

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>

À PROPOS

Ingénieur Logiciel Senior avec une solide expertise 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.

COMPÉTENCES TECHNIQUES

Catégorie	Technologies
Langages	JavaScript, TypeScript, Java 8/11/17/21, Python (IA/automatisation – 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
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)
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

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

EXPÉRIENCE PROFESSIONNELLE

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

Depuis septembre 2023

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.

J'ai travaillé au sein de **deux équipes** sur ce projet :

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.)
- Participé à la **conception de l'architecture du framework** afin de le rendre compatible avec des 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 :

- Conçu et développé un **Web Designer** permettant de construire des écrans visuellement, sans code, via une interface drag & drop
- Architecturé un **framework UI low-code et piloté par configuration** permettant de définir les écrans et comportements via JSON structuré
- Aidé à remplacer un système déclaratif rigide par une **architecture plus scalable et maintenable**
- Maintenu et fait évoluer la plateforme low-code (corrections de bugs, refactoring, améliorations de performance, nouvelles fonctionnalités)
- Piloté la **migration de plusieurs applications clientes majeures** vers la nouvelle plateforme low-code, incluant 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
- Intégré des **fonctionnalités IA dans le Web Designer**, permettant la **génération d'applications web dynamiques en langage naturel** — l'utilisateur décrit des écrans ou des flux en langage simple et le designer produit les vues, composants et bindings définis en JSON, prêts à être affinés

Équipe AI Tooling — Vibe Toolbox (en cours) :

- Développe actuellement la **Vibe Toolbox**, une plateforme interne qui **provisionne des environnements de développement prêts pour l'IA** pour les ingénieurs de l'organisation, réduisant le time-to-productivity sur les workflows assistés par IA de plusieurs jours à quelques 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

Environnement Technique :

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 — Développeur Java Senior

Avril 2023 – Septembre 2023

Client : Société Générale

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 & Contributions :

- Participé à la **conception et l'implémentation de services backend** utilisant 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
- Rédigé des tests unitaires et d'intégration pour assurer la fiabilité et la conformité
- Participé aux revues de code et cérémonies Scrum
- Travaillé dans un environnement conteneurisé et cloud

Environnement Technique :

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

Colas CRM — Développeur Full Stack Java / Angular Senior

Janvier 2023 – Avril 2023

Client : Colas

Contexte du projet :

Un **système CRM global** déployé dans 55 pays pour gérer les activités commerciales, opportunités et données clients du groupe Colas.

Responsabilités & Contributions :

- Développé des services backend utilisant Java et Spring Boot
- Construit des fonctionnalités frontend utilisant Angular et Ant Design
- Implémenté une authentification sécurisée utilisant Azure Active Directory (SSO)
- Conçu et consommé des APIs REST
- Développé des processus batch pour la migration de données utilisant Spring Batch
- Sécurisé les APIs avec Spring Security
- Déployé l'application sur **Azure** sur plusieurs environnements — App Services, Front Door, CDN, Blob Storage et Azure Database for PostgreSQL — avec des pipelines CI/CD
- Participé activement aux événements Scrum et revues de code

Environnement Technique :

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 — Développeur Full Stack Java / React

Octobre 2022 – Janvier 2023

Client : Zsoft Consulting

Contexte du projet :

Une **plateforme de gestion d'entreprise multi-tenant** permettant aux consultants de gérer les absences, demandes de congés, notes de frais, certifications et formations.

Responsabilités & Contributions :

- Maintenu et étendu les services backend existants
- Développé de nouvelles APIs REST utilisant Spring Boot
- Construit des fonctionnalités frontend utilisant React et Ant Design
- Amélioré les performances des modules de reporting
- Collaboré avec l'équipe dans les cérémonies Agile Scrum

Environnement Technique :

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

RiskReveal — Développeur Full Stack Java / Angular

Juin 2019 – Septembre 2022

Client : Scor

Contexte du projet :

Une application dédiée à l'**évaluation des risques de catastrophes naturelles**, utilisée pour analyser et visualiser les données de risque pour la prise de décision liée à l'assurance.

Responsabilités & Contributions :

- Conçu et développé des services backend avec Spring Boot

- Développé des fonctionnalités frontend utilisant Angular
- Créé et maintenu des APIs REST
- Implémenté des fonctionnalités en temps réel utilisant WebSockets
- Participé aux revues de code et sessions de pair programming
- Contribué à la personnalisation UI utilisant Ant Design et bibliothèques de visualisation de données

Environnement Technique :

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

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

Mars 2019 – Juin 2019

Client : Scor

Contexte du projet :

Une plateforme de gestion de tarification pour les équipes de tarification internes.

Responsabilités & Contributions :

- 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
-

PROJETS PERSONNELS

Orrery — IDE d'Orchestration Multi-Agents Git (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. Le nom évoque un planétaire (*orrery*) : des agents en orbite autour d'un projet, comme les corps d'un modèle mécanique du système solaire.

Responsabilités & Contributions :

- 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 via **xterm.js (rendu 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, Tailwind CSS 4, xterm.js (WebGL), CodeMirror 6, Lexical, RxJS, plugins Tauri (updater, dialog, notification, process, opener), ccusage, Vitest, Playwright, pnpm, GitHub Actions, Render

AI Portal — Plateforme d'Assistant IA d'Entreprise Auto-Hébergeable

Rôle : Auteur Unique — Développeur Full Stack

Type : Projet Personnel Open Source

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

Contexte du projet :

Plateforme d'assistant IA auto-hébergeable — application de type *ChatGPT interne* combinant chat en streaming, retrieval-augmented generation (RAG) sur des bases de connaissances privées, gestion d'un catalogue de modèles et mémoire par utilisateur. Projet mené de bout en bout en solo pour valider des patterns de production pour l'IA d'entreprise : abstraction des providers, ingestion documentaire, authentification hybride et déploiement en une commande.

Responsabilités & Contributions :

- Conçu et implémenté le backend en **FastAPI**, structuré en modules à frontières claires (assistants, conversations, bases de connaissances, catalogue de modèles, mémoires, clés d'API) pour isoler les préoccupations IA des fonctionnalités produit
- Construit le **chat en streaming** via **LangChain** avec **abstraction des providers** (compatibles OpenAI + Anthropic), permettant la substitution de modèles et le routage à travers un proxy compatible OpenAI sans aucune modification du code applicatif
- Implémenté les **bases de connaissances RAG** sur **PostgreSQL + pgvector** — upload, découpage, embedding, récupération et injection de contexte côté conversation, avec ingestion déchargée du chemin de la requête
- Construit le client web sur **TanStack Start (React + Vite)** avec routing par fichiers et TanStack Query, livrant les écrans chat, bases de connaissances et mémoires alimentés par des réponses streamées
- Implémenté un **mode d'authentification dual** — token bearer local pour itération rapide, et validation JWT **Microsoft Entra ID** avec app roles pour la production — ainsi que des clés d'API émises par le portail (`aiportal_...`) pour l'accès programmatique

- Industrialisé le déploiement : **Docker Compose** pour le dev local, un profil Docker full-stack, et un **blueprint Render + Supabase en une commande** avec migrations Alembic appliquées automatiquement à chaque déploiement
- Conçu un **flux de configuration self-hosted** (assistant au premier démarrage, provisionnement mono-organisation) avec verrouillage 503 de l'API tant que la configuration n'est pas terminée, ciblant les organisations souhaitant exécuter la plateforme en privé

Environnement Technique :

Python 3.12, FastAPI, LangChain, APIs OpenAI & Anthropic, 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 — Design System Angular & Bibliothèque de Composants

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.

Responsabilités & Contributions :

- 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 / design tokens, GitHub Actions

Pages — Plateforme Unifiée de Gestion de Projet & Documentation

Rôle : Développeur Full Stack

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**.

Responsabilités & Contributions :

- Établi une **architecture monorepo scalable** utilisant les principes de **Domain-Driven Design (DDD)**, permettant des frontières de domaine claires et une structure de codebase maintenable entre frontend et backend
- Livré un **frontend hautement performant et réactif** avec **Angular 21** et **Angular Signals**, atteignant une détection de changements efficace et une expérience utilisateur fluide avec des composants réutilisables et responsives suivant la méthodologie BOM
- Ingénieré une **API REST prête pour la production** avec **FastAPI (Python 3.12)** exploitant `async/await` pour des performances I/O optimales, intégrée avec **PostgreSQL** via **SQLAlchemy 2.0** et **Redis** pour des temps de réponse améliorés
- 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 des capacités de **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 — Plateforme de Services Gaming

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.

Responsabilités & Contributions :

- 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 pour des performances optimales
- 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 de base de données 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 — Plateforme de Gestion de Projets pour Cabinets Comptables

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.

Responsabilités & Contributions :

- 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 pour collaborateurs, superviseurs, managers et experts-comptables
- Conçu et implémenté un **modèle de données hiérarchique complexe** utilisant **Prisma ORM** avec **PostgreSQL**, supportant les relations portefeuilles → dossiers → missions → tâches avec dépendances interconnectées et workflows de validation progressive
- Construit des **fonctionnalités de visualisation multi-dimensionnelles** incluant des vues calendrier pour tâches et missions, 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, Docker Compose, pnpm, Git

Chartify — Plugin Figma

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.

Responsabilités & Contributions :

- 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
- Français
- Anglais