NAJIH Driss

Senior Software Engineer – AI Driven Apps

📍 France · najih.driss73@gmail.com · +33 7 49 52 96 00
🔗 LinkedIn: https://www.linkedin.com/in/najih-driss/
💻 GitHub: https://github.com/kouji-dev
🌐 Portfolio: https://www.kouji.dev
📅 Book a meeting: https://calendar.app.google/KTPtgPtkgrbJUEjm7

ABOUT

Senior Software Engineer with strong expertise in designing and building large-scale web platforms and backend frameworks. Extensive experience across JavaScript/TypeScript, Java, Angular, React, and cloud-native architectures.

Specialized in **complex frontend frameworks, backend systems, and platform-level development**, with experience working on **shared frameworks, low-code-like systems, and enterprise tooling** used by multiple teams and products.

Experienced in **AI-driven application design**, particularly for **backend systems prepared for AI integration**, with hands-on usage of **Python and LLM APIs in personal and proof-of-concept contexts** to design AI-enabled workflows, automation, and agent-based capabilities.

Strong focus on clean architecture, scalability, maintainability, and developer experience.

TECHNICAL SKILLS

Category	Technologies
Languages	JavaScript, TypeScript, Java 8/11/17/21, Python (AI/automation – PoC), Go
Front-End	Angular, NGRX, NGXS, Angular Material, React, Redux, Redux-Saga, Redux-Thunk, TailwindCSS, Ant Design, MUI, HTML5, CSS3, SCSS, Quill, Lexical, CKEditor, AG-Grid, Highcharts, ECharts
Back-End	Spring Boot, Spring MVC, Spring Data, Spring Security, Spring Batch, Hibernate, JPA, Node.js, Express.js, NestJS, Prisma, Quarkus
APIs	REST, GraphQL
Databases	PostgreSQL, Oracle DB, SQL Server, MongoDB, MySQL, Redis
Messaging	RabbitMQ, Kafka
Security	OAuth2, JWT, Keycloak, Azure AD (SSO)
Cloud	AWS (Lambda, Step Functions, S3, DynamoDB, EC2, Elastic Beanstalk, CloudFront), Azure (AD, App Services, CDN, Front Door, Blob Storage, PostgreSQL)
DevOps	Docker, Kubernetes, GitLab CI/CD, Jenkins
Testing	JUnit 5, AssertJ, Mockito, WireMock, Jest, Karma, Testing Library, Cypress, Playwright, JaCoCo, SonarQube
Tools & Practices	Agile Scrum, Git, GitHub, GitLab, Jira, Maven, NPM, Yarn, IntelliJ, Figma, TDD, DDD, BDD, Clean Architecture, Hexagonal Architecture

AI & PYTHON CAPABILITIES

- Use of **Python as a general-purpose language** for AI-driven applications, automation, and backend services (personal and PoC contexts)
- Hands-on experimentation with **LLM APIs**
- Design and testing of **AI agents**, prompt engineering strategies, and agent workflows
- Strong interest in integrating AI capabilities into existing enterprise systems rather than isolated ML projects

PROFESSIONAL EXPERIENCE

Amundi ITS — Senior Full Stack Developer (Angular / Java)

Since September 2023

Project context:
Amundi ITS develops shared platforms and frameworks used by multiple internal and client-facing

applications. A major product is the **Maestro framework**, an Angular-based solution enabling backend teams to rapidly integrate consistent, enterprise-grade user interfaces.

I worked within **two teams** on this project:

Framework Team:

- Maintained and evolved the **Maestro Angular framework** used by multiple internal and client-facing applications
- Actively participated in **major Angular version migrations (15 → 18 → 20)**, including component upgrades, dependency updates, and adoption of new framework capabilities (Signals, Standalone Components, etc.)
- Designed and developed **major reusable Angular components** integrated into the framework (forms, data grids, navigation, layouts, etc.)
- Participated in the **framework architecture design** to make it compatible with **micro-frontend environments**, enabling smoother integration into modular and distributed applications
- Managed versioning and delivery of framework releases, ensuring stability for consuming teams
- Ensured component quality, accessibility, and adherence to Angular best practices
- Provided **level 3 support** to application teams integrating the framework
- Worked closely with product owners and UX designers to translate business needs into scalable UI components

Lowcode Team:

- Designed and built a **Web Designer** enabling teams to compose screens visually, without code, through a drag & drop interface
- Architected a **low-code, configuration-driven UI framework** allowing screens and behaviors to be defined via structured JSON
- Helped replace a rigid declarative system with a **more scalable and maintainable architecture**
- Maintained and evolved the low-code platform (bug fixes, refactoring, performance improvements, new features)
- Led the **migration of several major client applications** to the new low-code platform, including critical business components (Employee Savings Plan) from legacy tools
- Contributed to the **architecture of client applications** from other teams, providing technical expertise and best practices
- Integrated **AI features into the Web Designer**, enabling **natural-language-driven generation of dynamic web applications** — users describe screens or flows in plain language and the designer scaffolds the corresponding JSON-defined views, components and bindings, ready for further refinement

AI Tooling Team — Vibe Toolbox (current):

- Currently building the **Vibe Toolbox**, an internal platform that **provisions AI-ready development environments** for engineers across the organisation, reducing time-to-productivity on AI-assisted workflows from days to minutes
- Implemented **skills management** — installation, versioning and sharing of reusable AI skills (prompts, workflows, agent definitions) so teams adopt validated patterns instead of rebuilding them per project
- Built **marketplace mechanics** for AI skills and agents, supporting discovery, packaging and lifecycle management with role-based governance suited to enterprise use

- Designed the toolbox as a **single entry point** for developers to bootstrap model access, credentials, MCP configuration and skill bundles, bringing consistency to how AI tooling is adopted across teams

Cross-team contributions:

- Designed, built and published multiple **internal Java and JavaScript packages** (libraries, shared utilities, framework modules) consumed by Maestro and downstream application teams
- Set up and maintained **automated CI/CD pipelines** (GitLab CI, Jenkins) covering build, lint, tests, coverage and release of framework and low-code artifacts
- Ensured **strong test coverage** across frontend and backend with **JaCoCo** (Java), **Karma** and **Jest** (Angular/TypeScript), plus **Cypress** and **Playwright** for end-to-end scenarios
- Daily use of **agentic AI tooling** (Claude, Codex) for code generation, refactoring, code reviews, debugging, and accelerating exploratory work on framework and platform features

Technical Environment:
Angular, TypeScript, Java 8/11/21, Spring Boot, Quarkus, Keycloak, Oracle DB, MongoDB, Karma, Jest, JaCoCo, Cypress, Playwright, Webpack, Jenkins, GitLab CI, Claude, Codex, Agile Scrum

---

KYC — Senior Java Developer

April 2023 – September 2023
Client: Société Générale

Project context:
A set of backend services dedicated to **Know Your Customer (KYC)** processes, used to evaluate customer trust levels and eligibility, and to support fraud detection workflows.

Responsibilities & Contributions:

- Participated in the **design and implementation of backend services** using Java and Spring Boot
- Developed and exposed REST APIs consumed by other internal systems
- Implemented security mechanisms using Spring Security and OAuth2
- Wrote unit and integration tests to ensure reliability and compliance
- Participated in code reviews and Scrum ceremonies
- Worked within a containerized, cloud-based environment

Technical Environment:
Java 17, Spring Boot, Spring Security, OAuth2, PostgreSQL, Quartz, Flyway, Docker, Kubernetes, RabbitMQ, Jenkins, GitHub CI/CD

Colas CRM — Senior Full Stack Java / Angular Developer

January 2023 – April 2023
Client: Colas

Project context:
A **global CRM system** deployed across 55 countries to manage commercial activities, opportunities, and customer data for the Colas group.

Responsibilities & Contributions:

- Developed backend services using Java and Spring Boot
- Built frontend features using Angular and Ant Design
- Implemented secure authentication using Azure Active Directory (SSO)
- Designed and consumed REST APIs
- Developed batch processes for data migration using Spring Batch
- Secured APIs with Spring Security
- Deployed the application on **Azure** across multiple environments — App Services, Front Door, CDN, Blob Storage and Azure Database for PostgreSQL — with CI/CD pipelines
- Actively participated in Scrum events and code reviews

Technical Environment:
Java 11, Spring Boot, Spring Batch, Angular 15, TypeScript, Ant Design, Azure (AD, App Services, CDN, Front Door, Blob Storage, PostgreSQL), GitLab CI/CD

Gloody — Full Stack Java / React Developer

October 2022 – January 2023
Client: Zsoft Consulting

Project context:
A **multi-tenant business management platform** enabling consultants to manage absences, leave requests, expense reports, certifications, and training.

Responsibilities & Contributions:

- Maintained and extended existing backend services
- Developed new REST APIs using Spring Boot
- Built frontend features using React and Ant Design
- Improved performance of reporting modules
- Collaborated with the team in Agile Scrum ceremonies

Technical Environment:
Java 8, Spring Boot, Spring Security, JWT, React, TypeScript, AWS, GitLab CI/CD

RiskReveal — Full Stack Java / Angular Developer

June 2019 – September 2022
Client: Scor

Project context:
An application dedicated to **natural disaster risk assessment**, used to analyze and visualize risk data for insurance-related decision making.

Responsibilities & Contributions:

- Designed and developed backend services with Spring Boot
- Developed frontend features using Angular
- Created and maintained REST APIs
- Implemented real-time features using WebSockets
- Participated in code reviews and pair programming sessions

- Contributed to UI customization using Ant Design and data visualization libraries

Technical Environment:
Java 8, Spring Boot, Spring Batch, Angular 12, SQL Server, Azure DevOps, Jenkins

---

SoliT — Angular Front-End Developer (End-of-Study Internship)

March 2019 – June 2019
Client: Scor

Project context:
A pricing management platform for internal pricing teams.

Responsibilities & Contributions:

- Developed frontend features using Angular
- Designed and styled UI components
- Integrated REST APIs
- Applied best practices in HTML, CSS, and TypeScript

SIDE PROJECTS

Orrery — Multi-Agent Git Orchestrator IDE (Desktop)

Role: Solo Author — Full Stack Developer
Type: Personal Open-Source Project
Repository: https://github.com/kouji-dev/orrery-releases

Project context:
Orrery is a futuristic, IntelliJ-inspired **multi-agent git orchestrator IDE**, built as a native desktop application with **Tauri (Rust) and Angular 20**. Each project is a git repository, and a project can **spawn multiple AI coding agents** (Claude Code, Codex, Cursor, Gemini) that each run in their own **git worktree, branch, terminal and conversation** — letting a developer drive several agents in parallel from a single cockpit. The name evokes an orrery: agents orbiting a project like bodies in a mechanical model of the solar system.

Responsibilities & Contributions:

- Architected the application as a **Tauri 2 desktop shell (Rust)** wrapping an **Angular 20** front-end, with a clean **module-per-domain** structure (projects, agents, workspace, orchestrator dashboard, right panel, sidebar, settings, …)
- Built the **orchestrator dashboard** — a hero overview with live stats and multiple visualization metaphors for monitoring agents across projects
- Implemented the **per-agent workspace** with Diff / Terminal / Chat panes: integrated terminals via **xterm.js (WebGL renderer)**, a diff/merge editor on **CodeMirror 6**, and rich-text / markdown editing via **Lexical**
- Designed the **spawn-agent flow** (project, branch, tool, model, effort, prompt) and an **agent-worktree runtime** so each agent is isolated on its own branch and worktree
- Integrated **agent cost tracking** (ccusage) and an **agent hooks / permissions** model to govern what agents are allowed to do

- Built a **design-token-driven design system** with theming, density, colour palettes and motion controls, plus real-time streaming logs and live timers
- Wrote the **Rust backend** pieces (auto-updater, app-icon generation, hooks CLI) and wired the **auto-update** flow via Tauri's updater plugin
- Established a **quality & release pipeline**: **Vitest** unit tests, **Playwright** end-to-end suites, a **performance smoke-test** harness with assertions, semantic version-bump scripts, and **GitHub Actions** workflows (including a Render-deployed landing site)

Technical Environment:
Tauri 2, Rust, Angular 20, TypeScript, Tailwind CSS 4, xterm.js (WebGL), CodeMirror 6, Lexical, RxJS, Tauri plugins (updater, dialog, notification, process, opener), ccusage, Vitest, Playwright, pnpm, GitHub Actions, Render

---

AI Portal — Self-Hostable Enterprise AI Assistant Platform

Role: Solo Author — Full Stack Developer
Type: Personal Open-Source Project
Repository: https://github.com/kouji-dev/ai-portal

Project context:
A self-hostable AI assistant platform — an *internal-ChatGPT-style* application combining streaming chat, retrieval-augmented generation (RAG) over private knowledge bases, model catalog management and per-user memory. Built end-to-end as a single-author project to validate production patterns for enterprise AI: provider abstraction, document ingestion, hybrid authentication and one-command deployment.

Responsibilities & Contributions:

- Designed and implemented the backend in **FastAPI**, structured into clear module boundaries (assistants, conversations, knowledge bases, model catalog, memories, API keys) to keep AI concerns isolated from product features
- Built **streaming chat** over **LangChain** with **provider abstraction** (OpenAI-compatible + Anthropic), enabling drop-in model substitution and routing through an OpenAI-compatible proxy with no application changes
- Implemented **RAG knowledge bases** on **PostgreSQL + pgvector** — upload, chunking, embedding, retrieval and conversation-side context injection, with ingestion offloaded from the request path
- Built the web client on **TanStack Start (React + Vite)** with file-based routing and TanStack Query, delivering chat, knowledge bases and memories surfaces backed by streamed responses
- Implemented **dual authentication modes** — local dev bearer token for fast iteration, and **Microsoft Entra ID** JWT validation with app roles for production — plus portal-issued API keys (`aip_…`) for programmatic access
- Productionised deployment: **Docker Compose** for local dev, a full-stack Docker profile, and a **one-command Render + Supabase blueprint** with Alembic migrations applied automatically on each deploy
- Designed a **self-hosted setup flow** (first-boot wizard, single-org provisioning) with API-wide 503 lockout until configuration completes, targeting organisations that want to run the platform privately

Technical Environment:
Python 3.12, FastAPI, LangChain, OpenAI & Anthropic APIs, PostgreSQL 17 + pgvector, Redis, SQLAlchemy 2.0, Alembic, Pydantic, TanStack Start, React, Vite, TanStack Router, TanStack Query, Tailwind CSS 4, Microsoft Entra ID (Azure AD), JWT, Docker, Docker Compose, Render, Supabase, pnpm

---

Kouji UI — Angular Design System & Component Library

Role: Solo Author & Maintainer
Type: Personal Open-Source Library
Repository: https://github.com/kouji-dev/kouji-ui

Project context:
An open-source **Angular 21** design system providing **60+ accessible, themable UI primitives** covering inputs, overlays, navigation, data display, feedback and layout. Maintained as a single-author library to demonstrate end-to-end *frontend platform engineering* — monorepo discipline, publishable package boundaries, accessibility-by-default and rigorous release management.

Responsibilities & Contributions:

- Architected a **pnpm + Turborepo monorepo** with three publishable packages — `components`, `core`, `themes` — plus a dedicated documentation app, sharing tooling, lint rules and a Tailwind v4 foundation
- Authored **60+ standalone Angular 21 components** (calendar, combobox, command palette, date/time pickers, dialog, drawer, file upload, form, OTP input, popover, stepper, toast, tooltip, tree-select, …) designed for tree-shakability and signal-based change detection
- Built a **decoupled theming layer** (`@kouji-ui/themes`) driven by design tokens, enabling theme switching, dark mode and per-product overrides without forking components
- Set up a **strict quality pipeline**: **Vitest** unit tests, **Playwright** end-to-end suites, automated **accessibility audits** (axe-based) producing JSON + HTML reports under `reports/a11y`, plus ESLint, Prettier and Husky pre-commit hooks
- Established **release engineering** with **Changesets** for semantic versioning and changelog generation across the three packages, wired into GitHub Actions workflows
- Wrote contributor documentation and **AI-collaboration guidelines** (`CLAUDE.md`, `RULES.md`) so the library can be extended consistently by both humans and AI coding agents

Technical Environment:
Angular 21, TypeScript, Angular CLI, pnpm workspaces, Turborepo, Vitest, Playwright, axe-core, ESLint, Prettier, Husky, Changesets, Tailwind CSS / design tokens, GitHub Actions

---

Pages — Unified Project Management & Documentation Platform

Role: Full Stack Developer
Type: Personal Side Project
Repository: https://github.com/kouji-dev/pages

Project context:
Pages is an ambitious full-stack platform designed to **replace the traditional Jira + Confluence stack** with a single, cohesive solution.

The goal is to eliminate context switching by unifying **issue tracking, sprint management, and technical documentation** into one modern, developer-friendly platform, while laying the foundation for **AI-enhanced workflows**.

Responsibilities & Contributions:

- Established a **scalable monorepo architecture** using **Domain-Driven Design (DDD)** principles, enabling clean domain boundaries and maintainable codebase structure across frontend and backend
- Delivered a **highly performant, reactive frontend** with **Angular 21** and **Angular Signals**, achieving efficient change detection and seamless user experience with reusable, responsive components following BOM methodology
- Engineered a **production-ready REST API** with **FastAPI (Python 3.12)** leveraging async/await for optimal I/O performance, integrated with **PostgreSQL** via **SQLAlchemy 2.0** and **Redis** for enhanced response times
- Implemented **end-to-end project management workflows** including sprint planning, issue tracking, Kanban boards, and backlog management with **real-time collaboration** capabilities via WebSockets
- Enabled **rich documentation authoring** through **Lexical editor** integration and ensured code quality with comprehensive test coverage (unit, integration, functional) and automated CI/CD workflows

Technical Environment:
Angular 21, TypeScript 5.7+, Python 3.12, FastAPI, PostgreSQL 17, Redis 8, SQLAlchemy 2.0, Lexical, Tailwind CSS 4, Docker, Docker Compose, Poetry, pnpm, WebSockets, JWT, Pydantic, Alembic, Git, GitHub Actions

---

GamerGG — Gaming Service Platform

Role: Full Stack Developer
Type: Personal Side Project
Repository: https://github.com/kouji-dev/GamerGG

Project context:
GamerGG is a **full-stack gaming service marketplace** designed to connect players with professional boosters and coaches.
The platform enables gamers to purchase **rank boosting, coaching services, and gaming packages** while providing boosters with tools to manage orders, track performance, and engage with clients through integrated **Discord** communication.

Responsibilities & Contributions:

- Architected a **Turborepo monorepo** structure enabling shared components and utilities across multiple Next.js applications with optimized build caching
- Built a **modern, responsive web application** with **Next.js 13**, **TypeScript**, and **Tailwind CSS**, implementing server-side rendering and API routes for optimal performance
- Designed and implemented **authentication and authorization** using **NextAuth.js** with OAuth providers and secure session management
- Developed a **robust database layer** using **Prisma ORM** with **PostgreSQL**, implementing migrations, seeding, and type-safe database queries

- Created **comprehensive order management system** with real-time status tracking, active boost monitoring, and order history for both clients and boosters
- Integrated **Discord server** connectivity for seamless communication between clients and boosters within the platform
- Implemented **user analytics dashboard** featuring top boosters leaderboard, user spending tracking, and performance metrics
- Established **reusable UI component library** shared across the monorepo with consistent design system and TypeScript types

Technical Environment:
Next.js 13, React 18, TypeScript, Prisma, PostgreSQL, NextAuth.js, Tailwind CSS, Turborepo, Yarn Workspaces, OAuth, Discord API, Git, GitHub Actions

---

Acta — Accounting Firm Project Management Platform

Role: Full Stack Developer
Type: Personal Side Project

Project context:
Acta is a **comprehensive project management platform** designed specifically for accounting firms to streamline **cross-functional production tracking** across multiple teams.
The platform enables real-time task planning, time tracking, and ticketing for complex accounting workflows, managing hierarchical structures of **portfolios, files, missions, and tasks** with progressive validation workflows and multi-dimensional progress visualization.

Responsibilities & Contributions:

- Architected a **full-stack monorepo** with **Angular 21** frontend and **Express.js** backend, implementing **role-based access control (RBAC)** with granular permissions for collaborators, supervisors, managers, and chartered accountants
- Designed and implemented a **complex hierarchical data model** using **Prisma ORM** with **PostgreSQL**, supporting portfolios → files → missions → tasks relationships with interconnected dependencies and progressive validation workflows
- Built **multi-dimensional visualization features** including calendar views for tasks and missions, progress tracking by collaborator or file, and time tracking dashboards with daily/weekly/monthly/yearly aggregations
- Developed **real-time task management** with status workflows, due date change request system with approval workflows, and automatic task assignment based on role-based folder assignments
- Implemented **comprehensive time tracking system** with detailed reporting capabilities, enabling analysis by collaborator, file, task, or mission with flexible filtering and aggregation
- Created **responsive, data-rich UI components** using **TanStack Table** for complex data grids, **Luxon** for date/time handling, and **Tailwind CSS** for modern, accessible design
- Established **robust authentication and authorization** using **JWT** and **Passport.js** with role-based route guards and permission-based feature access

Technical Environment:
Angular 21, TypeScript, Express.js, Prisma, PostgreSQL, JWT, Passport.js, TanStack Table, Luxon, Tailwind CSS, Docker, Docker Compose, pnpm, Git

---

Chartify — Figma Plugin

Role: Plugin Developer
Type: Figma Community Plugin
Link: https://www.figma.com/community/plugin/1361309122482309141/chartify

Project context:
Chartify is a **Figma community plugin** designed to streamline the creation of data visualizations and charts directly within Figma's design environment.
The plugin enables designers to quickly generate various chart types, customize data visualizations, and integrate charts seamlessly into their design workflows without leaving Figma.

Responsibilities & Contributions:

- Developed a **Figma plugin** using TypeScript and the Figma Plugin API to extend design tool capabilities
- Implemented **chart generation functionality** supporting multiple chart types and customizable data visualization options
- Created an **intuitive user interface** within Figma's plugin environment for easy chart configuration and customization
- Ensured **seamless integration** with Figma's design system, allowing charts to be easily incorporated into design files
- Published and maintained the plugin in the **Figma Community**, making it accessible to designers worldwide

Technical Environment:
Angular, TypeScript, Taiga UI, Figma Plugin API, Figma Community

---

EDUCATION

State Engineering Diploma in Computer Science
National School of Applied Sciences of Tangier — 2019

LANGUAGES

- Arabic
- French
- English