

NAJIH Driss

Ingénieur Logiciel Senior – Applications Pilotées par l'IA

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>

Prendre rendez-vous: <https://calendar.app.google/KTPtgPtkgrbJUEjm7>

RÉSUMÉ PROFESSIONNEL

Ingénieur Logiciel Senior avec **plus de 7 ans d'expérience** dans la conception et le développement de plateformes web à grande échelle et de frameworks backend. Expérience approfondie en JavaScript/TypeScript, Java, Angular, React et architectures cloud-native.

Spécialisé dans les **frameworks frontend complexes, systèmes backend et développement au niveau plateforme**, avec une expérience dans le travail sur **frameworks partagés, systèmes de type low-code et outils d'entreprise** utilisés par plusieurs équipes et produits.

Expérimenté dans la **conception d'applications pilotées par l'IA**, notamment pour les **systèmes backend préparés pour l'intégration IA**, avec une utilisation pratique de **Python et d'APIs LLM dans des contextes personnels et de preuve de concept** pour concevoir des workflows activés par l'IA, l'automatisation et des capacités basées sur des agents.

Fort accent mis sur l'architecture propre, la scalabilité, la maintenabilité et l'expérience développeur.

CAPACITÉS IA & PYTHON

- Utilisation de **Python comme langage généraliste** pour les applications pilotées par l'IA, l'automatisation et les services backend (contextes personnels et PoC)
- Expérimentation pratique avec les **APIs LLM**
- Conception et test d'**agents IA**, stratégies d'ingénierie de prompts et workflows d'agents
- Fort intérêt pour l'intégration de capacités IA dans les systèmes d'entreprise existants plutôt que dans des projets ML isolés

COMPÉTENCES TECHNIQUES

Langages: JavaScript, TypeScript, Java 8/11/17/21, Python (IA/automatisation – PoC), Rust, Go

Front-End: Angular (15 → 22), RxJS, NGRX, NGXS, Angular Material, React, Redux, TailwindCSS, Ant Design, MUI, HTML5, CSS3/SCSS, Lexical, AG-Grid, 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

Bases de Données: PostgreSQL, Oracle DB, SQL Server, MongoDB, MySQL, Redis

Messagerie: RabbitMQ, Kafka

Sécurité: 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), OVHcloud

DevOps: Docker, Kubernetes, GitLab CI/CD, Jenkins

Tests: JUnit 5, AssertJ, Mockito, WireMock, Jest, Karma, Testing Library, Cypress, Playwright, JaCoCo, SonarQube

Outils & Pratiques: Agile Scrum, Git, GitHub, GitLab, Jira, Maven, NPM, Yarn, IntelliJ, Figma, TDD, DDD, BDD, Clean Architecture, Hexagonal Architecture

EXPÉRIENCE PROFESSIONNELLE

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

Septembre 2023 – Aujourd'hui (3 ans)

Contexte du projet: Amundi ITS développe des plateformes et frameworks partagés utilisés par plusieurs applications internes et clientes. Un produit majeur est le **framework Maestro**, une solution basée sur Angular permettant aux équipes backend d'intégrer rapidement des interfaces utilisateur cohérentes et de niveau entreprise.

Équipe Framework

- Maintenu et fait évoluer le **framework Angular Maestro** utilisé par plusieurs applications internes et clientes
- Participé activement aux **migrations de versions majeures d'Angular (15 → 18 → 20)**, incluant la mise à niveau des composants, des dépendances et l'adoption des nouveautés du framework (Signals, Standalone Components, etc.)
- Conçu et développé des **composants Angular majeurs et réutilisables** intégrés au framework (formulaires, grilles de données, navigation, layouts, etc.)
- Co-conçu l'**architecture du framework** pour la compatibilité avec les environnements **micro-frontends**, facilitant son intégration dans des applications modulaires et distribuées
- Géré le versioning et la livraison des releases du framework, garantissant la stabilité pour les équipes consommatrices
- Garanti la qualité, l'accessibilité et le respect des meilleures pratiques Angular sur l'ensemble des composants
- Fourni un **support niveau 3** aux équipes applicatives intégrant le framework
- Collaboré étroitement avec les product owners et designers UX pour traduire les besoins métier en composants UI scalables

Équipe Lowcode

- Livré une **interface drag & drop, no-code** permettant aux équipes de composer des écrans visuellement sans écrire de code
- Créé la **bibliothèque low-code** qui alimente le Web Designer — un **moteur à base de descripteurs** où écrans, composants et comportements sont définis déclarativement via un **JSON** structuré
- Refondu un système déclaratif rigide en une **architecture plus scalable et maintenable**
- Maintenu et fait évoluer la plateforme low-code (corrections de bugs, refactoring, améliorations de performance, nouvelles fonctionnalités)
- Piloté ou accompagné la **migration de trois applications clientes existantes** vers la plateforme low-code, en tant que référent plateforme auprès de chaque équipe
- **Migration Siebel (équipe ESR)** — migré une **application legacy Oracle Siebel** vers la plateforme low-code, en reconstruisant ses écrans sous forme de descripteurs déclaratifs
- **Équipe Alto** — migré l'application de l'**ancien format de descripteurs vers la nouvelle architecture de descripteurs**, en définissant le chemin de migration et en accompagnant l'équipe sur les ruptures de compatibilité
- **Dashboard** — livré l'application comme **candidate du Web Designer**, validant l'outil de bout en bout, de la composition visuelle aux descripteurs JSON générés
- Également re-plateformé un **client lourd Java legacy vers le web** et migré des composants métier critiques (Plan d'Épargne Salariale) depuis des outils legacy

- Contribué à l'**architecture d'applications clientes** d'autres équipes, en apportant expertise technique et bonnes pratiques
- Développé le **Web Designer de zéro** avec un **agent IA de génération d'applications web en langage naturel** — l'utilisateur décrit un écran, l'outil produit vues, composants et bindings JSON
- Défini l'**UX du produit en étroite collaboration avec le Product Owner**, façonnant les parcours utilisateurs et les patterns d'interaction de bout en bout

Équipe AI Tooling — Vibe Toolbox (en cours)

- Développe la **Vibe Toolbox**, application desktop interne **Electron + Angular** provisionnant des **environnements de dev prêts pour l'IA** (skills d'agents, MCP), réduisant le time-to-productivity de jours à minutes
- Implémenté la **gestion de skills** — installation, versioning et partage de skills IA réutilisables (prompts, workflows, définitions d'agents) pour que les équipes adoptent des patterns validés au lieu de les recréer à chaque projet
- Construit les **mécanismes de marketplace** pour les skills et agents IA, supportant découverte, packaging et gestion du cycle de vie avec une gouvernance par rôles adaptée à l'usage entreprise
- Conçu la toolbox comme **point d'entrée unique** pour les développeurs pour configurer accès aux modèles, credentials, configuration MCP et bundles de skills, apportant une cohérence dans l'adoption de l'outillage IA à travers les équipes

Contributions transverses

- Conçu, créé et publié plusieurs **packages internes Java et JavaScript** (bibliothèques, utilitaires partagés, modules de framework) consommés par Maestro et les équipes applicatives en aval
- Mis en place et maintenu des **pipelines CI/CD automatisés** (GitLab CI, Jenkins) couvrant le build, le lint, les tests, la couverture et la livraison des artefacts framework et low-code
- Garanti une **forte couverture de tests** côté frontend et backend avec **JaCoCo** (Java), **Karma** et **Jest** (Angular/TypeScript), ainsi que **Cypress** et **Playwright** pour les scénarios end-to-end
- Utilisation quotidienne d'**outils d'IA agentique** (Claude, Codex) pour la génération de code, le refactoring, les revues de code, le débogage et l'accélération du travail exploratoire sur les fonctionnalités du framework et de la plateforme

Défis Techniques:

- Intégration micro-frontends et **rétrocompatibilité du framework** à travers les migrations majeures d'Angular
- Remplacement d'un système déclaratif rigide par un **moteur low-code à base de descripteurs** scalable, sans casser les applications clientes existantes
- Conception d'APIs de framework **assez génériques pour de nombreuses équipes** tout en restant simples à adopter rapidement

Environnement Technique: *Angular, TypeScript, Node.js, Electron, SQLite, PowerShell, Java 8/11/21, Spring Boot, Quarkus, Keycloak, Oracle DB, MongoDB, Karma, Jest, JaCoCo, Cypress, Playwright, Webpack, Jenkins, GitLab CI, Claude, Codex, Agile Scrum*

Zsoft Consulting — Consultant Full Stack (Java / React)

Octobre 2022 – Septembre 2023 (1 an)

Contexte du projet: Société de conseil IT. Consultant à temps plein sur des missions successives pour des clients grands comptes (Société Générale, Colas).

Client : Société Générale — Projet : KYC

Contexte du projet: Un ensemble de services backend dédiés aux processus **Know Your Customer (KYC)**, utilisés pour évaluer les niveaux de confiance et d'éligibilité des clients, et pour supporter les workflows de détection de fraude.

Responsabilités:

- Piloté une **mission full-backend d'extraction d'un microservice autonome et réutilisable** depuis un système **KYC legacy**
- Découplé le service d'une **base de données détenue par une autre équipe fournisseur**, en définissant des frontières et une propriété de service claires
- Conçu et implémenté des **services backend** avec Java et Spring Boot
- Développé et exposé des APIs REST consommées par d'autres systèmes internes
- Implémenté des mécanismes de sécurité utilisant Spring Security et OAuth2
- Assuré le **support N2**, en investiguant et résolvant les incidents de production
- Assuré la qualité avec des tests unitaires et d'intégration ainsi que la **couverture de code (JaCoCo)**, pour la fiabilité et la conformité

Défis Techniques:

- Extraire un service autonome depuis un **système KYC legacy fortement couplé** dont la base de données appartenait à une autre équipe fournisseur

Environnement Technique: *Java 17, Spring Boot, Spring Security, OAuth2, PostgreSQL, Quartz, Flyway, JaCoCo, Docker, Kubernetes, RabbitMQ, Jenkins*

Client : Colas — Projet : CRM

Contexte du projet: **CLASS** — un **CRM global / Gestionnaire d'Opportunités** construit de zéro et déployé dans 55 pays pour gérer les activités commerciales, opportunités et données clients du groupe Colas, en remplacement d'une application legacy **Power Apps / Teams** par une stack moderne React + Java Spring Boot.

Responsabilités:

- Construit l'application **de zéro**, des maquettes jusqu'au déploiement en production
- Piloté la **migration depuis Power Apps / Teams** vers une application React + Spring Boot maintenable, en reconstruisant les workflows de suivi d'opportunités (filtrage par type, région et autres critères)
- Implémenté l'interface à partir des maquettes des designers avec **React et le design system Ant Design**, avec des composants cohérents et réutilisables
- Implémenté l'authentification sécurisée avec **Azure AD (IAM / SSO)**, intégrée via Spring Security
- Développé les services backend et APIs REST utilisant Java et Spring Boot
- Développé des processus batch pour la **migration de données** utilisant Spring Batch
- Déployé l'application sur **Azure** (App Services, Front Door, CDN, Blob Storage, PostgreSQL) avec des **pipelines CI/CD**

Défis Techniques:

- Remplacer une application legacy **Power Apps / Teams** sans perturber les workflows commerciaux dans **55 pays**

Environnement Technique: *React, TypeScript, Ant Design, Java 11, Spring Boot, Spring Security, Spring Batch, Azure (AD, App Services, CDN, Front Door, Blob Storage, PostgreSQL), GitLab CI/CD*

Client : gloody.co — **Projet :** SaaS RH & Gestion d'Entreprise

Contexte du projet: gloody.co — un SaaS multi-tenant RH & gestion d'entreprise centralisant les workflows quotidiens des sociétés clientes : absences et congés, facturation, notes de frais et réservation de salles, servis depuis une base de code unique.

Responsabilités:

- Livré des **fonctionnalités full-stack** sur l'ensemble de la suite — absences et congés, facturation, notes de frais et réservation de salles — avec des front-ends **React / Ant Design** sur des APIs REST **Java / Spring Boot**
- Piloté l'**architecture front-end**, en particulier un **calendrier haute performance** affichant des données de planning denses (absences, congés, réservations de salles) avec une navigation fluide
- Structuré le front-end en **composants Ant Design réutilisables** et patterns d'état partagés, garantissant la cohérence entre les modules de la suite
- Conçu et exposé des **APIs REST avec Java et Spring Boot** sur un modèle de données multi-tenant, en isolant les données de chaque organisation cliente
- Travaillé sur un déploiement hébergé sur **AWS**, en contribuant à la chaîne de livraison

Défis Techniques:

- Maintenir un **calendrier fluide** tout en affichant de gros volumes de données de planning multi-sources sur un modèle multi-tenant

Environnement Technique: *React, TypeScript, Ant Design, SCSS, Java, Spring Boot, REST APIs, AWS*

Scor — Développeur Full Stack → Lead de Projet (Java / Angular)

Juin 2019 – Septembre 2022 (3 ans 4 mois)

Contexte du projet: RiskReveal — une application de traitement de données lourd pour l'évaluation des risques de catastrophes naturelles, ingérant et traitant de gros fichiers d'événements de sinistres historiques (ELT — Event Loss Tables), chacun rattaché à une région géographique et un type de péril (inondation, ouragan, etc.) ; les réassureurs utilisent ces données pour évaluer et tarifer le risque d'un portefeuille donné.

Responsabilités:

- Évolué de développeur full-stack à **lead technique / de projet**, validant les livraisons backend (Spring) et pilotant l'implémentation au sein de l'équipe
- Construit l'**UI de risque data-heavy** sur **AG Grid** — grilles virtualisées, groupement et agrégation sur de gros volumes de sinistres **PLT**
- Piloté le **modèle de calibration** — calculs statistiques appliqués aux événements de sinistres stockés dans des fichiers **PLT**
- Développé des **dashboards statistiques** sur les données de sinistres **PLT** — indicateurs de perte agrégés par péril et par région, alimentés par les calculs de calibration
- Conçu et développé des services backend et APIs REST avec Spring Boot

Réalisations & Impact:

- Optimisé et fait évoluer le **système d'orchestration batch** (Spring Batch) sur de gros volumes de données de risque, avec **WebSockets** pour la progression temps réel
- Mis en place l'**authentification SSO** avec Spring Security, en intégrant **Azure AD** et **Keycloak** comme fournisseurs d'identité
- Refactorisé le frontend vers un **design system unique et unifié (Ant Design)**, en supprimant les composants **PrimeNG** legacy
- Rendu les **gros volumes de données de risque exploitables dans le navigateur** — rendu de grilles virtualisées et visualisation de données sur les écrans de sinistres lourds

Environnement Technique: *Java 8, Spring Boot, Spring Batch, Spring Security, Angular 12, Ant Design, AG Grid, WebSockets, Azure AD, Keycloak, SQL Server, Azure DevOps, Jenkins*

Scor — Développeur Front-End Angular (Stage de Fin d'Études)

Mars 2019 – Juin 2019 (4 mois)

Contexte du projet: Une plateforme de gestion de tarification pour les équipes de tarification internes.

- Développé des fonctionnalités frontend utilisant Angular
- Conçu et stylisé des composants UI
- Intégré des APIs REST
- Appliqué les meilleures pratiques en HTML, CSS et TypeScript

Orrery — Multi-Agent Git Orchestrator IDE (Desktop)

Rôle: Auteur Unique — Développeur Full Stack · Type: Projet Personnel Open Source

Dépôt: <https://github.com/kouji-dev/orrery-releases>

Contexte du projet: Orrery est un IDE d'orchestration multi-agents Git futuriste, inspiré d'IntelliJ, conçu comme une application desktop native avec Tauri (Rust) et Angular 20. Chaque projet est un dépôt Git, et un projet peut lancer plusieurs agents de code IA (Claude Code, Codex, Cursor, Gemini) qui s'exécutent chacun dans leur propre **worktree Git, branche, terminal et conversation** — permettant à un développeur de piloter plusieurs agents en parallèle depuis un cockpit unique.

- Architecturé l'application comme un **shell desktop Tauri 2 (Rust)** encapsulant un frontend **Angular 20**, avec une structure propre **par domaine** (projets, agents, workspace, dashboard d'orchestration, panneau de droite, sidebar, paramètres, ...)
- Construit le **dashboard d'orchestration** — une vue d'ensemble avec statistiques en direct et plusieurs métaphores de visualisation pour superviser les agents à travers les projets
- Implémenté le **workspace par agent** avec panneaux Diff / Terminal / Chat : **terminaux intégrés accélérés WebGL**, éditeur diff/merge sur **CodeMirror 6**, et édition rich-text / markdown via **Lexical**
- Conçu le **flux de lancement d'agent** (projet, branche, outil, modèle, effort, prompt) et un **runtime de worktree par agent** afin que chaque agent soit isolé sur sa propre branche et son worktree
- Intégré le **suivi des coûts par agent** (ccusage) et un modèle de **hooks / permissions d'agent** pour gouverner ce que les agents sont autorisés à faire
- Construit un **design system piloté par design tokens** avec thématisation, densité, palettes de couleurs et contrôles de motion, ainsi que des logs en streaming temps réel et des timers en direct
- Écrit les éléments du **backend Rust** (auto-updater, génération d'icône d'application, CLI de hooks) et câblé le flux de **mise à jour automatique** via le plugin updater de Tauri
- Établi une **pipeline qualité & release** : tests unitaires **Vitest**, suites end-to-end **Playwright**, un harnais de **smoke-test de performance** avec assertions, scripts de bump de version sémantique, et workflows **GitHub Actions** (incluant un site vitrine déployé sur Render)

Environnement Technique: Tauri 2, Rust, Angular 20, TypeScript, CodeMirror 6, Lexical, RxJS, Vitest, Playwright, GitHub Actions

Vortex — Enterprise LLM Gateway

Rôle: Auteur Unique — Développeur Full Stack · Type: Projet Personnel Open Source

Dépôt: <https://github.com/kouji-dev/vortex>

Contexte du projet: Vortex est une **passerelle LLM compatible OpenAI/Anthropic, multi-tenant (SaaS) ou auto-hébergeable**, dotée d'une gouvernance d'entreprise — budgets par membre, contrôle d'accès basé sur les rôles, journaux d'audit et attribution des coûts — offrant aux organisations un point d'entrée unique et gouverné vers les grands modèles de langage.

- Architecturé un **monorepo pnpm** (feature-per-folder, hexagonal-lite) avec une API + passerelle **Hono**, deux consoles **Angular 22 (zoneless)** — tenant et super-admin plateforme — et des packages partagés `core`, `db`, `shared`, `sdk` et `cli`
- Construit une **passerelle compatible OpenAI/Anthropic** avec un **registre de providers**, routant tout l'usage à travers un endpoint unique et gouverné avec substitution de modèles transparente
- Implémenté une **gouvernance d'entreprise** — **budgets par membre**, **RBAC**, journalisation d'audit et **attribution des coûts** par utilisateur / application — appliquée à travers les tenants
- Conçu une **couche de données multi-tenant** sur **PostgreSQL + Drizzle** avec **row-level security (RLS)** pour l'isolation des tenants, et **Redis** pour le cache et les compteurs d'usage temps réel
- Intégré la **facturation Stripe** (mode SaaS) et des **DTOs validés par Zod** partagés entre les apps, et livré un **CLI Vortex** ainsi qu'un **SDK** léger avec helpers `acting-user` / `acting-app`
- Livré les consoles tenant et plateforme avec **@kouji-ui** (mon propre design system) et **Angular SSR**, couvertes par des tests end-to-end **Playwright** et une infrastructure locale basée sur Docker

Environnement Technique: *Angular 22 (zoneless), TypeScript, Hono, Node.js, PostgreSQL, Drizzle ORM (RLS), Redis, Zod, Stripe, @kouji-ui, Angular SSR, Playwright, Docker, pnpm, Render*

Kouji UI — Angular Design System & Component Library

Rôle: Auteur Unique & Mainteneur · Type: Bibliothèque Personnelle Open Source

Dépôt: <https://github.com/kouji-dev/kouji-ui>

Contexte du projet: Design system **Angular 21** open source fournissant **plus de 60 primitives UI accessibles et thématables** couvrant inputs, overlays, navigation, affichage de données, feedback et layout. Maintenu en solo pour démontrer une maîtrise complète du *frontend platform engineering* — discipline monorepo, frontières de packages publiables, accessibilité par défaut et gestion de releases rigoureuse.

- Architecturé un **monorepo pnpm + Turborepo** avec trois packages publiables — `components` , `core` , `themes` — ainsi qu'un site de documentation dédié, partageant outillage, règles de lint et fondation Tailwind v4
- Écrit **plus de 60 composants Angular 21 standalone** (calendar, combobox, command palette, date/time pickers, dialog, drawer, file upload, form, OTP input, popover, stepper, toast, tooltip, tree-select, ...) conçus pour le tree-shaking et la détection de changement par signals
- Construit une **couche de thématisation découplée** (`@kouji-ui/themes`) pilotée par design tokens, permettant le changement de thème, le mode sombre et les overrides par produit sans forker les composants
- Mis en place une **pipeline qualité stricte** : tests unitaires **Vitest**, suites end-to-end **Playwright**, audits d'**accessibilité** automatisés (basés axe) produisant des rapports JSON + HTML dans `reports/a11y` , plus ESLint, Prettier et hooks Husky pré-commit
- Établi le **release engineering** via **Changesets** pour le versioning sémantique et la génération de changelogs sur les trois packages, intégré aux workflows GitHub Actions
- Rédigé la documentation contributeur et des **directives de collaboration IA** (`CLAUDE.md` , `RULES.md`) pour que la bibliothèque puisse être étendue de manière cohérente par humains et agents IA

Environnement Technique: *Angular 21, TypeScript, Angular CLI, pnpm workspaces, Turborepo, Vitest, Playwright, axe-core, ESLint, Prettier, Husky, Changesets, Tailwind CSS, GitHub Actions*

Pages — Unified Project Management & Documentation Platform

Rôle: Développeur Full Stack · Type: Projet Personnel

Dépôt: <https://github.com/kouji-dev/pages>

Contexte du projet: Pages est une plateforme full-stack ambitieuse conçue pour **remplacer la stack traditionnelle Jira + Confluence** par une solution unique et cohésive. L'objectif est d'éliminer le changement de contexte en unifiant le **suivi de tickets, la gestion de sprints et la documentation technique** dans une plateforme moderne et conviviale pour les développeurs, tout en posant les bases pour des **workflows améliorés par l'IA**.

- Établi une **architecture monorepo scalable** utilisant les principes de **Domain-Driven Design (DDD)** entre frontend et backend
- Livré un **frontend hautement performant et réactif** avec **Angular 21** et **Angular Signals** avec des composants réutilisables et responsives
- Ingénieré une **API REST prête pour la production** avec **FastAPI (Python 3.12)**, intégrée avec **PostgreSQL** via **SQLAlchemy 2.0** et **Redis**
- Implémenté des **workflows de gestion de projet de bout en bout** incluant la planification de sprints, le suivi de tickets, les tableaux Kanban et la gestion de backlog avec **collaboration en temps réel** via WebSockets
- Activé la **rédaction de documentation riche** grâce à l'intégration de l'**éditeur Lexical** et assuré la qualité du code avec une couverture de tests complète (unitaire, intégration, fonctionnel) et des workflows CI/CD automatisés

Environnement Technique: *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

Rôle: Développeur Full Stack · Type: Projet Personnel

Dépôt: <https://github.com/kouji-dev/GamerGG>

Contexte du projet: GamerGG est un **marché de services gaming full-stack** conçu pour connecter les joueurs avec des boosters et coachs professionnels. La plateforme permet aux joueurs d'acheter des **services de boost de rang, coaching et packages gaming** tout en fournissant aux boosters des outils pour gérer les commandes, suivre les performances et interagir avec les clients via une communication **Discord** intégrée.

- Architecturé une structure **monorepo Turborepo** permettant des composants et utilitaires partagés à travers plusieurs applications Next.js avec un cache de build optimisé
- Construit une **application web moderne et responsive** avec **Next.js 13**, **TypeScript** et **Tailwind CSS**, implémentant le rendu côté serveur et les routes API
- Conçu et implémenté l'**authentification et autorisation** utilisant **NextAuth.js** avec des providers OAuth et une gestion de session sécurisée
- Développé une **couche de base de données robuste** utilisant **Prisma ORM** avec **PostgreSQL**, implémentant les migrations, seeding et requêtes type-safe
- Créé un **système de gestion de commandes complet** avec suivi de statut en temps réel, monitoring de boost actif et historique de commandes pour les clients et boosters
- Intégré la **connectivité serveur Discord** pour une communication fluide entre clients et boosters au sein de la plateforme
- Implémenté un **tableau de bord d'analytics utilisateur** présentant un classement des meilleurs boosters, suivi des dépenses utilisateur et métriques de performance
- Établi une **bibliothèque de composants UI réutilisables** partagée à travers le monorepo avec un système de design cohérent et des types TypeScript

Environnement Technique: *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

Rôle: Développeur Full Stack · Type: Projet Personnel

Contexte du projet: Acta est une **plateforme complète de gestion de projets** conçue spécifiquement pour les cabinets comptables afin de rationaliser le **suivi de production inter-fonctionnel** à travers plusieurs équipes. La plateforme permet la planification de tâches en temps réel, le suivi du temps et le ticketing pour des flux de travail comptables complexes, gérant des structures hiérarchiques de **portefeuilles, dossiers, missions et tâches** avec des workflows de validation progressive et une visualisation multi-dimensionnelle de la progression.

- Architecturé un **monorepo full-stack** avec frontend **Angular 21** et backend **Express.js**, implémentant un **contrôle d'accès basé sur les rôles (RBAC)** avec des permissions granulaires
- Conçu un **modèle de données hiérarchique complexe** utilisant **Prisma ORM** avec **PostgreSQL**, supportant les relations portefeuilles → dossiers → missions → tâches avec workflows de validation progressive
- Construit des **fonctionnalités de visualisation multi-dimensionnelles** incluant des vues calendrier, suivi de progression par collaborateur ou dossier, et tableaux de bord de suivi du temps avec agrégations quotidiennes/hebdomadaires/mensuelles/annuelles
- Développé une **gestion de tâches en temps réel** avec workflows de statut, système de demandes de modification de dates d'échéance avec workflows d'approbation, et attribution automatique de tâches basée sur les affectations de dossiers par rôle
- Implémenté un **système complet de suivi du temps** avec capacités de reporting détaillées, permettant l'analyse par collaborateur, dossier, tâche ou mission avec filtrage et agrégation flexibles
- Créé des **composants UI réactifs et riches en données** utilisant **TanStack Table** pour des grilles de données complexes, **Luxon** pour la gestion des dates/heures, et **Tailwind CSS** pour un design moderne et accessible
- Établi une **authentification et autorisation robustes** utilisant **JWT** et **Passport.js** avec gardes de route basés sur les rôles et accès aux fonctionnalités basé sur les permissions

Environnement Technique: *Angular 21, TypeScript, Express.js, Prisma, PostgreSQL, JWT, Passport.js, TanStack Table, Luxon, Tailwind CSS, Docker, npm*

Chartify — Figma Plugin

Rôle: Développeur de Plugin · **Type:** Plugin Communautaire Figma

Lien: <https://www.figma.com/community/plugin/1361309122482309141/chartify>

Contexte du projet: Chartify est un **plugin communautaire Figma** conçu pour rationaliser la création de visualisations de données et graphiques directement dans l'environnement de design Figma. Le plugin permet aux designers de générer rapidement différents types de graphiques, personnaliser les visualisations de données et intégrer des graphiques de manière fluide dans leurs workflows de design sans quitter Figma.

- Développé un **plugin Figma** utilisant TypeScript et l'API Plugin Figma pour étendre les capacités de l'outil de design
- Implémenté une **fonctionnalité de génération de graphiques** supportant plusieurs types de graphiques et options de visualisation de données personnalisables
- Créé une **interface utilisateur intuitive** dans l'environnement plugin de Figma pour une configuration et personnalisation faciles des graphiques
- Assuré une **intégration fluide** avec le système de design de Figma, permettant aux graphiques d'être facilement incorporés dans les fichiers de design
- Publié et maintenu le plugin dans la **Communauté Figma**, le rendant accessible aux designers du monde entier

Environnement Technique: *Angular, TypeScript, Taiga UI, Figma Plugin API, Figma Community*

FORMATION

Diplôme d'Ingénieur d'État en Informatique

École Nationale des Sciences Appliquées de Tanger — 2019

LANGUES

- Arabe — Natif
- Français — Courant
- Anglais — Professionnel