HI

PRODUCT PORTFOLIO · 2025 TO 2026

Huzaifa Islam

Product Manager, Payments & Fintech

I help businesses get paid. I work out what to build, show it as a working prototype before anyone writes code, and then see it through to launch with the engineering team. This portfolio tells five of those stories in plain words.

2 weeks → 1 hour
to plan a new payment provider

30+
integration plans written for engineers

10
prototypes, each with a narrated video

Product Manager at Yuno, a global payments platform, since December 2025
Hyderabad, India · huzafaisam16@gmail.com · +91 9304538340 · linkedin.com/in/whozz

The prototypes use a made up brand (Acme Pay) and sample data. No customer data or company figures are shown.

IN SHORT

What changed because of my work

I work on the parts of online payments that decide whether money moves: what we know about a card, the security check a bank runs, and the connections to payment companies around the world.

2 weeks → 1 hour
Planning a new payment provider used to take two weeks. My AI assisted tool does it in under an hour.

30+
Clear integration plans for engineers, across five regions of the world.

1.1M+
Card ranges answered by the working prototype I built, not just slides.

23
Security and approval ideas turned into one roadmap you can click through.

3
Test tools that proved what payment providers really do, and unblocked their launch.

10
Prototypes and testers, each with a narrated, captioned video.

The five stories

#	Story	What it shows about me
1	**BIN Intelligence:** know a card before you charge it	Building a new product from nothing: research, pricing, prototype, requirements
2	**A planning assistant** that turns weeks into an hour	Designing an AI assisted internal tool that removes a bottleneck
3	**Connecting new ways to pay,** from plan to launch	Delivery, scope discipline and proof that things work
4	**A security roadmap** you can play like a video	Making a complicated plan simple enough for everyone to agree on
5	**Finding where money and time leak**	Using data and small tools to make problems visible

How I work

1 Understand
Talk to people, read the data first.

2 One page
The problem, the bet, the decisions.

3 Show it
A clickable prototype, early.

4 Detail it
Requirements and the test plan.

5 Check it
A review and a named sign off.

6 Launch it
Test evidence, then go live.

Story 1 · New product · Prototype + API

BIN Intelligence: know a card before you charge it

The first digits of every card number are called the BIN. They reveal what kind of card it is (credit, debit, prepaid, business) and which bank issued it. I designed a product that turns those digits into useful answers for online shops and the companies that process their payments.

The problem

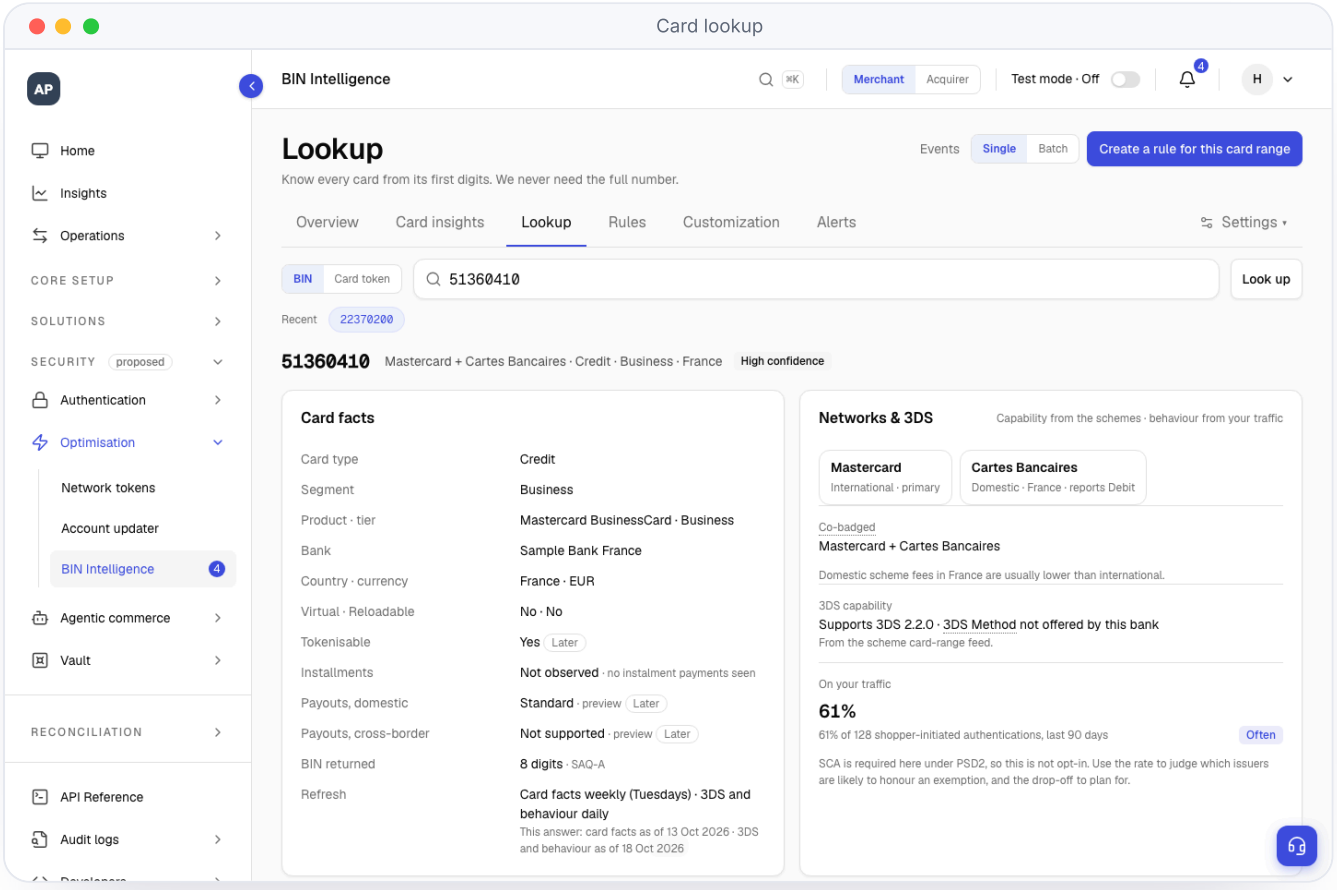
Shops only learn what a card is after they charge it. The card data they buy is often out of date. And nobody tells them how a bank is likely to behave, for example whether it will ask the shopper for an extra security check.

What I did

I researched the market, wrote a one page plan and the full requirements, designed the pricing, and built a clickable console plus a working API. People could try the product before any budget was committed.

The result

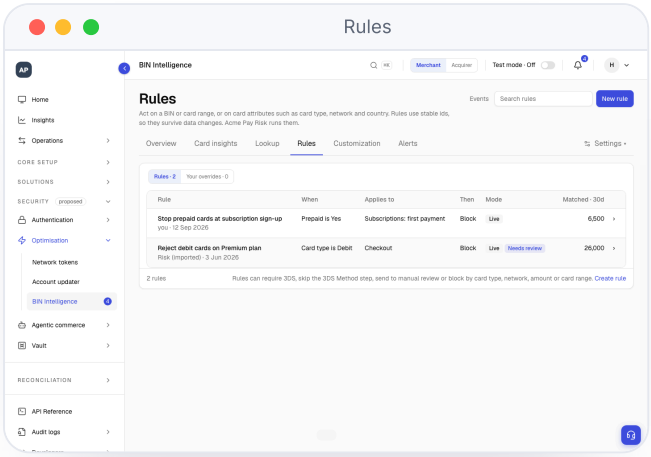
A clear plan in three stages and a simple privacy rule: every shared number describes a group of cards, never another shop. That rule protects shops and also makes the product different from what others sell.



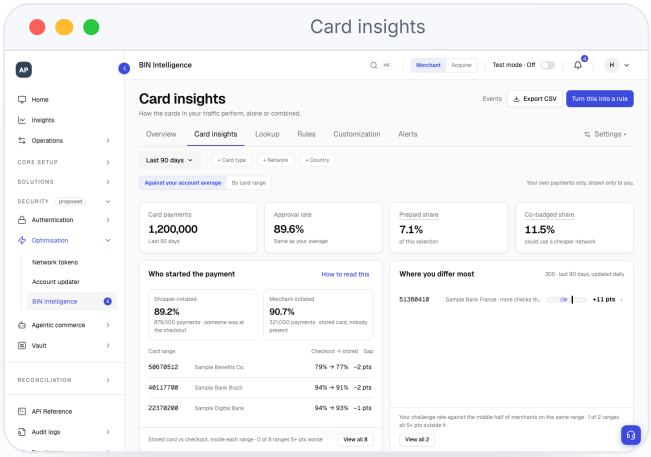
Card lookup: type the first digits and see the card type, the bank, the card networks and which security checks the bank supports. Sample data.

Story 1, continued

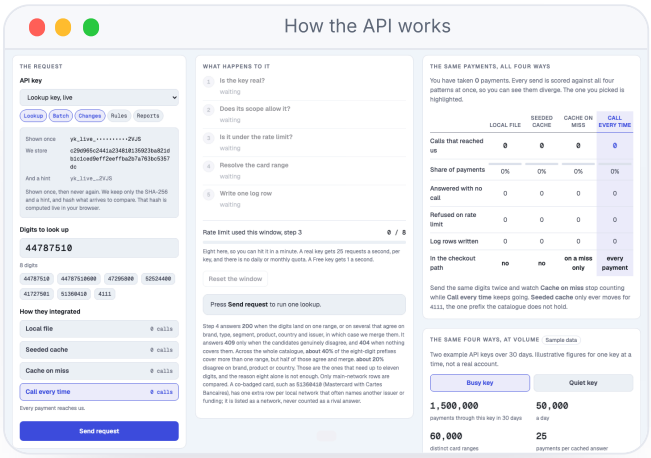
From a one page plan to a working product



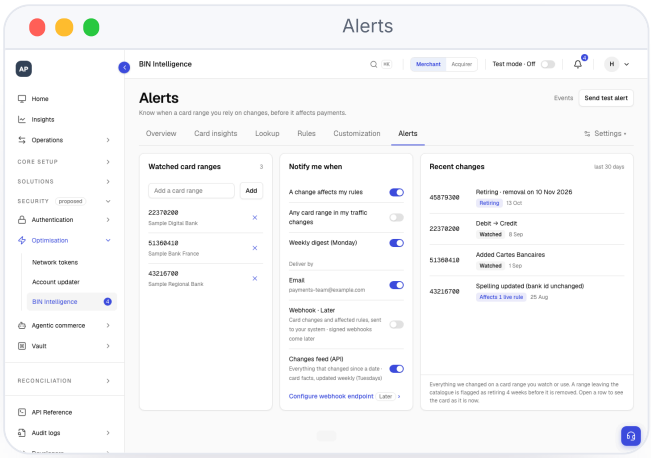
Rules: block or flag certain cards, tested on the last 30 days of payments before going live.



Card insights: how each group of cards performs compared with the wider market.



One API request followed from start to finish: the key, the limit, the answer and the record.



Alerts: hear first when card data changes or affects one of your rules.

What I produced	Why it mattered
One page plan	The problem, why now, the three big ideas, and the decisions leaders needed to make, each with my recommendation
Full requirements	What to build, user stories, the test plan and which teams are involved
Prototype and API	A console and a real API on over a million card ranges, with keys, limits and request records
Pricing and business case	Three plans, from free to premium, built from research with possible buyers

What I learned: the strongest part of the plan was not a feature, it was a boundary. Deciding early what the product would never show made it safer and easier to sell.

Story 2 · Internal tool · AI

A planning assistant that turns weeks into an hour

Before a business can accept payments through a new payment company, someone has to read that company's documents, compare it with what already exists, find who is affected, and write a plan for engineers. That reading was slowing the whole roadmap down.

The problem

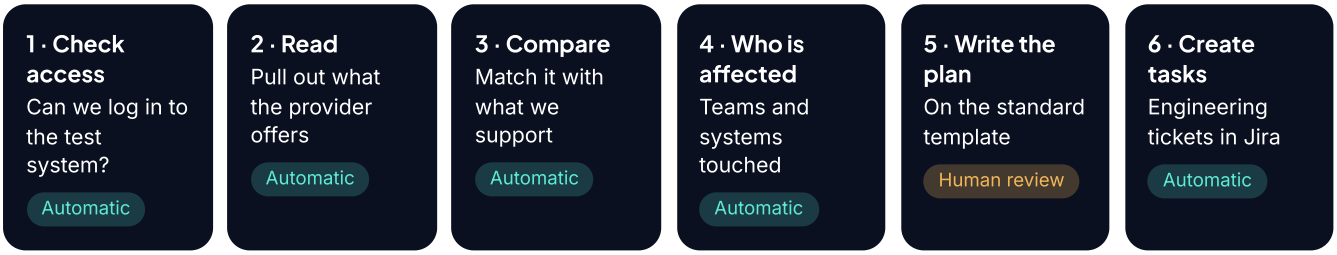
Each plan took one to two weeks. The quality depended on who wrote it, and engineers sometimes got plans with missing pieces.

What I did

I designed a six step AI assisted process that runs from a single command and ends with a plan and engineering tasks, always checked by a person.

The result

A plan now takes under an hour, and more than 30 have been produced this way, each reviewed by a technical lead.



Time to plan one provider — **2 weeks → under 1 hour**

Decisions that made it work

- **A model answer.** One perfect plan sets the standard, and every new plan is checked against it.
- **Steps in order.** Who is affected depends on the comparison, so the steps never run out of order.
- **A person decides.** Any step can be re run with feedback, and a named reviewer signs off.

Payment companies planned (a selection)

Stripe, Adyen, PayPal, Nuvei, Airwallex, dLocal, Worldpay, Moyasar, Paymob, Tap, Tazapay, PayU, Unlimint, Emerchantpay, PagBank, Pagar.me, Redeban, Netcetera, Belvo, TrustPayments and Xendit.

What I learned: letting the tool handle the reading freed my time for the parts that need judgement, such as what to leave out and how to handle the tricky cases.

Story 3 | Delivery | Launches

Connecting new ways to pay, from plan to launch

Every country has its own favourite ways to pay: cards, digital wallets, instant bank transfers. I plan each connection, test it, and get it ready to launch together with engineering and quality assurance. Some examples:

Clearer reasons when a payment fails

Airwallex · worldwide

Mapped 147 bank refusal codes by card brand and tested 22 scenarios, so shops can see why a payment was refused.

Bring your own security check

Moyasar, Paymob, Tap, Worldpay, Antom

Proved in test systems that each one accepts a bank security check carried out elsewhere, and wrote down exactly what data to send.

Automatic Pix payments

Adyen · Brazil

Turned the central bank's rules for repeat payments into a simple charging plan and requirements.

Safer stored cards

Stripe · worldwide

Mapped every switch needed on both sides to turn on card tokens, so shops could roll them out.

Live in production

Redeban · Colombia, NomuPay · Europe

Took connections from requirements through certification to live payments.

Local favourites

Nuvei · Poland, Brazil, Netherlands

Tested BLIK, Pix, iDEAL and Klarna, and mapped how refunds are reported.

India

PayU, BillDesk, Razorpay

Worked through local rules on invoices and keeping data in India, then tested the payment flows.

Fraud reports

Unlimint, Airwallex, Nuvei

Checked how each one reports fraud to shops, and defined what to build.

How I keep scope tight

Build what people need now. Extra options wait until there is a real business reason. Every extra switch is another thing that can go wrong.

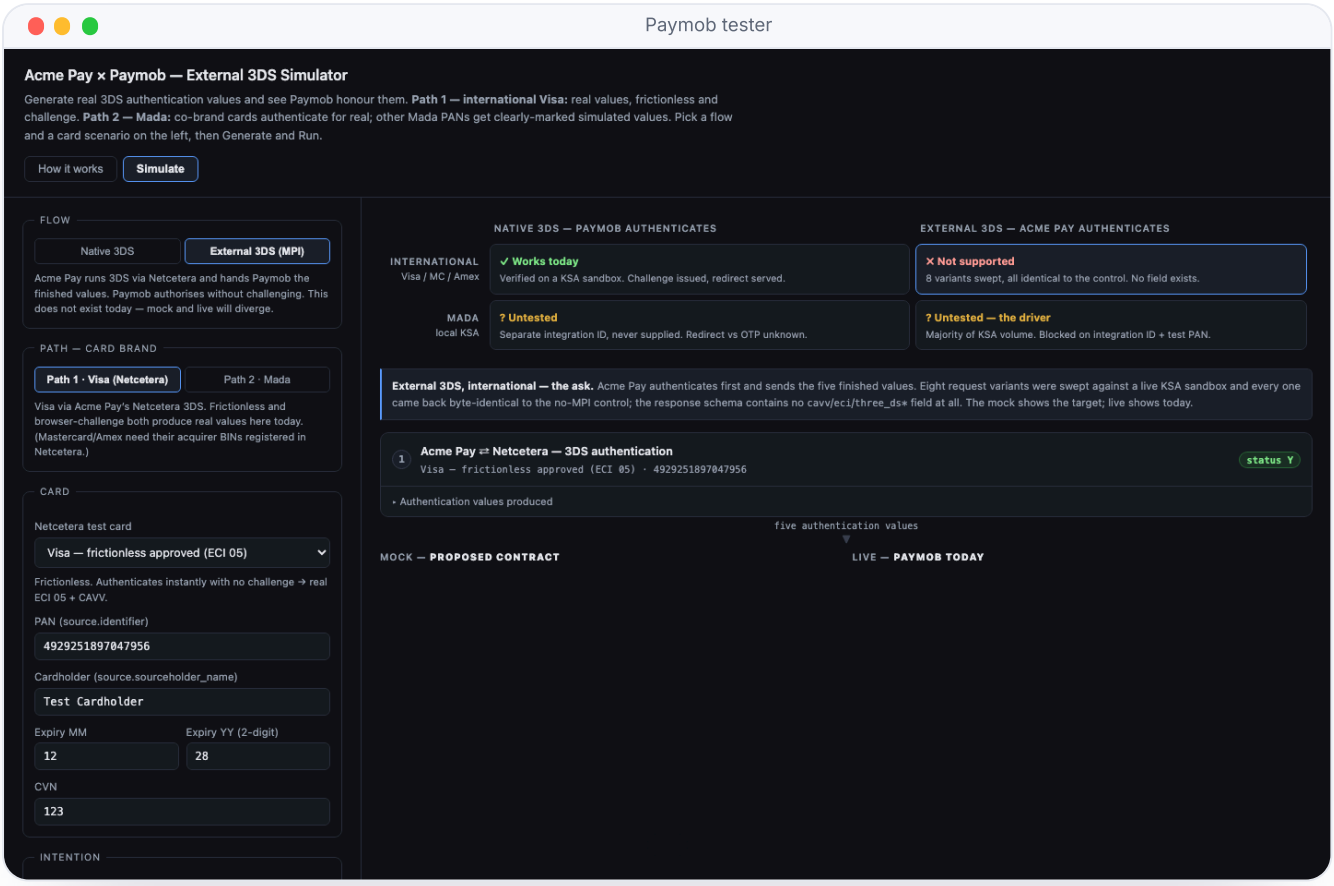
How I prove it works

Nothing is called done without test evidence and a quality sign off. When a test showed the plan was wrong, I corrected the plan.

Story 4 | Roadmap | Simulator

Making card security easy to see, then proving it

3-D Secure is the extra check your bank sometimes asks for when you pay online, such as a code sent to your phone. It stops fraud, but too many checks make people give up. There were 23 ideas to improve it, and no shared picture of how they fit together.



One of the three provider testers. It runs the same test payment two ways and shows, side by side, what works today and what does not.

What I did

Linked all 23 ideas to owners and tasks, and built a simulator that plays each one with its result, cost and who pays if fraud happens.

Testers for providers

Built three small testers for Paymob, Moyasar and Worldpay. Each sends a test payment two ways and shows where the answers differ.

The result

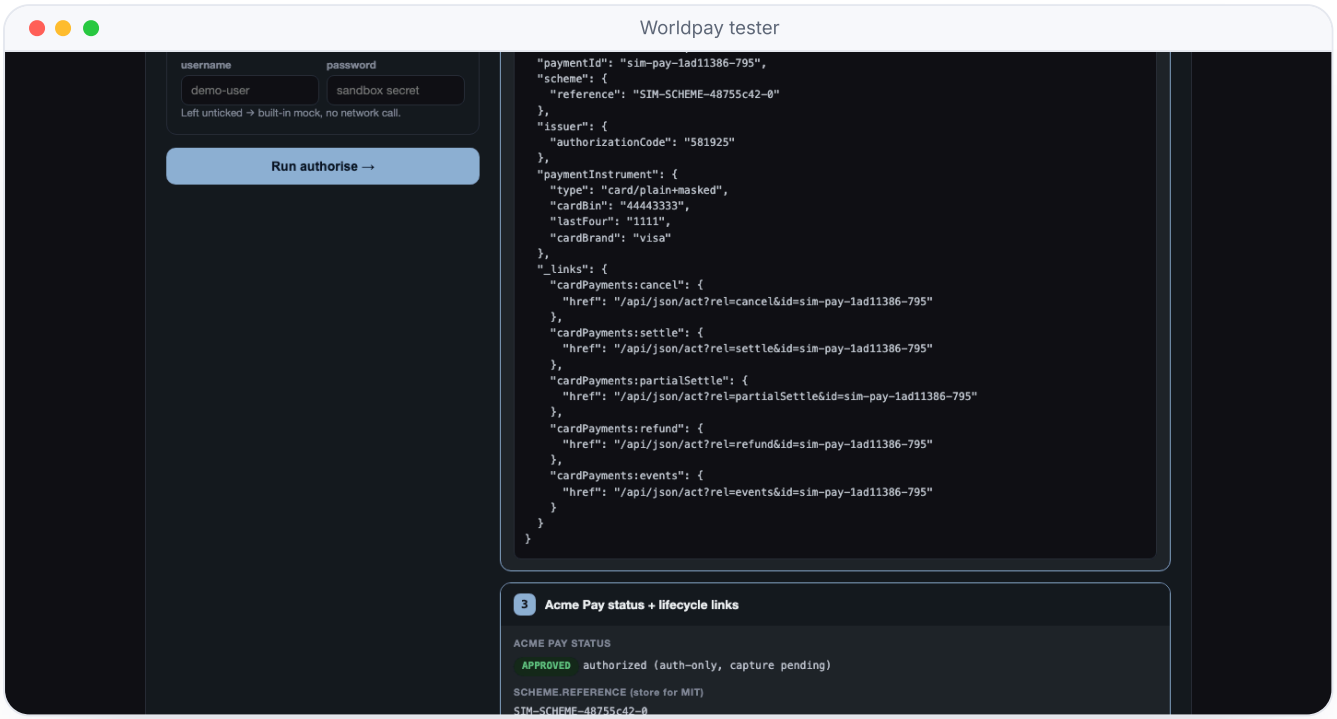
Long debates got shorter because everyone could see the difference. The testers turned guesses into proof and unblocked launches.

What I learned: a plan people can click beats a plan people have to read.

Story 5 | Data | Internal tools

Finding where money and time quietly leak

Payment companies lose money in places nobody owns, and lose time asking "what is the status?". I use data and small tools to make both visible, sized and owned.



Proof over opinion: a tester shows exactly what a provider sends back, so nobody has to guess.

- **A revenue leak finder (hackathon idea).** I designed an assistant that spots money lost in the payment process, works out how much, explains why, and sends it to the right team.
- **A ranked list of leaks.** I compared approval rates across card types, retries and card tokens with SQL, then ranked each gap and gave it an owner.
- **Workload health.** I measured how much work came in against how much went out, and where time really went, and used it to make the case for automation.
- **Live team dashboards.** I built a company wide work desk and a team desk fed straight from Jira, so people look status up instead of chasing it.
- **Why a shop's payments dropped.** I wrote SQL checks that pinpointed the cause, ready for account managers to take to the shop.

Skills across the portfolio

Product	Payments	Tools
Research, strategy, one page plans, requirements, pricing, business cases, roadmaps, getting people to agree	Card security (3-D Secure), card data, card tokens, Pix, UPI, BLIK, digital wallets, refusal codes, fraud reports, PCI DSS	APIs, SQL, test systems, prototypes in HTML and Node, AI assistants, Jira and Confluence

WATCH AND TRY

Prototypes and videos

Every prototype opens in a normal web browser, with no login. The three testers are small apps with their code included. Each has a narrated, captioned video of under two minutes.

Prototype	What you will see	Video
BIN Intelligence console	Card lookup, card insights, rules tested on past payments, corrections and alerts	1:56
How the card API works	One request followed from the key, to the limit, to the answer and its record	1:14
Security roadmap simulator	23 ideas for online card security, each played step by step	1:36
Paymob tester	The same test payment sent two ways, with the differences side by side	1:29
Moyasar tester	A first payment, then a repeat charge on the saved card	1:18
Worldpay tester	Two ways to connect to Worldpay, tested side by side	1:37

Four more pieces are internal analysis, so I show them in interviews rather than online: the card data discovery explorer, the system design, the online security deep dive, and the revenue leak finder.

See everything online

Website: **add your Render link here**
Email: huzafaisam16@gmail.com · LinkedIn: linkedin.com/in/whozz

This is the public edition. The prototypes use a made up brand (Acme Pay) and sample data. Customer and colleague names, internal systems and company figures have all been removed.