

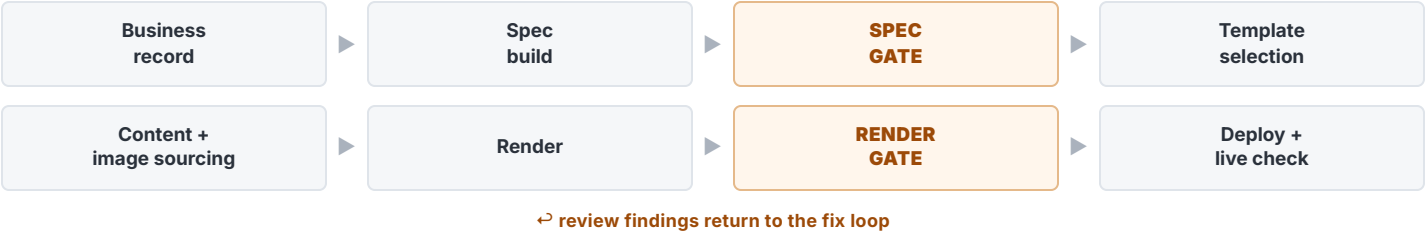
Automated Site Generation & Deployment Pipeline

A system that builds, reviews, and ships production marketing sites from a single business record — and **reviews every page against the record it came from** before it ships.

THE PROBLEM

Every client site was a hand-build. Shipping them at volume meant either quality collapsing or a human eyeballing every page before it went live — and the failures that matter are the ones that still look finished. A wrong photo, the wrong template, stale hours, a price nobody quoted — generated content fails silently: it renders perfectly and reads as true.

ARCHITECTURE



THE GATES

Spec gate

Validates the input before anything renders — business-type-to-template match, hours, service list, review attribution. Catches the failures that become invisible once a page looks finished.

Render gate

Automated visual and data review of the rendered page. Wrong imagery, wrong template, broken layout, and claims on the page that are not backed by the source record.

The omission rule

Where a fact cannot be traced to the source record — a price, a rating, an hours entry — the rule is to drop it or fall back to neutral wording rather than fill the gap. An unpriced tier renders as “Custom Quote” instead of a number nobody quoted.

BEFORE MERGE

Every change to the pipeline goes through automated review before it lands — CodeRabbit on the diff, plus formatters and static checks that fix what they can and flag the rest. Two layers by design: the render gate reviews what the pipeline produces; this reviews the code that produces it.

DURING THE BUILD

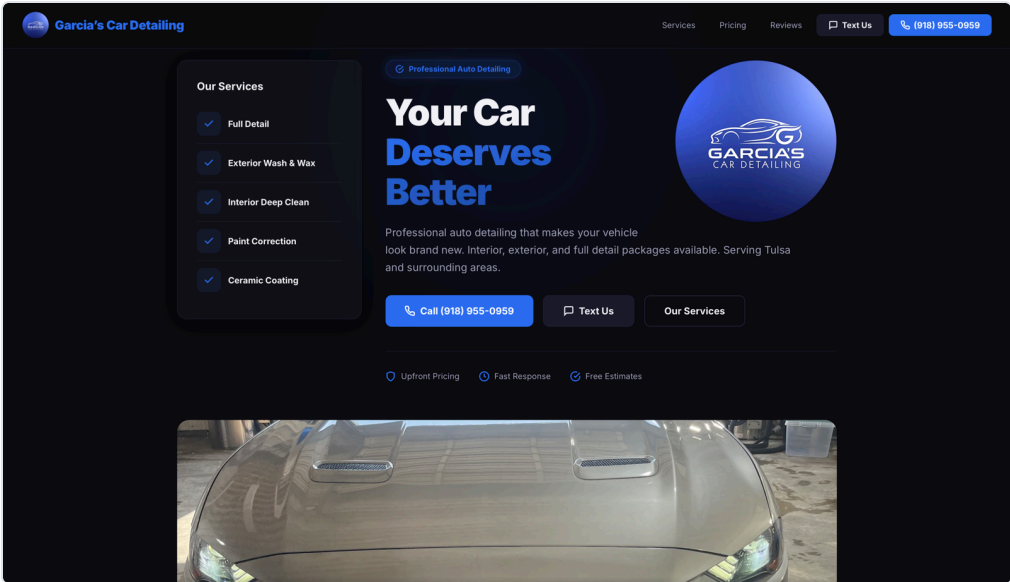
Every build runs in its own git worktree and deploys to its own isolated host target. A failing or half-finished build cannot reach another client's live site, and two builds can run at once without touching each other.

AFTER DEPLOY

Uptime, TLS certificate, and lead-delivery monitors run continuously — including a form wiring check that catches a submission handler silently dropping leads while still showing the visitor a success message.

STACK

Python · headless Chrome · Google Places & PageSpeed Insights APIs · Netlify deploy API · git worktrees for build isolation · Linux (cron, systemd)



GENERATED OUTPUT

Garcia's Car Detailing — built, reviewed, and deployed by the pipeline from a single business record. Services, contact details, hours and photography all resolved from that record, then checked against it again after render.

- 15+ production client sites shipped through the pipeline
- 2 review gates every build clears before it reaches a live URL
- 0 manual steps from business record to verified live site