

Diya Solanki

Software Engineer | Language Servers, Developer Tooling & Backend Systems

Rust · TypeScript · LSP/JSON-RPC · WebGL · Tokio · PostgreSQL

Phone: +91 7726952683

Email: diya.solanki.31@gmail.com

Location: Bangalore, India

LinkedIn: linkedin.com/in/diyasolanki

GitHub: github.com/diyaayay

Portfolio: diyasolanki.com

EXPERIENCE

Oracle

Bengaluru, India

Software Developer

Jul 2025 – Present

- Built a VS Code extension that authenticates to Oracle DBs, runs SQL via the Oracle client, and commits changes — removing the team's dependency on SQL Developer.
- Automated pre-drop VM preparation over SSH (repo sync, process cleanup) and wired it into Jenkins release-pipeline triggers, replacing the team's manual per-release checklist.
- Raised MFCS automated test coverage, migrated TortoiseSVN workflows to Oracle's internal VCS, and generated Postman suites from Swagger REST definitions.

p5.js — Processing Foundation

Remote

Contractor, Core Library

Nov 2023 – Jun 2025

- Extended the WebGL/3D pipeline with `.obj/.mtl` mesh export and texture/material bindings, so DCC-exported assets (e.g., Blender) render in-browser with the expected materials.
- Hardened `p5.Geometry` and camera paths — `ortho()` projection edge cases, curve caps, and vertex-normal wiring — eliminating silent draw failures in real-world sketches.
- Reworked the input stack for p5.js 2.0+: corrected touch vs. mouse event ordering and restructured `mouseButton` into an object model.
- Drove the JavaScript→TypeScript migration via hand-curated `.d.ts` declarations for public APIs. [🔗 12 merged PRs](#)

Processing Foundation Software Development Grant (pr05)

US, Remote

Software Developer (Grant-Funded)

Jul 2024 – Oct 2024

- Designed a client/server language server for Java-based Processing over JSON-RPC (LSP/LSIF) — hover, go-to-definition, references, completion, diagnostics, and real-time code analysis.
- Built a platform-independent project layout in Rust using symbolic links to bind protocol, server, client, and the Processing runtime into one workspace.
- Built custom Java JARs with Apache Ant and embedded the Processing runtime headlessly for VS Code sketches.
- Released on the [VS Code Marketplace](#); 1,900+ installs.

Google Summer of Code — JSON Schema (@json-schema-org)

US, Remote

Software Engineer

May 2024 – Aug 2024

- Implemented a JSON Schema language server in TypeScript with a pub/sub JSON-RPC core; exposed AST/diagnostics, completion, hover, and go-to-definition to VS Code and Neovim clients.
- Extended the server for draft/version skew and custom dialects: `$vocabulary` registration, default dialect resolution, `if/then/else` schemas, dialect-aware completion, and live workspace config reloads without re-init.
- Stabilized editor integrations on macOS/Windows/Linux; wired npm release steps and Vitest suites that lifted line coverage **23%→85%**; merged **19 PRs** and closed **20+** issues. [final report](#)

Hewlett Packard Enterprise

India, Remote

Software Engineer Intern

Feb 2024 – Jun 2024

- Flask + MongoDB API for HPE protection policies (snapshots, on-prem/cloud backups) with JSON schema validation on ingest.
- Recovery-point forecasting from `backupFrequency` and `retainFor`: normalized schedules to minutes and projected active copy counts at future times.
- Schedule dependency tree (DFS) and overlap simulator with source lineage; pandas + pickled XGBoost for delta/cost-size estimates. [🔗](#)

PROJECTS

Better Uptime Monitor

- Rust, Tokio, Poem, Diesel, PostgreSQL, JWT (HS256), Argon2, Docker, GitHub Actions
- Multi-crate workspace: JWT-protected REST API, embedded Diesel migrations, and a Tokio worker for scheduled HTTP checks with persisted history and latest status.
- Liveness/readiness endpoints, **tracing** request logs, and per-site webhooks (POST JSON on up/down transitions).
- CI with fmt/Clippy/build and Postgres-backed integration tests for the API and store.

Federated Medical MNIST

- Rust (Exonum 1.0-rc, MerkleDB proofs), Protobuf, Tokio; Node (**exonum-client**); Python training; React UI
- Permissioned-chain ML service: clients submit signed **ShareUpdates** gradient txs; **SyncBarrier** coordinates BSP/SSP/BAP-style release of aggregated global weights on-chain.
- Public API exposes model state with Merkle/block proofs; Python computes local updates from MedMNIST-style tiles, Node submits txs to the Exonum explorer REST API.

Cove — Cross-Platform Rust Bitcoin Wallet (open source)

- Rust; bitcoinpl/cove, 110+ stars
- Merged regression tests for the SATSCARD (NFC tap-card) parser.

ACHIEVEMENTS

Processing Foundation Advisory Committee Member — roadmap input for p5.js, Processing, and editor/tooling.

GSoC 2025 & 2026 Mentor, Processing Foundation — mentoring [Context-Aware Autocomplete for the p5.js Editor](#) (2025) and [full texture \(.mtl\) support for .obj 3D geometries](#) (2026).

GSoC Mentor Summit 2025 (Germany) — Processing Foundation delegate; open-source sustainability, governance, and release hygiene.

TECHNICAL SKILLS

Backend & APIs: Poem, Tokio, Flask, REST, JSON-RPC, LSP/LSIF, Swagger/OpenAPI, Protobuf

Languages: Rust, TypeScript, JavaScript, Java, Python, GLSL

Graphics: WebGL, GLSL, p5.js, Canvas, Three.js, Blender, glTF, .obj/.mtl, vertex/fragment shaders

Data: PostgreSQL, MongoDB, Diesel, SQL, JSON Schema, Exonum (MerkleDB)

CI/CD & build: Docker, GitHub Actions, Jenkins, Maven, Cargo, Ant, npm

Quality: Vitest, Mocha, pytest

ML: PyTorch, NumPy, pandas, XGBoost

Tools: Git, Linux, macOS, VS Code, Neovim, Postman, SSH, Jira

EDUCATION

BMS College of Engineering

Bachelor of Engineering in Information Science and Engineering, GPA: 8.9/10

Bangalore, India

Dec 2021 – Jun 2025

High School (CBSE)

Class XII — 94.5%

Jodhpur, India

2020