MoMo Data Analysis System — Report

Author: Sydney Erik Wamalwa

Date: June 2025

1. Introduction

The report includes an overview of the design, implementation, and deployment of the MoMo Data Analysis System, the web application that displays the data of the mobile money transactions. The system supports users in analysis of transaction patterns, viewing aggregations, and an index-paginated list of the transactions.

The technologies employed for this project include Flask (Python) for the application, SQLite for the database, HTML/CSS for the structuring and styling of the UI, and JavaScript and Chart.js for the visualization. The project aimed to make the system lightweight and educational and to showcase full-stack development with minimal dependencies.

2. Project Approach

The process of development centered on the following:

- Lightweight Full-Stack Development with Flask for backend operations, SQLite for storing data, and dynamic visualization with Chart.js
- RESTful Architecture: The application is separated concern-wise with the provision of endpoints for frontend JavaScript for dynamically retrieving data for summaries and charts.
- User-Centered Design: This interface is intuitive and responds effectively, presenting information in the form of charts and tables with minimal user intervention.

3. System Architecture

The system employs an architecture that includes:

- Frontend Layer
- CSS and HTML for styling and structure
- JavaScript (vanilla) for data retrieval and display.
- Chart.js for Dynamically Rendering Bar and Pie Charts from API Data
- Back-end Layer
- Flask handles the routing, logic, and JSON API endpoints
- They are rendered through Jinja2.

- The backend endpoints serve pages and JSON responses both.

Data Layer

- SQLite is an embedded, lightweight database.
- Flask directly interfaces with the SQLite DB using SQL queries to retrieve and aggregate data.

4. Database Design

The system employs a single table named transactions that is described as

Field	Type	Description
`id`	INTEGER	Primary key, auto-incremented
`date`	TEXT	Date of the transaction
`transaction_type`	TEXT	Type (e.g. Deposit, Withdrawal, Transfer)
`amount`	REAL	Value of the transaction
`recipient`	TEXT	Receiver's name or identifier

Such a structure supports extensible summary inquiry, filtering, and visualization.

5. Challenges Encountered

Integration of Chart.js with Flask

Chart.js graphs needed to be dynamically updated through Flask APIs, whose views and data needed to be separated clearly. Raw JSON had to be rendered through backend endpoints, which were then called by JavaScript for chart updates.

Responsive frontend without frameworks

Not using Bootstrap or frontend frameworks, it was a time-consuming but fulfilling experience to create a clean, responsive design with pure CSS.

Grouping and Data Aggregation

SQLite transaction-type group totals queries were designed with care to produce tidy JSON for summary totals as well as for chart data sets.

6. Key Design Decisions

- **Chart.js for Visualization**: Chosen for its simplicity and beauty, Chart.js simplified the process of creating high-quality charts with less code.
- **Modular Flask Routes**: The application routes were separated into UI-rendering endpoints and API-serving endpoints for modularity and readability reasons.
- **Progressive Enhancement**: Users can still access tabular data when JavaScript does not work or is turned off, enhancing accessibility.
- **Simple, easy to scale**: The database schema was made simple so that it could scale quickly with user tables, transaction categories, or additional metadata as needed.

7. Conclusion MoMo Data Analysis

The system succeeds in its quest to offer the visualization of the mobile money transactions in an easy-to-use, interactive interface. By using Flask, SQLite, HTML, CSS, JavaScript, and Chart.js, the application remains lightweight yet delivers the full-stack experience. Future possible improvements include: User authentication Export to CSV Multi-user dashboards Email/messaging notifications for high-value transactions.