Projektplan: DailyHacker – Smarter Health & Life Tracker

# 1. Einleitung

Ziel des Projekts ist die Entwicklung einer Web-Anwendung zur täglichen Erfassung von Gesundheits- und Lifestyle-Daten. Nutzer sollen ihre Schritte, Schlafdauer, Supplements (inkl. mg-Angabe), Trainingseinheiten sowie externe Daten wie Wetter und Mondphase dokumentieren können. Die Anwendung wird mit Java Spring Boot im Backend, einer relationalen Datenbank (PostgreSQL), und einem React-Frontend realisiert.

# 2. Funktionsübersicht

- Nutzer kann tägliche Daten eingeben: Schritte, Schlaf, Training, Supplements  
- Supplements werden inkl. mg-Angabe erfasst  
- Wetterdaten (Nacht) und Mondphase werden automatisch via API ergänzt  
- Daten sind als Verlauf abrufbar  
- Optional: Statistikmodul für Analyse über Zeiträume  
- Optional: Authentifizierung bei mehreren Nutzern  
- Optional: Schritt-Import von iPhone Health API in Zukunft

# 3. Technologiestack

- Backend: Java mit Spring Boot (REST-API)  
- Datenbank: PostgreSQL  
- Frontend: React (mobilfähig)  
- APIs: OpenWeather, Mondphase-API  
- Optional später: Swift/iOS-App für automatische Datenerfassung

# 4. Datenmodell (vereinfacht)

Tabelle: User  
- id, name  
  
Tabelle: DailyEntry  
- id, user\_id, datum, schritte, schlaf\_minuten, wetter\_temp, mondphase  
  
Tabelle: SupplementEntry  
- id, daily\_entry\_id, supplement\_name, menge\_mg, genommen  
  
Tabelle: TrainingEntry (optional)  
- id, daily\_entry\_id, typ, intensität, notiz

# 5. Projektphasen & Vorgehen

Phase 1 – Planung:  
- Datenmodell entwerfen  
- API-Schnittstellen skizzieren  
- Anforderungen definieren  
  
Phase 2 – Backend-Umsetzung:  
- Spring Boot Projekt erstellen  
- Entity-Klassen + Repository + Controller anlegen  
- Datenbankanbindung (PostgreSQL oder H2 lokal)  
- CRUD-Endpunkte für DailyEntry, Supplements etc.  
- API-Tests mit Postman  
  
Phase 3 – API-Integration:  
- Wetter-API anbinden  
- Mondphasen-API integrieren  
- Daten automatisch ergänzen beim Tages-Eintrag  
  
Phase 4 – Frontend-Umsetzung:  
- React-Projekt starten (z.B. mit Vite)  
- Formular für Tagesdaten bauen  
- REST-Aufrufe ans Backend  
- Verlauf & Übersichtskomponente erstellen  
  
Phase 5 – Optional & Erweiterung:  
- Authentifizierung (JWT)  
- Mobile App / iOS-Schnittstelle vorbereiten  
- Statistikmodul (Diagramme, Reports)

# 6. Ziel

Am Ende soll eine voll funktionsfähige Web-App entstehen, die dem Nutzer eine tägliche Übersicht seiner Gesundheitsdaten gibt und langfristig Trends erkennen lässt. Das Projekt eignet sich als praktischer Beweis für Backend- und API-Kompetenz mit Java Spring Boot und zeigt gleichzeitig einen persönlichen Anwendungsfall.