



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

STACK

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

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

AI-augmented dev
Claude Code · custom subagents · tiered CLAUDE.md · agentic adversarial review

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

LANGUAGES

English — C1 / fluent
Italian — native
Arabic — conversational

EDUCATION

Level 3 Diploma · Computing
HCUC, United Kingdom · 2021

PROFILE

Full-stack software engineer with ~2.5 years building production systems on ASP.NET Core and React for a hospitality software company. I designed an exactly-once Nexi payment flow over a non-idempotent downstream (MediatR state machine + idempotency lock), lead the migration of a multi-tenant SaaS from legacy ASP.NET MVC/Razor to React 19 behind a BFF, and enforce multi-tenant data isolation across the API. I also built the team's AI-augmented review workflow on Claude Code. Core stack: ASP.NET Core 10, C#, MediatR, SQL Server, React 19, TypeScript.

EXPERIENCE

Software Engineer

Sibylla S.r.l. — Rome, Italy

SEP 2023 — PRESENT

Network — hospitality e-commerce

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

MAR 2025 — PRESENT

- Made quote-payment confirmation *exactly-once* over a non-idempotent downstream booking system: a local payment-lock keyed by a unique transaction code and a MediatR-handled state machine (pending → nexi_paid → confirmed | failed) that no-ops on retransmitted gateway callbacks — preventing double bookings and lost confirmations, with a guard against silent "HTTP 200 but failed" responses so no payment is confirmed without a real booking.
- Principal contributor and top author by commit volume on the hospitality e-commerce platform, rebuilt in-house after prior external development missed requirements. Sole author of the online quote-payment flow, founder of the React self-service check-in kiosk, and dominant author of the marketplace SPA — booking, cart, checkout, wallet and account — from zero to production.
- Architected the React 19 frontend as three independent SPAs (marketplace, kiosk, quote landing) on Redux Toolkit + RTK Query: a *single-flight* 401-refresh auth layer that collapses concurrent token refreshes and replays XSRF transparently, a ports-and-adapters payment architecture (Nexi / wallet / discount behind one provider interface), and react-hook-form + Zod over inconsistent legacy payloads. Tested with Vitest and Playwright (e2e + a11y); coordinated the testing → staging → production release train on Azure DevOps.

Portal — multi-tenant hospitality SaaS

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

SEP 2023 — PRESENT

- Leading the *page-by-page* migration of a multi-tenant hospitality-management SaaS from legacy ASP.NET MVC / Razor / jQuery to React 19, mounted into the existing app behind a backend-for-frontend; a route manifest decides per page whether to SPA-navigate or full-reload, so pages move incrementally with no big-bang rewrite. Re-built in React several features I had originally shipped in jQuery, and hardened the deploy pipeline against stale frontend bundles.
- Audited and hardened multi-tenant data isolation (tenant claim filtered on the row, not via a parent join), and led a security and tenancy audit that surfaced cross-tenant access (IDOR) and an issued-but-never-validated CSRF token, among other findings.

AI-AUGMENTED ENGINEERING

- Designed the framework that makes the team's AI-assisted migration reliable: per-layer instruction files so each agent loads only the conventions for the layer it touches, narrow tool-scoped specialist subagents for blast-radius isolation, an orchestration skill that classifies a task and injects the right invariants, and a *plan-first* process where the first step is always analysis-with-no-code and a human approves before anything mutates — turning a page migration into a fixed, repeatable pipeline.
- Built read-only, project-tuned review subagents (adversarial architecture review, SOLID audit, TS↔C# contract-drift, payment-flow tracing) over the ASP.NET Core / React codebase, backed by a tiered CLAUDE.md knowledge base codifying the domain glossary, the payment state machine and non-obvious gotchas — surfacing three latent payment-flow bugs before release.

Frontend Developer · Intern

BetterTogether S.r.l. — Rome, Italy

OCT 2022 — MAY 2023

- Front end for automated invoicing (B2C / B2B2C) with HTML, CSS, JS and C# / ASP.NET MVC; integrated Aruba APIs for electronic invoicing; standardised UI and refactored legacy code.

PROJECTS

- **laboratoire** — solo React 19 / TypeScript monorepo (pnpm + Turbo). Built the server-side core of a Next.js App Router booking flow: idempotent order processing and a checkout that re-validates prices server-side rather than trusting the client, plus a Cal.com webhook with timing-safe HMAC verification — root-causing a production-breaking *bare-hex* signature bug against the sender's source. Surrounded by iron-session admin auth, CORS / origin allowlists, a Neon / Drizzle Postgres schema, and a secretless, cache-aware CI gate.