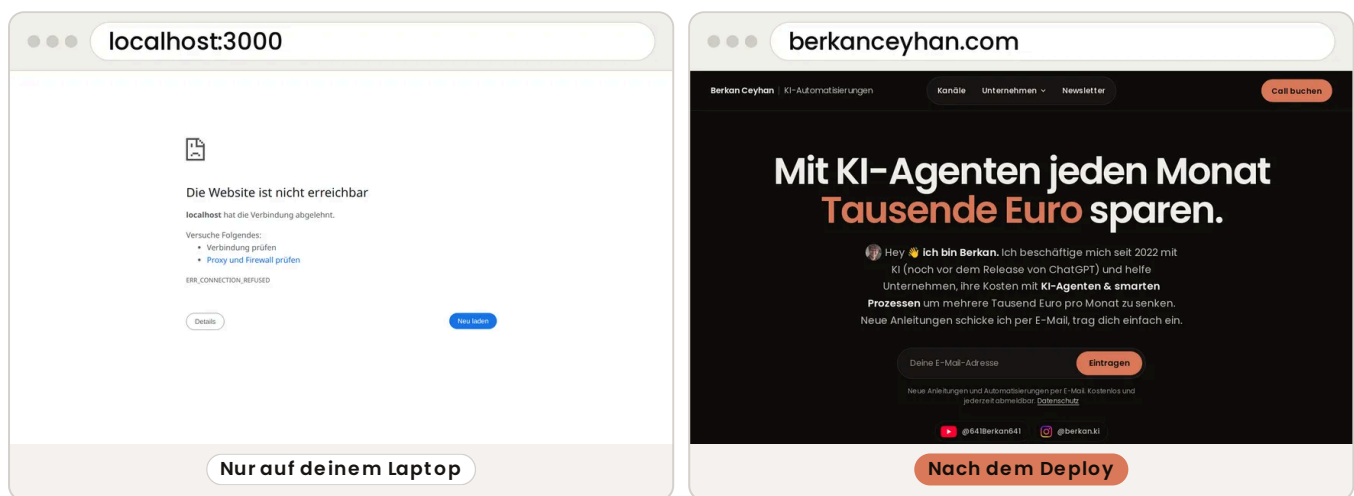


Lass Claude deine App online stellen, auf deiner eigenen Domain

Drei Prompts für Website, Web-App und Datenbank. Claude lädt dein Projekt hoch und verbindet deine Domain.



Berkan Ceyhan · Stand Oktober 2026



Links die Fehlerseite, die jeder andere sieht, wenn er dein localhost öffnet. Rechts berkanceyhan.com, von Claude gebaut und bei Vercel veröffentlicht.

AUF EINEN BLICK

- ✓ **Website** aus HTML, CSS, JS: GitHub Pages
- ✓ **Web-App** mit Login oder Daten: Vercel
- ✓ **Datenbank** dazu: Supabase
- ✓ **Ein Prompt** pro Weg, Claude macht den Rest



Website oder Web-App? Das entscheidet deinen Weg

- 1 Reine Website aus HTML, CSS und JavaScript: **GitHub Pages**
- 2 Anmeldung oder gespeicherte Daten: **Vercel**
- 3 Dazu eine Datenbank: **Supabase**, verknüpft mit Vercel

Du bist nicht sicher, was du gebaut hast? Frag Claude Code in deinem Projektordner:

Welcher Weg passt?

Sieh dir dieses Projekt an und sag mir in drei Sätzen:

- Ist das eine reine Website (nur HTML, CSS und JavaScript, ohne Server) oder eine Web-App (mit Anmeldung, Server-Code oder gespeicherten Daten)?
- Brauche ich eine Datenbank?
- Welchen Weg empfiehlst du: GitHub Pages, Vercel oder Vercel mit Supabase?

Ändere nichts.

📄 KURZ ERKLÄRT

Drei Dinge machst du selbst

Dich beim Anbieter anmelden, die DNS-Einträge bei deinem Domain-Anbieter anlegen und, falls nötig, bezahlen. Claude sagt dir jedes Mal genau, was du eintragen oder ausführen musst.

🔗 Website: Mit **GitHub Pages** online, ohne Server

Dein Projekt liegt auf GitHub, und Pages macht daraus eine Website. Das ist kostenlos für öffentliche Repositories.

Auf GitHub Pages veröffentlichen

Stell dieses Projekt als Website auf GitHub Pages online. Es ist eine reine Website aus HTML, CSS und JavaScript, ohne Server.

Prüf zuerst:

1. Liegen Schlüssel, Passwörter oder eine .env im Projekt? Dann halt an und sag mir Bescheid, denn das Repository wird öffentlich.
2. Ist die GitHub-CLI installiert und angemeldet (gh --version, gh auth status)? Wenn nicht, sag mir den Befehl, den ich selbst im Terminal ausführen muss, und warte.

Dann:

3. Leg ein öffentliches Repository an, committe alles und pushe auf main.
4. Aktiviere GitHub Pages für den Branch main (Ordner / oder /docs, je nachdem, wo die index.html liegt).
5. Meine Domain ist [meine-domain.de]. Trag sie in den Pages-Einstellungen ein und aktiviere HTTPS. Sag mir, welche DNS-Einträge ich bei meinem Domain-Anbieter anlegen muss (Apex-Domain und www). Nimm die Werte aus der aktuellen GitHub-Doku. Wenn HTTPS noch nicht geht, sag mir, wann ich es nachholen muss.
6. Prüf am Ende, ob die Seite unter der github.io-Adresse antwortet, und nenn mir die Adresse.

⚠️ ACHTUNG

Dein Code ist dann öffentlich

Im kostenlosen GitHub-Konto braucht Pages ein öffentliches Repository, jeder kann den Code lesen. Leg dort nie Schlüssel oder Passwörter ab. Für Shops und Bezahlseiten ist Pages laut GitHub nicht gedacht.

▲ Web-App: Mit **Vercel** online, inklusive Domain

Du legst einen kostenlosen Account an, Claude lädt dein Projekt hoch und verbindet die Domain.

Auf Vercel veröffentlichen

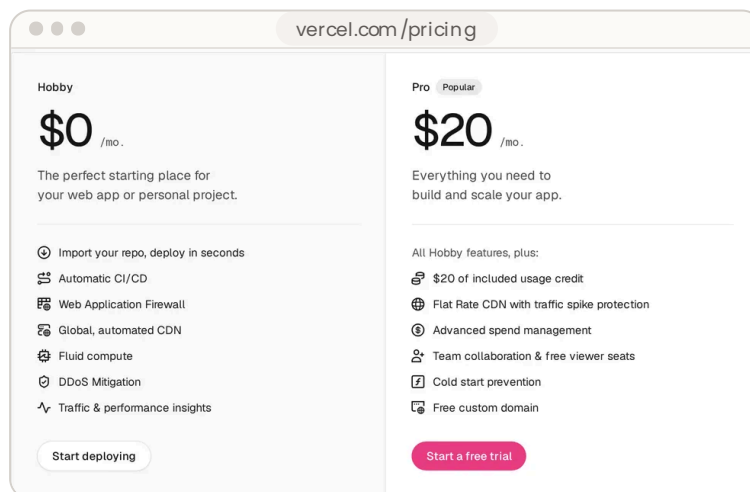
Stell dieses Projekt auf Vercel online und verbinde meine Domain [meine-domain.de].

Prüf zuerst:

1. Welche Art Projekt ist das (Framework, Build-Befehl, Ausgabeordner)? Sag es mir in einem Satz.
2. Liegen Schlüssel oder Zugangsdaten im Code oder in einer .env, die ins Repository kämen? Dann halt an. Die gehören als Umgebungsvariablen zu Vercel, nie ins Repository.
3. Ist die Vercel-CLI installiert (vercel --version)? Wenn nicht, installier sie. Bin ich angemeldet (vercel whoami)? Wenn nicht, sag mir den Befehl, den ich selbst ausführen muss, und warte.

Dann:

4. Verknüpf das Projekt (vercel link) und stell es online (vercel --prod).
5. Leg die Umgebungsvariablen aus meiner .env bei Vercel an (vercel env add, für Production). Zeig mir nur die Namen, nie die Werte.
6. Füg die Domain hinzu (vercel domains add) und sag mir, welche DNS-Einträge ich bei meinem Domain-Anbieter anlegen muss. Prüf danach mit vercel domains verify.
7. Öffne die Produktions-Adresse, prüf, ob die Seite lädt, und nenn mir die Adresse.



Hobby kostet nichts, Pro 20 \$ im Monat. Stand Oktober 2026, ohne Steuern.

⚠ ACHTUNG

Kostenlos gilt nur für nicht-kommerzielle Projekte

Der Hobby-Plan ist für persönliche Projekte. Sobald die Seite Geld bringt, brauchst du Pro für 20 \$ pro Entwickler und Monat. Das gilt schon, wenn du für etwas wirbst, das du verkaufst, oder wenn jemand für den Bau bezahlt wird.



Datenbank: Mit Supabase Daten speichern

Leg das Supabase-Projekt in Vercel als Storage-Ressource an. Dann stehen die Zugangsdaten automatisch in deinem Vercel-Projekt. Der kostenlose Plan erlaubt 2 aktive Projekte und pausiert sie nach einer Woche ohne Aktivität, Pro startet bei 25 \$ im Monat (Stand Oktober 2026).

Supabase anbinden

Verbinde dieses Projekt mit meiner Supabase-Datenbank. Das Projekt läuft auf Vercel, das Supabase-Projekt habe ich dort angelegt.

Plan zuerst:

1. Welche Daten speichert die App? Schlag mir Tabellen mit Spalten vor und warte auf mein Okay.

Dann:

2. Ist die Supabase-CLI da? Wenn nicht, installier sie im Projekt (`npm install supabase --save-dev`). Bin ich angemeldet und ist das Projekt verknüpft (`supabase login`, `supabase link`)? Wenn nicht, sag mir, was ich selbst ausführen muss, und warte.

3. Schreib die Tabellen als Migration. Schalte Row Level Security für jede Tabelle ein und schreib Regeln, nach denen jede Person nur ihre eigenen Zeilen sieht.

4. Zeig mir `supabase db push --dry-run`. Nach meinem Okay wende die Migration an.

5. Im Browser-Code darf nur der Publishable Key stehen. Der Secret Key bleibt auf dem Server. Prüf das im ganzen Code.

6. Prüf, dass die Supabase-Variablen bei Vercel stehen (nur Namen zeigen, nie Werte). Stell neu online (`vercel --prod`) und teste: Eintrag anlegen, Eintrag lesen.

⚠ ACHTUNG

Nur der Publishable Key gehört in den Browser

Der Secret Key umgeht alle Zugriffsregeln und darf nur auf dem Server liegen. Schalte Row Level Security für jede Tabelle ein, bevor du online gehst.



Automatisierungs- Call für Agenturen, Coaches und Dienstleister

In 30 Minuten schauen wir gemeinsam, wo KI-Agenten und automatisierte Prozesse in deinem Betrieb zuerst Zeit und Geld sparen. Kostenlos.

FÜR WEN IST DAS?

- ✓ Agenturen, Coaches und Dienstleister
- ✓ ab 10.000 € Monatsumsatz
- ✓ ein klarer Flaschenhals im Tagesgeschäft
- ✓ du willst KI im Betrieb einsetzen

Call buchen

[calendly.com/
berkanceyhan/
automatisierungscall](https://calendly.com/berkanceyhan/automatisierungscall)



Scannen & Termin wählen