High-Level Design (HLD)

Project Name: Cryptocurrency Liquidity Prediction for Market Stability

Objective:
Predict future liquidity of cryptocurrencies based on historical and real-time market data using machine learning models.

1. System Components

1.1 Data Ingestion Module

- Collect historical crypto data (price, volume, market cap) via CoinGecko CSV file.

- Store data as CSVs in /data/raw/

.

1.2 Preprocessing & Feature Engineering

- Clean missing values, remove outliers.

- Engineer time-based features, rolling statistics, volatility indicators

.

- Store processed data in /data/processed/.

1.3 Exploratory Data Analysis (EDA)

- Visualizations: volume trends, price movements, liquidity indicators.

- Generate and save EDA report in /reports/ EDA_Workfiow_Report.pdf

1.4 Modeling & Evaluation

- Algorithms: Random Forest, XGBoost, or LSTM (for time-series).

- Hyperparameter tuning using GridSearchCV

- Save best model to /outputs/models/.

1.5 Prediction Engine

- Load trained model

- Predict liquidity score or label for future date inputs

- Output stored in /outputs/predictions/.

1.6 Web Application (UI)

- Frontend using Streamlit
- User input: Date or crypto token
- Output: Predicted liquidity and confidence level
- Main app in app.py

2. Data Flow Diagram

[CSV]
↓
[DATA INGESTION]

↓
[PREPROCESSING + FEATURE ENGINEERING]

↓
[EDA + REPORT GENERATION]

↓
[MODEL TRAINING]

↓
SAVED MODEL]

↓
[PREDICTION MODULE]

↓
[STREAMLIT]

3. Tools & Technologies

Component	Tool/Library
Data Collection	CoinGecko csv

Processing	Pandas, NumPy
Visualization	Matplotlib, Seaborn
Modeling	Scikit-learn, XGBoost
Web App	Streamlit
Reporting	PDF, Markdown
Version Control	Git + GitHub

4. Key Deliverables

- Clean & processed dataset
- EDA Report (reports/EDA_Workfiow_Report.pdf)
- Trained and validated model (outputs/models/)
- Prediction script
- Streamlit interface
- Final Project Report
- README with usage instructions