

Aidan Davis

HARMONY

FULL-STACK ENGINEER · WEB, MOBILE, AND OFFLINE SYSTEMS

aidanmichaeldavis@gmail.com · capybaraharmony.com · github.com/harmonyvt · twitch.tv/capybaraharmony

I build software people use at work, end to end — field apps, office tools, sync, permissions, and the document pipelines underneath them.

I'm a full-stack engineer at RamRod Software, building inspection software for a housing business while I finish a CS degree at Swinburne. I've been there since 2022 and I own the whole stack.

Since 2025 I've been rebuilding the company's toolchain from scratch. It's a Convex, Next.js, and React Native monorepo, and about 1,100 of its 1,150 commits are mine. Three parts were hard: a multi-org permission model I had to design, not adopt, an offline sync layer that resolves conflicts without losing a day of field work, and a document compiler that turns a typed content model into the Typst PDFs clients get. None of those had a clean off-the-shelf answer.

Before the rebuild I maintained what it replaced: nineteen C# projects, three Office add-ins, and a Xamarin field app passing SQLite files between office and field with no central server. It taught me to reason about data consistency with no network to arbitrate, to make emitters injection-safe, and to write permission boundaries down rather than assume them.

SKILLS

Primary TypeScript · C# · React · React Native · Next.js · Convex
Working .NET · SQLite · PowerSync · Expo · Bun · Zod · Typst · Clerk · Azure · Azure
OpenAI · VSTO · Xamarin · Elysia · Effect · Daytona
Familiar Blazor · Auth0 · PGLite · libSQL · Drizzle

EXPERIENCE

RamRod Software Pty Ltd 2022 — present
Full-stack engineer — principal author of the current inspection platform

EDUCATION

Swinburne University of Technology expected 2028
Bachelor of Computer Science, in progress

 **Gus** the inspection platform

WRITTEN ON SITE

Inspectors capture as they inspect, so hours of report writing left the working day.

IN DAILY USE

Darbecca's inspectors and office staff run every job through it, booking to delivery.

PRINCIPAL AUTHOR

I wrote 1,100 of 1,150 commits, about 372,000 lines across fifteen packages.

ONE SYSTEM

Three Office add-ins, a field app, and a file share collapsed into one place to look.

RECORDS CAME ACROSS

Two decades of legacy records migrated in, so nothing had to be re-keyed by hand.

BACKEND SCALE

131 tables, 304 indexes, and 452 functions behind one web app and one field client.

PARTS OF GUS

Choosing the infrastructure — Two stacks were proven unfit before either could become the thing to maintain.

Typed document compiler — The business stopped paying a third party to render the reports it sends clients.

Offline field capture — Buildings with no reception and whole disconnected days still produce a finished report.

In-house agent runtime — Agents resolve through the same access rules as people, so nothing gains reach by automating.

Database I/O reduction — One query was consuming 74 GB against a 50 GB monthly database budget.

Office and SQLite transport what Gus replaced

KEPT IT RUNNING

Nineteen C# projects stayed live daily while their replacement was built alongside.

REPORTS STAYED RIGHT

A wrong report comes from a bad merge, so correctness there was the whole job.

NOTES ON PHOTOS

Inspectors could annotate evidence on site, delivered across schema, capture, sync, and upload.

FAILURES GOT VISIBLE

Crash and usage telemetry turned a silent field app into one the business could debug.

OFFICE AUTOMATION

Word and Outlook generated and delivered reports without anyone re-typing them.

NINETEEN PROJECTS

Nineteen C# projects, about 84,000 lines, across seven .NET target frameworks since 2019.