

# Vibecode Quickscan

10-Pijler Review — geanonimiseerde voorbeeldrapportage

Blis Digital — Jorden Willemsen
maart 2026
Geanonimiseerd voorbeeld

Blis Digital

# VAN 1.75 NAAR 3.3 — PRODUCTIEWAARDIG, NOG NIET ONDERHOUDBAAR

**3.3/5**

**HUIDIGE SCORE**

Solide basis voor productie — CI/CD en iteratief scannen ontbreken nog voor duurzaam beheer

**1.75/5**

**SCORE FEB 2026**

Een eerdere externe review signaleerde terechte security-issues, maar onderschatte de volwassenheid van AI development en ging te diep in op code styling

**8/10**

**PIJLERS ≥ 3/5**

Kern van de applicatie staat — beveiliging, privacy en architectuur zijn productiewaardig

**3×**

**BLOKKEERDERS VOOR SCHAALBAAR BEHEER**

Geen CI/CD-pipeline, geen proactieve alerting, geen staging-omgeving — zonder deze drie is doorontwikkelen risicovol

# SCOREOVERZICHT — 10 PIJLERS

| PIJLER | SCORE | STATUS |
|---|---|---|
| 1. Authenticatie & Autorisatie | 4/5 | • Custom Claims JWT, Firestore-regels, auto-logout |
| 2. Beveiliging | 4/5 | • Secret Manager, strikte CSP/HSTS headers, input-validatie |
| 3. Privacy & AVG | 4/5 | • GDPR Art. 17, consent-logging, EU-datacenter (europe-west1) |
| 4. Code Kwaliteit | 4/5 | • Goede structuur, scheiding van concerns, ESLint geconfigureerd |
| 5. Testen | 2/5 | • Backend-tests aanwezig, geen frontend-tests of CI/CD |
| 6. Hosting | 4/5 | • Firebase Hosting CDN, automatisch schaalbaar, EU-regio |
| 7. Back-ups | 3/5 | • Dagelijkse Firestore-backup, Auth & Storage niet gedekt |
| 8. Licenties & Dependencies | 3/5 | • Weinig deps, xlsx AGPL-licentie controleren |
| 9. Monitoring | 3/5 | • Sentry + Firestore logs, geen proactieve alerting |
| 10. Ontwikkelproces | 2/5 | • Geen CI/CD, geen staging-omgeving |

Goed (4–5/5) Voldoende (3/5) Aandacht nodig (1–2/5)

# AUTHENTICATIE & AUTORISATIE (4/5) • BEVEILIGING (4/5)

## Authenticatie & Autorisatie

**Wat werkt goed:**

- Admin-rechten via Firebase Custom Claims (JWT) — niet meer via client-side Firestore-veld
- `checkIsAdmin()` in Cloud Functions verifieert `auth.token.admin === true`
- Firestore-regels blokkeren schrijven naar `role`, `saldo` en `points` vanuit de client
- `AdminRoute` herleidt niet-admins direct naar de homepage
- Auto-logout na 10 minuten inactiviteit

**Aandachtspunten:**

- Firebase API-sleutel staat in `firebase.js` — beperk deze in Google Cloud Console tot het productiedomein
- Rate-limiting op login-pogingen is niet expliciet geconfigureerd

## Beveiliging

**Wat werkt goed:**

- Secrets (API-sleutels van externe koppelingen) in Firebase Secret Manager — niet hardcoded in de code
- HTTP headers geconfigureerd: HSTS, strikte CSP, X-Frame-Options, X-XSS-Protection, Referrer-Policy
- CSP: `default-src 'none'` met expliciete whitelist per bron
- Input-validatie aanwezig in Cloud Functions voor gevoelige mutaties

**Aandachtspunten:**

- CSP bevat `style-src 'unsafe-inline'` — overweeg nonces of hashes voor inline styles
- `react-router-dom` v7.9.4: voer `npm audit` uit om te bevestigen dat eerder gevonden XSS/CSRF CVEs zijn verholpen
- `nodemailer` staat in `functions/package.json` maar wordt niet gebruikt — verwijder deze dode dependency

# PRIVACY & AVG (4/5) · HOSTING (4/5)

## Privacy & AVG

**Wat werkt goed:**

- Cookie-banner aanwezig, privacybeleidspagina op `/privacy`
- Toestemmingsregistratie in registratieformulier: `consent_accepted_at`, versienummer en expliciete toestemmingsbevestiging
- GDPR Art. 17 (recht op vergetelheid) geïmplementeerd: `deleteUserAccount` Cloud Function anonimiseert PII, bewaart productdata voor garantie
- Cloud Functions draaien in `europe-west1` (België) — data blijft in de EU

**Aandachtspunten:**

- Login-logs bevatten `userAgent` — dit is technisch PII; documenteer in privacybeleid of verwijder
- Geen bewaarbeleid gedefinieerd voor `logs_login`; definieer maximale bewaartermijn met automatische opruiming

## Hosting

**Wat werkt goed:**

- Firebase Hosting op Google Cloud: CDN wereldwijd, automatisch schaalbaar
- Cloud Functions in `europe-west1` — AVG-conform voor EU-data
- SPA-routing geconfigureerd via rewrites in `firebase.json`
- HTTPS verplicht via Firebase Hosting (automatisch)

**Aandachtspunten:**

- Geen expliciete quota-limieten op Cloud Functions — bij onverwacht hoog gebruik kunnen kosten oplopen; stel budgetalerts in via Google Cloud Console
- Geen staging- of acceptatieomgeving aantoonbaar aanwezig — wijzigingen gaan direct naar productie

# CODE KWALITEIT (4/5) · LICENTIES & DEPENDENCIES (3/5)

## Code Kwaliteit

**Wat werkt goed:**

- Duidelijke scheiding van concerns: `hooks/`, `services/`, `utils/`, `components/`, `pages/`
- Herbruikbare utilities: `dateUtils.js`, `statusUtils.js`, `usePagination.js`
- PropTypes-validatie aanwezig — was 42× ontbrekend per eerdere externe review
- ESLint geconfigureerd: `no-eval`, `no-unused-vars`, react-hooks regels
- `AdminLayout` voor consistente admin-UI zonder herhaling

**Conclusie:** Code Kwaliteit scoort 4/5 — goed fundament aanwezig, geen kritieke verbeterpunten voor productie.

## Licenties & Dependencies

**Wat werkt goed:**

- Meeste dependencies zijn MIT of Apache 2.0 (firebase, react, papaparse, prop-types)
- Weinig externe dependencies — beperkt aanvalsoppervlak
- Overrides in `functions/package.json` voor bekende kwetsbaarheden in transitive dependencies

**Aandachtspunten:**

- **xlsx (SheetJS)** heeft een dual-license model: de community-versie is AGPL-3.0; voor commercieel gebruik is een betaalde licentie vereist — controleer of dit van toepassing is
- **nodemailer** in `functions/package.json` wordt nergens gebruikt maar is wel een actieve dependency — verwijder dit om aanvalsoppervlak te minimaliseren
- **react-router-dom v7.9.4** — voer `npm audit` uit; eerder gevonden CVEs (XSS, CSRF) dienen geverifieerd te zijn

# TESTEN (2/5) · ONTWIKKELPROCES (2/5)

## Testen

**Wat werkt goed:**

- Cloud Functions unit-tests aanwezig (`functions/test/functions.test.js`) met Jest en uitgebreide mocks voor firebase-admin, axios en nodemailer
- Firestore security rules tests aanwezig (`functions/test/rules.test.js`) met `@firebase/rules-unit-testing`
- Test-scripts geconfigureerd: `npm run test:functions`, `npm run test:rules`, `npm run test:all`

**Verbeterpunten:**

- Geen frontend-tests — geen test-framework geconfigureerd in de hoofd `package.json`
- Geen integratietests die end-to-end flows testen (registratie → extern partnersysteem → e-mail)
- Geen CI/CD-pipeline die tests automatisch draait bij elke commit of pull request

## Ontwikkelproces

**Wat werkt goed:**

- Vite voor snelle lokale ontwikkeling met Hot Module Replacement
- Firebase Emulators geconfigureerd voor lokaal testen (auth, firestore, functions op poort 9099/8080/5001)
- ESLint pre-deploy check geconfigureerd in `firebase.json`
- Duidelijke npm-scripts: dev, build, lint, preview, test

**Verbeterpunten:**

- Geen CI/CD-pipeline — geen automatische test-run of build-validatie bij push
- Geen staging- of acceptatieomgeving — wijzigingen gaan direct naar productie
- Geen formeel code-review-proces of PR-workflow
- Geen gedocumenteerde deploy-procedure of rollback-plan

# BACK-UPS (3/5) · MONITORING (3/5)

## Back-ups

**Wat werkt goed:**

- Automatische dagelijkse Firestore-backup via `scheduledFirestoreBackup` Cloud Function om 03:00 Amsterdam-tijd
- Backups worden opgeslagen in Google Cloud Storage met datum-gebaseerde folder (`gs://<bucket>/YYYY-MM-DD`)
- Backup dekt alle Firestore-collecties

**Verbeterpunten:**

- Firebase Auth-gebruikersdata (accounts) is **niet** opgenomen in de backup — bij verlies van het Firebase-project zijn accounts weg
- Firebase Storage-bestanden (beeldbank) worden **niet** geback-upt
- Geen aantoonbaar herstelproces getest (disaster recovery test ontbreekt)
- Controleer of de backup-bucket voldoende beveiligd is met juiste IAM-permissies

## Monitoring

**Wat werkt goed:**

- Sentry geïntegreerd in `main.jsx` voor frontend error-tracking in productie
- Performance sampling geconfigureerd (`tracesSampleRate: 0.1`)
- Login-logs in Firestore (`logs_login`) voor audit-trail
- Admin-actie logs in Firestore (`logs_admin`) voor beheeractiviteiten
- Cloud Functions loggen via Firebase Logging

**Verbeterpunten:**

- Geen proactieve alerting geconfigureerd — bij crashes of kritieke fouten wordt niemand automatisch gewaarschuwd
- Geen uptime-monitoring (geen melding als Firebase Hosting onbereikbaar is)
- Sentry DSN is hardcoded in `main.jsx` — overweeg een environment variable voor portabiliteit

# ACTIEPLAN — TOP 5 PRIORITEITEN

| ACTIE | PRIORITEIT | OMSCHRIJVING |
|---|---|---|
| **CI/CD-pipeline** | HOOG | • Stel **GitHub Actions** in: automatische test-run + build-validatie bij elke push naar main. Voorkomt regressions en maakt deploys reproduceerbaar. |
| **npm audit — react-router CVEs** | HOOG | • Voer **npm audit** uit op het project. Bevestig dat react-router-dom v7.9.4 de eerder gevonden XSS- en CSRF-kwetsbaarheden heeft verholpen. |
| **Firebase API-sleutel restrictie** | MIDDEL | • Beperk de publieke Firebase-sleutel in **Google Cloud Console** tot alleen het productiedomein. |
| **Verwijder dode dependency nodemailer** | MIDDEL | • Verwijder **nodemailer** uit functions/package.json — het wordt niet gebruikt maar vergroot het aanvalsoppervlak onnodig. |
| **Auth + Storage backups** | MIDDEL | • Voeg periodieke export van **Firebase Auth-gebruikers** en **Storage-bestanden** (beeldbank) toe aan het backup-plan. |

Hoog — direct oppakken  Middel — plannen

# CONCLUSIE — PRODUCTIEKLAAR MET ACTIEPLAN

### Van prototype naar productie

De applicatie heeft een **sterke verbetering** doorgemaakt. In februari 2026 scoorde de applicatie 1.75/5 — onvoldoende voor productie. Vandaag staat de score op **3.3/5**.

De fundamenten zijn nu solide gelegd:

- Beveiligingsarchitectuur is correct opgezet met backend-validatie
- AVG-compliance is aantoonbaar geborgd
- Monitoring en backups zijn aanwezig
- Code heeft een duidelijke, onderhoudbare structuur

**Streefniveau:** met de top-5 acties is een score van 4.0/5 haalbaar.

### Wat nu?

**Direct oppakken:**

1. CI/CD-pipeline via GitHub Actions instellen
2. `npm audit` draaien — react-router CVEs verifiëren

**Plannen (binnen 4 weken):**

1. Firebase API-sleutel restricties instellen in Google Cloud Console
2. `nodemailer` verwijderen uit `functions/package.json`
3. Backup-scope uitbreiden: Firebase Auth-gebruikers + Storage-bestanden

**Langere termijn:**

- Frontend-tests toevoegen (React Testing Library of Vitest)
- Staging-omgeving opzetten voor veilige deploys
- Bewaarbeleid en automatische opruiming voor `logs_login`

De applicatie is productieklaar. Met dit actieplan groeit de applicatie naar een robuust, schaalbaar niveau.

# Vragen?

Neem contact op met ons team