

# Yumin Na

Seoul, South Korea

04nayumin@gmail.com | github.com/insam16 | LinkedIn | Portfolio

## Summary

---

Embedded systems developer bridging electrical engineering, electronics, and low-level programming to deliver efficient and reliable hardware–software integrated solutions.

## Education & Learning

---

### 42 Curriculum / Self-directed Programming Study

- Studied core C++ concepts including subtype polymorphism, abstract classes, interfaces, deep copy, and memory management.
- Practiced low-level debugging, implementation reasoning, and project-based problem solving.
- Built a habit of learning through direct implementation, revision, and iterative refinement.

### Seoul Robotics High School

- Studied fundamental electrical and electronic engineering concepts, including circuits, digital logic, and computer architecture
- Developed programming skills in C and a strong foundation in data structures
- Gained hands-on experience in electronic control and industrial hardware design and development
- Designed and implemented robot hardware and software control projects
- Built IoT-based systems and acquired experience in hardware–software integrated system design

## Projects

---

### minishell (Unix Shell Implementation)

#### *Description:*

Implemented a simplified Unix shell in C, replicating core Bash functionalities. Developed the full execution pipeline including command parsing, process creation, I/O redirection, and pipe handling.

#### *Tech:*

C, Linux, POSIX APIs (fork, execve, pipe, dup2), Makefile

#### *Role:*

- Implemented heredoc functionality
- Developed pipe and I/O redirection handling logic

- Managed process creation and execution flow using fork/exec
- Implemented environment variable handling and built-in commands

*Challenges & Solutions:*

- Resolved file descriptor mismanagement causing resource leaks in pipe execution
- Fixed execution order and precedence issues in multiple redirections
- Stabilized parent-child process synchronization and exit handling
- Refactored fragile parsing logic into a robust token-based structure for complex inputs

*What I Learned:*

- Memory leak detection and management using Valgrind
- Deep understanding of OS-level concepts such as processes, file descriptors, and system calls
- System-level program design through implementing shell internals
- Importance of exception handling and resource management for system stability
- Experience in designing structured parsing logic for complex inputs
- Writing test cases and performing systematic debugging

## **Inception (Docker-based Service Infrastructure)**

*Description:*

Built a containerized web service infrastructure using Docker and Docker Compose, integrating Nginx, WordPress, and MariaDB. Configured service isolation, networking, and HTTPS for a secure and modular environment.

*Tech:*

Docker, Docker Compose, Nginx, MariaDB, Linux

*Role:*

- Designed overall service architecture and Docker Compose configuration
- Wrote Dockerfiles and built images for each service
- Configured Nginx reverse proxy and HTTPS setup
- Designed container networking and volume structure

*Challenges & Solutions:*

- Resolved inter-container communication issues through proper network configuration and service name-based addressing
- Fixed HTTPS certificate application and port redirection conflicts
- Solved WordPress-MariaDB initialization failures by adjusting environment variables and startup order

- Debugged port mapping and external accessibility issues via Nginx configuration

*What I Learned:*

- Service isolation and dependency management in containerized environments
- Practical understanding of reverse proxy and HTTPS mechanisms
- Importance of network design and initialization order in infrastructure setup
- Experience in designing system architecture from a holistic perspective

## Skills

---

**Languages:** C, C++, GDScript

**Frameworks / Tools:** Git, Docker, Godot 4

**Hardware:** Circuit Design, Circuit Debugging, Oscilloscope, PCB Design, Soldering

**Areas:** Embedded systems, hardware–software integration, electronic control, debugging, system programming

## Languages

---

- Korean (Native)
- English (Intermediate / Technical reading)
- Japanese (Basic)