**Project Documentation: HLD & LLD**

# 1. High-Level Design (HLD)

The goal of the project is to predict cryptocurrency liquidity to ensure market stability using historical data and machine learning models. The system takes in features like trading volume, market cap, price, and social media sentiment to predict a liquidity score.

Major Components:

- Data Collection from CSV (historical data from CoinGecko)

- Data Preprocessing (handling missing values, normalization, feature creation)

- Exploratory Data Analysis (EDA)

- Feature Engineering (liquidity ratio, volume-to-market-cap, etc.)

- Model Building (Linear Models, Tree-based Models, XGBoost, AdaBoost, Catboost)

- Model Evaluation using MAE, and R²

- Model Deployment (Flask)

# 2. Low-Level Design (LLD)

🔽 Detailed Breakdown of Each Component:

## 2.1 Data Preprocessing

- Removed nulls using mean/mode  
- Encoded categorical columns (e.g., 'symbol')  
- Normalized numerical columns using MinMaxScaler

## 2.2 Feature Engineering

- Created new features:  
 \* liquidity\_score = volume / market\_cap  
 \* price\_change\_ratio = 24h / price  
 \* is\_stable\_coin = 1 if coin is USDT/USDC else

## 2.3 Model Training

- Models used:  
 \* Linear Regression (Baseline)  
 \* Random Forest Regressor  
 \* AdaBoost Regressor  
 \* CatBoost Regressor  
 \* SVR  
- Best model selected using GridSearchCV

## 2.4 Evaluation

- Metrics:  
 \* MAE (Mean Absolute Error)  
 \* R² Score

## 2.5 Model Saving

- Saved final CatBoost model using dill for future predictions

**3. Hosted Link:**

The Flask app is live at:

https://cryptocurrency-liquidity-prediction-a1qg.onrender.com