

Faris Alrusaini

Rust Fullstack Engineer | Dioxus & WebAssembly Systems Engineer

Riyadh, Saudi Arabia | +966552377038 | faris.it4@gmail.com

🌐 faris193 📘 Faris Alrusayni 📁 My Portfolio

(May 2026)

About Me

I build production systems in Rust — specializing in fullstack development with Dioxus, from database schema to WebAssembly frontend. I founded Sanad and built it alone, owning every architectural decision. I combine systems thinking with product engineering. Diploma in Programming & Development (GPA 3.98), self-directed in Rust, Dioxus, and WebAssembly. RoboRumble 2026: ESP32 robotics.

Engineering Approach

- Single-codebase architecture — the same Rust code compiles to WebAssembly on the client and native on the server through feature flags, keeping the entire system consistent.
- Domain-driven database design — schemas model business entities, not application features, which keeps the data layer stable as the product grows.
- Authentication as infrastructure — bcrypt, email OTP, session management, and role-based access control are designed as a security layer from the start.
- Production-first deployment — Docker multi-stage builds, CI/CD pipelines, and container registry deployment are part of the initial project setup.

Technical Toolkit

- Core Stack: Rust, Dioxus, Axum, Tokio, PostgreSQL, SQLx, WebAssembly, Tailwind CSS, Serde
- Infrastructure: Docker, GitHub Actions, CI/CD, Linux (Fedora)
- APIs & Integration: Twilio WhatsApp API, SMTP (lettre), Supabase, REST API Design
- Security & Auth: bcrypt, OTP, tower-sessions, Role-Based Access Control
- Frontend: HTML5/CSS3, JavaScript/ES6+, RTL/Arabic Layouts, Responsive Design, CSS Animations
- Embedded: ESP32, C/C++, Robotics, IoT Architecture

mylink - Link-in-Bio Platform

I built mylink because existing link-in-bio tools didn't serve the Arabic market well — they lacked RTL support, deep visual customization, and local integration. The platform lets users create fully customizable public profile pages with links across 29 platforms, shared at a personal URL. The architecture is a single Rust codebase: a Dioxus WebAssembly frontend with an Axum server. The core engineering challenge was the customization system — over 80 interdependent parameters controlling backgrounds, layouts, link styles, and header positioning. I solved this by generating CSS dynamically from PostgreSQL settings at render time,

treating each public profile as a database-driven style engine.

- Session-based authentication with bcrypt hashing, 30-day inactivity expiry, and admin/user role separation.
- Multipart file upload system with real-time XHR progress tracking, file-type validation, and size enforcement.
- Link management supporting 29 platforms with automatic URL generation and platform-specific icon rendering.
- 11 database migrations documenting the schema's evolution across profiles, links, wallets, and admin assets.

Sanad – Flagship Project

Sanad is a production-grade electronic invitation platform I built for the Saudi market. It manages the complete lifecycle — user registration, visual invitation design via a drag-and-drop editor, guest management, WhatsApp delivery and RSVP tracking, event management, and an administrative dashboard — all in a single Rust codebase.

The most technically demanding component is the visual invitation editor. I designed it as a layer-based rendering engine running entirely in the browser as compiled WebAssembly. The editor supports multiple layer types — text, icons, images, QR codes, decorative elements, and grouped templates — each with its own data model and rendering logic, unified under a shared abstraction. Interaction is real-time and server-independent, with bounds-constrained manipulation and client-side image compression.

Authentication uses bcrypt password hashing with email OTP verification and configurable security policies, backed by a PostgreSQL session store. Role-based access control separates users from administrators. The Twilio WhatsApp integration handles outbound delivery and inbound RSVP parsing through a webhook, replacing manual workflows with an automated auditable system. The platform includes a tiered subscription system, credits wallet, event management, support chat, and admin dashboard — deployed via Docker builds and GitHub Actions to GitHub Container Registry.

- Designed and built entirely solo: database architecture, authentication, frontend, APIs, and deployment.
- Layer-based visual editor running in WebAssembly with real-time canvas rendering and client-side image compression.
- Twilio WhatsApp integration with automated delivery, inbound RSVP webhook parsing, and tiered subscription system.

Keywords

Rust, Dioxus, Axum, WebAssembly, PostgreSQL, SQLx, Tailwind CSS, Twilio, Docker, GitHub Actions, Fullstack, SaaS, Authentication, WebAssembly Frontend, REST APIs, CI/CD, Systems Engineering, RTL, Supabase