Junaid Farooq

Senior / Lead Elixir Engineer. Elixir, Phoenix, Ruby on Rails, Node.js

fjunaid252@gmail.com | Pakistan, remote | ijunaidfarooq.com | github.com/ijunaid8989 | linkedin.com/in/iamjunnii

INTRODUCTION

Hello, I have been a programmer for more than 10 years in Elixir, Ruby and Node.js, with an appetite to learn and try new things. I mostly learned through R&D. I love to contribute to open source and I tend to poke my nose into strange and complicated problems. I still need to know a lot of stuff, but I think I have the answers.

WORKING EXPERIENCE

Senior Elixir Software Engineer

Apr 2025 to present

Whippy, messaging platform for staffing teams (remote)

I built the Bullhorn integration from scratch and do most of the ATS sync work (Bullhorn, TempWorks, Avionte, Aqore). 1,100+ commits and 200+ merged pull requests so far.

- Built the **Bullhorn ATS** sync in Elixir: proxy or direct auth with a fresh login on rejected tokens, stepped self onboarding, and live Bullhorn metadata so each customer syncs its own standard and custom objects.
- Split sync into separate **Oban** lanes: a 15 minute tip pull, page range backfills that claim a range before fetching it, a last N days window, and two push lanes into Whippy. Every queue limit is **per integration**, so one big customer cannot starve the rest.
- Worked within Bullhorn's limits (100k calls a month, 1,500 requests a minute): trimmed candidate pulls from 244 fields to 85 (500 rows in about 6 s instead of 42 s per 100) and split tip pulls into an ID scan plus parallel fetches.
- Found **silent data loss**: Bullhorn caps pages with one to many fields at 200 rows, which skipped 300 of every 500. Also fixed short pages ending backfills early and cursors saved from the wrong point.
- Stopped 429 responses from Whippy's API being stored as permanent errors, which had kept 400k+ contacts from syncing; new **Postgres indexes** took the tip query from 1,037 ms to 2.4 ms.
- Made it safe under concurrency: row locks, a Postgres **advisory lock** for gap recovery, claim before write for Bullhorn Notes, config version checks for edits after onboarding, and a job that restarts dead push chains.
- Hash based change detection so only changed records are pushed; custom data push path cut from seven regex passes per key to one.
- **TempWorks, Avionte, Aqore:** self onboarding with connection tests, per object sync toggles, last message activity sync, retention and archiving policies, Slack alerts on sync failures.
- Custom **roles and permissions** with invite time role assignment, backend permission checks, OAuth scopes on the public API and audit logging.
- **Forms Builder**: versioning and publishing, public APIs, responses turned into inbound conversation messages, email/SMS notifications.
- Moved usage and campaign reporting onto a unified messages model, with automated comparison runs delivered to Slack and S3.

Lead Elixir Software Engineer

Sep 2024 to Apr 2026

Truely.com, travel eSIM platform (remote)

- Built and extended a multi endpoint **Elixir/Phoenix** monolith (internal dashboard, public API, marketing site, background jobs) for **eSIM plans**, fulfilment and provider integrations.
- Implemented **Telna** as a first class provider: template sync, product and channel mapping, channel products v2 manager, and dashboard sync/resync from product type pages.
- Owned **Telna webhooks** end to end (eSIM status, package usage and alerts, package status, IMEI changes) and wired usage and status into plan updates, including ICCID resolution edge cases.
- Built **eSIM subscription sync** (Webbing and Telna flows, eUICC profile API, scheduled workers, rate limited batch sync) and organisation facing events for the install/enable lifecycle.
- Hardened operational behaviour: cancel/suspend flows, balance line activation rules, pause/resume plans, and legacy vs unlimited plans.

- Introduced and maintained **ClickHouse** event analytics (local cluster setup, schema and migrations, replication fixes, CDR style naming) and KPI dashboards.
- Added **GA4** conversion tracking and moved conversion handling onto **Oban** for reliable async processing.
- Added PostgreSQL **trigram fuzzy search** (queries, GiST indexes, tests) for coverage style endpoints.
- Expanded the V2 API (OpenAPI auth/login, orders, user eSIM APIs, pipelines) with **rate limiting** on sensitive endpoints.
- Implemented share group redemption and fixed eSIM ownership in group flows.
- Improved background jobs (Oban workers, throttling for plan updates, job cleaner, Oban Web visibility).
- Security: signup and login rate limits, IP allowlisting from the database, less password leakage in OAuth error paths, log sensitivity filtering, SQL injection hardening.

Senior Elixir Software Engineer Jul 2024 to Nov 2025

Yolo Group, B2B contract through my UK legal entity (remote)

- Architected and led distributed **microservices** focused on scalability, fault tolerance and high throughput data processing.
- Designed a search platform on **Elasticsearch/OpenSearch** with index lifecycle management (rollover, retention policies) for performance and storage at scale.
- Built a domain driven search abstraction layer for pattern based querying across multiple indices and services.
- Developed an internal observability and analytics platform (Kibana like) for search, debugging and operational insight across distributed systems.
- Built real time pipelines with Phoenix channels and **WebSockets** streaming live updates across clients and domains.
- Designed an analytics system on **ClickHouse** for large scale aggregation and reporting.
- Led cross service communication with **Protobuf** contracts for type safety, backward compatibility and efficient serialization.
- Built messaging and notification infrastructure, including **Telegram bots** and multi tenant broadcast systems.
- Improved reliability and observability with monitoring patterns inspired by Phoenix Observer and custom telemetry pipelines.

Senior Fullstack Elixir Contractor Sep 2021 to Apr 2025

Helium Health, hospital and clinic software (remote)

- Shipped **OpenAPI** documentation and Swagger UI so internal and partner teams could integrate consistently.
- Built institution bootstrap and **feature flags** so new tenants could be onboarded and capabilities toggled per customer.
- Delivered institution administration APIs: profile, locations and clinics, **roles**, tariffs, tags, teams, queues, dynamic forms, lab catalog mapping and audit visibility.
- Developed patient lifecycle APIs: registration, longitudinal record, visits, medical history, vitals and charting, immunizations and televisits.
- Built clinical documentation (**EMR** style entries), templates, referrals, tasks, communications and clinical libraries.
- Implemented antenatal care, labour and delivery (partograph style data) and televisit co-invitations.
- Owned appointments and queues: scheduling, screening, in-queue operations and front desk coordination.
- Delivered inpatient features: wards, rooms and beds, admissions and in-hospital documentation.
- Built lab and imaging order workflows from ordering to results, approvals, billing hooks and external result ingestion.
- Implemented pharmacy, inventory, store and procurement APIs (prescriptions, dispensary, requisitions, budgets, stock, batches, consumables).
- Delivered a patient portal API (profile, booking, appointments, claims, attachments, read only clinical and billing views).
- Added realtime channels for notifications, queues, prior authorization requests and invoices, plus broad reporting and CSV exports.

Senior Fullstack Elixir Contractor Sep 2022 to Feb 2024

Nectarine Credit, credit application SaaS (remote)

- Built a **dynamic form system** (custom fields, step based fields, toggle and contact fields, multi file uploads, required/show logic, branching).
- Implemented **fingerprint and identity risk** checks end to end: visitor capture, repeated submission checks, IP and fingerprint scans, smart signals, Slack alerts.
- Developed company level email notification settings with admin controls and **parent/child inheritance**.
- Added **role and permission** features: company roles, user/company role sections, parent users on child companies, mutation access rules.
- Integrated **OpenCorporates** (API integration, payload details, result limits, company edit sync).
- Built opt-out and referral controls, application audit history with request agent logging, and EIN / Federal Tax Number support in UI and PDF exports.
- Delivered **NetSuite** and external integration work, omnichannel improvements (attachments, auto replies, cloud convert fallback), workflow alerts and CSV exports.

Elixir Developer (Contract)

Jun 2022 to Aug 2022

SafeBoda, mobility and fintech, Uganda (remote)

- Designed and developed the company's **e-wallet** integrated with the payments processor.
- Integrated payment drivers with **Interswitch** and **Flexipay**.
- Built an RMQ publisher on **RabbitMQ** with topic exchanges binding multiple queues, consumed with **Broadway**.

Lead Elixir Engineer (Fullstack: Node.js, Elixir, Ruby, Vue, React)

Dec 2015 to Jun 2022

Evercam, construction cameras, Ireland (Islamabad, remote)

- Rewrote a Ruby **Sinatra** API into an **Elixir Phoenix** API with GraphQL, Swagger docs and unit tests; maintained Rails admin and dashboard apps (RSpec, Capybara).
- Moved the platform from a monolith to **microservices**, separating business and camera logic; migrated the Rails frontend to **Nuxt** on the Phoenix API.
- Built and maintained an Elixir service running **1000+ camera workers** on RabbitMQ with producer/consumer logic and DynamicSupervisor.
- Built Snapshot Extractor on a **RabbitMQ fan out** topology with Oban, GenServer and BroadwayRabbitMQ; ingested Kafka streams with BroadwayKafka and Protobuf.
- Built embedded Elixir (**Nerves**) at the edge with Phoenix backends for telemetry and device health next to Hikvision and Axis cameras.
- Implemented **hot code upgrades** with Distillery and Ansible (write up on Medium).
- Ran SeaweedFS clusters for camera snapshots, RAID and Nagios on Hetzner, and AWS (Beanstalk, Amplify, EC2, S3, Route53, Lambda).
- Live streamed JPEGs with Phoenix channels and PubSub; connected services with Erlang RPC; Redis cache on the Phoenix API.
- Built a Node API with React, Apollo GraphQL and MongoDB, and device management tooling (edge-commander).
- Created a Rails app for the Eyes of Things computer vision project (Mobile Camera Demonstrator).

Development Executive (JavaScript Engineer)

Aug 2014 to Aug 2015

Highrise Technologies, IBM Business Partner (Islamabad)

- Worked around IBM Security Access Manager for Web, Security Identity Manager and Access Manager for SSO.
- Developed a ticketing system through three stages: PM2 with Node.js cluster, then RabbitMQ, then Redis cache with an Nginx reverse proxy.

Web Developer (Node.js)

Jan 2012 to Aug 2014

Websol Technologies (Faisalabad)

- Built and maintained Node.js and Express APIs with Mongoose; worked in Backbone.js, Webpack, ES6 and Handlebars.

OPEN SOURCE

- **prawn_ex**: PDF generation in pure Elixir, no Chrome needed. I maintain it (hex.pm).

- **Queue Sentinel**: groups RabbitMQ dead letter failures and explains each group in plain English.
- **Vision Sentinel**: camera health monitoring (lens blocked, frozen feed, tampering, exposure faults) with classical computer vision.

SKILLS

Backend: Elixir, Phoenix, Oban, Broadway, Absinthe GraphQL, Ruby on Rails, Sidekiq, Node.js, Express, TypeScript, SQL

Data and messaging: PostgreSQL, ClickHouse, OpenSearch/Elasticsearch, Redis, RabbitMQ, Kafka, Protobuf

Frontend: React, Apollo Client, Vue, Nuxt

Infra: AWS, Hetzner, Fly.io, Heroku, Docker, Terraform, Ansible, GitHub Actions, Nagios, SeaweedFS

Languages: English, advanced (spoken and written)