Customer Referral Platform

Replacing a manual, spreadsheet-run referral program with an automated platform for a multi-trade home-improvement contractor

Hugo · October 2026 · Case study

Summary

A major roofing, siding, windows, gutters and solar contractor ran its customer referral program by hand in a spreadsheet. Staff typed in every referral, checked it against the CRM and paid it out manually, and duplicate and fraudulent payouts slipped through.

I designed, built and run a referral platform that handles the program from first submission to final payout, with staff stepping in only for flagged exceptions. It has been live since January 2026 and has since expanded across the client's subsidiaries, including solar.

At a glance

Client	Major US home-improvement contractor (roofing, siding, windows, gutters, solar)
My role	Sole developer, end to end
Status	Live since January 2026; rebuilt version in production since March 2026
Stack	Python, FastAPI, HTMX, Tailwind, SQLite
Integrations	HubSpot CRM, JobProgress, Resend email

Key results

- **Fully automated.** Hundreds of referrals and thousands of dollars in payouts processed since January 2026.
- **No double payouts.** Three layers of protection, ending with a database guarantee, mean a referral can never be paid twice.
- **Admin work cut to exceptions.** Staff now handle only flagged referrals instead of every one.
- **Expanded without new code.** Solar and other subsidiaries were added through configuration.
- **Shipped fast and safely.** About 540 commits in under four months, backed by 246 automated tests.

The challenge

Every referral was typed into a spreadsheet, checked and paid out by staff. As the program grew, that process broke down.

Duplicate payouts. Nothing stopped the same customer being referred, and paid for, more than once.

Fraudulent claims. Self-referrals and repeat claims had to be caught by eye, and some got through.

No link to the CRM. For each referral, staff cross-checked HubSpot and the job-management system to see whether the job had sold and reached the stage where a payout was owed.

No visibility. Referrers had no easy way to submit referrals, and management had no reliable view of what was pending, owed or paid.

The solution

How a referral flows

Submission → Eligibility checks → CRM stage reached → Payout gate → Referrer paid

1. **Submission.** A customer submits a friend through their personal referral link.
2. **Eligibility checks.** Five automatic checks run before the referral is accepted (see below).
3. **CRM stage reached.** The CRM drives payouts: HubSpot for solar, JobProgress for every other trade.
4. **Payout gate.** Both CRMs feed one shared gate, which creates the payout once the job reaches a qualifying stage.
5. **Referrer paid.** The referrer gets an emailed claim link and picks a payout method.

What it includes

Public referral portal. Customers sign up once, get a personal referral link and submit friends in a few clicks.

Automatic fraud and duplicate checks. Five checks run on every submission:

Check	What it catches
Self-referral	Customers referring themselves
Blacklist	People barred from the program
Already a customer	Referred people who already exist in the job-management system
Lifetime prior payout	Anyone already paid for in the past
Name-based duplicates	The same person submitted more than once

CRM-driven payouts. The platform listens to HubSpot and JobProgress in real time and creates the payout automatically when a referred job reaches a qualifying stage.

Multi-company routing. Each trade routes to the right company, with its own commission amounts and payout triggers. This is what let the client add subsidiaries without new code.

Admin portal. Staff manage referrers, referrals, payouts, blacklists, email templates and settings, with a full audit trail of every action.

Automated email. Referrers get a transactional email at each step. Templates are editable in the portal, with delivery and open tracking.

Bulk enrollment. Existing customers can be imported from a CSV, and completed customers are enrolled automatically.

Results

Area	Before	After
Referral intake	Typed into a spreadsheet by staff	Self-serve portal with personal referral links
Duplicates and fraud	Caught by eye; some paid out	Five automatic checks on every submission, plus a lifetime blacklist
Payout decisions	Staff cross-checked the CRM by hand	Created automatically when the job reaches a qualifying stage
Paying referrers	Manual follow-up	Emailed claim link; referrer picks a payout method
Growing the referrer base	One by one	CSV bulk import, plus automatic enrollment of completed customers
Sales leads	Not captured	Every submission becomes a HubSpot lead, even declined ones
Coverage	One program	Roofing, siding, windows, doors and solar subsidiaries

What this means for the business

- **More referrals.** A simple portal, personal links and automatic enrollment of happy customers increased referral volume.
- **No duplicate or fraudulent payouts.** A database-level guarantee means a referral can never be paid twice.
- **Hours of admin work removed.** Staff handle only flagged exceptions.
- **More sales leads.** Every submission reaches the sales team in HubSpot, including referrals that don't qualify for a payout.
- **Built to expand.** New subsidiaries, including solar, were added through configuration instead of new development.

Engineering highlights

Three layers of payout protection. Submission checks, status checks and a database unique constraint mean a referral can never be paid twice, even when webhooks arrive at the same time.

Secure integrations. Every webhook is signature-verified: HubSpot v3 with a 5-minute replay window, JobProgress with HMAC, and Resend with Svix. Public endpoints are rate-limited per IP address.

Fails safe. If JobProgress is unreachable, a referral goes to manual review instead of being approved. HubSpot sync retries with exponential backoff and respects API rate limits.

Safe to deploy. 246 automated tests run before every deploy, with separate staging and production environments. CRM writes and emails are disabled outside production.

Auditable. Every significant action is written to an activity log, and every error is captured in a searchable error log.

Deliberately simple. About 15,000 lines of Python, no front-end build step, and a single server. Database migrations are versioned, six architecture decisions are written down, and every production incident has a postmortem.

My role

I was the sole developer, responsible for everything from requirements to production support.

- **Requirements and design.** Worked directly with the client to map their referral rules, payout policy and CRM workflow. Wrote the specs and recorded six architecture decisions.
- **Re-platforming.** Audited the first version (React and Supabase), which had become over-engineered for what it does. Rebuilt it on a far simpler stack with a feature-by-feature parity check, then cut over to production in March 2026.
- **Integrations.** Built the HubSpot, JobProgress and Resend integrations, including signed webhooks, rate-limit handling and retries.
- **Delivery.** Shipped about 540 commits in under four months, including an admin redesign, email analytics, auto-enrollment and payout tooling.
- **Production operations.** Run deployments, backups and incident response. Every production problem gets a postmortem and a safeguard so it cannot happen again.

Appendix: Glossary

- **Webhook:** a real-time message one system sends another when something happens, such as a job changing stage.
- **Signature verification:** confirming a webhook really came from the stated sender and was not altered. HMAC and Svix are two methods of doing this.
- **Replay window:** the time limit after which a captured webhook can no longer be resent to trigger the same action.
- **Rate limiting:** capping how many requests one source can make, to block abuse.
- **Unique constraint:** a database rule that makes it impossible to store the same record twice.