Technical Document on React Dashboard

1. Overview & Purpose

This project is an interactive financial dashboard built with **React** and **Plotly.js** for the frontend, and **Flask** (with **Flask-CORS**) for the backend API. The goal is to provide a visual analysis of key financial metrics for two companies:

- **DIPPED PRODUCTS PLC** (abbreviated as DIPD)
- **Richard Pieris Exports PLC** (abbreviated as REXP)

The dashboard allows users to:

- View summarized Key Performance Indicators (KPIs) for a selected quarter.
- Explore time-series charts that display trends for metrics such as Revenue, Gross Profit, Gross Margin, Net Income, Operating Income, and Cost Breakdown.
- Switch between a single-company view and a "Compare Both" view, where KPI cards and charts for both companies are shown side by side.

---

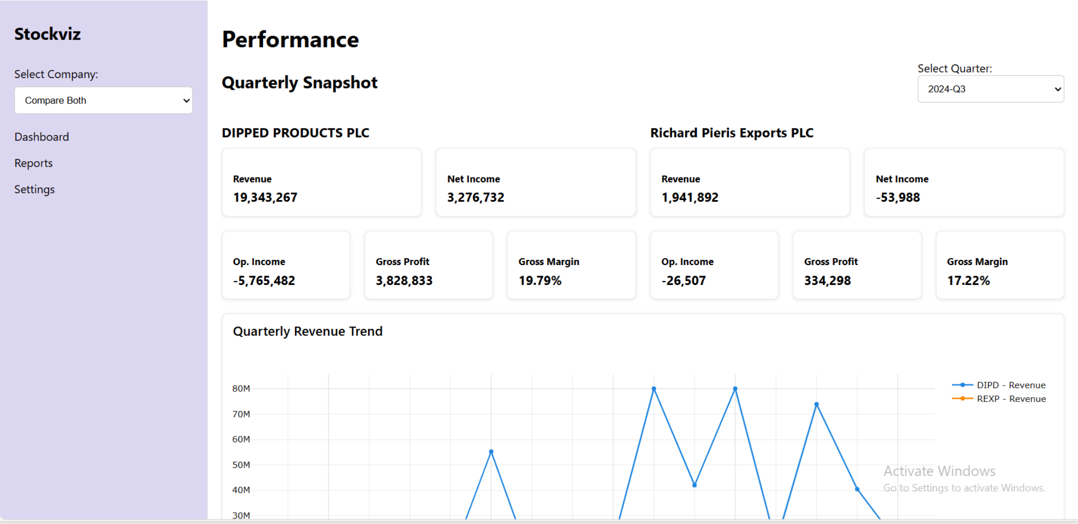
2. Functionalities & User Options

User Interface & Filters

- **Sidebar (Navigation & Company Selector):**
    - **Company Selector:**
        1. Choose between “DIPPED PRODUCTS PLC”, “Richard Pieris Exports PLC”, or “Compare Both”.
        2. Changing the selection updates the charts to display data accordingly.
    - **Navigation Menu:**

        1. Basic navigation links (e.g., Dashboard, Reports, Settings). These are placeholders and can be extended.

- **Main Content Area:**
    - **Quarter Filter (KPI Section):**
        1. A dropdown allows the user to select a specific quarter.
        2. This filter only affects the KPI cards (the snapshot metrics) at the top of the page; the charts still display a multi-quarter timeline.
    - **KPI Cards:**
        1. Show key metrics:
            - **Revenue:** Total income from sales.
            - **Net Income:** Final profit after all expenses.
            - **Operating Income:** Profit from core operations (excluding non-operational costs).
            - **Gross Profit:** Revenue minus the Cost of Goods Sold (COGS).
            - **Gross Margin:** Gross Profit as a percentage of Revenue.
        2. In "Compare Both" mode, two columns of KPI cards are shown—one for each company.



Stockviz

Select Company: Compare Both

Dashboard

Reports

Settings

Performance

Quarterly Snapshot

Select Quarter: 2024-Q3

DIPPED PRODUCTS PLC

Revenue 19,343,267

Net Income 3,276,732

Op. Income -5,765,482

Gross Profit 3,828,833

Gross Margin 19.79%

Richard Pieris Exports PLC

Revenue 1,941,892

Net Income -53,988

Op. Income -26,507

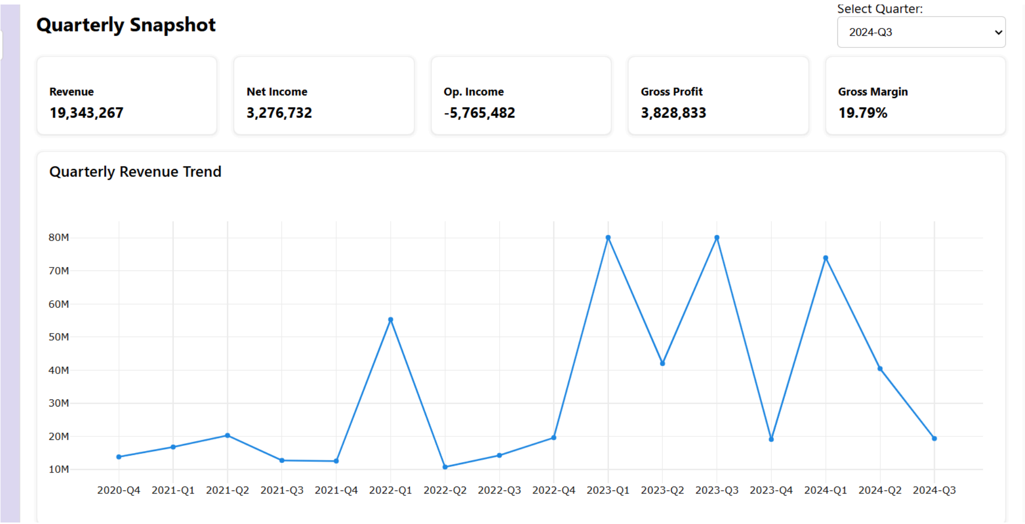
Gross Profit 334,298

Gross Margin 17.22%

Quarterly Revenue Trend

- **Charts:**
  The dashboard features multiple charts arranged in the following order:

  1. **Revenue Trend (Line Chart):**
     - Tracks the revenue across quarters.
     - In "Compare Both" mode, two lines are displayed (one for each company) using contrasting colors.
  2. **Gross Profit & Margin (Combo Chart):**
     - Bars display Gross Profit while a line plots Gross Margin (%).
     - Provides insight into cost efficiency.
  3. **Net vs. Operating Income (Grouped Bar Chart):**
     - Displays both Net Income and Operating Income side by side.
     - Highlights differences between operating performance and overall profitability.
  4. **Cost Breakdown (Stacked Bar Chart):**
     - Shown only when a single company is selected.
     - Displays a breakdown of costs into COGS and Operating Expenses (OPEX).

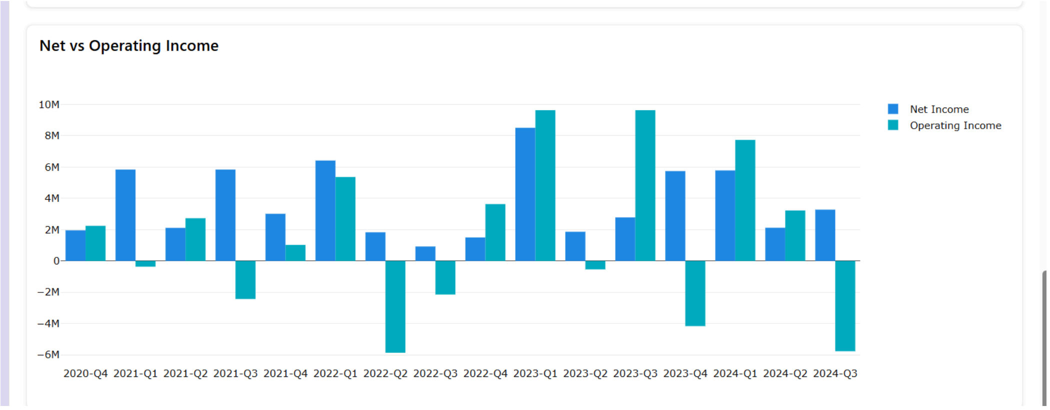


Quarterly Snapshot

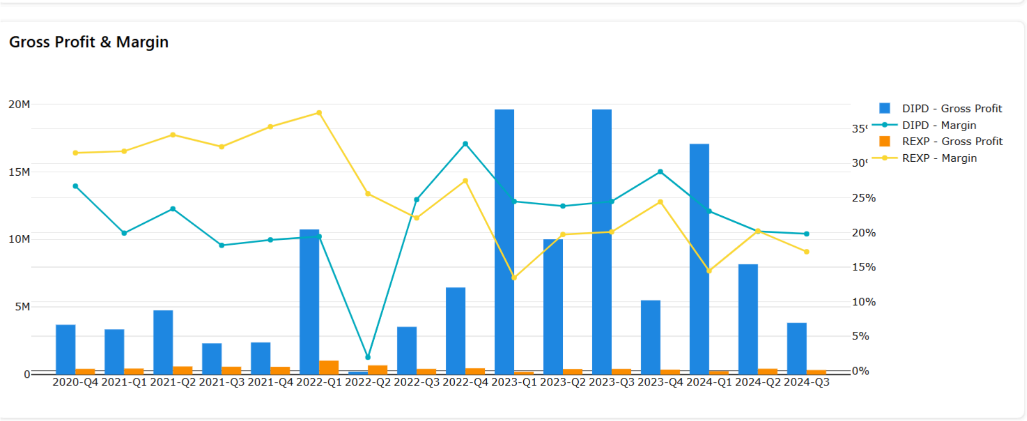
Select Quarter: 2024-Q3

Revenue	Net Income	Op. Income	Gross Profit	Gross Margin
19,343,267	3,276,732	-5,765,482	3,828,833	19.79%

Quarterly Revenue Trend



Net vs Operating Income



Gross Profit & Margin

3. Technologies Used

- **React:**
  - The frontend is built using React, a JavaScript library for building user interfaces with a component-based architecture.
- **Plotly.js & react-plotly.js:**
  - Plotly.js is used to create interactive, responsive charts.
  - The `react-plotly.js` library is a React wrapper for Plotly.js, which simplifies the integration of these charts into React components.
- **Flask:**
  - A lightweight Python web framework that powers the backend API.
  - Provides an endpoint (`/get-data`) to serve the dataset as JSON.
- **Flask-CORS:**
  - A Flask extension to handle Cross-Origin Resource Sharing, allowing the React frontend (typically running on a different port) to access the Flask API.
- **HTML/CSS:**
  - Basic HTML and CSS (or CSS-in-JS) are used to style the dashboard.
  - The theme is light and pastel, with a focus on readability and creative color contrasts for chart elements.

4. Key Metrics & Financial Jargon Explained

- **Revenue:**
  The total income generated from the sale of goods or services in a quarter.

- **COGS (Cost of Goods Sold):**
  The direct costs attributable to the production of the goods sold by the company.

- **OPEX (Operating Expenses):**
  The day-to-day expenses required to run the business (e.g., salaries, rent).

- **Gross Profit:**
  Calculated as Revenue minus COGS. It represents the profit made after covering direct production costs.

- **Operating Income:**
  Derived by subtracting Operating Expenses from Gross Profit. It reflects the profit from core business operations.

- **Net Income:**
  The final profit after all expenses, including non-operating items like taxes and interest.

- **Gross Margin:**
  Expressed as a percentage, it is calculated as (Gross Profit / Revenue) × 100%. It indicates how efficiently a company produces its goods relative to revenue.

---

5. Challenges Encountered & Solutions

Data Loading & CORS Issues

- **Challenge:**
  Initially, data was loaded directly from the public folder. This was replaced by a Flask API for more flexibility and scalability.

- **Solution:**

    - Set up a Flask backend to serve the dataset at `/get-data`.
    - Enabled Flask-CORS to allow cross-origin requests from the React app.

File Path & Route Mismatches

- **Challenge:**
  There were occasional mismatches between the file names/paths in the Flask backend and what the React frontend expected.
- **Solution:**
    - Standardized the JSON file name and ensured the Flask code correctly points to its location.
    - Updated the React fetch URL to match the Flask endpoint.

Color Palette & Visual Appeal

- **Challenge:**
  Finding a creative, high-contrast color palette that is both engaging and compatible with a light pastel theme.
- **Solution:**
    - Experimented with multiple palettes.
    - Settled on a combination where DIPD uses royal blue and bright cyan, while REXP uses deep orange and warm yellow.
    - Adjusted the cost breakdown colors to vivid red and purple.
    - Ensured text is mostly black for readability on a light background.

---

6. Future Enhancements

- **Implement an option to download reports**
  Through the reports tab we can provide the option for the user to directly download the

relevant report.

- **Authentication & Authorization:**
  Secure sensitive financial data by adding user authentication.

Conclusion

This document outlines the core functionalities of the React Dashboard, describes the financial metrics and jargon used, lists the technologies behind the solution, and explains the challenges faced during development. The dashboard offers a flexible interface for viewing and comparing key financial metrics over time, making it a valuable tool for decision-makers and analysts. Enjoy exploring the data and feel free to extend the application with new features!