Trading Engine Platform

prezentare intemediara

Ideea proiectului & Backlog

O aplicatie web in care utilizatorii isi pot crea cont, pot vizualiza date reale de piata, pot plasa ordine de cumparare/vanzare si pot urmari evolutia portofoliului in timp real.

Sprint 1: *Configurare & Date Piata*

- Configurare backend Spring Boot + PostgreSQL
- Integrare API Binance/TradingView pentru preturi reale
- WebSockets pentru transmiterea preturilor live
- Afișare grafice istorice de pret

Sprint 2: *Order Book & Matching Engine*

- Modelare si stocare Order Book
- Plasare ordine BUY/SELL (market & limit)
- Implementare motor de matching (automat cand preturile se potrivesc)
- Istoric executii pentru fiecare ordin

Sprint 3: *Autentificare & KYC*

- Inregistrare/autenti ficare securizata
- Upload documente pentru KYC si workflow de aprobare (admin)
- Two-Factor Authentication (2FA)

Sprint 4: *Dashboard & Notificari*

- Panou utilizator: solduri, pozitii deschise
- Notificari pentru confirmare tranzactii si modificari de status
- Actualizari instant (WebSocket) pentru executii partiale/finale

Arhitectura

Front-end React

- Conexiune HTTP pentru API calls
- WebSocket client pentru live updates

API Spring Boot (Java)

- Controller → Service → Repository (JPA)
- Gestioneaza autentificare, ordine si portofoliu

Microserviciu Python (FastAPI)

- Endpoint POST /order pentru matching engine
- Endpoint GET /orderbook pentru order book

PostgreSQL

- Entitati: User, Order, Trade, Portfolio, UserAsset, Asset, KYCRequest.

Containerizare Docker

Aspecte tehnice

1. **Containerizare Docker**

– Docker pentru pornire simultana si testing local

2. **Integrare Python ↔ Java**

– Motorul de matching e scris in Python (FastAPI)

– Consum REST prin Spring WebClient, mapping DTO-URI

3. **Logica Matching EngineL**

– Structuri de date in-memory: orderbook cu prioritati

– Gestionare partial-fills, update cantitati, generare Trade

4. **Simulare de Piata**

– Apel unitar la API extern (TradingView) pentru snapshot

– Simulare locala a efectului ordinelor plasate

Lucruri ramase de implementat

- Simulare de piata dinamica

  – Generare evenimente de pret (ticks) la fiecare tranzactie si difuzare prin WebSocket catre client

- Front-End complet

  – Formular ordin (market/limit), vizual Order Book, grafice live, portofoliu etc.

- Workflow KYC & Admin

  – Dashboard admin pentru aprobare documente si notificari email

- Testare & CI/CD

  – Unit si integration tests pentru Java si Python

  – Pipeline GitHub Actions + Docker Compose pentru mediu CI