

# Lastenheft: Social-Media-Video-Prompt-Generator

## 1. Einleitung

### 1.1 Ziel des Dokuments

Dieses Lastenheft beschreibt die Anforderungen und Rahmenbedingungen für die Entwicklung einer Web-Plattform, die Content-Creators und Unternehmen ermöglicht, maßgeschneiderte Prompts für Social-Media-Videos (animierte Videos und fortlaufende Serien) zu generieren.

### 1.2 Projektname

Social-Media-Video-Prompt-Generator

### 1.3 Projektauftraggeber

Trainer / Betreuer des Projekts

### 1.4 Projektumfeld

- Hosting und Verwaltung über GitHub
- No-Code-/Low-Code-Prototyping (Replit) bis MVP
- KI-Integration mittels OpenAI GPT API

## 2. Ist-Zustand

Aktuell existiert lediglich das Konzept und ein Prototyp-Mockup zur Benutzeroberfläche. Es gibt keine lauffähige Anwendung.

## 3. Soll-Zustand

Eine Online-Plattform mit Frontend und Backend, die folgende Funktionen im MVP-Zustand erfüllt: - Eingabemaske für Kategorie und Schlagworte - Backend-Service, der per API drei Prompts generiert - Anzeige der generierten Prompts im Frontend - Deployment über GitHub Actions auf Vercel/Heroku/Netlify

## 4. Zielsetzung und Abgrenzung

### 4.1 Projektziele

- Schnelle Generierung relevanter Video-Prompts
- Einfache Bedienbarkeit
- Klare MVP-Grenze zur frühzeitigen Freigabe

## 4.2 Nicht-Ziele im MVP

- Nutzer-Authentifizierung und -Verwaltung
- Persistente Speicherung von Prompts
- Ausgereiftes UI/UX-Design
- CI/CD jenseits automatischem Deployment

## 5. Stakeholder

Rolle	Verantwortlichkeit
Projektleiter (Du)	Gesamtkoordination, GitHub-Organisation
Trainer/Betreuer	Fachliche Abnahme, Feedback
Entwickler-Team	Umsetzung Frontend, Backend, Deployment
Endanwender	Content-Creators, Social-Media-Manager

## 6. Fachliche Anforderungen

### 6.1 Eingabeparameter

- Kategorie (z. B. Tutorial, Produkt-Demo, Storytelling)
- Schlagworte/Thema (Freitext)

### 6.2 KI-Generierung

- Schnittstelle zu OpenAI GPT
- Parameter: Anzahl der Prompts = 3, Maximale Länge

### 6.3 Ausgabe

- Darstellung von zwei klar formulierten Prompt-Vorschlägen
- Option zum Kopieren in die Zwischenablage

### 6.4 Projektmanagement

- Repository-Struktur: `/api`, `/web`, `/docs`
- Branch-Strategie: `main` (protected), `develop`, `feature/*`
- Issues und Kanban-Board in GitHub Projects

## 7. Nicht-funktionale Anforderungen

Anforderung	Beschreibung
Performanz	Antwortzeit < 2 Sekunden
Verfügbarkeit	99,5% (MVP-Stufe)
Usability	Einfache, selbsterklärende UI
Wartbarkeit	Saubere Code-Struktur, Dokumentation ( <code>README</code> )

Anforderung	Beschreibung
Skalierbarkeit	Trennung von Front- und Backend für spätere Skalierung
Sicherheit	HTTPS, sichere Aufbewahrung von API-Schlüsseln

## 8. Technische Rahmenbedingungen

- **Frontend:** React (Vite/Create React App) im Ordner `/web`
- **Backend:** Node.js/Express (Alternativ Python/FastAPI) im Ordner `/api`
- **CI/CD:** GitHub Actions für automatisches Deployment
- **Hosting:** Vercel oder Netlify für Frontend; Heroku/Vercel für Backend
- **API-Keys:** Verwaltung über GitHub Secrets

## 9. MVP-Plan (Minimum Viable Product)

1. **Backend-Implementierung** ( `POST /generate-prompts` )
2. **Frontend-Formular** (Kategorie + Schlagworte)
3. **KI-API-Integration** (OpenAI GPT)
4. **Anzeige der Ergebnisse** im Frontend
5. **Deployment** via GitHub Actions

## 10. Zeit- und Meilensteinplanung

Meilenstein	Datum
Kickoff & Repo-Setup	KW 29 (Juli)
Backend MVP fertig	Ende KW 29
Frontend MVP fertig	Ende KW 30
Integration & Testing	Ende KW 30
Deployment & Abnahme	Mitte KW 31

## 11. Abnahmekriterien

- Funktion `POST /generate-prompts` liefert valide Antworten
- Frontend zeigt Prompts korrekt an
- Automatisches Deployment bei Merge in `main`
- Dokumentation in `README.md` und `/docs`

## 12. Glossar

- **MVP:** Minimum Viable Product
- **CI/CD:** Continuous Integration / Continuous Deployment
- **HTTP:** Hypertext Transfer Protocol
- **API:** Application Programming Interface

*Version: 1.0*