app.guidetranslator.com

Update_V7_build: 26: Februar, 2026, 8:36

💡 Ulrich-Erklärung: Was bedeutet das?

Alle 7 Bauphasen der App sind fertig und auf dem Server. Der 'Build' ist wie das Zusammenpacken aller Dateien zu einem fertigen Paket — und das Paket ist jetzt 26% kleiner geworden (512KB → 380KB). Das bedeutet: die App lädt schneller für jeden Besucher.

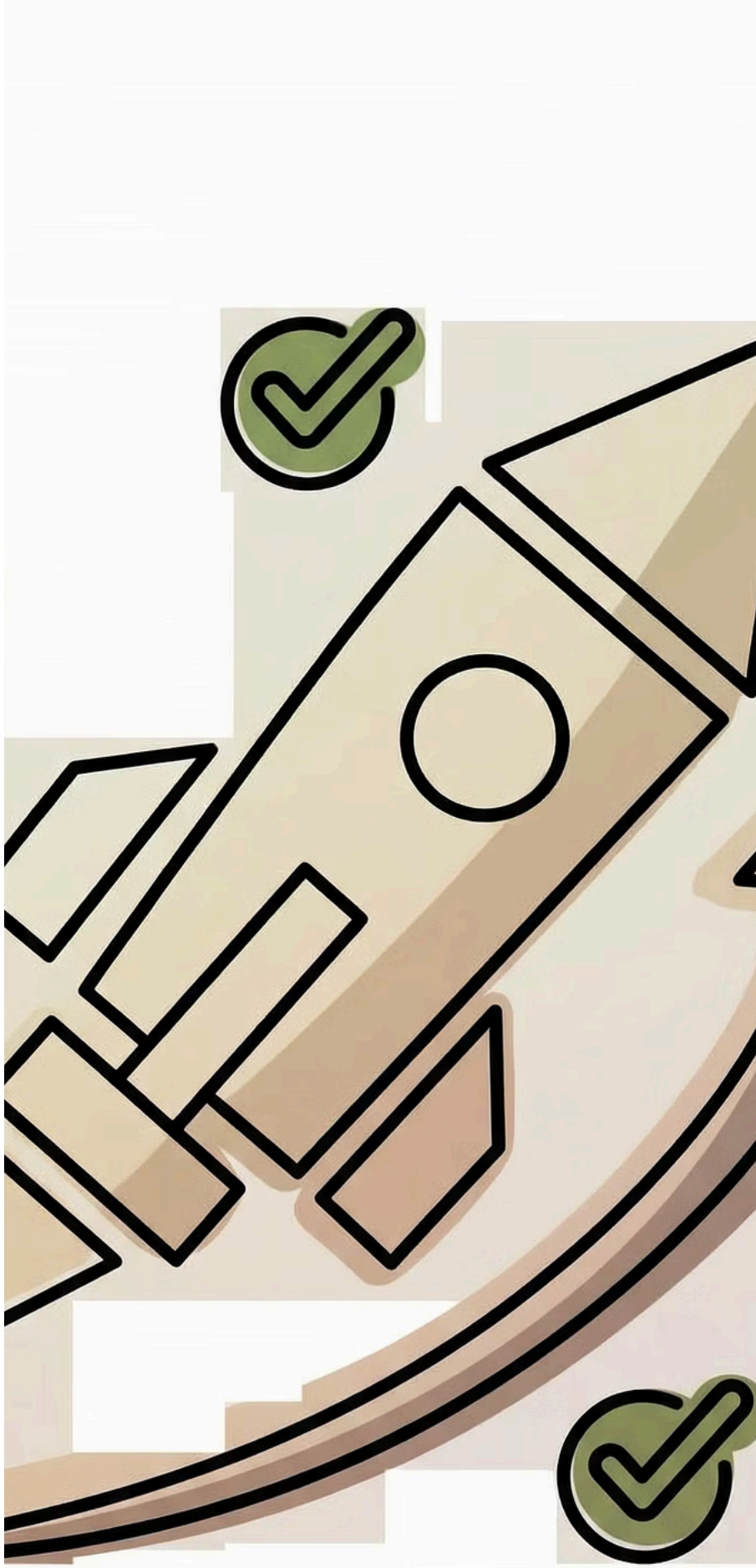
STATUS

Alle 7 Phasen: Implementiert & Deployed 🎉

Build erfolgreich (512KB → 380KB nach Code-Splitting). Alle Phasen sind implementiert und gepusht.

Commit	Phase	Inhalt	Status
Phase 1–3	Router + Modularisierung	Router, 6 Segmente, Sales-Flow	✅ Live
Phase 4	Supabase Auth	Rollen, Accounts-Tab	✅ Live
Phase 5–7	Pricing + Stripe + Dashboard	Pricing, Checkout, Dashboard	✅ Live
Jetzt	UX + Cron + Code-Splitting	Cron, Lazy-Loading, Nav-Updates	✅ Live

> Von 512KB auf 380KB — 17 separate Chunks, alle Seiten laden on-demand. Kein Warning mehr.



NEUE FEATURES

Was gerade gebaut wurde

Diese Runde hat vier wichtige Lücken geschlossen — automatische Follow-Ups, schnellere Ladezeiten, bessere Navigation und Vercel Cron.

💡 Ulrich´s-Erklärung: Was wurde gebaut?

In dieser Runde wurden 4 Dinge fertiggestellt: (1) Ein automatischer E-Mail-Erinnerer für Leute, die 'später testen' geklickt haben. (2) Die App lädt jetzt schneller. (3) Auf jeder Segment-Seite gibt es jetzt einen 'Preise ansehen' Button. (4) Eingeloggte User sehen ihr Dashboard direkt in der Navigation.

Follow-Up Cron

api/check-followups.js — läuft täglich 8 Uhr. Sendet automatisch Erinnerung wenn jemand vor 7+ Tagen 'später testen' gewählt hat.

Code-Splitting

Alle 12 Seiten lazy-loaded. Haupt-Chunk: 512KB → 380KB. 17 separate Chunks — Seiten laden nur wenn sie gebraucht werden.

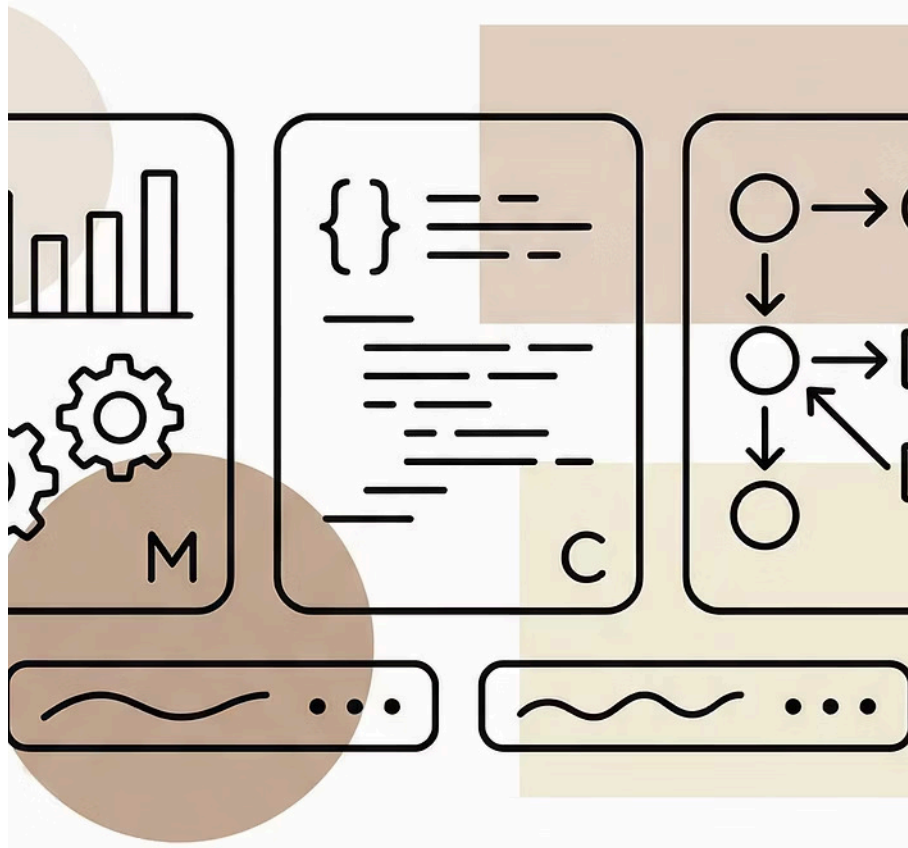
Pricing CTAs

'Preise ansehen' Button auf allen Segment-Landings — im Hero und im Footer. Direkter Weg zum Checkout.

Auth-Navigation

'Dashboard' Link in Nav für eingeloggte User. 'Admin' Link für Admins. Suspense mit Lade-Indikator.

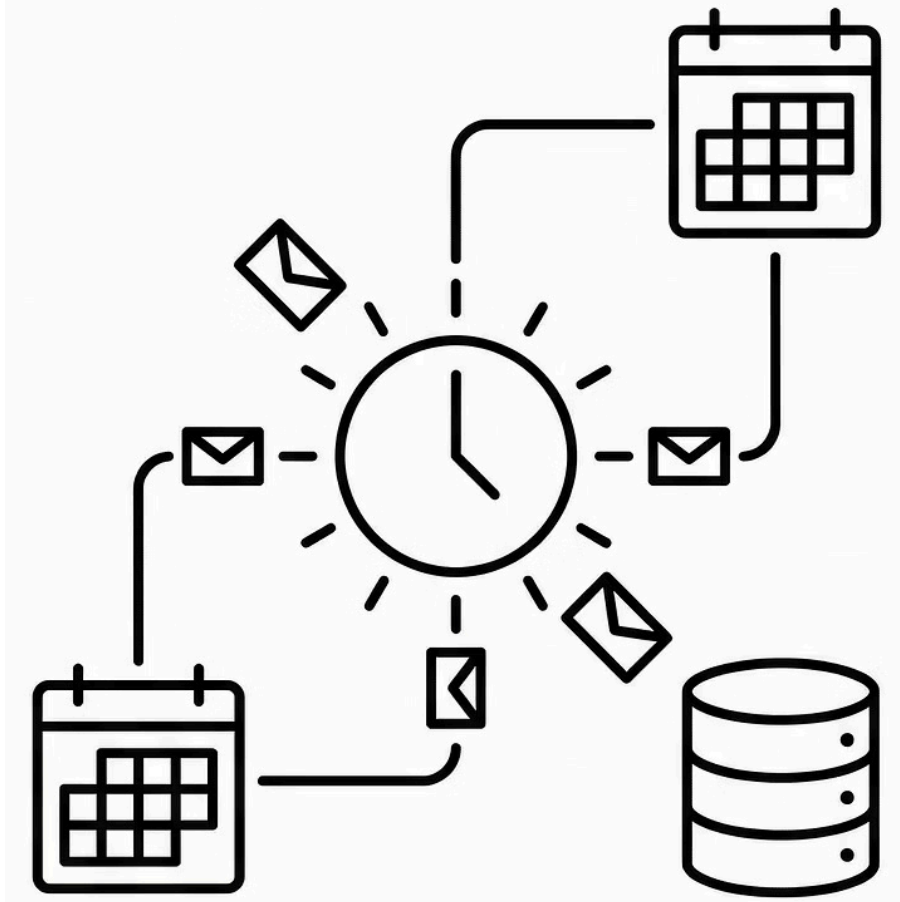
> Kein Lead geht mehr verloren — wer 'Später' klickt, bekommt nach 7 Tagen automatisch eine Erinnerung.



PHASE 3 ERWEITERUNG

Follow-Up Cron — Kein Lead geht verloren

Der Cron-Job läuft automatisch jeden Tag um 8 Uhr und kümmert sich um alle Leads, die 'Später testen' gewählt haben — ohne manuellen Aufwand.



💡 Ulrich´s-Erklärung: Was ist ein Cron-Job?

Ein Cron-Job ist wie ein automatischer Wecker für den Server. Du sagst ihm: 'Mach das jeden Tag um 8 Uhr' — und er tut es, ohne dass jemand einen Knopf drücken muss. In diesem Fall schaut er jeden Morgen nach: Wer hat vor 7+ Tagen 'Später testen' geklickt und noch keine Erinnerung bekommen?

Täglich 8 Uhr

Vercel Cron startet api/check-followups.js automatisch

Leads prüfen

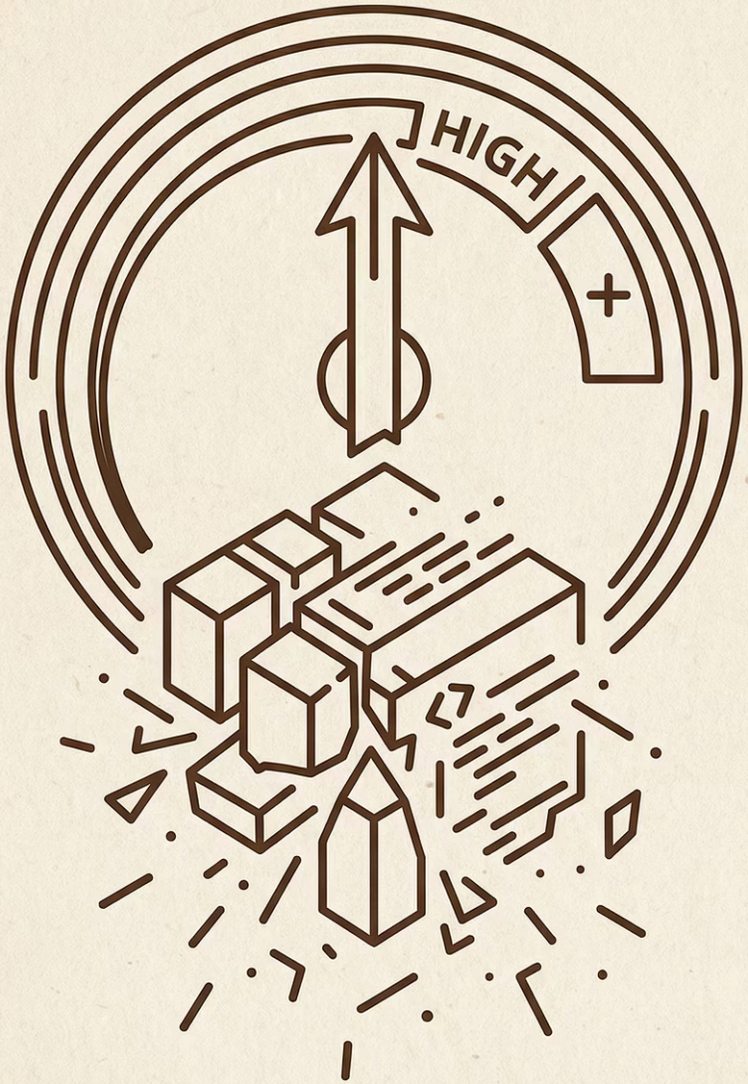
Sucht alle Leads mit pipeline_stage: testet_spaeter die 7+ Tage alt sind

E-Mail senden

Automatische Test-Erinnerung per E-Mail an den Lead

Admin-Notiz

Erstellt Admin-Notiz für überfällige Wiedervorlagen im Dashboard



PERFORMANCE

Code-Splitting — Schnellere Ladezeiten

Alle 12 Seiten werden jetzt lazy-loaded — der Haupt-Chunk schrumpfte von 512KB auf 380KB. Seiten laden nur wenn sie wirklich gebraucht werden.

💡 Ulrich´s-Erklärung: Was ist Code-Splitting / Lazy Loading?

Stell dir vor, du bestellst ein Buch — aber statt das ganze Buch auf einmal zu liefern, bekommst du nur das Kapitel, das du gerade liest. Lazy Loading macht genau das: Die App lädt nur den Code für die Seite, die du gerade öffnest. Das macht den ersten Ladevorgang viel schneller.

Vorher

- Haupt-Chunk: 512KB
- Alles auf einmal geladen
- Browser-Warning: Chunk zu groß
- Langsamer erster Ladevorgang

Nachher ✅

- Haupt-Chunk: 380KB (−26%)
- 17 separate Chunks
- Kein Warning mehr
- Seiten laden on-demand

> **Jede der 12 Seiten ist jetzt ein eigener Chunk** — der Browser lädt nur was er braucht.

⚠ ACTION REQUIRED ⚠️

Noch offen: Vercel Setup für Go-Live

Der Code ist fertig und gepusht. Bevor die Plattform live geht, müssen 4 Dinge in Vercel und Stripe konfiguriert werden — kein Code, nur Einstellungen.

💡 Ulrich´s-Erklärung
Was sind Env-Variablen?

Env-Variablen (Environment Variables) sind geheime Passwörter und Einstellungen, die der Server kennen muss — aber die nicht im Code stehen dürfen. Stell dir vor, du hast einen Tresor (Stripe, Supabase) und brauchst den Schlüssel. Der Schlüssel steht nicht im Code, sondern wird separat in Vercel hinterlegt.

SQL-Migrationen ausführen

Beide SQL-Dateien im Supabase SQL Editor ausführen: phase4-auth.sql + phase5-7-stripe-usage.sql

Env-Variablen in Vercel setzen

SUPABASE_SERVICE_KEY, STRIPE_SECRET_KEY, STRIPE_WEBHOOK_SECRET, CRON_SECRET, SEED_SECRET, APP_URL

Stripe Webhook einrichten

URL: https://sales.guidetranslator.com/api/stripe-webhook — Events: checkout, subscription, invoice

Stripe Price IDs eintragen

Aus dem Stripe Dashboard → Products → Price IDs in src/config/pricing.js eintragen

> Danach: Seed-Admin ausführen → Login testen → Admin → Accounts → Dashboard → Pricing → Checkout.