Scope of Work: Results Roofing WebDev Sprint

Partner: CBHC
Client: Results Roofing
Project Type: Website Overhaul

Engagement Term: Q4 2025 - Q1 2026

Project Goal

Replicate and improve a competitor's online roofing estimate and checkout experience, delivering a production-ready flow: address → packages/pricing → financing → schedule → e-signature → deposit → portal, with reliability, security, and analytics baked in.

Scope of Work (Inclusions)

1. Product and UX
   - Customer flow: address capture with coverage decision, package selection, financing pre-qual, booking, signature, payment, confirmation, and customer portal.
   - Package content: materials database, warranty rules, good/better/best pricing presentation.

2. Integrations
   - Measurement vendor (Renoworks/HOVER/EagleView) job queue + polling.
   - CRM (JobNimbus/AccuLynx/HubSpot), E-signature (DocuSign/XodoSign), Payments (Stripe), Booking (Cal.com/Calendly), SMS (Twilio/Sakari), Email (SES/SendGrid).
   - Google Places for address validation and service-area screening.

3. Data & Analytics
   - Event contract implemented server-side; Server-side GTM and conversions APIs.
   - Offline conversion uploads and dashboards for Lead → Install attribution.
4. Platform & Ops
   - Next.js app with ISR for marketing pages, feature flags, perf budgets in CI.
   - SLOs, synthetic checks, canary deploys with auto-rollback; security headers (CSP/HSTS), secrets management, compliance for TCPA consent.

Out of Scope (Change Order If Needed)

- Full brand redesign, net-new CMS build, custom ERP, on-prem workloads, geographic routing beyond single region, call-center tooling, native mobile apps.

Deliverables

- UX wireframes and clickable prototype for the full funnel.
- Technical design doc: API contracts, data model, integration diagrams.
- Working Next.js application with production build pipelines.
- Event contract JSON and tracking plan; sGTM container and server.

- Configured integrations (payments, signature, CRM, measurement vendor, booking, SMS/email).
- Customer portal (view quote, status timeline, reschedule, pay balance).
- Security hardening: CSP/HSTS headers, secret rotation, roles and access.
- SLOs and synthetic checks; canary + auto-rollback policies.
- Go-Live runbook and rollback plan; training handoff.

Acceptance Criteria (sample, abbreviated)

- Address validation: p95 < 500 ms; coverage decision returned with reason.
- Measurement request: queued and acknowledged; result persisted; graceful fallback to manual estimate if vendor SLA > 90 seconds.
- Price presentation: three packages rendered with total and monthly payment; financing widget returns a tokenized pre-qual.
- Booking: time slot held and confirmed; CRM updated within 30 seconds.
- Signature: envelope created, signed, and callback updates agreement status.
- Payment: deposit captured via Stripe; receipt emailed; ledger recorded.
- Analytics: server events fire for all key actions; sGTM receives and forwards; daily offline conversions uploaded; dashboard shows Lead → Signed → Scheduled funnel.
- Reliability: Estimator availability ≥ 99.9%; quote p95 ≤ 3 s; checkout success ≥ 98.5%.
- Security: CSP nonces on scripts; HSTS 1 year, includeSubDomains, preload; cookies Secure/HttpOnly/SameSite=Lax; no secrets in repo; all env via manager.

Assumptions

- One CRM chosen. One measurement vendor chosen for MVP.
- Single payment processor (Stripe). Single e-signature provider.
- Single production region; standard business hours for go-live.
- Existing brand/CMS assets are available.

Roles and Responsibilities (RACI)

- Product Owner (Client): requirements, package rules, financing options, go-live approval.
- Tech Lead (Us): architecture, integrations, security, performance.
- Front-end Dev (Us): UX flows, portal, perf budgets.
- Back-end/Integrations (Us): APIs, webhooks, data model, vendor adapters.
- Analytics Engineer (Us): event contract, sGTM, dashboards, offline conversions.
- DevOps/SRE (Us): CI/CD, canary, observability, SLOs, runbooks.
- CRM Admin (Client): pipeline stages, fields, auth, test org.

Implementation Timeline and Project Plan

Phase 0 — Kickoff and Discovery (Week 1)

- Workshops: goals, constraints, vendor selection, compliance.
- Confirm packages, warranty rules, financing provider.
- Outputs: Project charter, finalized scope, risk register, milestone dates.

Phase 1 — UX and Architecture (Weeks 1–2)

- Wireframes and prototype for full funnel.
- Technical design: API endpoints, data model, integration map, event contract, security headers, SLOs, deployment strategy.
- Acceptance: sign-off on prototype and technical design.

Phase 2 — Foundations and Enablers (Weeks 2–3)

- Repo setup, CI/CD, branch protections, feature flags, environments.
- Perf budgets in CI (LCP, CLS, INP thresholds per template).
- sGTM server set up; base event contract scaffolding.
- Security headers middleware; secrets manager wired.

Phase 3 — Core Funnel Build (Weeks 3–6)

- Address capture and coverage decision with Google Places.
- Measurement service adapter + queue + polling + persistence.
- Pricing engine (rules, material DB, warranty logic) and package UI.
- Financing pre-qual integration.
- Booking integration and hold logic.
- E-signature templates and callback handler.
- Stripe deposits and receipts.
- Customer portal MVP (status, docs, payments).
- Unit tests and initial integration tests for each module.

Phase 4 — Analytics, Reliability, and Hardening (Weeks 5–7, overlaps)

- Full event contract implemented server-side; sGTM mappings; conversions APIs.
- Synthetic checks for happy path; SLO budgets with burn alerts.
- Canary deploys with automated rollback conditions.
- CSP locked with nonces; HSTS and cookie policies verified.
- Load and latency tuning; image pipeline and lazy-loading for heavy widgets.

Phase 5 — UAT, Training, and Content (Week 7–8)

- UAT scripts for each acceptance criterion.
- Training for ops and sales on portal and CRM handoffs.
- Final content pass on copy, disclosures, TCPA, and FAQs.

Phase 6 — Launch and Stabilization (Week 9)

- Staged roll-out: 10% → 50% → 100% traffic with error-budget guardrails.
- 7-day stabilization window; defect triage; performance watch.

Phase 7 — Enhancements and Experiments (Weeks 10–12)

- A/B tests: pricing clarity, financing placements, trust elements.
- Manual-estimate fallback SLA refinements; vendor timeouts.
- Dashboard fine-tuning; offline conversions QA; attribution reviews.

Gantt-style Overview (Weeks 1–12)

- W1: Kickoff, vendor picks, scope lock.
- W1–2: UX + Architecture.
- W2–3: Foundations (CI/CD, sGTM, security headers).
- W3–6: Core funnel build end-to-end.
- W5–7: Analytics, SLOs, canary, hardening.
- W7–8: UAT, training, content and legal.
- W9: Launch + stabilize.
- W10–12: Experiments and enhancements.

Work Breakdown Structure (high level)

1. Product and UX
    - User stories, wireframes, prototype, content and disclosure copy.

2. Front-end
    - Next.js routes, components, state, portal, accessibility pass, performance.

3. Back-end and Integrations
    - API layer, webhooks, vendor adapters, data model and persistence.

4. Data & Analytics
    - Event contract, sGTM server, conversions API, warehouse and dashboard.

5. DevOps/SRE
    - CI/CD, IaC where applicable, observability, SLOs, canary and rollback, security headers, secrets, runbooks.
6. QA and UAT
    - Unit/integration tests, synthetic monitoring, acceptance testing.

Risks and Mitigations

- Vendor latency or outage → manual estimate fallback; circuit breakers; queue with retries.
- Financing partner changes → provider abstracted behind a simple interface.
- CRM schema mismatches → mapping doc and test org; schema contracts.
- Performance regressions → perf budgets in CI; synthetic checks gating deploys.
- Consent/compliance gaps → legal review of TCPA copy; event-level consent flags.

Success Metrics (post-launch)

- Funnel: lead → qualified quote ≥ 35%, quote → signed ≥ 15%, signed → scheduled ≥ 90%.

- Speed: p95 quote response ≤ 3 s; mobile LCP ≤ 2.5 s on key routes.
- Reliability: availability ≥ 99.9%; auto-rollback when error budget burns.
- Attribution: ≥ 90% of signed jobs mapped to a campaign/source via server-side events + offline conversions.
- CSAT/NPS on the portal flow ≥ 8/10.

Change Control

- Any scope additions are tracked as change requests with impact on cost and time; critical path changes require sign-off.

Optional Acceleration Tracks

- **MVP in 3–4 weeks** by narrowing to one measurement vendor, one CRM, and deposit-only payments, while deferring portal features and experiments to Week 5+.
- **Parallel team lanes** to compress core build to 3–4 weeks: Front-end + Integrations + Analytics + SRE running concurrently with daily integration tests.

Hours by Phase (SOW-Aligned, FULL Custom)

Phase	What happens	Baseline hours	With code assist
0. Kickoff & Discovery	Goals, constraints, vendor picks, risk register, plan	30–40	n/a
1. UX & Architecture	Wireframes/proto; API & data model; event contract; SLOs; security headers plan	80–110	60–85
2. Foundations & Enablers	Repos, CI/CD, envs, feature flags, perf budgets, sGTM server scaffold, secrets	90–120	70–95
3. Core Funnel Build	Address/coverage, measurement queue+poll, pricing engine & packages, financing, booking, e-sig, Stripe deposit, portal MVP	260–340	185–250
4. Analytics, Reliability & Hardening	Server-side events, CAPI, dashboards, synthetic checks, canary+rollback, CSP/HSTS, load tuning	120–160	95–125
5. UAT, Training & Content	Test scripts, acceptance runs, ops training, disclosures/consent copy, fixes	60–80	55–70
6. Launch & Stabilization	Staged rollout, monitoring, hotfixes, post-launch review	40–60	36–50
7. Enhancements & Experiments	A/Bs, manual-estimate fallback polish, dashboards tuning	80–120	65–95
Total		**760–1,030**	**594–802**

Transparent Role Mix Assumptions (for staffing/rate cards)

- Front-end (Next.js/UX): 25–30%
- Back-end & Integrations: 30–35%
- Analytics/MarTech (sGTM, CAPI, dashboards): 12–15%
- DevOps/SRE (CI/CD, SLOs, canary, security headers): 12–15%
- PM/QA/Training: 12–15%

Code-assist savings primarily hit Phases 1–4; Phases 5–7 compress less because they're test, ops, and stakeholder-heavy.

MVP fast-track option (3–4 weeks)

Scope: 1 measurement vendor, 1 CRM, deposit-only payments, minimal portal, defer experiments.

Bundle	Baseline	With code-assist
MVP (3–4 wks)	320–450	250–340

Contingency & assumptions

- +10% contingency for vendor quirks (financing/e-sig/measurement), legal copy/consent reviews, and CRM schema drift.

- If you add a second CRM or financing provider, or expand the portal (change orders, warranty docs, crew calendar), expect +40–80 hours each.

- Data warehouse work beyond dashboards (e.g., LTV modeling, attribution pipeline hardening) is typically +40–80 hours.

Quick pricing notes

- Fixed-fee core: Phases 0–4 with clear acceptance criteria.
- T&M: Phases 5–7 and any experiments.
- Milestones tied to: Design sign-off, Core Funnel dev complete, Hardening complete, UAT pass, Go-Live, Post-launch review.

Hours by Phase (AI-assisted MVP)

Option A — 2-week sprint MVP (target ~140 - 160 HRS)

Phase	What happens (trimmed for MVP)	Hours (A)
0. Kickoff & Discovery	Rapid vendor picks, scope lock, risks	10–14
1. UX & Architecture	Light wires/proto, minimal data model & APIs	18–24
2. Foundations & Enablers	Single env CI/CD, basic perf budget, secrets, basic GA4	14–20
3. Core Funnel Build	Address/coverage, simple pricing (3 tiers), financing, booking, DocuSign, Stripe deposit	58–78
4. Analytics/Reliability/Hardening	Minimal: server events stub, security headers, smoke test	14–20
5. UAT/Training/Content	Single UAT pass, disclosures/consent copy	8–12
6. Launch & Stabilization	Staged cutover, bugfix window	6–8
7. Enhancements/Experiments	Deferred (only critical polish if needed)	0–4
Total		**140–160**

Investment: $18,900 - $21,600 (blended rate @ 135/hr)

What was cut/deferred: portal beyond receipt page, robust sGTM + CAPI mapping, canary/auto-rollback + SLOs, manual-estimate fallback, dashboards, multi-vendor support, advanced pricing rules, extensive a11y/localization.

Option B — 3-week robust MVP (target 180 - 240 HRS; adds measurement + light sGTM)

Phase	What happens (adds vs A)	Hours (B)
0. Kickoff & Discovery	As above	12–16
1. UX & Architecture	As above + queues/webhooks sketch	24–32
2. Foundations & Enablers	As above + sGTM server scaffold	18–26

3. Core Funnel Build	+ Measurement vendor adapter + queue/poll + simple manual fallback	78–108
4. Analytics/Reliability/Hardening	+ Map 3–4 key events in sGTM, one synthetic check	24–34
5. UAT/Training/Content	As above	10–14
6. Launch & Stabilization	As above	8–12
7. Enhancements/Experiments	Small polish/A-B hooks	6–8
Total		**180–240**

Investment: $24,300 - $32,400 (blended rate @ 135/hr)

Glossary: Key Terminology

Product & Flow

- **Good/Better/Best pricing**: Showing three preset package tiers with clear trade-offs (materials, warranty, lead time).

- **Pre-qual (financing)**: A soft credit check to estimate available loan terms without affecting credit score.

- **Manual-estimate fallback**: If the auto-measurement vendor is slow/down, switch the user to a short form and commit to an SLA to deliver a quote manually.

- **Measurement vendor adapter**: The code module that talks to HOVER/EagleView/Renoworks APIs to request, poll, and store roof measurements.

- **Booking hold**: Temporarily reserving a timeslot during checkout before final confirmation.

- **Envelope (DocuSign)**: A signed document package; your app creates an “envelope,” sends it, and listens for completed callbacks.

Data & Analytics

- **Event contract**: The canonical list of analytics events (names, fields, types) that every app must emit so dashboards and attribution work reliably.

- **Server events**: Analytics events sent from your server (not the browser), improving accuracy and ad-platform match rates.

- **sGTM (Server-side Google Tag Manager)**: A GTM container that runs on your server (often in Cloud Run/App Engine). It receives your server events and forwards to GA4, Meta, etc., reducing client-side tags and cookie loss.

- **sGTM server scaffold**: The initial setup of that server container (hosting, DNS, tag templates, secrets, permissions) plus a minimal mapping of a few events to downstream tools.

- **CAPI (Conversions API)**: Server-to-server conversion endpoints (Meta/Google/Microsoft) to backstop browser pixel losses.
- **Offline conversions**: Uploading down-funnel outcomes (signed deal, install) back to ad platforms or GA so they can learn and optimize.
- **Data layer**: A structured JS object on pages that pushes standardized event data for tags/trackers.
- **Attribution**: Mapping which channels/campaigns caused a conversion across touchpoints.

Web Performance & UX

- **Core Web Vitals**: Google's UX metrics—**LCP** (Largest Contentful Paint), **CLS** (Cumulative Layout Shift), **INP** (Interaction to Next Paint).
- **Perf budget**: Hard limits you enforce (e.g., LCP < 2.5s mobile) that fail a build if exceeded.
- **Lazy-load**: Defer loading heavy components (like a 3D visualizer) until user intent is clear.

Deployment & Reliability

- **CI/CD**: Continuous Integration / Continuous Delivery; automated build, test, and deploy pipelines.
- **Canary deploy**: Release to a small % of users/traffic first; auto-rollback if errors/latency spike.
- **Blue/green**: Maintain two identical environments; switch traffic instantly to the new one after checks pass.
- **Synthetic check**: A scripted robot that runs the "happy path" (e.g., address → quote) continuously and alerts on failure.
- **Smoke test**: A quick set of tests to confirm basic functionality after deployment.
- **SLO / SLA**: **Service Level Objective** (your internal target, e.g., 99.9% uptime) vs **Agreement** (what you promise customers).
- **Error budget**: The allowable failure time before you must slow or stop releases.
- **Runbook**: Step-by-step instructions to diagnose and fix incidents; includes rollback steps.
- **Chaos day**: Intentionally break a dependency to validate resilience and incident response.
- **P95 latency**: The 95th-percentile response time—how slow the slowest 5% of requests are.

Security & Compliance

- **CSP (Content-Security-Policy)**: HTTP header that locks down which scripts/styles can run; prevents XSS.
- **HSTS**: Forces HTTPS (and subdomains) for a set duration; mitigates downgrade attacks.
- **X-Frame-Options**: Prevents your site from being embedded in iframes (clickjacking).
- **Referrer-Policy**: Controls what referrer info the browser sends on navigation.
- **SameSite / Secure / HttpOnly cookies**: Cookie flags that reduce CSRF, MITM, and XSS risk.
- **OIDC (OpenID Connect)**: Modern auth for servers and CI to get short-lived, scoped credentials—no long-lived keys.
- **Secrets manager**: Central encrypted storage for API keys/tokens; accessed at runtime only.
- **TCPA consent**: Clear, logged consent (checkbox text, timestamp, IP) required before marketing SMS/auto-dialing.

Infrastructure & DevOps

- **IaC (Infrastructure as Code)**: Declarative infra (VPCs, DBs, policies) in code, not clickops.
- **Terraform / Terragrunt**: Popular IaC tool and its orchestration layer for DRY, multi-env setups.
- **GitOps**: Store desired state (manifests/configs) in Git; tools like ArgoCD sync clusters to Git automatically.
- **SBOM (Software Bill of Materials)**: A manifest of all packages in your build (used for supply-chain security).
- **SAST / SCA / DAST**: Code and dependency scanners—**Static**, **Composition** (libraries), **Dynamic** (live app).
- **WAF (Web Application Firewall)**: Filters malicious web traffic before it hits your app.
- **CDN / Edge**: Content Delivery Network serving assets and sometimes running logic close to users for speed.

- **PITR (Point-in-Time Recovery)**: Database backups that let you restore to an exact timestamp.

- **DR (Disaster Recovery)**: Strategy and drills to restore service after catastrophic failures.

App Architecture (Next.js / Headless)

- **Headless CMS**: CMS used only as a content API; a separate front end consumes it.

- **Atlas (WP Engine)**: Platform for headless WordPress (content backend) with a modern JS front end.

- **SSR / SSG / ISR**: Next.js render modes—**Server-Side Render**, **Static Site Generation**, **Incremental Static Regeneration** (rebuild pages on demand).

- **Hydration**: The browser attaching interactivity to server-rendered HTML.

- **App Router**: Next.js' newer routing system (layouts, server components, streaming).

APIs & Integrations

- **Webhook**: An HTTP callback your app receives when an external system changes state (e.g., "DocuSign completed").

- **Idempotency**: Re-sending a request yields the same result (prevents duplicate charges on retries).

- **Queue/poll**: Asynchronous workflow—enqueue a job then poll for completion (common with measurement vendors).

- **Rate limiting / backoff**: Throttling requests and retrying slowly when vendors say "too many/slow down."

Testing & Quality

- **Unit test / Integration test**: Testing small pieces vs testing multiple components end-to-end.

- **Accessibility (a11y)**: Ensuring the UI works with screen readers/keyboard; legally relevant in many markets.

SEO & Local

- **Structured data / schema**: JSON-LD markup (Organization, LocalBusiness, Service, FAQ) to enrich search results.
- **Sitemaps / robots.txt**: Machine maps for crawlers; robots.txt tells bots what not to fetch.
- **NAP consistency**: Name/Address/Phone uniform across pages and directories for local SEO.

Misc / AI-Assist

- **Code-assist (Claude/Lovable)**: Using AI to generate scaffolds, boilerplate, tests, and docs; humans still handle integration edge cases, security, and acceptance.
- **Scaffold**: A minimal but working skeleton of a service/component (routes, configs, CI hooks) you grow from.