1. Introduction

- **Project Name:** Cryptocurrency Liquidity Prediction
- **Purpose:** To build a machine learning system that predicts liquidity in cryptocurrency markets for stability and risk management.
- **Scope:** Covers data ingestion, preprocessing, model training, evaluation, and deployment.

---

2. System Architecture

Insert architecture diagram here (hld_architecture.png).
(Save the diagram in reports folder and insert into the Word file.)

---

3. Components Overview

1. **Data Source**
    - Raw cryptocurrency datasets (price, volume, order books, external APIs).
    - Stored in `data/raw/`.
2. **Data Processing**
    - Cleaning, transformation, and feature engineering.
    - Outputs stored in `data/processed/`.
3. **Modeling**
    - ML algorithms: Regression, Random Forest, XGBoost, etc.
    - Model training notebooks in `notebooks/`.
4. **Evaluation**
    - Performance metrics: RMSE, MAPE, R^2.

    - Visualization of results stored in `reports/eda/`.

5. **Deployment**
    - API endpoints for predictions.
    - Templates and static files for front-end.

---

4. Module Interaction

- **Data → Processing → Model → Evaluation → Deployment**
- Data flows sequentially between these modules with feedback loops for model improvement.

---

5. Technology Stack

- **Programming:** Python (Pandas, NumPy, Scikit-learn, Flask)
- **Visualization:** Matplotlib, Seaborn
- **Deployment:** Flask/Django + HTML/CSS/JS
- **Storage:** CSV/Database

---

6. Constraints & Assumptions

- Requires updated crypto data sources.
- Models retrained periodically to adapt to volatility.

---

7. Conclusion

The HLD ensures clarity on how data flows through the system, how models are trained, and how the system will be deployed for end-users.