

1. Projektidee

Die PortfolioApp adressiert Schwächen vieler bestehender Krypto-Portfolio-Tools: Sie sind oft entweder benutzerunfreundlich oder zentralisiert, wodurch Nutzer ihre Vermögenswerte faktisch in die App einzahlen müssen. Das erhöht das Risiko. Diese Anwendung verfolgt einen Ansatz, bei dem Nutzer ihre Wallets selbst behalten und nur deren Bestände analysiert und verwaltet werden – mit klarer, einfacher Oberfläche.

2. Funktionalitäten

- Übersicht aller Wallets inkl. USD-Gesamtwert (Net Worth).
- Detailansicht je Wallet mit Token-Holdings (Symbol, Name, Amount, USD-Wert).
- Verwaltung von Zielallokationen (Target Allocations) je Asset inkl. Toleranz.
- Automatische Erkennung von Abweichungen gegenüber Targets.
- E-Mail-Benachrichtigung bei Drift (Cooldown konfigurierbar).
- Admin-Bereich: Benutzerübersicht, Löschen, Detailansicht.
- 2FA-Unterstützung per Code (E-Mail), optionale Aktivierung pro Nutzer.

3. Technische Details

- Frontend: Blazor Server (Bootstrap-basiertes UI).
- Backend: C#-Services mit Repository-Pattern (z. B. UserService, WalletService, TargetAllocationService, DashboardService).
- Datenbank: MySQL via Entity Framework Core; Entities für User, Wallet, CoinHolding, TargetAllocation, Rebalance-Sessions.
- Authentifizierung: Cookie-Auth, Rollen (User/Admin), optional 2FA per E-Mail-Code.
- E-Mail: SMTP-Versand (Verifikation, Passwort-Reset, Drift-Alerts).
- Externe Schnittstellen (APIs): SolanaTracker (Wallet-/Token-Daten) und Jupiter (DEX/Swaps).
- Rebalancing: Aktuell Greedy-Ansatz zur Berechnung minimaler Swaps.

4. Verbesserungsvorschläge

- Multi-Chain-Support (Ethereum, Arbitrum, Base, BSC, u. a.).
- Verbessertes Logging (zentraler Log-Store, UI-Viewer im Admin-Bereich).
- Erweiterte Admin-Seite (Wallet-/Chain-Management, Audit-Logs).
- Optimierte Algorithmen für Swap-Berechnungen (z. B. Min-Cost-Flow, ILP, heuristische Verfahren) statt rein Greedy.
- Dynamisches Hinzufügen neuer Chains über die Admin-Oberfläche (konfigurierbare Endpunkte/Provider).