**🧠 Projektidee: "FinCrypto – Dein persönliches Finanz- & Krypto-Dashboard"**

**📌 Ziel:**

Erstelle eine Web-App, mit der Nutzer sowohl ihre **persönlichen Finanzen verwalten** als auch eine **individuelle Kryptowährungs-Watchlist** führen können. Die App liefert Live-Daten zu Coins und Wechselkursen, erlaubt Notizen und analysiert Einnahmen und Ausgaben über Zeit.

**🛠️ Verwendete Technologien:**

* **Frontend:** Vue.js mit Chart-Komponenten (z. B. Chart.js)
* **Backend:** Java Spring Boot
* **Datenbank:** MySQL oder H2, JPA mit mindestens zwei Entitäten pro Modul
* **Externe REST-APIs:**
  + CoinGecko API (für Kryptowährungen)
  + [ExchangeRate-API](https://exchangerate.host) (für Wechselkurse)

**📦 Datenmodell (Entitäten & Beziehungen):**

**🪙 Krypto-Modul:**

* CryptoEntry: Name, Symbol, aktueller Preis, Zeitstempel, gespeicherter Kursverlauf
* UserNote: Text, Bezug auf CryptoEntry
* Beziehung: **One-to-Many** (ein Eintrag hat mehrere Notizen)

**💰 Finanz-Modul:**

* Transaction: Betrag, Datum, Beschreibung, Kategorie, Währung
* Category: Name (z. B. "Lebensmittel", "Freizeit", "Miete")
* Beziehung: **Many-to-One** (Transaktionen gehören zu einer Kategorie)

**🔗 API-Integration:**

* **CoinGecko:** Aktuelle Krypto-Preise, Kursverläufe (für Diagramme)
* **ExchangeRate-API:** Umrechnung von Transaktionen in verschiedene Währungen oder Visualisierung von Wertentwicklungen

**🎨 Frontend-Funktionalität:**

* Krypto-Übersicht mit Preisdiagrammen (letzte 7 Tage)
* Notizen zu jedem Coin
* Finanzübersicht mit Filter nach Kategorie, Monat, Währung
* Diagramm (z. B. Tortendiagramm für Kategorien, Linienchart für Ausgabenverlauf)

**📋 Projektplanung & Dokumentation:**

* **Mindmap oder Kanban-Board:** Aufteilung in Krypto-Teil, Finanz-Teil, API-Integration, Frontend
* **Datenbankdiagramm:** Zwei Teilbereiche: Crypto & Finanzen
* **Kommentierter Code:** Saubere Trennung von Modulen, Repositories, DTOs

**✨ Bonus-Ideen (für Zusatzpunkte):**

* Wechselkurs-Auswahl: Automatische Umrechnung von Transaktionen in EUR/USD/BTC
* "Top-Gainer"-Liste aus CoinGecko (Tages- oder Wochengewinner)
* Export der Finanzdaten (CSV)
* Dark Mode fürs Dashboard

**Fazit:**  
Diese App kombiniert sinnvoll **persönliche Finanzplanung** mit dem aktuellen Trendthema **Kryptowährungen** und bietet sowohl technisches als auch gestalterisches Potenzial für eine herausragende Abschlussarbeit.

Möchtest du eine Vorlage für die **Datenbankstruktur** oder ein **Mockup für das UI-Layout**?

**Projektbeschreibung: Finazmanager – Dein persönliches Finanz- & Krypto-Dashboard**

**Ziel des Projekts:**  
Das Ziel dieser Anwendung ist es, eine kombinierte Web-App zu entwickeln, mit der Nutzer ihre persönlichen Finanzen verwalten und gleichzeitig eine individuelle Kryptowährungs-Watchlist führen können. Die Anwendung verbindet klassische Haushaltsbuch-Funktionalitäten mit aktuellen Daten aus dem Kryptomarkt und bietet einen Überblick über die finanzielle Lage des Nutzers sowie über Entwicklungen bei Kryptowährungen.

**Funktionsweise der Anwendung:**  
Die Web-App besteht aus zwei Hauptmodulen: einem Finanzmodul und einem Krypto-Modul.  
Im Finanzmodul können Benutzer Einnahmen und Ausgaben erfassen, bestimmten Kategorien wie "Miete", "Freizeit" oder "Essen" zuordnen. Zusätzlich ermöglicht die App die Anzeige aktueller Wechselkurse, um Ausgaben in verschiedenen Währungen zu vergleichen.

Im Krypto-Modul kann der Nutzer Kryptowährungen zur persönlichen Watchlist hinzufügen. Die App ruft dafür aktuelle Preisdaten von der CoinGecko-API ab und zeigt Preisverläufe der letzten Tage als Diagramm an. Zu jedem Krypto-Eintrag können eigene Notizen hinterlegt werden, etwa um Investmententscheidungen zu dokumentieren.

**Verwendete Technologien:**

* **Frontend:** Vue.js mit Chart.js für die grafische Darstellung von Ausgaben und Kursverläufen
* **Backend:** Java Spring Boot
* **Datenbank:** MySQL oder H2, Zugriff über Spring Data JPA
* **Externe REST-APIs:**
  + CoinGecko API zur Abfrage von Kryptowährungsdaten
  + ExchangeRate API zur Umrechnung und Anzeige von Wechselkursen  
    Die Kommunikation zwischen Frontend und Backend erfolgt über HTTP mit JSON.

**Mögliche Herausforderungen:**  
Die größte Herausforderung besteht in der sinnvollen Integration von zwei verschiedenen APIs und der asynchronen Verarbeitung ihrer Daten. Zusätzlich ist das Zeitmanagement kritisch, da sowohl Backend als auch Frontend eine klare Trennung und saubere Architektur benötigen.