Project: Cryptocurrency Liquidity Predictor

# High-Level Design (HLD)

## 1. Project Overview

The Cryptocurrency Liquidity Predictor project is a machine learning-based solution that forecasts the liquidity (24h trading volume) of cryptocurrencies. It is designed to help investors, analysts, and financial institutions assess the future liquidity of digital assets based on current market indicators. The application uses a pre-trained regression model and is deployed via a Streamlit-based interactive web app.

## 2. System Architecture

The system consists of the following major components:  
- **Data Layer**: Raw data in CSV format containing coin metrics.  
- **Preprocessing Layer**: Cleansing, transforming, and feature engineering of market data.  
- **Model Training**: Machine learning model training using various regressors and hyperparameter tuning (GridSearchCV).  
- **Model Inference Layer**: A Streamlit interface where the user inputs market metrics, and the model predicts liquidity.  
- **Model Artifacts**: Pickled model and scaler (`.pkl` files) are reused for fast predictions.

## 3. Technology Stack

- Python (pandas, numpy, sklearn, matplotlib, seaborn, joblib, streamlit)  
- Jupyter/Google Colab for training & experimentation  
- Streamlit for deployment and UI  
- Joblib for model serialization  
- Git (recommended for version control)

## 4. Deployment Strategy

The model is trained separately and deployed through a lightweight Streamlit web application. It can be hosted locally or tunneled via services like ngrok for public access. Google Colab can also be used for quick demonstrations.

# Low-Level Design (LLD)

## 1. Data Preprocessing Steps

- Read and clean the dataset (`coin\_gecko\_2022-03-17.csv`)  
- Convert string percentage columns to float  
- Rename columns for clarity  
- Create new features like:  
 - `price\_to\_marketcap = price / market\_cap`  
 - `volume\_to\_marketcap = volume\_24h / market\_cap`  
 - `volatility\_score = abs(7d%)`  
- Drop rows with all null values  
- Sort coins by market\_cap for visualization

## 2. Feature Engineering

The following features are used for prediction:  
- price  
- pct\_change\_1h  
- pct\_change\_24h  
- pct\_change\_7d  
- volume\_24h  
- market\_cap  
- price\_to\_marketcap  
- volume\_to\_marketcap  
- volatility\_score

## 3. Model Training & Selection

- Tried multiple regressors (RandomForest, GradientBoosting, Linear Regression, etc.)  
- Evaluated using R^2 Score and RMSE  
- Used GridSearchCV to find best parameters for each model  
- Selected best-performing model based on cross-validation  
- Saved final model and scaler using `joblib.dump()`

## 4. Streamlit Application Flow

- Load model and scaler using joblib  
- Load and clean latest dataset for EDA  
- Sidebar switch for "EDA" and "Predict Liquidity" modes  
- EDA mode: Shows market cap, volume, and correlation heatmap  
- Prediction mode: User inputs values -> transformed using scaler -> model predicts next-period liquidity

## 5. Folder Structure

project-root/  
│  
├── data/  
│ └── coin\_gecko\_2022-03-17.csv  
│  
├── models/  
│ ├── best\_crypto\_liquidity\_model.pkl  
│ └── scaler.pkl  
│  
├── notebooks/  
│ └── training\_and\_analysis.ipynb  
│  
├── streamlit\_app.py  
├── requirements.txt  
└── README.md

## 6. Future Enhancements

- Automate data ingestion from live APIs (e.g., CoinGecko, Binance)  
- Deploy with Docker or Streamlit Cloud  
- Extend model to classify illiquid vs. highly liquid coins  
- Add logging, monitoring, and error-handling modules