Low-Level Design (LLD) - Cryptocurrency Volatility Prediction

1. Module-wise Breakdown

preprocess.py: Loads dataset, cleans missing/invalid values, filters Bitcoin, generates new features like volatility, liquidity_ratio, and rolling_volatility.
main.py: Loads processed dataset, trains Random Forest Regressor, evaluates model using RMSE, MAE, and R^2, and saves model using joblib.
streamlit_app.py: Loads model, takes user input, and predicts volatility in a web UI.
generate_eda.py: Creates correlation heatmap, volatility trend, and distribution plots.

2. Feature Engineering

volatility = (high - low) / open liquidity_ratio = volume / marketCap rolling_volatility = 7-day moving average of volatility

3. Model Configuration

RandomForestRegressor with default hyperparameters; trained on numerical features.

4. Evaluation Metrics

RMSE = sqrt(mean_squared_error) MAE = mean absolute error R^2 Score = coefficient of determination

5. Data Flow

Dataset.csv → preprocess.py → processed_dataset.csv → main.py → model.joblib → streamlit_app.py