

MOMO DATA ANALYSIS Group 42

1. Introduction

Project Overview: We had a collective task to create an MTN enterprise-level full stack application designed to process SMS data in XML format, clean and categorize the data, store it in a relational database, and build a frontend interface to analyze and visualize the data.

2. Approach to Development

Methodology: The MTN Transactions Dashboard follows a structured end-to-end development methodology that integrates data processing, backend API development, database management, and frontend visualization.

Team Structure:

- a. Amanda Inema: Back end developer of the project.
- b. Louis-Marie Tona: Joint Front end developer of the project.
- c. Olivier Itangishaka: Joint Front end developer of the project.
- d. Ange Umutoni: Documentarian of the project.

Tools and Technologies:

- **Visual Studio Code** for the front end and back end development.
- **Github** for repository creation.
- **Swagger API** to test API endpoints.
- **PgAdmin** to view, query and manage the database.
- **Supabase** to store the transactions in a managed cloud database

3. Design Choices

User Experience (UX) Design: We went with a simplistic corporate style that looks the most like an classic MTN color design so that the users immediately know which company is related to the interface.

User Interface (UI) Design: For this part, the design for the dashboard was also made to look as much as possible as an MTN dashboard from its classic era where the colourways were always dark blue and amber yellow.

4. Functionality Overview

Core Features:

The system extracts transaction details from XML-based SMS messages, categorizes them into various transaction types (e.g., Payments, Withdrawals, Bank Transfers), and stores them in a database.

The frontend provides an interactive dashboard where users can:

- Search and filter transactions by type, date, or amount.
- View visual analytics using charts and graphs for transaction trends.
- Examine detailed transaction records in a structured table.
- Click on transactions to view additional details.

User Flow:

This project showcases end-to-end development skills, including backend API creation (Node.js, Express, Sequelize, PostgreSQL), frontend visualization (HTML, CSS, JavaScript, Chart.js), and real-time search filtering.

5. Challenges Faced

Technical Challenges:

- It was a bit troublesome to work with the chart.js to retrieve information from the backend.
- Integrating the graphs in the dashboard was a challenge as well.
- It was a challenge to host the postgres database on the supabase.
- Making the table responsive was an exhaustive task.

Design Challenges:

- Finding color variations that match in the charts or graphs.
- Making the table responsive was an exhaustive task.

6. Lessons Learned

What Worked Well:

- Hosting the server on swagger API was very straightforward.
- The API integration was very smooth.

What Could Be Improved:

- The User Interface needs a bit more work in order for it to be as interactive as possible as some of the buttons still face some errors.
- The design is a bit too colourful which quite does not match well with the current day trends of simplistic white backgrounds web platforms

7. Conclusion

So in summary again, This project showcases end-to-end development skills, including backend API creation, interactive frontend dashboard design and integration of databases and APIs in our web platform.