Seungwon Shin

Backend Engineer / Software Engineer

Email: ssws1005@gmail.com | GitHub: https://github.com/wondde | Portfolio: https://wondde.com

Summary

Backend engineering candidate studying Industrial Engineering and Cybersecurity as a double major at Ajou University. I connect user-facing flows to Go/PostgreSQL APIs, data integrity, and the operational work needed to ship and improve services. My recent work spans production delivery, NL2SQL quality evaluation, WebRTC signaling, and a Unicode-aware Dart package published on pub.dev. I distinguish team outcomes from personal ownership and verified results from untested hypotheses.

Education & Languages

- Ajou University, B.S. in Industrial Engineering / Double Major in Cybersecurity, expected February 2028
- Korean: Native | Japanese: JLPT N1 (July 2025) | English: Technical reading and documentation

Experience

Software Engineering Intern, LSware

2026.06.29 - 2026.08.31

- Began with OMNIGuard as the only product domain in scope within a poorly documented PostgreSQL environment of roughly 600 tables. After the initial results were recognized, my scope expanded to another product domain. Mapped product screens to runtime logs, DDL, and existing SQL to recover logical-deletion, latest-history, and product-scope rules without exposing internal schemas.
- Separated reusable domain and safety rules into the system prompt while keeping question-specific JOIN and aggregation patterns in worked examples; authored roughly 176 validated question-SQL examples, compared with 30-40 in similar internal work I reviewed.
- Fixed a separate set of 80 questions that were not added to the examples, then compared the final configuration's generated SQL against predefined expected-SQL criteria as a holdout evaluation. A configuration without shared rules met roughly 30-40% of the criteria; the rules-plus-examples configuration met 80/80. This is a scoped internal result, not general NL2SQL accuracy.
- Did not own model training, embeddings, reranking, serving infrastructure, or customer deployment.

Full-stack Developer / Code Reviewer, Blueberry Team

2025.10 - Present

- Contribute to a roughly 15-person team of engineers, designers, and marketers across Umaso, Lingding, PokeTree, and internal developer tooling.
- Expanded from Flutter implementation into Go APIs, PostgreSQL, admin tooling, internal operations, code review, and Go-server onboarding documentation.
- Coordinate feature work through GitHub Issues, PRs, Notion, and Asana; proposed a monorepo and feature-owner approach so one change can be followed from UI through API and operation.

Selected Projects

Umaso - Content-first Korean Recipe App

2026.02 - Present | Flutter, Go, chi, pgx, sqlc, PostgreSQL, R2, Next.js, Docker

- The original ingredient-and-substitute concept accumulated features before its target problem was clear. Proposed freezing new scope, reduced v1 to team-managed recipe content, and shipped it on Google Play instead of delaying indefinitely.
- Implemented Flutter features and Go/chi APIs for recipes, community interactions, blocking/filtering, likes, and R2 presigned uploads using pgx/sqlc and PostgreSQL.
- Built the Next.js admin workflow for recipes, ingredients, translations, users, and posts, and operate it on a personal home server for the team’s internal work.
- Traced repeated user logouts to dependency injection that wired admin JWT settings into the user refresh path, then restored separate user/admin secrets and lifetimes.
- Release proves delivery, not validation of the original product hypothesis; public usage and retention metrics are not yet available, and the next user/value direction is under discussion.

Lingding - Korean-Japanese Language Exchange App

2026.05 - Present | Flutter, Go, WebRTC, WebSocket, STUN/TURN, PostgreSQL

- Own the one-to-one audio path and Go WebSocket signaling lifecycle, including room membership and SDP offer/answer and ICE-candidate relay.
- Prefer direct STUN paths with Cloudflare TURN fallback and issue short-lived TURN credentials from the server rather than embedding long-lived secrets in the client.
- Production connection rate, tail latency, relay ratio, reconnection, and soak-test results have not yet been measured.

PolyLog - AI Diary Learning App

2025.10 - 2025.12 | Flutter, Firebase, Gemini API, Hive | Solo

- Turned a study-group problem into a 12-hour MVP and continued through App Store v1.0.1 release; the available aggregate shows roughly 30 downloads.
- Kept the Gemini key behind Firebase Functions, constrained responses with JSON mode, and mapped them into Flutter models with JSON.parse. Runtime schema validation, retries, and error logging remain backlog.
- Kept Firestore as the remote source of truth while serving routine vocabulary and review reads from Hive; the source repository is private.

Unicode Typing Grader - Unicode-aware Typing OSS

2026.08.24 - Present | Dart, Unicode, Grapheme Clusters, GitHub Actions | Solo OSS

- Extracted a typing grader first needed in a personal app into a Pure Dart package and published `unicode_typing_grader` 0.1.2 on pub.dev with English, Korean, and Japanese documentation.
- Defined NFC normalization, grapheme-level edit distance, deterministic edit paths, versioned policies, and conformance fixtures; formatting, static analysis, 39 tests, and publish dry-run pass.
- Adopted the public package in the app through a small adapter, removing duplicate comparison and metrics implementations while preserving its existing storage and display models.

Additional Evidence

- AJOU ARCADE: Built four browser eye-tracking games solo in three weeks / 58 public commits; operated them at an outdoor fair and redeployed a brighter mode on site. Applications at the same stage changed from 9 to roughly 150, but the result included audience, staffing, prize, and promotion changes.
- PokeTree: Proposed an 80+1 collection loop as an alternative to push notifications and handled image assets and sharing UI in a 12-person seasonal service with roughly 15,000 GA4 visitors. Server and core APIs were owned by other members; the collection feature's independent retention effect was not measured.
- Figma MCP Bridge: Contributed to architecture and MCP server work in a 3-person OSS team; owned Codex setup and multilingual documentation, and validated the bridge across Claude Code, Windsurf, and Codex. A single sample produced a 2,307-line / 142 KB REST response, not a controlled benchmark.

Technical Skills

- Backend: Go, chi, pgx, sqlc, PostgreSQL, SQL, goose, JWT, REST, WebSocket
- Client / Web: Flutter, Dart, Riverpod, Next.js, React, TypeScript
- Infrastructure: Docker, Docker Compose, Cloudflare Tunnel, R2 presigned uploads, Firebase, Vercel
- AI / Data: NL2SQL evaluation, prompt and few-shot design, MCP integration, JSON-mode LLM responses
- Collaboration: GitHub PRs, code review, onboarding documentation, issue and phase management