

Virtual Makeover v3 — Analyse & Rebuild-plan

Sjakie — 12 juni 2026

DEEL 1 — HOE DE HUIDIGE MAKEOVER IN ELKAAR ZIT

Het concept (wat het idee is)

Klant uploadt een selfie → AI analyseert het gezicht (vorm, geslacht, advies) → klant kiest kapsel + kleur uit een bibliotheek → Gemini genereert fotorealistische "jij met dit kapsel"-beelden → before/after-slider, delen via WhatsApp/e-mail, poll onder vrienden → betaalde credits via Mollie iDEAL voor meer generaties. Doel: leads + omzet voor de salon, en data over kapselvoorkeuren.

De flow in de huidige app

1. **Landing** → selfie upload of camera
2. **Gezichtsanalyse** (Gemini flash: gezichtsvorm, geslacht, advies)
3. **"3 looks" automatisch** — direct 3 AI-suggesties (incl. 1 gedurfde) als hook
4. **Volledige modus:** kapselbibliotheek (27 dames / 20 heren, met 2.5D multi-angle via MediaPipe), kleurkiezer, lengte (korter/zelfde/langer), kleurintensiteit, inspiratiefoto meesturen
5. **Moodboard:** make-up looks + baard looks + PDF-export
6. **Paywall/credits:** 1 gratis, daarna e-mail capture → Mollie iDEAL
7. **Poll:** resultaat delen, vrienden stemmen

Techniek (v1, AI Studio + Claude-verbeteringen)

- React 19 + Vite + TypeScript + Tailwind 4, één grote `App.tsx`
- Gemini direct vanuit de browser (`gemini-2.5-flash-image` voor beelden, `gemini-2.5-flash` voor analyse, deels al `gemini-3-flash-preview`)
- MediaPipe FaceLandmarker in-browser voor 2.5D hoeken

- Supabase (analytics, auth, polls) + extern backend-tje voor credits/Mollie (Render/VPS)
- Netlify hosting

Waarom hij "corrupt" voelt — de echte problemen

1. **Gemini API-key zit in de browser-bundel.** Typisch AI Studio-export: iedereen die de site opent kan jouw key uit de code vissen en op jouw kosten genereren. Dit is het grootste lek.
2. **App.tsx is een monoliet van 2.334 regels met 40+ losse state-variabelen.** Elke feature (3-looks, 2.5D, moodboard, poll, credits, paywall) is er later bovenop gelast. Eén wijziging breekt onverwacht iets anders — dáár komt het "corrupte" gevoel vandaan.
3. **Drie backends tegelijk:** Supabase + Render-webhook + VPS. Niemand weet meer wat waar leeft.
4. **Verouderde modellen** (gemini-2.5-flash-image) terwijl Nano Banana 2 / Gemini 3 veel beter gezichten behoudt — en gezichtsbehoud is prioriteit #1.
5. **Geen nette job-queue:** lange generaties lopen in de browser-tab; tab dicht = resultaat weg.

DEEL 2 — WAT BEHOUDEN, WAT ANDERS

Behouden (bewezen goed)

- Het kernconcept: selfie → analyse → looks → delen → betalen
- De "3 looks direct"-hook (sterke eerste ervaring)
- Before/after-slider, WhatsApp-delen, poll
- Mollie iDEAL + creditmodel
- Kapselbibliotheek-idee (eventueel beelden hergebruiken)

Anders in v3

V1	V3
API-key in browser	Key alléén server-side (Netlify Functions — zelfde bewezen patroon als de krullen-tool)
1 monoliet-App.tsx	Schermen als losse componenten met één duidelijke state-machine (upload → analyse → kies → resultaat → deel/betaal)
3 backends	1 backend: Netlify Functions + Blobs (+ Supabase alleen voor analytics/polls)
gemini-2.5-flash-image	Nano Banana 2 / gemini-3-flash-image, met fallback-engine (zelfde opzet als krullen-tool)
Generatie in de tab	Job-queue met polling — tab dicht = resultaat blijft (gefikt patroon van vanochtend)
MediaPipe 2.5D complexiteit	v3.0 simpel: 1 foto. 2.5D pas in v3.1 als upsell

DEEL 3 — DE BOUWMETHODE (zoals jij wil)

Stap A — Plan vaststellen (dit document; jij keurt goed of past aan)

Stap B — AI Studio prompt-loop (jouw credits, mijn prompts):

1. Ik geef jou prompt 1 → jij plakt 'm in Google AI Studio (Build-modus)
2. AI Studio bouwt → jij stuurt mij screenshot/link of beschrijft wat je ziet
3. Ik controleer tegen het plan → geef prompt 2 (verbetering of volgende scherm)
4. Herhalen tot je credits op zijn — we bouwen zo de complete **frontend + UX**
5. Jij downloadt de code (zip) en geeft 'm aan mij

Stap C — Ik maak 'm productieklaar (op de Mac Mini):

- API-key naar Netlify Functions (server-side), job-queue + Blobs
- Mollie/credits opnieuw aansluiten, Supabase analytics
- Deploy op Netlify, staging-test, dan DNS

AI Studio bouwt altijd met de key in de browser — dat is daar onvermijdelijk en prima voor het bouwen/testen. De beveiliging regel ik in stap C; zo gebruiken we AI Studio waar het sterk in is (snel UI bouwen op jouw credits) zonder het v1-lek te herhalen.

Volgorde van bouwen in AI Studio (per prompt-ronde):

1. Skelet + design-systeem (Haarvisie-stijl: stone-kleuren, premium, mobile-first) + scherm-flow
2. Upload + camera + gezichtsanalyse-scherm
3. "3 looks"-hook met nette laad-states
4. Kapselbibliotheek + kleurkiezer + opties
5. Resultaat: before/after-slider + galerij
6. Delen (WhatsApp/e-mail/poll) + e-mail capture
7. Paywall/credit-UI (frontend-only; echte betaling komt in stap C)
8. Moodboard (laatste — grootste feature)

DEEL 4 — OPEN KEUZES (graag jouw antwoord)

1. **Kapselbibliotheek:** bestaande 47 kapsels hergebruiken, of nieuwe genereren met Nano Banana 2?
2. **2.5D multi-angle:** meteen in v3, of eerst simpel met 1 foto en 2.5D later als upsell?
3. **Moodboard:** meenemen in de rebuild of eerst de kern perfect maken?
4. **Domein:** blijft het haarvisie-makeover.netlify.app of wordt het makeover.haartips.nl?

(Het oude v2-blueprint van 27 mei — Next.js/Vercel/multi-tenant SaaS — parkeer ik; de white-label-SaaS-droom kan later altijd nog bovenop deze v3.)