scope resolution, enabling proper variable shadowing and function-level environments.
- **Technologies Used**: Go, Recursive Descent Parsing, Git

Skills Summary

Languages:	Rust, Go, C/C++, Python, Java, JavaScript
Frameworks/Libraries:	ImGui, Iced (Rust GUI), Flask, React
Systems & Simulation Tools:	gem5, Bash, Make, Shell Scripting
Version Control & DevOps:	Git, GitHub, GitLab, Docker
Databases:	MySQL, PostgreSQL, SQLite
Operating Systems & Platforms:	Linux (Debian, Ubuntu), WSL, Windows