

Claude Code MVP Checkliste: Starte dein erstes Produkt in 48 Stunden

Der Non-Coder Guide: Von der Idee zum Live-Produkt — Keine Tutorials, kein Overwhelm

Von Julian, codecoast labs GmbH — Praktiker, der täglich mit Claude shipped

[Visual: 48-Stunden-Timeline mit Rakete am Ende]

Kostenloser Download per E-Mail. Nutze die 20% von Claude, die 80% der Ergebnisse bringen.

Warum du feststeckst — Und wie das hier alles ändert

Du hast die Idee. Vielleicht lebt sie seit Monaten in deiner Notizen-App. Du hast die Tutorials geschaut, die Kurse gebookmarked und die "How I Built X"-Threads gelesen.

Aber jedes Wochenende endet gleich: halb-fertige Projekte, verwirrende Fehlermeldungen und dieses sinkende Gefühl von "Ich bin einfach nicht technisch genug."

VORHER

- ✗ Vergeudete Wochenenden mit Tutorials
- ✗ "Noch ein Kurs"-Syndrom
- ✗ Angst vor dem leeren Editor
- ✗ Fehlermeldungen = Panik
- ✗ Ideen bleiben im Kopf

NACH DIESER CHECKLISTE

- ✓ Funktionierendes Produkt in 48 Stunden
- ✓ Copy-Paste Prompts, die funktionieren
- ✓ Claude erklärt alles
- ✓ Fehler werden zu Lernmomenten
- ✓ Ideen werden zu shipped Produkten

Meine Geschichte

Ich bin kein "echter" Entwickler. Kein Informatik-Studium. Kein Bootcamp-Zertifikat. Aber ich shippe ständig Produkte — Landing Pages, Web Apps, Automatisierungen — alles mit Claude Code als meinem Co-Piloten.

Diese Checkliste ist genau das, was ich mir gewünscht hätte, als ich anfing. Keine Theorie. Keine "Best Practices." Nur ein zeitgesteuerter, Schritt-für-Schritt-Weg von der Idee zum shipped MVP.



Du musst kein Entwickler werden. Du musst nur EINE Sache shippen.

Willst du geführte Unterstützung?

Tritt der Launchpad-Kohorte bei für Live-Hilfe und Accountability

Mehr über Launchpad →

Start

25% Fertig

Dein einziges Ziel heute Morgen: Claude Code zum Laufen bringen und beweisen, dass es funktioniert. Das ist alles. Noch kein Bauen.

15 min Installiere Claude Code von claude.ai/code (folge den Installer-Prompts)

5 min Öffne Claude Code im Terminal und überprüfe, dass es startet

10 min Teste deinen ersten Prompt: "Erkläre mir wie für einen 5-Jährigen: Was ist eine Web App?"

20 min Erstelle einen Projektordner auf deinem Computer (z.B. ~/Projects/my-mvp/)

15 min Öffne Claude Code in diesem Ordner und frage: "Hilf mir, hier ein Basisprojekt aufzusetzen"

30 min Baue etwas Winziges: "Erstelle eine Webseite, die Hello World sagt"

DEIN ERSTER ECHTER PROMPT

[Kopiere & füge das ein](#)

Erkläre mir wie für einen 5-Jährigen: Ich möchte eine einfache Todo-App bauen, in der ich Aufgaben hinzufügen und als erledigt markieren kann. Welche Teile muss ich bauen? Schreib noch keinen Code – erkläre nur die Teile.



Tipp: Ignoriere alle erweiterten Features, die du siehst. Du brauchst heute keine Agents, Plugins oder MCP Server. Nur die Basics.

Stunden 13-24

Core Building

Samstag Nachmittag & Abend

Foundation ✓

50% Fertig

Jetzt bauen wir. Der Schlüssel: Alles in winzige Stücke aufteilen und das "Erkläre wie für einen 5-Jährigen" (ELI5) Muster für jeden Prompt nutzen.

20 min Schreib die 3 Kern-Features deines MVPs auf (nicht mehr als 3!)

30 min Bitte Claude, Feature #1 mit dem ELI5-Prompt-Muster zu bauen

30 min Teste Feature #1 — funktioniert es? Wenn nicht, paste den Fehler zu Claude

30 min Baue Feature #2 auf die gleiche Weise

30 min Baue Feature #3 auf die gleiche Weise

30 min Bitte Claude, alle Features miteinander zu verbinden

DER ELI5 BUILDER PROMPT

Nutze dieses Muster für jedes Feature

Ich bin ein kompletter Anfänger. Ich brauche, dass du [FEATURE] für meine [ART VON APP] baust.

Bitte:

1. Erkläre in einfachen Worten, was du gleich tust

2. Schreibe den Code mit vielen Kommentaren, die jeden Teil erklären

3. Sag mir genau, wo ich diesen Code hinpicken soll

4. Zeig mir, wie ich testen kann, ob es funktioniert

Mein aktuelles Projekt hat: [BESCHREIBE, WAS DU BISHER HAST]

[Diagramm: Prompt → Claude schreibt Code → Du testest → Funktioniert? Ship it. Kaputt?
Paste Fehler zu Claude → Wiederholen]



Tipp: "Gut genug" schlägt "perfekt" jedes Mal. Du kannst es nächstes Wochenende verbessern. Heute shippen wir.

Core gebaut ✓

75% Fertig

Du hast funktionierende Features. Jetzt räumen wir die rauen Kanten auf und stellen sicher, dass nichts kaputt geht.

- 30 min Geh durch deine App wie ein User — notiere alles Verwirrende oder Kaputte
- 30 min Bitte Claude, jedes Problem zu fixen, eins nach dem anderen
- 45 min Bitte Claude, die UI zu verbessern: "Mach das professioneller aussehen"
- 30 min Teste Edge Cases: leere Eingaben, komische Zeichen, Doppelklicks
- 15 min Bitte Claude, einfaches Error Handling hinzuzufügen

Das "Steckst du fest?" Framework

< 15 Minuten festgesteckt

Formuliere deinen Prompt um. Sei spezifischer darüber, was falsch ist.

15-30 Minuten festgesteckt

Kopiere die exakte Fehlermeldung. Paste sie zu Claude und frage "Warum passiert das?"

30-60 Minuten festgesteckt

Bitte Claude, "diesen Code Zeile für Zeile zu erklären" — Verstehen schlägt Raten.

> 60 Minuten festgesteckt

Überspringen. Weitermachen. Frag die Community (@codecoast) oder heb es fürs nächste Wochenende auf.



Tipp: Denk an Claude als deinen Junior Developer. Es ist schnell, macht aber Fehler. Dein Job: reviewen, testen und klärende Fragen stellen.

Stunden 37-48

Ship & Iterate

Sonntag Nachmittag

Poliert ✓

100% — SHIP IT!

Das ist es. Zeit, deine Kreation ins Internet zu stellen, wo echte Menschen sie nutzen können.

30 min Frag Claude, wie du deine spezifische App deployen kannst (siehe Prompt unten)

45 min Folge den Deployment-Schritten — frag Claude bei Fehlern

15 min Besuche deine Live-URL. Mach einen Screenshot. Du hast es geschafft.

15 min Teile es irgendwo: Twitter, LinkedIn, einem Freund, oder schick es mir!

30 min Notiere 3 Dinge, die du beim nächsten Mal verbessern würdest (fürs nächste Wochenende)

DEPLOYMENT PROMPT (DOCKER + VPS)

Nutze genau diesen Prompt

Ich muss diese App auf meinem VPS mit Docker deployen.

Mein Setup:

- Ich habe einen VPS mit installiertem Docker
- Ich kann per SSH auf meinen Server zugreifen
- Die App ist ein [BESCHREIBE DEINEN APP-TYP: React, Node, etc.]

Bitte:

1. Erstelle ein Dockerfile für dieses Projekt

2. Erkläre jede Zeile des Dockerfiles einfach

3. Gib mir die exakten Befehle zum Bauen und Starten

4. Zeig mir, wie ich einen Reverse Proxy (nginx) aufsetze, falls nötig

Ich bin Anfänger – erkläre alles wie für einen 5-Jährigen.

```
# Beispiel Docker Deployment Befehle docker build -t my-mvp . docker run -d -p 3000:3000 --name my-app my-mvp
```



Du hast shipped. Das ist mehr als 99% der Leute mit "App-Ideen."

Das Erste ist immer das Schwerste. Es wird leichter von hier an.

Die Essentials: 5 Features + 3 Plugins

Du musst nicht alles wissen, was Claude Code kann. Hier sind die 20%, die 80% der Ergebnisse bringen.

5 Features, die du wirklich nutzen wirst



Chat Modus

Fragen stellen, Erklärungen bekommen



Dateien schreiben

Claude erstellt Code-Dateien für dich



Dateien bearbeiten

Claude modifiziert existierenden Code



Dateien lesen

Claude analysiert deinen Code



Befehle ausführen

Claude führt Terminal-Befehle aus

3 Plugins, die sich lohnen

1. Error Explainer

Erklärt automatisch Fehlermeldungen in einfachem Deutsch und schlägt Fixes vor. Perfekt für kryptische Fehler.

2. Code Sanity Checker

Prüft deinen Code auf häufige Fehler, bevor sie zu Bugs werden. Wie eine Rechtschreibprüfung für Code.

3. Quick Docs

Generiert einfache Dokumentation für deinen Code. Hilft dir später zu erinnern, was du gebaut hast.



Wochenend-MVP-Blueprint (One-Pager)

Samstag

- VM: Installieren + erster Prompt
- NM: 3 Kern-Features bauen
- Abend: Features verbinden

Sonntag

- VM: Testen + UI polieren
- NM: Auf Server deployen
- Abend: Teilen + feiern!

Wenn Dinge schief gehen (und das werden sie)

Jeder Builder trifft auf Wände. So kommst du raus, ohne in eine Spirale zu geraten.

Der Debug-Entscheidungsbaum

Fehlermeldung? → Kopiere den exakten Fehler → Paste zu Claude → "Warum passiert das und wie fixe ich es?"

Nichts passiert? → Frag Claude: "Führe mich Schritt für Schritt durch, was dieser Code macht"

Funktioniert lokal, bricht beim Deploy? → Frag Claude: "Welche Umgebungsunterschiede könnten das verursachen?"

Keine Ahnung, was falsch ist? → Fang bei diesem Feature von vorne an. Manchmal ist ein frischer Ansatz schneller.

Komplett festgesteckt? → Überspringen. Weitermachen. Ship ohne dieses Feature.

Mindset-Shifts

"Claude ist keine Magie — es ist ein Multiplikator. Es multipliziert deine Fragen in Antworten, deine Ideen in Code. Aber du musst trotzdem die richtigen Fragen stellen."

"Shipped schlägt perfekt. Eine live-geschaltete hässliche App lehrt dich mehr als ein schönes Konzept in deiner Notizen-App."

"Fehler sind keine Misserfolge. Sie sind nur Claudes Art zu sagen 'Ich brauche mehr Informationen.'"

Join the Community

Bei etwas Spezifischem hängen geblieben? Fragen zu deinem MVP?

[Finde mich auf X/Twitter](#)

Du hast shipped — Jetzt skalieren



Glückwunsch. Du hast Claude Code freigeschaltet.

Du bist von Idee → shipped Produkt in 48 Stunden gegangen.

Was du gelernt hast

- Claude Code ist nicht gruselig — es ist ein geduldiger Mit-BUILDER
- Das ELI5-Prompt-Muster funktioniert für jedes Feature
- "Gut genug" heute schlägt "perfekt" niemals
- Fehler sind nur Konversationen, die darauf warten zu passieren
- Du kannst shippen. Das hast du gerade bewiesen.

Was kommt als Nächstes?

Jetzt, wo du ein MVP shipped hast, hast du Optionen:

1. **Iterieren** — Nimm User-Feedback und verbessere dieses Wochenende
2. **Nochmal shippen** — Baue eine andere Idee (du weißt jetzt wie)
3. **Tiefer gehen** — Lerne die erweiterten Features (Agents, Plugins, MCP)

Tritt der Launchpad-Kohorte bei

8 Wochen geführtes Bauen mit Live-Support

- Wöchentliche Live-Sessions mit Q&A
- Private Community von Buildern
- Ship 3 echte Produkte bis zum Ende
- Direktes Feedback auf deine Builds

€397 — Inklusive lebenslangem Community-Zugang

[Launchpad beitreten →](#)

Kostenloser Bonus

Schick mir deinen MVP-Link per E-Mail (julian@codecoast.ch) und ich sende dir personalisiertes Feedback, was du verbessern kannst. Echte Menschen bauen echte Dinge — darum geht es hier.

codecoast labs GmbH

julian@codecoast.ch | @codecoast

codecoast.ch
