## 💡 1. ****The Main Three (Core Financial Statements)****

These are mandatory and foundational:

| Report | Purpose | Key Users |
| --- | --- | --- |
| **Trial Balance** | Internal check that total debits = total credits | Accountants |
| **Income Statement (Profit & Loss)** | Shows performance — revenue, expenses, and profit | Management, investors |
| **Balance Sheet (Statement of Financial Position)** | Shows financial position — assets, liabilities, and capital | Owners, lenders |

## 💼 2. ****Other Important Financial Reports****

Here are **other common ways** to present or analyze a business’s financial information:

| Report / Method | Description | Example / Use |
| --- | --- | --- |
| **Cash Flow Statement** | Shows how cash moves **in and out** of the business during a period. Divided into **Operating**, **Investing**, and **Financing** activities. | Used to track liquidity and ensure the business can pay its bills. |
| **Statement of Changes in Equity** | Shows how owner’s equity changed over time — e.g., new capital introduced, profits retained, or drawings. | Used to explain changes in owner’s capital or shareholders’ funds. |
| **Budgets & Forecasts** | Show **expected future performance** — projected income and expenses. | Used for planning and decision-making. |
| **Management Reports / KPI Dashboards** | Internal reports showing key metrics (sales trends, margins, ratios, etc.) often visualized in charts. | Used by management for quick insight. |
| **Notes to Financial Statements** | Explanations or details supporting figures in the main statements. | Required in formal financial reporting. |
| **Ratio Analysis** | Uses financial ratios (profitability, liquidity, solvency, efficiency) to assess performance. | Example: Current Ratio = Current Assets / Current Liabilities |
| **Comparative Statements** | Show two or more periods side-by-side (e.g., 2024 vs 2023) to analyze trends. | Useful for management and investors. |
| **Segment Reports** | Break down financials by department, product line, or branch. | For large organizations with multiple operations. |
| **Break-even Analysis** | Determines the sales level where revenue = costs. | Helps pricing and production decisions. |

## 📊 3. ****Visual and Analytical Presentations****

Modern businesses often go beyond tables and use **visual tools** for clarity:

| Method | Description | Tool Examples |
| --- | --- | --- |
| **Dashboards** | Interactive visual summaries (charts, KPIs, trends). | Power BI, Excel, Tableau |
| **Graphs & Charts** | Visual representation of trends (e.g., bar charts for revenue growth). | Excel, Google Sheets |
| **Infographics** | Simplified, graphical summaries for presentations. | Canva, Figma |
| **Financial Models** | Spreadsheets linking all financial statements dynamically. | Excel models for startups or projections |

## 🧠 Summary

| Category | Examples | Purpose |
| --- | --- | --- |
| **Core Statements** | Trial Balance, Income Statement, Balance Sheet | Fundamental financial reporting |
| **Supporting Statements** | Cash Flow, Equity Changes, Notes | Completeness and compliance |
| **Analytical Reports** | Ratios, Trends, Comparative | Performance analysis |
| **Visual Tools** | Dashboards, Charts | Quick insights and presentations |

Perfect — you already have a very strong **data foundation** for financial tracking in your coffee system.  
Now, let’s build a **clear, structured implementation plan** for how to use your models to show **complete financial information** using **the top 3 methods** (reports, dashboards, statements) **plus additional analytics**.

## 🧩 1. OVERVIEW OF OBJECTIVES

We want to show a **full financial view** of the business using your models:

* **Revenue tracking** (via Sale and SaleDetail)
* **Cost of goods sold (COGS)** (via Supply and SupplyDetails)
* **Operating expenses** (via Expenditure and Expense)
* **Advance payments / liabilities** (via AdvancePayment)
* **Profitability** per batch, per category, and overall
* **Cash flow & trends** over time

## ⚙️ 2. IMPLEMENTATION PLAN BY CATEGORY

### ****A. Reports (Printable / Exportable Summaries)****

**Goal:** Summarized financial documents for accounting & management.

#### 📄 1. Profit & Loss Report

**Data Sources:**

* Income → from Sale.amount\_received
* Cost of Sales → from sum of SupplyDetails.cost\_of\_net
* Operating Expenses → from Expenditure total

**Implementation Plan:**

* Create reports.py or services/reports\_service.py
* Function: generate\_profit\_loss(start\_date, end\_date, batch=None)
* Query examples:

total\_sales = Sale.objects.filter(sale\_date\_\_range=(start\_date, end\_date)).aggregate(total=Sum(F('sale\_details\_\_net\_kgs') \* F('sale\_details\_\_price\_per\_kg')))['total'] or 0

total\_cogs = SupplyDetails.objects.filter(supply\_\_date\_\_range=(start\_date, end\_date)).aggregate(total=Sum(F('net\_kgs') \* F('price\_per\_kg')))['total'] or 0

total\_expenses = Expenditure.objects.filter(date\_incurred\_\_range=(start\_date, end\_date)).aggregate(total=Sum('amount'))['total'] or 0

profit = total\_sales - total\_cogs - total\_expenses

**Outputs:**

* Total Revenue
* Cost of Goods Sold
* Gross Profit
* Operating Expenses
* Net Profit

**Display Options:**

* HTML printable page
* Download as PDF or Excel (via reportlab or openpyxl)

#### 📊 2. Batch Performance Report

**Goal:** Show profitability per coffee batch.

**Logic:**

* For each Batch, show:
  + Total supply cost (SupplyDetails.cost\_of\_net)
  + Total sales value (Sale.amount\_received)
  + Expenses (Expenditure.amount)
  + Profit/Loss

**Implementation:**

Batch.objects.annotate(

total\_sales=Sum(F('sale\_\_sale\_details\_\_net\_kgs') \* F('sale\_\_sale\_details\_\_price\_per\_kg')),

total\_supply=Sum(F('supply\_\_supply\_details\_\_net\_kgs') \* F('supply\_\_supply\_details\_\_price\_per\_kg')),

total\_expenses=Sum('expenditure\_\_amount'),

)

### ****B. Dashboards (Real-Time Visualization)****

**Goal:** Managers can see quick KPIs and charts.

#### 📈 KPI Widgets

**Top Cards:**

* Total Revenue (month-to-date)
* Total Expenses
* Net Profit
* Active Batches
* Advance Payments (Pending)

Use your Batch, Sale, and Supply models:

from django.db.models import Sum, F

from decimal import Decimal

total\_sales = Sale.objects.aggregate(total=Sum(F('sale\_details\_\_net\_kgs') \* F('sale\_details\_\_price\_per\_kg')))['total'] or Decimal('0')

total\_supplies = Supply.objects.aggregate(total=Sum(F('supply\_details\_\_net\_kgs') \* F('supply\_details\_\_price\_per\_kg')))['total'] or Decimal('0')

net\_profit = total\_sales - total\_supplies

**Frontend:**

* Use **Alpine.js** + **Chart.js** or **Plotly.js**
* Charts:
  + Sales vs Supplies over time
  + Expenses breakdown by category
  + Profit per batch
  + Coffee category contribution (% revenue share)

### ****C. Financial Statements (Formal Accounting View)****

**Goal:** Provide standardized statements.

#### 📜 1. Income Statement (P&L)

* Revenue → total from Sale.amount\_received
* Less COGS → from SupplyDetails
* = Gross Profit
* Less Expenses → from Expenditure
* = Net Profit

#### 📜 2. Balance Sheet (simplified)

* Assets = Cash + Receivables (Sales not yet cleared)
* Liabilities = Advance Payments pending
* Equity = Retained Profit

You can build this dynamically from:

assets = Sale.objects.filter(...).aggregate(total=Sum('amount\_received'))

liabilities = AdvancePayment.objects.filter(status='Pending').aggregate(total=Sum('amount'))

equity = assets['total'] - liabilities['total']

#### 📜 3. Cash Flow Statement

* Inflows: Sales revenue
* Outflows: Supplies + Expenditures + Advances
* Closing Cash = Opening Cash + Net Flow

You can generate this monthly or batch-wise.

### ****D. Additional (4th & 5th Methods)****

#### 🧠 4. Data Analytics & Ratios

* **Gross Profit Margin** = Gross Profit / Sales × 100
* **Net Profit Margin** = Net Profit / Sales × 100
* **Cost Efficiency Ratio** = Supplies Cost / Sales × 100
* **Category Performance** → profit per CoffeeCategory
* **Supplier Performance** → total value supplied per supplier

#### 💡 5. Forecasting / Trend Analysis

Use sales and supply trends to project:

* Expected next-month revenue
* Inventory requirements
* Seasonality analysis

If desired, integrate **pandas** or **scikit-learn** for trend predictions.

## 🧭 3. STEP-BY-STEP IMPLEMENTATION ROADMAP

| Step | Feature | Description | Tools |
| --- | --- | --- | --- |
| 1 | **Data Aggregation Services** | Create Python services for profit/loss, cash flow, and expenses. | Django ORM, Aggregates |
| 2 | **API Endpoints** | Create DRF or Django views to return metrics as JSON. | Django REST Framework |
| 3 | **Dashboard Frontend** | Alpine.js + Chart.js for visual KPIs. | Alpine.js, Chart.js |
| 4 | **Report Pages** | Printable reports (P&L, Batch Performance). | Django Templates, ReportLab |
| 5 | **Export Features** | PDF & Excel exports. | reportlab, openpyxl |
| 6 | **Analytics Layer** | Add financial ratios and category performance. | Django ORM |
| 7 | **Forecasts (optional)** | Predict trends using past data. | pandas, sklearn |

## 🚀 4. FUTURE IMPROVEMENTS

* Add **Cash Book model** for actual bank/cash transactions.
* Add **Receivables/Payables tracking**.
* Integrate **audit logs** for financial changes.
* Use **Celery** to auto-generate weekly financial snapshots.

Would you like me to write the **actual Django service layer code** (e.g. financial\_reports.py) that computes these metrics and returns a JSON summary ready for your dashboard?  
I can create it batch-wise, or overall business-wide depending on your preference.