# **Review Rocket Demo Front-End (Sample Funnel) – Development Handoff**

**Purpose:** Build a lightweight, static "try-it-yourself" demo of the Review Rocket flow (no database, no email, no webhooks). It’s purely for prospects/buyers to experience:

1. The "How did we do?" page (star selector + optional name/email fields)
2. The "We’re sorry" page for ≤3 stars
3. The "Thanks! Please post publicly" page for ≥4 stars (with fake redirect)

This will live as a separate mini-app (or folder) from the production client system. It must be trivial to fork in Replit and deploy to Vercel.

## **0. Constraints & Principles**

* **Zero backend required.** All logic is client-side (JavaScript). No persistence.
* **No external services.** No APIs, no SMTP, no Sheets.
* **One HTML/CSS/JS bundle** (vanilla or a tiny React/Vite app – your call; faster to do vanilla).
* **Must work on Vercel static hosting.** (No serverless functions needed.)
* **Easy to rebrand.** Logo, business name, primary color should be editable in one config block.

## **1. Tech Stack**

* **Option A (preferred for speed):** Plain HTML/CSS/JS (ES6 modules). Parcel or Vite optional for bundling.
* **Option B:** Next.js static export (but overkill – avoid unless you want built-in routing).

Routing: Use client-side routing via hash (/#/feedback) or simple separate HTML files.

## **2. Repo Structure**

****/demo

/public

logo.png # default demo logo

/src

config.js # branding + links

main.js # star logic + navigation

ui.css # minimal styling

pages/

index.html # page 1 – rating form

feedback.html # page 2 – low rating page

thank-you.html # page 3 – high rating page (redirect button)

vercel.json # optional (static config)

README.md

If using pure static HTML files, you can place them at root and keep config.js + ui.css in /assets.

## **3. Pages & Logic Specs**

### **3.1 Page 1 – /index.html ("How Did We Do?")**

**Fields:**

* Name (text, optional)
* Email (email, optional)
* Star rating (1–5). Use clickable stars or radio buttons styled as stars.
* Submit button

**Behavior:**

* On submit:  
  + If rating ≤ 3 → window.location = '/feedback.html?rating=2'
  + If rating ≥ 4 → window.location = '/thank-you.html?rating=5'
* Preserve name/email in query params if provided (for demo feel only).

**Copy Blocks (hard-coded, but configurable via config.js):**

* Headline: config.texts.landingHeadline
* Subhead: config.texts.landingSubhead
* Button label: config.texts.submitBtn

### **3.2 Page 2 – /feedback.html (Low Rating Path)**

**Content:**

* Headline: e.g., "We’re Sorry to Hear That…"
* Paragraph explaining private feedback
* Textarea (optional): "How can we improve?" (No storage; just demo)
* Button: "Send Feedback" (shows success toast or small modal; no backend)
* Link back to start ("Try again")

**Behavior:**

* On click "Send Feedback": show a simple alert('Thanks! We’ll be in touch.') or inline message.
* No network calls.

### **3.3 Page 3 – /thank-you.html (High Rating Path)**

**Content:**

* Headline: "Thanks for the kind words!"
* Paragraph: prompt to leave a public review
* Button: "Leave a Google Review" (url from config.publicReviewUrl)
* Optional 2-second fake "redirecting…" with setTimeout → then window.location = config.publicReviewUrl.
* Link back to start.

## **4. Configurability (config.js)**

****export const config = {

businessName: "Joe's Plumbing (Demo)",

primaryColor: "#0ea5e9", // Tailwind sky-500 as default

logoPath: "/public/logo.png",

publicReviewUrl: "https://www.google.com/",

texts: {

landingHeadline: "Tell Us How We Did",

landingSubhead: "Your feedback helps us serve you better.",

submitBtn: "Submit Review",

lowHeader: "We’re Sorry to Hear That…",

lowBody: "Help us make things right. Tell us how we can improve.",

lowSubmit: "Send Feedback",

highHeader: "Thanks for the Kind Words!",

highBody: "Would you be willing to leave this review on our public profile?",

highCta: "Leave a Google Review"

}

};

*   
  JS will inject these values into DOM on load.
* Basic CSS variables for color theming: --primary: config.primaryColor.

## **5. Styling**

* Keep CSS minimal. Simple utility classes or hand-rolled styles.
* Mobile-first, responsive layout. Single column. Large tap targets.
* Star widget: SVG or Unicode stars, filled on hover/click.

## **6. Acceptance Criteria**

* **AC1:** Selecting ≤3 stars routes to /feedback.html; ≥4 to /thank-you.html.
* **AC2:** All texts/colors/logo driven by config.js.
* **AC3:** No network requests made (verified in DevTools).
* **AC4:** Works when deployed to Vercel (static export).
* **AC5:** Codebase can be forked in Replit and run via npm run dev (if using bundler) or just "Run" if plain static.
* **AC6:** Lighthouse score ≥90 for Performance & Accessibility on mobile.

## **7. QA Checklist**

* Rate 1,2,3 → lands on feedback page every time
* Rate 4,5 → lands on thank-you page every time
* Buttons work on mobile (touch events)
* Config change (businessName, colors) reflects without code changes elsewhere
* No 404s when navigating directly to pages
* Query params handled gracefully (no crashes if missing)

## **8. Dev Tasks Breakdown**

1. **Scaffold repo & files** (structure above)
2. **Implement config loader** & DOM injector
3. **Build star rating component**
4. **Wire submit logic + redirects**
5. **Create feedback & thank-you pages** with placeholder text
6. **Implement basic routing (file-based)**
7. **Branding variables & CSS**
8. **README with deploy steps**
9. **Test on Replit & Vercel**

## **9. README Content (for Agent to Write)**

* Project purpose
* Quick start (Replit IDE mode + npm install && npm run dev if using bundler)
* How to change config
* How to deploy to Vercel (drag/drop or GitHub connect)

## **10. Future Enhancements (NOT in v1 scope)**

* Optional query param capture & localStorage to fake persistence
* Animated transitions between states
* CAPTCHA/rate limit (if you ever open it publicly and care about spam)
* Multi-language support via config

## **11. Deliverables**

* GitHub repo with code
* Deployed demo at https://<project>.vercel.app
* README with handoff instructions

## **12. Hand-Off Notes to Replit Agent**

* Follow this doc strictly—no server code.
* Keep everything in one repo folder /demo separate from main app.
* After completion, provide:  
  + Live URL
  + Zip of repo (if needed)
  + Short Loom explaining where to change config.js

**End of Handoff**