

## Projektübersicht

Dieses Projekt zielt darauf ab, eine umfassende Lernplattform zu entwickeln, die Neueinsteigern und Fortgeschrittenen den Kryptowährungshandel näherbringt. Die Plattform bietet eine Vielzahl an interaktiven Tools und Lernmaterialien, um das Lernen spannend und effektiv zu gestalten.

## Ausgangsidee

Die ursprüngliche Idee war eine Single-Page-Anwendung, die ein simuliertes Crypto-Wallet bietet, um den Umgang mit Kryptowährungen zu erlernen, ohne echtes Geld zu benutzen. Benutzer können verschiedene Kryptowährungen "kaufen", "verkaufen" und "verwalten".

## Ursprüngliche Hauptfunktionen:

**Lernen durch Handeln:** Spielerisches Lernen durch interaktive Transaktionen mit simulierten Kryptowährungen.

**Sicherheit und Risikofreiheit:** Kein echtes Geld erforderlich, daher keine finanziellen Risiken.

**Benutzerfreundlichkeit:** Einfache und intuitive Benutzeroberfläche.

**Realitätsnahes Erlebnis:** Authentische Simulation von Krypto-Wallets und -Transaktionen.

Diese ursprüngliche Idee erfüllte jedoch nicht vollständig die Anforderungen und wurde daher angepasst.

## Überarbeitete Projektidee

Die überarbeitete Idee fokussiert sich auf eine interaktive Lernplattform mit folgenden Hauptkomponenten:

**Quiz und Prüfungen:** Regelmäßige Tests zur Überprüfung des Wissensstandes.

**Lernblätter:** Strukturierte Lernmaterialien zu spezifischen Themen.

**Simulationsspiele:** Spielerische Handelsplattform, um das Erlernte praktisch anzuwenden.

**Belohnungssystem:** Punkte und Abzeichen für abgeschlossene Module und bestandene Tests.

**Benutzerrollen:** Unterschiedliche Benutzerrollen (z.B. Anfänger, Fortgeschrittene) mit angepassten Inhalten.

## Technologien und Architektur

### Frontend:

**Jinja Template Engine:** Jinja wird verwendet, um HTML-Seiten dynamisch zu generieren. Es ermöglicht die einfache Integration von Daten in HTML, unterstützt bedingte Anweisungen und Schleifen innerhalb der Templates und sorgt so für eine flexible und leistungsfähige Template-Verarbeitung. Jinja2 wurde gewählt, weil es gut mit Flask integriert werden kann und eine robuste Lösung für die Generierung dynamischer Webseiten bietet.

### Backend:

**Flask:** Flask ist ein leichtgewichtiges Python-Webframework, das sich hervorragend für die Entwicklung von Webanwendungen eignet. Es folgt dem MVC (Model-View-Controller) Designmuster und ermöglicht eine einfache und flexible Entwicklung von Anwendungen.

### Datenbank:

**SQLite:** SQLite wurde gewählt, weil es eine leichtgewichtige, serverlose Datenbank ist, die dennoch alle erforderlichen Funktionen für eine Webanwendung bietet. Es ist einfach zu integrieren und erfordert keine komplizierte Konfiguration, was es ideal für die Entwicklung und das Testen macht.

### API:

Eine JSON-basierte API wurde entwickelt, um Daten zwischen dem Frontend und Backend auszutauschen. Diese API ermöglicht das Abrufen und Speichern von Benutzerdaten, Quiz-Ergebnissen und anderen relevanten Informationen.

## Login

### Benutzerregistrierung

**Rollenbasierte Zugriffskontrolle:** Implementierung verschiedener Benutzerrollen (z.B. Anfänger, Fortgeschrittene) um angepasste Inhalte anzubieten und den Zugriff auf bestimmte Bereiche der Plattform zu steuern.

## Umgesetzte und nicht umgesetzte Ideen

### Umgesetzte Ideen:

**Interaktives Lernen:** Einführung von Quiz und Prüfungen, um den Wissenserwerb zu testen und zu fördern.

**Simulationsspiele:** Entwicklung einer Handelsplattform, die es Nutzern ermöglicht, in einer risikofreien Umgebung Handelsstrategien zu üben.

**Belohnungssystem:** Implementierung eines Punktesystems, das den Fortschritt der Benutzer belohnt und motiviert.

**Benutzerfreundliche Oberfläche:** Erstellung einer intuitiven Benutzeroberfläche mit konsistenten Layouts und Design-Elementen.

## **Nicht umgesetzte Ideen:**

**Echtzeit-Datenintegration:** Die Integration von Echtzeit-Marktdaten für ein realistisches Handelserlebnis wurde aufgrund der Komplexität und der notwendigen Ressourcen nicht umgesetzt. Stattdessen wird eine statische Datenbank mit vorab festgelegten Kursen verwendet.

**Multiplayer-Modus:** Der ursprüngliche Plan, einen Multiplayer-Modus zu unterstützen, wurde verworfen. Der Fokus liegt nun auf individuellen Lernmodulen und Übungen.

**Umfangreiche Finanzielle Bildung:** Die Idee, umfangreiche Module zu verschiedenen Aspekten der finanziellen Bildung (z.B. Marktanalyse, Risikomanagement) anzubieten, wurde reduziert. Stattdessen konzentrieren sich die Lernmaterialien auf die grundlegenden Konzepte des Kryptowährungshandels.

## **Fazit**

Durch die Anpassung und Verfeinerung der ursprünglichen Idee haben wir eine solide Grundlage geschaffen, um eine lehrreiche und benutzerfreundliche Plattform zu entwickeln, die den Nutzern ein tiefes Verständnis des Kryptowährungshandels vermittelt. Die Verwendung von Flask, Jinja und SQLite ermöglicht eine flexible und effiziente Entwicklung, während die verschiedenen interaktiven Elemente und das Belohnungssystem die Benutzer motivieren und den Lernprozess unterstützen.

Diese Dokumentation bietet einen detaillierten Einblick in die Projektentwicklung und die getroffenen Entscheidungen und soll als Leitfaden für die weitere Entwicklung und Implementierung der Plattform dienen.