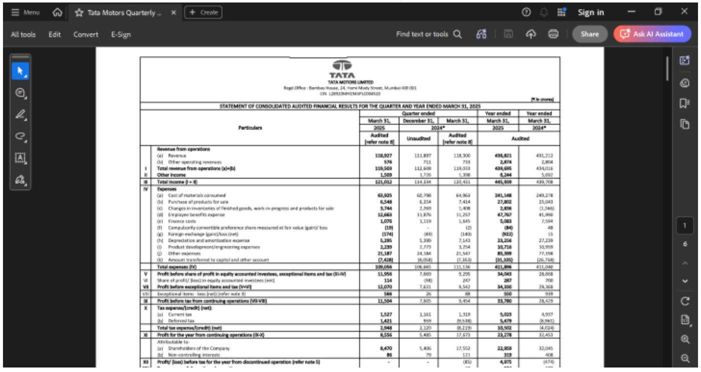
Option A: Financial Statement Extraction to Excel

By

Ankit Kumar (ak373714@gmail.com)

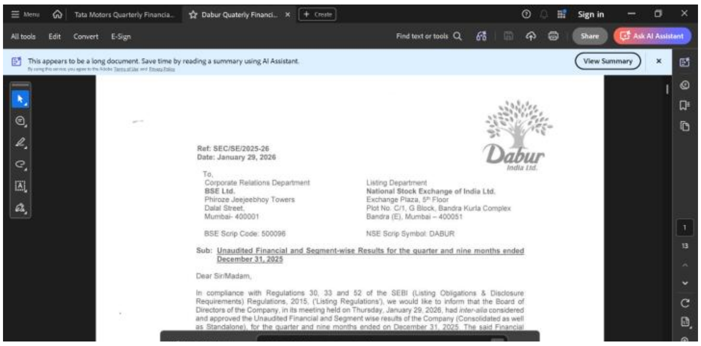
Input: Annual report / financial statement (PDF/Image)



TATA

TATA MOTORS LIMITED

STATEMENT OF CONSOLIDATED AUDITED FINANCIAL RESULTS FOR THE QUARTER AND YEAR ENDED MARCH 31, 2025



This appears to be a long document. Save time by reading a summary using AI Assistant.

Ref: SEC/SE/2025-26
Date: January 29, 2026

To,
Corporate Relations Department
BSE Ltd.
Phiroze Jeejeebhoy Towers
Dalal Street,
Mumbai- 400001

BSE Scrip Code: 500096

Listing Department
National Stock Exchange of India Ltd.
Exchange Plaza, 5th Floor
Plot No. C/1, G Block, Bandra Kurla Complex
Bandra (E), Mumbai – 400051

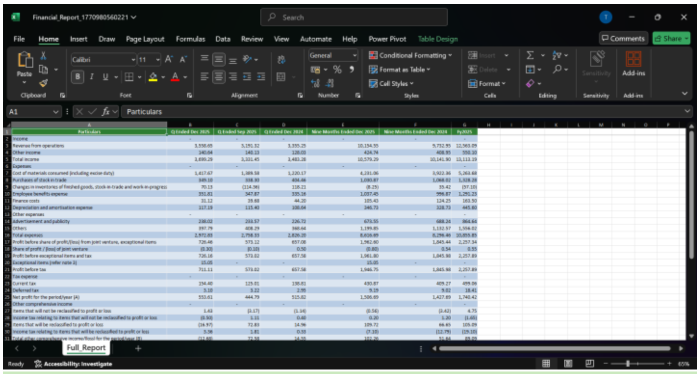
NSE Scrip Symbol: DABUR

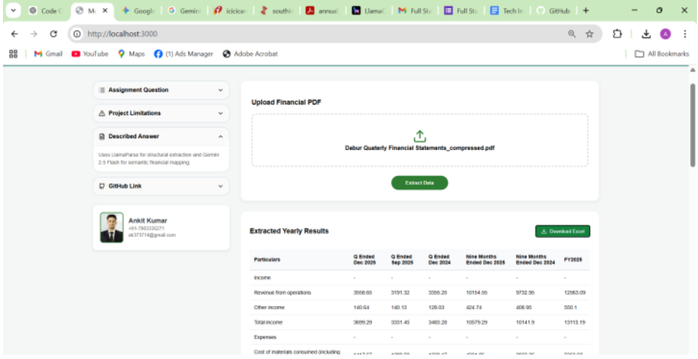
Sub: Unaudited Financial and Segment-wise Results for the quarter and nine months ended December 31, 2025

Dear Sir/Madam,

In compliance with Regulations 30, 33 and 52 of the SEBI (Listing Obligations & Disclosure Requirements) Regulations, 2015, ('Listing Regulations'), we would like to inform that the Board of Directors of the Company, in its meeting held on Thursday, January 29, 2026, had *inter-alia* considered and approved the Unaudited Financial and Segment wise results of the Company (Consolidated as well as Standalone), for the quarter and nine months ended on December 31, 2025. The said Financial

Output: Excel file with income statement line items extracted and ready for analysis and calculations.





Upload Financial PDF

Dabur Quaterly Financial Statements_compressed.pdf

Extract Data

Extracted Yearly Results

Download Excel

Particulars	Q Ended Dec 2025	Q Ended Sep 2025	Q Ended Dec 2024	Nine Months Ended Dec 2025	Nine Months Ended Dec 2024	FY2025
Income	-	-	-	-	-	-
Revenue from operations	3558.65	3191.32	3355.25	10154.55	9732.95	12563.09
Other income	140.64	140.13	128.03	424.74	408.95	550.1
Total income	3699.29	3331.45	3483.28	10579.29	10141.9	13113.19
Expenses	-	-	-	-	-	-

Technology and Work flow:-

Layer	Technology	Purpose in Project
Frontend	React.js	Built dynamic single-page UI
Frontend	Tailwind CSS	Modern responsive styling
Frontend	Flowbite	Pre-built UI components & tables
Frontend	Axios	API communication with Python backend
Backend	Python	Core backend logic & processing
Backend	FastAPI	REST API development
Backend	Uvicorn	Backend server execution
Backend	python-dotenv	Environment variable management
Backend	File handling libraries	PDF upload & processing
AI / LLM	Gemini 2.5 Flash	Intelligent financial data cleaning & structuring
AI / LLM	LlamaParse	Advanced document parsing
AI / LLM	Camelot	Table extraction from structured PDFs

User Uploads PDF
↓
React Frontend
↓
Express + Multer
↓
Camelot (Table Extraction)
↓
LlamaParse (Advanced Document Parsing)LLM
↓
Extracted Structured Data
↓
Gemini 2.5 Flash(AI Cleaning & Structuring) LLM
↓
Standardized Financial JSON
↓
React Visualization + Excel Export

Judgment calls you'll make

1. How do you find and extract these line items from unstructured text? (Pattern matching? LLM? Both?)

To find and extract line items from unstructured text, the project uses a hybrid approach combining both structural parsing and Large Language Model (LLM) reasoning:

- First, I use a tool called LlamaParse to see the document. It identifies where tables are located and converts the visual lines and rows into a text format (Markdown) that a computer can read without scrambling the numbers.
- Then, we use the Gemini AI as a "digital analyst". Instead of just looking for exact words, the AI understands the meaning of the text. For example, it knows that "Employee Costs" and "Staff Expenses" mean the same thing and should be put in the same row.
- The AI also cleans up the "noise." It can tell the difference between a row number (like "1. Revenue") and the actual financial data, and it handles special formatting like parentheses used for negative numbers.

2. What if the document has different line item names? (e.g., "Operating Costs" vs "Operating Expenses")

Here is how i handles :

- The AI is instructed to map varied line items to a standard chart of accounts. For example, it automatically recognizes that "Operating Costs," "Operating Expenses," and "Staff Costs" all belong to specific expense categories in your final Excel sheet.

- Traditional code (Regex) would look for "Operating Expenses" and miss "Operating Costs". By using an LLM, the system "reasons" that these terms represent the same financial concept.

- If an item name is highly unusual, the system is programmed to preserve the original name while still categorizing it correctly so an analyst can understand the source.

This ensures your Excel output is consistent and ready for calculations, regardless of how the original company chose to name its line items.

3. What if the document doesn't have all line items—how do you handle that?

Here is how missing data is handled:

- The AI is instructed to insert "N/A" (Not Available) into any cell where a value cannot be found in the text. This prevents an analyst from mistaking a missing value for a zero, which could drastically alter financial calculations.

- The system generates a "Note" column in your Excel output. If a line item like "Exceptional Items" is missing, the AI adds a comment explaining whether the data was simply not provided in the table or if it was mentioned elsewhere in the report.

- By using strict JSON schema enforcement, the LLM is prohibited from "inventing" numbers based on general knowledge; it must find the evidence within the provided document or label it as missing.

- In some cases, if "Profit Before Tax" and "Tax Expense" are present but "Net Profit" is missing, the AI can flag the relationship but will still mark the raw extraction as missing if the specific row doesn't exist.

4. How do you extract numeric values reliably without hallucination?

- We use "JSON Mode," which forces the AI to fill out a specific form rather than writing freely. This stops it from guessing or being creative.
- By using LlamaParse, the system "sees" exactly which column a number belongs to (like 2024 vs. 2023), so it doesn't mix them up.

- The AI is told to only extract what is on the page. If a number isn't there, it must write "N/A" instead of trying to calculate or guess it.
- It automatically fixes formatting, like turning numbers in brackets (500) into negative numbers -500 for your Excel formulas.
- The AI reads the top of the page first to confirm the currency and scale (like "Millions" or "Crores") before it ever touches a number.

5. How do you know what currency and units the numbers are in?

- The AI specifically looks at the top of the document for key phrases like "All amounts in ₹ Crores" or "USD Millions" before it reads the table.
- Instead of just looking for symbols, the AI understands financial language to know if a "100" means 100 dollars or 100 million dollars.
- Once found, this info is saved as "Metadata" and pinned to your Excel file so you always know the scale of the data.
- By identifying units first, the system ensures your financial models use accurate, scaled numbers rather than raw, incorrect figures.

6. What if the document has multiple years of data—do you extract all of them?

- If the report shows data for the current year, previous year, and even three-year trends, the AI extracts all of them into separate columns.
- It reads the headers to correctly assign each number to the right year (like FY2024 vs. FY2023) so the data stays organized.
- By capturing all years, the system allows you to immediately calculate growth rates and trends in your Excel sheet.
- If you specifically ask for a "Yearly" report, the AI is smart enough to ignore the smaller quarterly snapshots and only keep the full-year audited columns

7. How do you present missing or ambiguous data in the Excel file so an analyst can spot it?

- If a value is missing from the table, the system writes "N/A" instead of leaving a blank or entering a zero. This prevents calculation errors.
- Every row has an attached note. If a value is ambiguous or required a "judgment call," the AI explains its reasoning right next to the data.
- If the AI finds a number that doesn't seem to match the surrounding context (like a total that doesn't sum up), it adds a warning label in the notes for manual verification.
- The system often includes a "Source Line" column, showing exactly what the original text said before the AI cleaned it up for the Excel sheet.

Frontend Link (netlify)

Backend Link (render)