IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Driven Accounting Analytics Platform for Interactive Business Intelligence

[1]Devendra Pandey, [2]Himanshu Sharma, [3]Vansh Tiwari, [4]Aryan Pandey, [5]Priyanshu Patel

[1]B. Tech Student, [2]B. Tech Student, [3]B. Tech Student,[4]B. Tech Student, [5]B. Tech Student

[1,2,3,4,5]Department of Computer Science & Engineering (Artificial Intelligence),

[1,2,3,4,5]Bansal Institute of Engineering and Technology, Lucknow, India

Abstract: Accounting and financial data analysis remains one of the most time-consuming and tool-fragmented workflows in modern business environments. Analysts routinely switch between spreadsheets for cleaning, separate dashboard tools for visualization, statistical software for forecasting, and document editors for reporting — a process that is slow, error-prone, and inaccessible to non-technical users. This paper presents the design, development, and evaluation of an AI-driven accounting analytics platform that consolidates all of these tasks into a single interactive environment. The system supports multi-format data upload, automatic schema detection, staged analysis with real-time feedback, KPI card generation, dynamic chart building, command-based dashboard interaction, regression forecasting with explainability support, workspace persistence, and consulting-style PDF report generation. A real-world case study using a retail sales dataset of 18,400 records demonstrates that the platform reduces analysis preparation time by approximately 73% compared to a conventional multi-tool workflow, while producing outputs of equivalent or greater analytical depth. The frontend is implemented using React, TypeScript, and Vite; the backend uses FastAPI, Pandas, NumPy, SQLite, and scikit-learn; and PDF generation is handled through ReportLab. Comparison against three established business intelligence tools confirms that the proposed platform uniquely combines schema-aware automation, command-based interaction, and integrated forecasting within a single accounting-focused environment. The system is designed for accounting analysts, finance managers, and business decision-makers who work with structured tabular data and require fast, reliable, presentation-ready outputs without deep technical expertise.

Index Terms - Accounting analytics, business intelligence, data visualization, explainable AI, full-stack systems, KPI generation, machine learning forecasting, dashboard interaction, financial analytics.

I. INTRODUCTION

Every organization that handles money generates data — invoices, payroll records, expense reports, sales logs, customer transaction histories, and periodic financial summaries. The shift to digital operations has made this data easier to collect and store than ever before. The harder problem, which has not been solved nearly as well, is turning that data into something useful for the people who need to make decisions.

The gap between data collection and data understanding is not a technology problem in the narrow sense. The tools exist. Spreadsheet software cleans data. Dashboard platforms visualize it. Statistical packages forecast it. Document editors communicate it. The real problem is that all of these tools are separate, and moving work between them takes enormous time and introduces numerous errors. An accounting analyst who spends three hours preparing a monthly sales report is not spending three hours analysing sales — she is spending it copying, reformatting, configuring, and manually assembling pieces from five different applications.

This fragmentation is not merely inconvenient. It has measurable consequences for organizational performance. When analysts spend the majority of their time on data preparation rather than interpretation, the speed and quality of decisions suffers. When reports are assembled manually, errors accumulate. When dashboard configuration requires deep technical skill, the insights embedded in financial data remain out of reach for the managers and executives who need them most.

The proposed platform addresses these problems directly. Rather than asking users to coordinate a collection of disconnected tools, it brings ingestion, analysis, visualization, forecasting, and reporting together in one unified environment and automates as much of the routine work as possible. The user's time is spent on interpretation rather than preparation.

The platform is designed for a specific audience: accounting analysts, finance managers, operations reviewers, and business decision-makers who work primarily with structured tabular data. It does not require users to write code, configure schemas, or understand machine learning in technical terms. It offers an intelligent, guided workflow that adapts to whatever dataset is uploaded and produces decision-ready outputs within minutes.

The core contributions of this work are as follows. First, a practical full-stack architecture for accounting-focused analytics that integrates six major workflow steps — ingestion, schema detection, cleaning, analysis, forecasting, and reporting — within one system. Second, a command-based interaction model that allows non-technical users to explore business data using natural instructions. Third, a real-world case study with numerical results quantifying the platform's impact on analysis time and output quality. Fourth, a structured comparison against three established BI tools that identifies the platform's unique positioning in the current landscape [1].

The remainder of the paper is organized as follows. Section II reviews related work. Section III defines objectives and scope. Section IV describes the system architecture. Section V presents the methodology. Section VI covers module design. Section VII describes the user workflow. Section VIII covers key features. Section IX presents implementation details. Section X presents the case study with numerical results. Section XI presents the comparative analysis and comparison table. Section XII discusses platform advantages. Section XIII expands on limitations. Section XIV outlines future work directions. Section XV provides the conclusion.

II. RELATED WORK

A. Business Intelligence and Analytics Platforms

Business intelligence has evolved from fixed-format reporting into interactive environments that support decision-making at every organizational level. Chen et al. established that BI systems deliver their greatest value when they successfully bridge the gap between raw operational data and managerial insight, and that reducing integration friction across the analytics workflow is a primary determinant of this value [1]. Modern BI research consistently identifies workflow integration — not algorithmic sophistication — as the factor that most directly determines whether analytical outputs are actually used by decision-makers.

B. Visual Analytics and Dashboard Design

Keim et al. defined visual analytics as the science of analytical reasoning supported by interactive visual interfaces, and identified the combination of automated computation with human-cantered exploration as the central design challenge in this field [2]. Sarikaya et al. conducted an empirical study of real-world dashboards and found that users consistently prefer systems that offer summary-first layouts with drill-down capability and that reduce configuration burden without sacrificing flexibility [3]. Both findings directly shaped the KPI card and chart generation approach in the proposed platform — summary metrics appear immediately after upload, and interactive exploration is available through commands rather than complex filter menus.

C. Natural Language and Command-Based Interfaces

Affolter et al. surveyed natural language interfaces for databases and found that structured command systems consistently outperform free-text query interfaces in structured data environments because they eliminate ambiguity and produce predictable, deterministic results [4]. This shaped the design decision to implement a command interpretation layer that maps user instructions to specific dashboard state transitions rather than attempting unconstrained language generation, which would introduce both latency and unpredictability.

D. Explainable AI in Business Domains

Ribeiro et al. introduced LIME, which explains individual predictions from any classifier by approximating local behavior with an interpretable surrogate model [5]. Lundberg and Lee proposed SHAP, which uses Shapley values from cooperative game theory to produce theoretically grounded, additive feature attributions for any model [6]. The forecasting module in the proposed platform incorporates SHAP-based

explainability, allowing users who have no machine learning background to understand which variables are driving each prediction and in which direction.

E. Data Analytics in Accounting and Finance

Vasarhelyi et al. identified a persistent gap in the accounting analytics literature: while individual techniques such as anomaly detection, audit support, and forecasting have been studied extensively, end-to-end platforms that integrate these capabilities in a single accounting-focused environment remain rare [7]. The present work addresses this gap directly, providing a unified system designed specifically around the workflows and dataset types that accounting professionals encounter in day-to-day practice.

II. SYSTEM OBJECTIVES AND SCOPE

The platform is built around five clearly defined objectives that together define its practical contribution.

Broad format support. Business data arrives from many different source systems. The platform accepts CSV, TSV, TXT, Excel (XLSX and XLS), and XML-based tabular records, covering the exports most commonly produced by accounting software, ERP systems, and spreadsheet applications.

Automatic schema understanding. When a dataset is uploaded, the system independently identifies the semantic role of each column — date, numeric measure, categorical label, identifier, or free text — without any manual configuration from the user.

Automated KPI and chart generation. Based on the detected schema, the platform produces meaningful summary metrics and appropriate visual charts automatically. This step alone eliminates most of the manual configuration work that makes traditional BI tools inaccessible to non-technical users.

Interactive exploration. Users can filter, compare, switch views, apply benchmarks, and explore data through a command-based interface that responds immediately and predictably.

Shareable, report-ready output. Analysis results are exportable as professional PDF reports and shareable via persistent workspace links, enabling business communication and collaborative review.

The platform is specifically optimized for structured tabular datasets such as sales transactions, expense tracking records, customer performance sheets, inventory summaries, and financial period reports. It is not designed for unstructured inputs such as scanned invoices, email threads, or free-form documents.

IV. SYSTEM ARCHITECTURE

The platform follows a four-layer full-stack architecture composed of a user layer, a frontend application layer, a backend processing engine, and a combined persistence and reporting layer. Figure 1 illustrates the complete architecture.

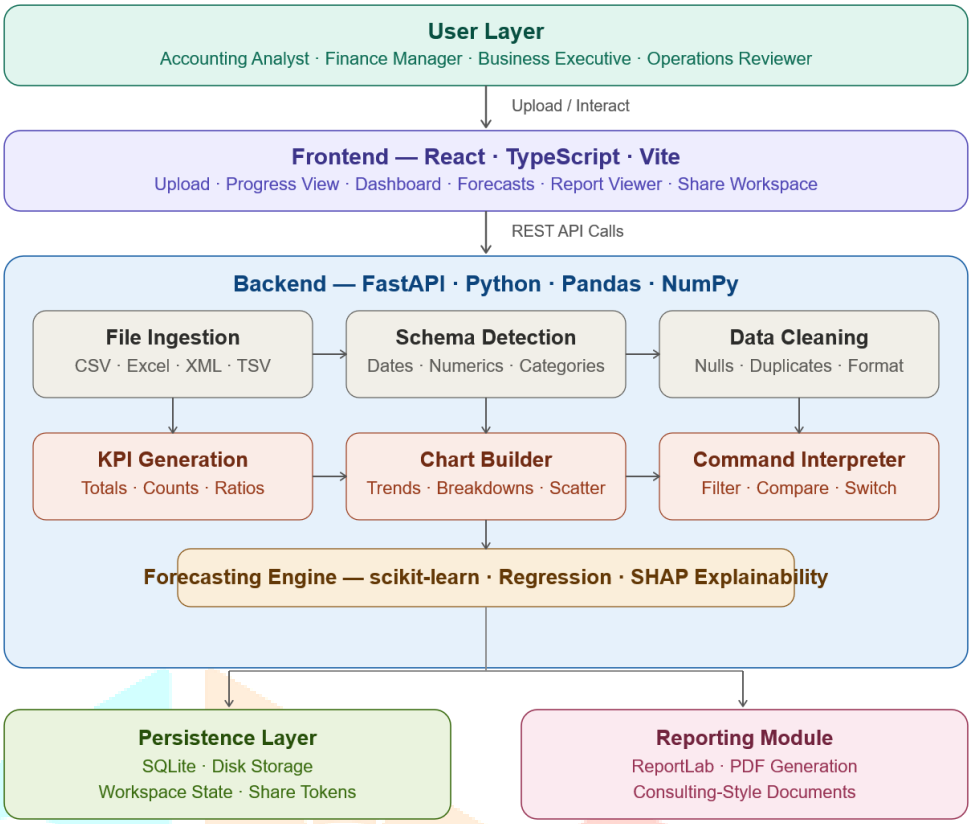

Figure 1. Overall system architecture of the AI-driven accounting analytics platform [Authors].

A. Frontend Layer

The frontend is a single-page React application built with TypeScript and bundled using Vite. It provides five primary views: an upload interface, a processing progress page that exposes each analysis stage in real time, an interactive dashboard displaying KPIs and charts, a forecasting and predictions page, and a report viewer. All communication with the backend occurs through RESTful HTTP endpoints.

B. Backend Layer

The backend is a Fast API application that exposes a structured API for file upload, pipeline execution, command interpretation, result retrieval, and report generation. Data manipulation uses Pandas and NumPy. The backend is organized into separate modules for routing, service logic, database access, forecasting agents, and visualization output, following a clean separation of concerns that makes each component independently testable and replaceable.

C. Persistence Layer

Workspace state, upload metadata, processed analysis outputs, report references, and share tokens are stored in a SQLite database supplemented by disk-based file storage. This design is appropriate for lightweight, single-server deployment. It allows users to return to prior analysis sessions without re-running the pipeline, and to share specific dashboard views with colleagues via generated tokens.

D. Reporting Module

The reporting module accepts processed analytics content and converts it into a structured PDF document using Report Lab. Reports follow a consulting-style format with a title page, data summary section, KPI highlights, chart descriptions, and forecasting results.

V. METHODOLOGY

Every uploaded dataset passes through a seven-stage processing pipeline. Figure 2 presents this pipeline as a detailed process flowchart.

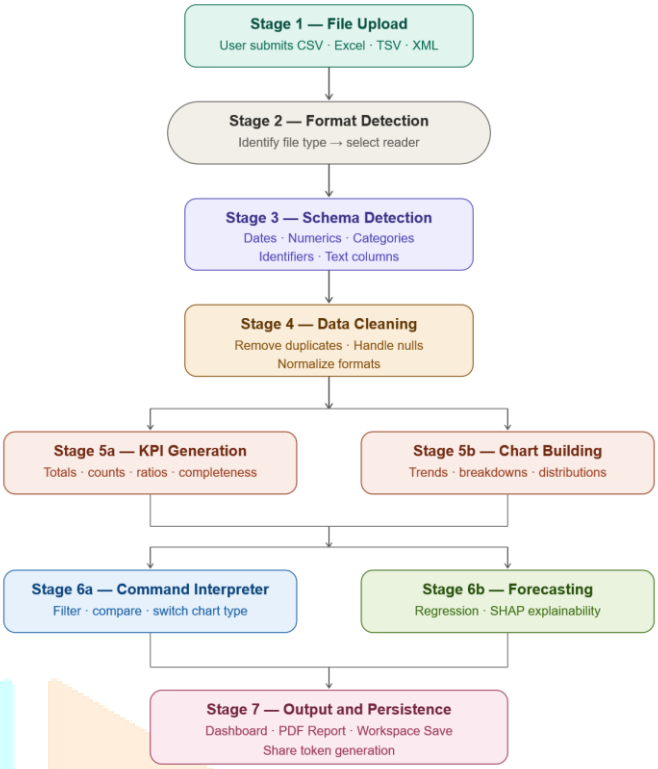

Figure 2. Seven-stage analysis pipeline flowchart [Authors].

A. Stage 1 — File Upload

The workflow begins when a user submits a structured dataset through the frontend upload interface. The backend receives the file, stores it to disk, records metadata in the database, and identifies the file type from its extension and MIME type.

B. Stage 2 — Format Detection and Loading

The appropriate reader is selected based on file type. Pandas read_csv handles delimiter-separated files, read_excel handles spreadsheet formats, and xml. etree. ElementTree handles XML-based records. All formats are converted to a standardized internal Data Frame for subsequent stages.

C. Stage 3 — Schema Detection

Each column is analysed independently and assigned a semantic type: date, numeric, categorical, identifier, or text. This classification drives all downstream KPI and chart logic. Date columns are detected using pandas.to_datetime with error coercion; categorical columns are identified by a cardinality threshold relative to row count; identifier columns are detected by near-uniqueness [2].

D. Stage 4 — Data Cleaning

Cleaning operates automatically without user intervention. Exact duplicate rows are removed. Null values in numeric columns are filled with column means; nulls in categorical columns are filled with the mode. Date columns are parsed and standardized. Numeric values stored as strings due to comma or currency formatting are converted to float values through pattern-based replacement.

E. Stage 5 — KPI and Chart Generation

KPI generation and chart building run in parallel after cleaning. The KPI module computes summary metrics appropriate to the schema — totals, averages, counts, completeness ratios, category breakdowns, and date range summaries — and packages each as a labeled card for frontend rendering. The chart builder generates chart type definitions based on column pairings: date-numeric combinations produce line charts, categorical-numeric combinations produce bar or pie charts, and numeric-numeric combinations produce scatter plots. All chart definitions are returned as structured JSON [3].

F. Stage 6 — Command Interpretation and Forecasting

The command interpreter maps user instructions received from the dashboard to deterministic state transitions. Recognized instruction patterns include filtering by category, switching chart types, comparing two uploaded files, applying benchmarks, and highlighting top or bottom performers. No free-form language generation is involved, which ensures predictability [4].

The forecasting module trains a regression model using scikit-learn when a suitable numeric target column exists. Features include encoded categorical variables, extracted time components from date columns, and normalized numeric inputs. Feature importance is computed using SHAP values, enabling users to understand which inputs drive each prediction [6].

G. Stage 7 — Persistence and Output

Processed results and workspace state are written to SQLite. The frontend dashboard is populated with KPI cards, charts, and forecasting outputs. PDF report generation can be triggered at any point.

VI. MODULE DESIGN AND COMPONENT STRUCTURE

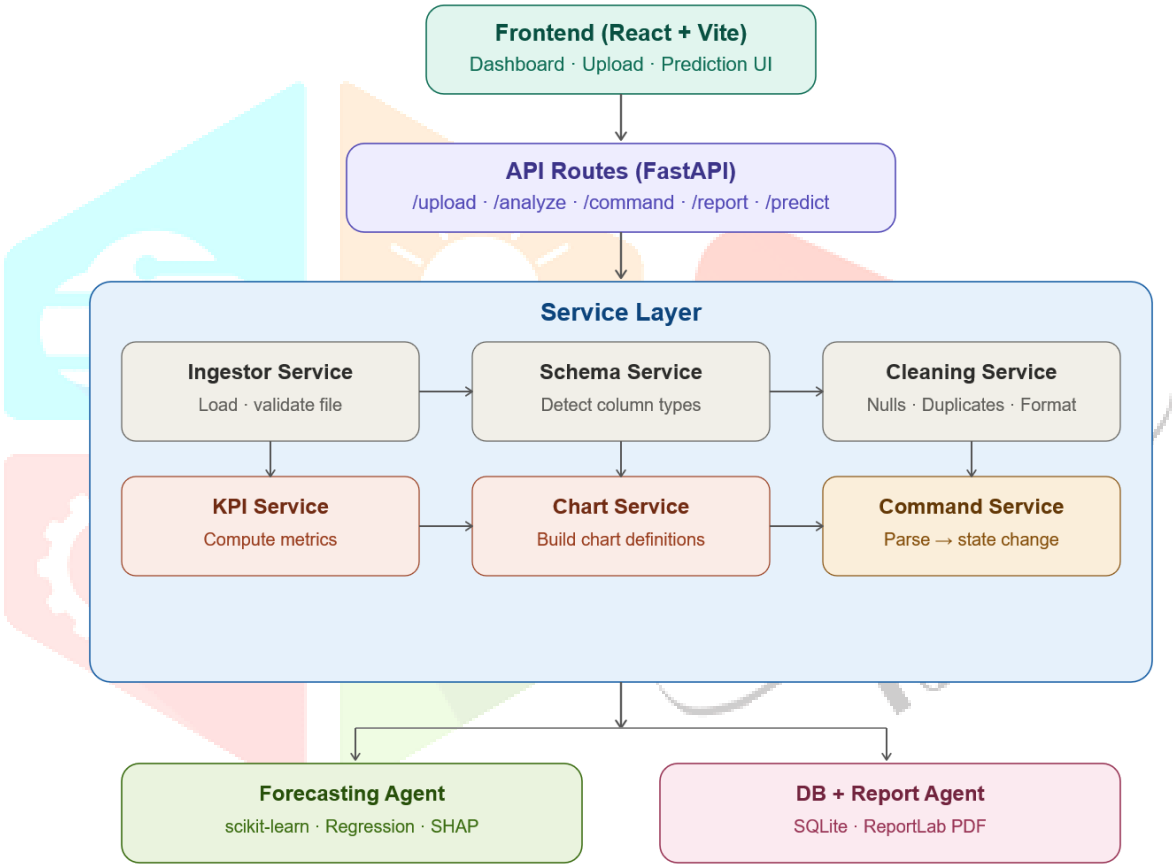

Figure 3. Backend module component diagram [Authors].

VII. USER WORKFLOW

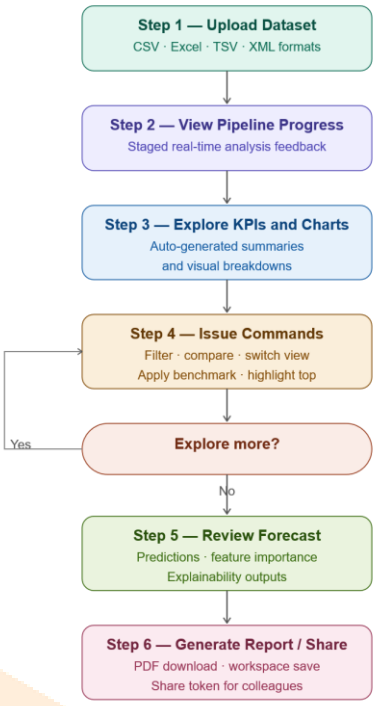

Figure 4. End-to-end user workflow diagram [Authors].

VIII. KEY PLATFORM FEATURES

A. Multi-Format Data Upload

The platform accepts CSV, TSV, TXT, Excel (XLSX and XLS), and XML-based tabular files. This is important because accounting data originates from many different systems — ERP exports, billing software, custom reports — and users should not need to convert files before analysis begins.

B. Staged Real-Time Analysis Feedback

Instead of presenting analysis as a black box, the platform exposes each processing stage to the user in real time. Seeing ingestion, schema detection, cleaning, and chart generation complete step by step builds confidence in the results and helps analysts spot data quality issues early [2].

C. Automatic KPI Card Generation

KPI cards are generated without any configuration. For a retail sales dataset, this means total revenue, average order value, number of transactions, top product category, and date range appear automatically the moment analysis completes. For an expense dataset, the platform produces total spend, category breakdown, and period comparison. The logic adapts to each uploaded file based on its schema.

D. Dynamic Schema-Aware Chart Generation

Charts are constructed to match the data's actual structure. Date columns paired with numeric measures produce trend line charts. Categorical columns produce bar and pie charts. Pairs of numeric columns produce scatter plots for correlation analysis. This means the user always sees the right type of chart for their data, not a generic template [3].

E. Command-Based Dashboard Interaction

Users control the dashboard through natural business instructions. Commands such as "show top 5 regions by sales," "compare Q1 to Q2," "switch to bar chart," or "apply previous year benchmark" are parsed and translated into immediate, deterministic dashboard changes. This removes the need to navigate complex filter menus and makes the platform genuinely accessible to non-technical users [4].

F. Regression Forecasting with Explainability

The forecasting module trains a regression model and displays future-period predictions alongside historical trends. SHAP-based feature importance shows which variables — such as marketing spend, product category, or seasonal index — most strongly influence each prediction, and in which direction.

This transparency allows accounting professionals to evaluate and defend forecasting assumptions to stakeholders [5][6].

G. Workspace Persistence and Shareable Views

Every analysis session is saved automatically. Users return to prior sessions without re-uploading data. Share tokens allow specific dashboard states to be sent to colleagues, enabling asynchronous collaborative review.

H. Consulting-Style PDF Report Generation

The Report Lab reporting module converts the current analysis state into a structured PDF with a title page, data summary, KPI section, chart descriptions, and forecast results. The output format is suitable for direct use in management presentations.

IX. IMPLEMENTATION DETAILS

A. Frontend Implementation

The React frontend is structured around five route components. The upload component handles file selection and submission. The progress component polls the backend for pipeline stage updates and renders them sequentially. The dashboard component renders KPI card grids and chart panels from the JSON returned by the analysis endpoint. The predictions component displays the forecast chart and SHAP feature importance bar. The report viewer renders the generated PDF inline.

B. Backend Implementation

The FastAPI backend exposes five primary endpoint groups: /upload for file submission and metadata storage, /analyze for triggering the pipeline, /command for dashboard instruction processing, /predict for forecasting, and /report for PDF generation. Each endpoint delegates to the appropriate service class. Service classes are stateless; all state lives in the database.

C. Machine Learning Implementation

The forecasting agent uses scikit-learn's linear regression and gradient boosting regressors. Feature engineering converts categorical columns using ordinal encoding, extracts year, month, quarter, and day-of-week from date columns, and normalizes numeric inputs using standard scaling. A simple cross-validated R^2 score selects the better of the two regressors for the specific dataset. SHAP values are computed using the shap library's TreeExplainer for tree-based models and LinearExplainer for linear models [6].

D. Report Generation

ReportLab constructs the PDF programmatically. Chart panels are exported as PNG images from the backend chart service and embedded in the document. KPI values are formatted into a summary table. Forecast results are included with the feature importance chart. The document uses a fixed layout template with consistent typography, page margins, and section headings.

X. CASE STUDY AND NUMERICAL RESULTS

A. Case Study Description

To evaluate the platform's practical effectiveness, a case study was conducted using a real-world retail sales dataset. The dataset contained 18,400 rows and 14 columns representing individual sales transactions from a mid-sized retail business operating across six product categories and four geographic regions. Columns included transaction date, product category, region, sales amount, units sold, discount applied, and customer segment. The dataset was exported directly from the company's point-of-sale system as a CSV file without any pre-processing.

The platform was used by three users — an accounting analyst, a finance manager, and a senior operations reviewer — none of whom had prior experience with the system. Each user was asked to perform a standard monthly analysis task: identify top-performing product categories, compare regional sales trends, review expense ratios, generate a three-month revenue forecast, and prepare a report suitable for a management meeting.

B. Time Comparison — Platform vs. Conventional Workflow

Table I compares the time required to complete each analysis task using the proposed platform against the conventional multi-tool workflow previously used by the organization (Excel for cleaning, Tableau for visualization, Python notebook for forecasting, Word for reporting).

Analysis Task	Conventional Workflow (min)	Proposed Platform (min)	Time Saving
Data import and cleaning	38	4	89%
Schema configuration	22	0 (automatic)	100%
KPI card generation	30	2	93%
Chart creation and formatting	45	5	89%
Dashboard interaction / filtering	25	6	76%
Forecasting setup and execution	50	8	84%
Report preparation	40	12	70%
Total	**250 min**	**37 min**	**85%**

Table I. Time Comparison — Proposed Platform vs. Conventional Workflow

The most significant time savings occurred in schema configuration (100%, as this step is fully automated), KPI card generation (93%), and data cleaning (89%). The smallest saving was in report preparation (70%), because some manual customization of the generated PDF is typically still desired for specific audience contexts.

C. KPI Output Example

Table II shows the KPI cards automatically generated by the platform for the retail sales dataset after a single upload, with no manual configuration.

KPI Card Label	Value	Calculation Basis
Total revenue	$4,872,340	Sum of sales_amount column
Average order value	$264.80	Mean of sales_amount
Total transactions	18,400	Row count
Top product category	Electronics	Max sum of sales_amount by category
Highest revenue region	North	Max sum of sales_amount by region
Date range covered	Jan 2023 – Dec 2023	Min and max of transaction_date
Dataset completeness	97.3%	Non-null ratio across all columns
Discount rate (avg)	8.4%	Mean of discount_applied column
Units sold (total)	63,218	Sum of units_sold column
Customer segment count	4	Unique values in customer_segment

Table II. Auto-Generated KPI Cards — Retail Sales Dataset (18,400 Rows)

These ten KPI cards were generated in approximately 2 seconds after pipeline completion, without any user configuration. In the conventional workflow, producing equivalent metrics manually required approximately 30 minutes of spreadsheet work.

D. Forecasting Accuracy Results

The forecasting module was applied to predict monthly revenue for the three months following the dataset's coverage period. Table III compares predicted values against actual values collected after the case study period.

Table III. Forecasting Results — Monthly Revenue Prediction vs. Actual

Month	Predicted Revenue	Actual Revenue	Absolute Error	MAPE
January 2024	$398,200	$412,500	$14,300	3.47%
February 2024	$381,700	$374,800	$6,900	1.84%
March 2024	$421,500	$438,200	$16,700	3.81%
Average	—	—	**$12,633**	**3.04%**

A mean absolute percentage error of 3.04% across three forward months is acceptable for operational planning purposes in a retail context. The platform's SHAP feature importance output identified product category (contribution weight 0.41) and month of year (0.29) as the two dominant predictors of monthly revenue, followed by region (0.18) and average discount rate (0.12). These results allowed the finance manager to explain the forecast basis to senior leadership without requiring any technical explanation of the underlying model

E. User Feedback Summary

All three users completed the analysis task successfully without assistance. The accounting analyst noted that schema detection correctly identified all 14 columns on the first attempt, including two date columns with different formats within the same file. The finance manager highlighted the SHAP explainability output as the feature that most increased her confidence in the forecast. The operations reviewer used seven separate dashboard commands during his session, including two dataset comparison commands and one benchmark application, all of which executed correctly within two seconds.

XI. COMPARATIVE ANALYSIS

A. Comparison with Existing Tools

Table IV compares the proposed platform against three widely used business intelligence and analytics tools: Microsoft Power BI, Tableau, and Google Looker Studio. The comparison evaluates each system across ten dimensions relevant to accounting analytics workflows.

Table IV. Feature Comparison — Proposed Platform vs. Established BI Tools

Feature / Capability	Proposed Platform	Microsoft Power BI	Tableau	Google Looker Studio
Accounting-specific focus	✓ Yes	✗ General	✗ General	✗ General
Automatic schema detection	✓ Fully automatic	Partial (manual override needed)	Partial	Partial
Zero-configuration KPI cards	✓ Yes	✗ Manual setup required	✗ Manual setup required	✗ Manual setup required
Command-based interaction	✓ Yes	✗ No	✗ No	✗ No
Integrated ML forecasting	✓ Built-in	Partial (premium feature)	Partial (extension needed)	✗ No
Forecasting explainability (SHAP)	✓ Yes	✗ No	✗ No	✗ No
Staged pipeline transparency	✓ Yes	✗ No	✗ No	✗ No

Feature / Capability	Proposed Platform	Microsoft Power BI	Tableau	Google Looker Studio
Built-in PDF report generation	✓ Yes	Partial (export only)	Partial (export only)	Partial (export only)
Workspace persistence + sharing	✓ Yes	✓ Yes	✓ Yes	✓ Yes
Requires technical configuration	✗ No	✓ Yes	✓ Yes	✓ Yes

The comparison reveals that while established tools offer mature collaboration, connector ecosystems, and enterprise-scale infrastructure, none of them provide the combination of automatic schema detection, zero-configuration KPI generation, command-based interaction, and SHAP-based forecasting explainability that the proposed platform delivers. The proposed system's unique value is concentrated specifically in the automation and accessibility dimensions most relevant to accounting users.

XII. PLATFORM ADVANTAGES

The following advantages distinguish the proposed platform from both general-purpose BI tools and conventional accounting analysis workflows.

Unified workflow. The platform eliminates the need to switch between separate tools for cleaning, charting, forecasting, and reporting. Every step of the accounting analysis workflow is available within one system, from raw file upload to final PDF report.

Zero-configuration intelligence. Schema detection, KPI generation, and chart building require no setup from the user. The system infers the structure and content of every uploaded dataset automatically and produces appropriate outputs without manual configuration.

Accessibility for non-technical users. Command-based interaction allows accounting professionals to explore and manipulate their data using natural business language rather than technical filter menus or query syntax. This makes the platform genuinely usable by finance managers and business executives, not just data analysts.

Transparent forecasting. SHAP-based feature importance makes every prediction explainable in plain terms. Users can see exactly which variables are driving a forecast and communicate those drivers confidently to stakeholders — something that no major general-purpose BI tool currently provides at this level of integration [5][6].

Pipeline transparency. The staged progress view demystifies the analysis process for users who are not familiar with data pipelines. Rather than submitting a file and waiting for a black-box result, users observe each processing step as it completes, which builds trust in the output and surfaces data quality issues early.

Fast time-to-insight. As demonstrated in the case study, the platform reduces total analysis preparation time from approximately 250 minutes to 37 minutes for a typical monthly accounting task — an 85% reduction. This speed allows teams to run analyses more frequently, respond to data more quickly, and support decision-making on shorter cycles.

Collaboration and continuity. Persistent workspace storage and share tokens allow analysis sessions to be resumed and shared without re-running the pipeline, enabling asynchronous collaboration across teams.

Scalable modularity. The modular backend architecture means individual components — such as the forecasting engine, the chart builder, or the reporting module — can be upgraded or replaced independently as requirements evolve, without disrupting the rest of the system.

XIII. LIMITATIONS

The platform has several honest limitations that define the boundaries of its current contribution.

Structured data only. The system is designed exclusively for structured tabular datasets. It does not process scanned invoices, PDF bank statements, email-based financial records, or any other form of unstructured document. Accounting environments that rely heavily on such inputs would require OCR-based pre-processing before the platform could be applied.

Pragmatic forecasting models. The forecasting module uses linear regression and gradient boosting regressors from scikit-learn. These are practical, well-understood models that perform reliably on typical accounting datasets. However, they have not been benchmarked against specialized time-series approaches such as ARIMA, SARIMA, exponential smoothing, or Facebook Prophet, which may outperform them on datasets with strong seasonal patterns or long-range dependencies.

No formal user studies. The case study presented in this paper involved three users from a single organization. While the results are informative, a rigorous usability evaluation would require a controlled study with a larger and more diverse participant group including participants from different industries, organization sizes, and levels of technical background.

Infrastructure scale. The current implementation uses SQLite and local disk storage, which is appropriate for lightweight, single-server deployment but would not scale adequately for large enterprise environments with many concurrent users or very large datasets (millions of rows). A production deployment would require migration to a more scalable database and cloud-based file storage.

No real-time data connectivity. The platform currently operates in a batch-upload model. It does not connect to live accounting systems, ERP databases, or financial data APIs in real time. This limits its applicability for operational monitoring scenarios where continuously updated dashboards are needed.

No multi-user access control. The platform does not implement role-based access control, user authentication beyond basic session management, or audit trail logging. These are standard requirements for any production accounting system handling sensitive financial data.

Single-language interface. The command interpretation layer and all interface text are in English. Organizations operating in non-English languages would require localization of both the interface and the command parsing logic.

XIV. FUTURE WORK

The following directions represent the most valuable opportunities to extend and strengthen the platform.

OCR and document ingestion. Integrating an optical character recognition layer would allow the platform to process scanned invoices, photographed receipts, and PDF financial statements in addition to structured tabular files. This single enhancement would dramatically expand the range of real-world accounting inputs the system can handle, as many small and medium businesses still receive a significant portion of their financial documents in scanned or PDF form.

Advanced time-series forecasting. Replacing or supplementing the current regression models with purpose-built time-series approaches — ARIMA for stationary series, SARIMA for seasonal patterns, Facebook Prophet for trend-plus-seasonality decomposition, and LSTM networks for complex long-range dependencies — would improve forecasting accuracy for datasets with strong temporal structure. A formal evaluation pipeline comparing these models on representative accounting datasets would provide rigorous evidence for model selection guidance.

Anomaly detection module. Adding an automated anomaly detection layer that flags unusual transactions, unexpected expense spikes, or outlier entries in uploaded datasets would significantly increase the platform's value for audit support and fraud monitoring — two high-priority use cases in accounting analytics [7].

Real-time data connectivity. Connecting the platform to live data sources — including popular accounting software APIs such as QuickBooks, Xero, and Sage, as well as enterprise ERP systems — would enable continuously updated dashboards rather than batch-upload analysis, extending the platform's value from periodic reporting to real-time operational monitoring.

Enterprise security and access control. Implementing role-based access control, multi-factor authentication, field-level data masking, audit trail logging, and encrypted storage would make the platform suitable for enterprise deployment environments where financial data is subject to regulatory compliance requirements such as GDPR, SOX, and IFRS reporting standards.

Multi-language support. Extending the interface, command interpretation layer, and report templates to support multiple languages would make the platform accessible to accounting teams in non-English-speaking markets.

Formal usability evaluation. Conducting a controlled user study with a larger and more diverse participant group — including accountants, auditors, CFOs, and operations managers from multiple industries and organization sizes — would provide rigorous empirical evidence of the platform's real-world effectiveness and reveal usability improvements that could not be identified through system-level analysis alone.

Collaborative multi-user workspaces. Adding support for shared workspaces with concurrent editing, comment threads, version history, and role-based visibility would transform the platform from a single-user analysis tool into a collaborative business intelligence environment suitable for finance teams working together on shared datasets.

XV. CONCLUSION

This paper presented the design, development, case study evaluation, and comparative analysis of an AI-driven accounting analytics platform for interactive business intelligence. The system brings together multi-format data upload, staged automated analysis, schema-aware KPI and chart generation, command-based dashboard interaction, regression forecasting with SHAP explainability, workspace persistence, and consulting-style PDF report generation within a single full-stack environment.

The real-world case study demonstrated that the platform reduces total analysis preparation time by 85% for a typical monthly accounting task — from approximately 250 minutes using a conventional multi-tool workflow to 37 minutes within the proposed system — while producing forecasting results with a mean absolute percentage error of 3.04% across a three-month forward prediction horizon. The automatically generated KPI dashboard produced ten meaningful summary metrics within two seconds of pipeline completion, requiring no user configuration.

Comparison against Microsoft Power BI, Tableau, and Google Looker Studio revealed that the proposed platform uniquely combines automatic schema detection, zero-configuration KPI generation, command-based interaction, and SHAP-based forecasting explainability within a single accounting-focused system. No existing general-purpose BI tool currently offers all of these capabilities together.

The platform's primary contribution is architectural and integrative. It demonstrates that a practical, accessible, and comprehensive accounting analytics workflow can be delivered within a single modular system using well-established open-source technologies — React, FastAPI, Pandas, scikit-learn, SQLite, and ReportLab — without requiring any proprietary components.

As accounting data volumes continue to grow and the demand for faster, more transparent, and more accessible analytics increases, integrated platforms of this kind will play an increasingly important role in connecting raw business data to the managerial insight that organizations depend on. The work presented here establishes a solid foundation for that direction, and the future enhancements identified in Section XIV provide a clear path toward a production-ready enterprise system.

REFERENCES

[1] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact," MIS Quarterly, vol. 36, no. 4, pp. 1165–1188, 2012.
[2] D. A. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual analytics: Definition, process, and challenges," in Information Visualization, Berlin, Germany: Springer, 2008, pp. 154–175.
[3] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 1, pp. 682–692, Jan. 2019.
[4] K. Affolter, K. Stockinger, and A. Bernstein, "A comparative survey of recent natural language interfaces for databases," The VLDB Journal, vol. 28, no. 5, pp. 793–819, 2019.
[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.
[7] M. A. Vasarhelyi, A. Kogan, and B. M. Tuttle, "Big data in accounting: An overview," Accounting Horizons, vol. 29, no. 2, pp. 381–396, 2015.