

// case study · 01/06

ToffeeOS: restaurant software two cafés run on every day.

How I designed, built, tested and shipped a multi-tenant POS while finishing my CS degree.

// the problem · 02/06

The internet drops. The till can't.

Cafés in Pakistan take orders through a rush hour where the connection can go at any moment, and they still have to charge the right tax on every bill.

A till that stops when the Wi-Fi does means lost orders and a queue at the counter.

So the brief was simple: keep selling, always, and get the numbers right.

// what it does · 03/06

One system from order to **report.**

POS

Takes orders and cash payments even while offline.

Kitchen display

Orders reach the kitchen screen as they're placed.

Inventory

Stock tracked against what's actually sold.

Delivery riders

Delivery orders assigned and tracked.

QR table ordering

Guests order from their table.

Tax & reports

Punjab PRA split built in: 16% cash, 8% card.

// how it's built · 04/06

Multi-tenant from day one.

Every café is its own tenant on one codebase, so a new client is a setup, not a fork.

frontend

Next.js App Router, TypeScript

data

Prisma on Supabase Postgres

live updates

Supabase Realtime

hosting

Vercel

resilience

Offline order & cash capture

// how I ship safely · 05/06

Live cafés don't get a beta.

333

Playwright end-to-end tests

228

Vitest unit tests

The end-to-end suite runs before every release. That's what lets one engineer push changes to a system people are ringing up sales on right now.

// in production · 06/06

200+

orders a day at SPILL and Loom,
two cafés in Bahawalpur.

ToffeeOS is one of the projects I've shipped through
Krimsons, alongside software for 20+ paying clients since
2024.

I'm open to remote full-stack and Python roles.

linkedin.com/in/amarabdurrehman ·
amarabdulrehman123@gmail.com