

Projektkonzept: Planify – Webapp für Kalender & Aufgabenplaner

1. Projektbeschreibung

Planify ist eine lokal einsetzbare Webanwendung zur Verwaltung von Terminen und Aufgaben. Die App kombiniert Kalender- und ToDo-Listen-Funktionen, ist benutzerfreundlich gestaltet und wird mittels Docker-Containern betrieben, um einfache Installation und Portabilität zu gewährleisten.

2. Zielsetzung

- Entwicklung einer lokalen Webapp mit Kalender- und Aufgabenmanagement
- Containerisierung aller Komponenten mit Docker
- Offline-Fähigkeit ohne Cloud-Anbindung
- Saubere Trennung von Frontend, Backend und Datenbank
- Erweiterbar für Benutzerverwaltung und Erinnerungsfunktionen

3. Systemarchitektur

- **Frontend:** React, responsives Design, Kommuniziert via REST API
- **Backend:** FastAPI, Python, API-Server mit Datenlogik
- **Datenbank:** PostgreSQL, persistente Speicherung
- **Zusatz:** Adminer als Datenbankverwaltungstool
- Alle Dienste laufen in eigenen Docker-Containern, orchestriert mit docker-compose.

4. Funktionale Anforderungen

- Termine erstellen, bearbeiten, löschen, mit Datum und Beschreibung
- Aufgaben anlegen, priorisieren, erledigen und terminieren
- Tages- und Wochenansicht des Kalenders
- (Optional) Benutzerregistrierung und Login mit JWT
- (Optional) Erinnerungen per lokalem Mailserver (Mailhog)

5. Technische Umsetzung

- **Frontend:** React + TailwindCSS, React Router, Axios
- **Backend:** FastAPI, Pydantic, SQLAlchemy, JWT Auth
- **DB:** PostgreSQL mit Adminer für Web-GUI
- **Containerisierung:** Dockerfiles für Frontend & Backend, docker-compose für Gesamtprojekt

6. Entwicklungsplan

Phase	Aufgaben	Zeitraumen
Planung	Architektur, Konzept, Tech-Stack definieren	1 Woche
Backend	API-Endpunkte, Datenmodell, Auth	2 Wochen
Frontend	UI-Design, Kalender- und Aufgaben-Views	2 Wochen
Integration	Docker-Setup, Zusammenspiel aller Container	1 Woche
Test & Doku	Tests, Fehlerbehebung, Projektdokumentation	1 Woche

7. Vorteile & Nutzen

- Einfaches Setup und Betrieb lokal ohne Internet
- Modularer Aufbau erleichtert Weiterentwicklung
- Praktisches Beispiel für moderne Webentwicklung mit Containern
- Gute Lernbasis für Teamarbeit, Deployment und DevOps