



HASSAN AKKARI

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

CONTATTI

+39 351 787 2307
hassan.akkari01@gmail.com
itshassan.it
in/hassan-akkari
github.com/hassan-akkari

STACK

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

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

Sviluppo AI-augmented
Claude Code · subagent personalizzati · CLAUDE.md a livelli · review avversariale con agenti

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

LINGUE

Inglese — C1 / fluente
Italiano — madrelingua
Arabo — conversazionale

FORMAZIONE

Level 3 Diploma · Computing
HCUC, Regno Unito · 2021

PROFILO

Software engineer full-stack con quasi 3 anni di esperienza su sistemi in produzione ASP.NET Core e React per un'azienda software nell'hospitality. Ho progettato un flusso di pagamento Nexi *exactly-once* sopra un downstream non idempotente (state machine MediatR + lock di idempotenza), guido la migrazione di un SaaS multi-tenant da ASP.NET MVC/Razor legacy a React 19 dietro un BFF e garantisco l'isolamento dei dati tra tenant lungo tutta l'API. Ho inoltre costruito il workflow di review AI-augmented del team su Claude Code. Stack principale: ASP.NET Core 10, C#, MediatR, SQL Server, React 19, TypeScript.

ESPERIENZA

Software Engineer

Sibylla S.r.l. — Roma, Italia

SET 2023 — PRESENTE

Network — e-commerce per l'hospitality

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

MAR 2025 — PRESENTE

- Implementata la conferma *exactly-once* dei pagamenti dei preventivi su un sistema di prenotazione downstream non idempotente: un payment lock locale con chiave sul codice transazione univoco e una state machine MediatR (pending → nexi_paid → confirmed | failed) trattano come no-op i callback ritrasmessi — evitando doppie prenotazioni e conferme perse. Un guard intercetta anche le risposte silenziose "HTTP 200 ma fallite", così nessun pagamento viene confermato senza una prenotazione reale.
- Contributore principale e autore #1 per volume di commit della piattaforma e-commerce, ricostruita in-house dopo che il precedente sviluppo esterno non aveva soddisfatto i requisiti. Unico autore del flusso di pagamento dei preventivi, ideatore del chiosco React di self check-in e principale autore della SPA marketplace — booking, carrello, checkout, wallet e account — da zero alla produzione.
- Progettato il frontend React 19 come tre SPA indipendenti (marketplace, chiosco, landing preventivi) su Redux Toolkit + RTK Query: layer di autenticazione con refresh 401 *single-flight*, architettura pagamenti ports-and-adapters (Nexi / wallet / sconti dietro un'unica interfaccia provider) e form react-hook-form + Zod sopra payload legacy incoerenti. Test con Vitest e Playwright (e2e + a11y); coordinato il release train testing → staging → produzione su Azure DevOps.

Portal — SaaS gestionale multi-tenant

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

SET 2023 — PRESENTE

- Alla guida della migrazione page-by-page di un SaaS gestionale per l'hospitality da ASP.NET MVC / Razor / jQuery legacy a React 19, montato nell'app esistente dietro un backend-for-frontend; un route manifest decide per ogni pagina tra navigazione SPA e full reload, permettendo una migrazione incrementale senza riscrittura big-bang. Ricostruite in React diverse funzionalità originariamente sviluppate in jQuery e irrobustita la pipeline di deploy contro bundle frontend non aggiornati.
- Verificato e irrobustito l'isolamento dei dati multi-tenant (claim del tenant filtrato direttamente sulla riga, non tramite join su un'entità padre); condotto un audit di sicurezza e tenancy che ha fatto emergere accessi cross-tenant (IDOR) e un token CSRF emesso ma mai validato lato server, oltre ad altri rilievi.

INGEGNERIA ASSISTITA DALL'AI

- Progettato il framework che rende affidabile la migrazione AI-assisted del team: file di istruzioni per layer, subagent specialistici con tool ristretti per isolare il blast radius, una skill di orchestrazione che classifica il task e inietta gli invarianti corretti, e un processo plan-first in cui il primo passo è sempre analisi senza codice e ogni modifica richiede approvazione umana — trasformando la migrazione di una pagina in una pipeline fissa e ripetibile.
- Costruiti subagent di review read-only calibrati sul progetto (review architetture avversariale, audit SOLID, contract drift TS↔C#, tracing del flusso pagamenti), sostenuti da una knowledge base CLAUDE.md a livelli che codifica glossario di dominio, payment state machine e gotcha non ovvi — facendo emergere tre bug latenti del flusso pagamenti prima del rilascio.

Frontend Developer · Stage

BetterTogether S.r.l. — Roma, Italia

OTT 2022 — MAG 2023

- Front-end per fatturazione automatica (B2C / B2B2C) con HTML, CSS, JS e C# / ASP.NET MVC; integrazione delle API Aruba per la fatturazione elettronica; standardizzazione della UI e refactoring del codice legacy.

PROGETTI

- laboratoire** — monorepo personale React 19 / TypeScript (pnpm + Turbo). Costruito il core server-side di un flusso di prenotazione Next.js App Router: order processing idempotente e checkout con ri-validazione dei prezzi lato server, più un webhook Cal.com con verifica HMAC timing-safe — individuato e corretto un bug di firma *bare-hex* che rompeva la produzione confrontandolo con il sorgente del mittente. Attorno: auth admin iron-session, allowlist CORS/origin, schema Postgres Neon/Drizzle e CI gate cache-aware senza segreti.