



HASSAN AKKARI

SOFTWARE ENGINEER · FULL-STACK
.NET / ASP.NET CORE

CONTACT

+39 351 787 2307
hassan.akkari01@gmail.com
itshassan.it
in/hassan-akkari
github.com/hassan-akkari
Citoyen italien (UE)

STACK

Backend & Données
ASP.NET Core 10 · C# · MediatR · Dapper · SQL Server

Frontend
React 19 · TypeScript · Redux Toolkit · RTK Query · Next.js · Zod

Dev augmenté par l'IA
Claude Code · subagents personnalisés · CLAUDE.md à niveaux · revue adversariale agentique

Tests & outils
Vitest · Playwright (e2e + a11y) · Azure DevOps · CI/CD (YAML) · Figma

LANGUES

Anglais — C1 / courant
Italien — langue maternelle
Français — A2 (en cours d'apprentissage)
Arabe — conversationnel

FORMATION

Level 3 Diploma · Computing
HCUC, Royaume-Uni · 2021

PROFIL

Ingénieur logiciel full-stack avec ~2,5 ans de systèmes en production sur ASP.NET Core et React pour un éditeur de logiciels pour l'hôtellerie. J'ai conçu un flux de paiement Nexi *exactly-once* sur un aval non idempotent (machine à états MediatR + verrou d'idempotence), je pilote la migration d'un SaaS multi-tenant depuis ASP.NET MVC/Razor legacy vers React 19 derrière un BFF, et je garantis l'isolation des données multi-tenant sur toute l'API. J'ai aussi construit le workflow de revue augmentée par l'IA de l'équipe sur Claude Code. Stack principal : ASP.NET Core 10, C#, MediatR, SQL Server, React 19, TypeScript.

EXPÉRIENCE

Software Engineer

Sibylla S.r.l. — Rome, Italie

SEPT. 2023 — PRÉSENT

Network — e-commerce hôtelier

ASP.NET Core 10 · C# · MediatR · Dapper · SQL Server · React 19

MARS 2025 — PRÉSENT

- Confirmation de paiement des devis rendue *exactly-once* sur un système de réservation aval non idempotent : un verrou de paiement local à clé sur un code de transaction unique et une machine à états gérée via MediatR (pending → nexi_paid → confirmed | failed) qui ignore les callbacks de gateway retransmis — pas de doubles réservations ni de confirmations perdues, avec un garde-fou contre les réponses silencieuses « HTTP 200 mais échec », de sorte qu'aucun paiement n'est confirmé sans réservation réelle.
- Contributeur principal et premier auteur en volume de commits de la plateforme e-commerce hôtelière, reconstruite en interne après qu'un développement externe précédent a manqué les exigences. Seul auteur du flux de paiement des devis en ligne, fondateur du kiosque React de self check-in et auteur dominant de la SPA marketplace — réservation, panier, checkout, wallet et compte — de zéro à la production.
- Frontend React 19 architecturé en trois SPA indépendantes (marketplace, kiosque, landing devis) sur Redux Toolkit + RTK Query : une couche d'auth 401-refresh *single-flight* qui fusionne les refresh de token concurrents et rejoue le XSRF de façon transparente, une architecture de paiement ports-and-adapters (Nexi / wallet / remises derrière une interface provider unique) et des formulaires react-hook-form + Zod sur des payloads legacy incohérents. Tests avec Vitest et Playwright (e2e + a11y) ; coordination du release train testing → staging → production sur Azure DevOps.

Portal — SaaS de gestion multi-tenant

ASP.NET Core 10 · BFF · multi-tenant · React 19

SEPT. 2023 — PRÉSENT

- Pilotage de la migration *page par page* d'un SaaS de gestion hôtelière multi-tenant depuis ASP.NET MVC / Razor / jQuery legacy vers React 19, monté dans l'application existante derrière un backend-for-frontend ; un manifeste de routes décide pour chaque page entre navigation SPA et rechargement complet, les pages migrent donc de façon incrémentale, sans réécriture big-bang. Plusieurs fonctionnalités livrées à l'origine en jQuery reconstruites en React, et pipeline de déploiement durcie contre les bundles frontend obsolètes.
- Isolation des données multi-tenant auditée et durcie (claim du tenant filtré au niveau de la ligne, et non via une jointure parente) ; conduite d'un audit de sécurité et de tenancy ayant révélé des accès cross-tenant (IDOR) et un token CSRF émis mais jamais validé côté serveur, entre autres constats.

INGÉNIERIE AUGMENTÉE PAR L'IA

- Conception du framework qui rend fiable la migration assistée par IA de l'équipe : fichiers d'instructions par couche (chaque agent ne charge que les conventions de sa couche), subagents spécialisés à outils restreints pour isoler le blast-radius, une skill d'orchestration qui classe la tâche et injecte les bons invariants, et un processus *plan-first* où la première étape est toujours une analyse sans code, validée par un humain avant toute modification — la migration d'une page devient une pipeline fixe et répétable.
- Construction de subagents de revue read-only calibrés sur le projet (revue d'architecture adversariale, audit SOLID, contract-drift TS↔C#, traçage du flux de paiement) sur la base de code ASP.NET Core / React, appuyés sur une base de connaissances CLAUDE.md à niveaux codifiant le glossaire métier, la machine à états des paiements et les pièges non évidents — 3 bugs latents du flux de paiement détectés avant la mise en production.

Frontend Developer · Stage

BetterTogether S.r.l. — Rome, Italie

OCT. 2022 — MAI 2023

- Front-end pour la facturation automatisée (B2C / B2B2C) en HTML, CSS, JS et C# / ASP.NET MVC ; intégration des API Aruba pour la facturation électronique ; standardisation de l'UI et refactoring du code legacy.

PROJETS

- **laboratoire** — monorepo personnel React 19 / TypeScript (pnpm + Turbo). Cœur server-side d'un flux de réservation Next.js App Router : traitement de commandes idempotent et un checkout qui revalide les prix côté serveur plutôt que de faire confiance au client, plus un webhook Cal.com avec vérification HMAC timing-safe — bug de signature *bare-hex* cassant la production identifié et corrigé à partir du code source de l'émetteur. Autour : auth admin iron-session, allowlists CORS / origin, schéma Postgres Neon / Drizzle et une CI gate cache-aware sans secrets.