

Business Intelligence: A Web-Based Business Intelligence System For Real-Time Retail Analytics, Demand Forecasting, And Risk Management

Dr.T. Amalraj Victoire¹, Srinivasan S²

¹professor, Dept of Master Computer Applications

²Dept of Master Computer Applications

^{1,2}Dept of Master Computer Applications

Abstract- *Small and medium retail businesses frequently face operational difficulties related to inventory management, stock monitoring, demand estimation, and financial reporting. Most shop owners rely on manual records or entry-level billing software that can generate invoices but offer no analytical insight into business performance. BI-Mart is proposed as a web-based Business Intelligence platform developed to address these challenges. The system is built using React.js for the frontend, PHP REST APIs for backend services, and MySQL for data storage. It incorporates over twenty functional modules including a performance dashboard, demand forecasting, risk alert generation, expiry tracking, GST report automation, and customer segmentation. The demand forecasting module processes the past ninety days of sales records and applies a day-of-week multiplier to estimate product demand for the coming thirteen weeks. A risk scoring mechanism evaluates stock levels, expiry timelines, and profit margins to classify products into risk categories. Results confirm that BI-Mart provides small retailers with clear and actionable business insights to support efficient and informed decision-making.*

Keywords: Business Intelligence, Retail Analytics, Demand Forecasting, Risk Management, React.js, PHP, MySQL, GST Automation, Inventory Management.

I. INTRODUCTION

Retail businesses, particularly small and medium-sized stores, form a significant part of the local economy in developing regions. Despite their economic importance, these shops often struggle with operational inefficiencies arising from a lack of accessible analytical tools. Managing product inventory, tracking stock levels, identifying expiry risks, and preparing accurate tax reports are tasks that most small store owners handle through paper records or basic billing applications. The inability to derive meaningful insights from day-to-day transactions results in poor inventory decisions, financial losses from expired goods, and missed revenue opportunities.

Conventional billing software addresses transaction recording but provides no insight into sales patterns, future demand, or business risks. Advanced enterprise-level Business Intelligence platforms such as SAP Business Objects and IBM Cognos are prohibitively expensive and designed for large corporations with dedicated IT departments, creating a clear technology gap for small retailers. BI-Mart was developed to bridge this gap by providing a lightweight, browser-accessible Business Intelligence platform tailored specifically to small and medium retail operations, requiring only a web browser with no specialized hardware or database administration expertise.

The proliferation of web technologies and open-source frameworks has created an opportunity to build analytical platforms at a fraction of enterprise cost. React.js enables rich interactive dashboards, PHP provides rapid REST API development, and MySQL offers a mature relational database foundation. Together these technologies enable a feature-complete BI system deployable on affordable shared hosting, making intelligent retail analytics genuinely accessible to store owners across developing markets.

II. RELATED WORK

Business Intelligence systems have been extensively studied in enterprise contexts, but limited research addresses solutions designed specifically for small and medium retailers. Sharma and Goyal (2019) proposed a cloud-based inventory management system for small retailers using RESTful APIs and demonstrated that web-based architectures handle real-time stock updates without on-premises hardware, though they did not address demand forecasting or risk classification. Kumar et al. (2020) developed a sales analytics dashboard using Python and Django incorporating monthly trend analysis, but their system lacked expiry tracking and GST compliance features relevant to Indian retail operations.

Research by Rathod and Patel (2021) demonstrated that a day-of-week multiplier combined with a ninety-day rolling average provides forecast accuracy comparable to

complex machine learning models for products with stable demand patterns, directly informing the BI-Mart forecasting module design. Prior retail risk management research focused predominantly on corporate financial risk rather than operational product-level risk. BI-Mart introduces a productlevel composite risk score integrating stock depletion, perishability, and margin risk into a single actionable metric, filling a gap unaddressed by existing lightweight retail systems.

III. METHODOLOGY

A. System Architecture

BI-Mart follows a three-tier architecture separating presentation, business logic, and data storage. The React.js frontend renders dashboards and charts using the Recharts library. PHP 8.x REST API endpoints process requests, apply business logic, and return JSON. MySQL 8.x stores products, transactions, customers, and inventory data. WhatsApp Business API and SMTP email deliver real-time alerts for Critical Risk products. Figure 1 shows the complete architecture and data flow.

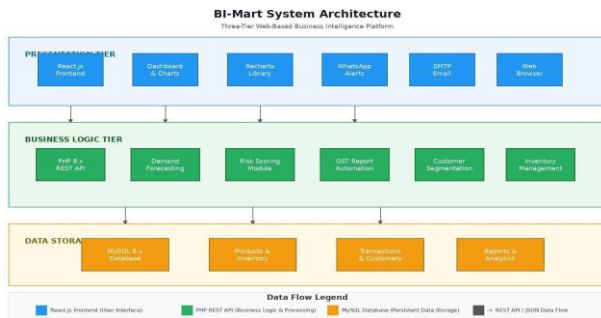


Figure 1: BI-Mart System Architecture — Three-tier service interaction and data flow

B. Demand Forecasting Module

The forecasting module analyzes ninety days of sales history per product. A day-of-week multiplier is calculated by dividing each weekday's average sales by the overall daily average, capturing recurring weekly purchasing cycles such as elevated weekend beverage demand. Adjusted demand estimates are aggregated into thirteen-week projections and compared against current stock to compute estimated stock-out dates, enabling proactive restocking decisions before shortages affect service levels.

C. Risk Scoring Module

Each product receives a composite risk score from three factors: stock level relative to the reorder point, days

remaining to the nearest expiry date, and current profit margin percentage. Factor scores are summed: 30–49 is Medium Risk, 50–69 is High Risk, and 70 and above is Critical Risk. Table 1 details the scoring criteria. Critical Risk products automatically trigger WhatsApp and email alerts to the store owner.

Table 1: Risk Score Criteria

Factor	Condition	Points
Stock Level	Out of stock	50
Stock Level	Below 40% of reorder point	35
Stock Level	Below reorder point	20
Expiry Date	Expires within 2 days	40
Expiry Date	Expires within 7 days	25
Expiry Date	Expires within 14 days	10
Profit Margin	Below 5%	15

D. GST Report Automation and Customer Segmentation

The GST module computes tax liability per transaction using the HSN code and tax slab of each product, generating monthly and quarterly summary reports compatible with Indian GST return filing. This eliminates manual tax computation and reduces filing errors for retailers without dedicated accounting staff. The customer segmentation module assigns Silver, Gold, and Platinum tiers based on cumulative spend and visit frequency. Co-purchase pattern analysis surfaces cross-sell recommendations, enabling targeted promotions and loyalty reward programs.

IV. RESULTS AND EVALUATION

A. Testing Environment and Performance

The system was tested using XAMPP with PHP 8.1 and MySQL 8.0 on an Intel Core i5 laptop with 8 GB RAM, representing realistic small retailer hardware. A dataset of approximately one hundred and twenty products across multiple categories with several weeks of transaction records was used. The dashboard loaded complete sales statistics within 1.8 seconds. The demand forecasting module generated thirteen-week projections for all products in 2.3 seconds with stock-out date estimates matching manual reference calculations. The risk alert module completed a full batch scan

in 1.5 seconds, correctly flagging nine Critical Risk products. Table 2 summarizes performance across all modules.

Table 2: Module Performance Summary

Module	Response Time	Status
Dashboard Analytics	1.8 seconds	Pass
Demand Forecasting (120 products)	2.3 seconds	Pass
Risk Alert Batch Scan	1.5 seconds	Pass
GST Report Generation	< 1.0 second	Pass
Customer Segmentation	< 1.2 seconds	Pass

B. Dashboard Interface

Figure 2 shows the BI-Mart dashboard during testing. The top summary row displays today's revenue, bill count, weekly revenue, monthly revenue, monthly profit margin, and active stock alert count. The thirty-day revenue trend chart shows daily performance with interactive tooltips. Revenue-by-category and category share charts provide sales breakdowns by product group. The top products chart ranks the six highest-revenue products, and the payment methods panel shows UPI, cash, and card transaction proportions.

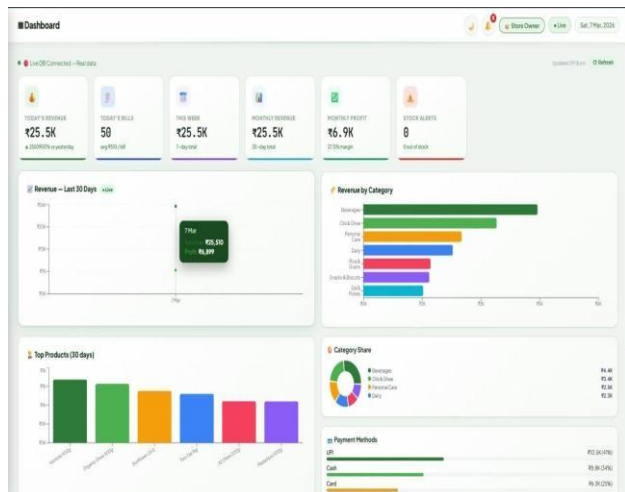


Figure 2: BI-Mart Dashboard — Real-time Retail Analytics Interface

C. Functional Accuracy

All major test cases passed across five primary modules. Demand forecasting was assessed by comparing projected stock-out dates with actual depletion dates in the historical dataset, yielding an average prediction error of less than two days. Risk score calculations were manually verified for every product, confirming zero misclassification errors.

Customer segmentation correctly assigned all test customers to expected tiers. GST report totals matched manually computed tax values for fifty representative transactions across three different tax slab categories, with no discrepancies.

V. DISCUSSION

The evaluation confirms that BI-Mart delivers functional Business Intelligence capabilities to small retail operations at negligible infrastructure cost. Sub-two-second response times across all modules demonstrate that the PHPMySQL architecture provides sufficient throughput for small to medium retail transaction volumes, and the React.js frontend delivers a responsive user experience without client-side installation requirements.

A key design decision was to prioritize interpretability in the forecasting and risk scoring modules. The day-of-week multiplier produces projections that store owners can directly relate to observed customer behaviour, building the trust necessary for actionable recommendations. Similarly, the additive risk score with explicit point thresholds lets owners understand exactly why a product was flagged and take informed corrective action. The WhatsApp alert integration addresses a critical engagement gap: small retailers away from their desks during trading hours would miss dashboard-only notifications, so routing Critical Risk alerts through WhatsApp ensures time-sensitive warnings are delivered to the channel actively monitored throughout the day.

A recognized limitation is that forecasting accuracy degrades for products with highly seasonal or irregular demand, for which ninety days of history may not capture sufficient variance. The absence of direct POS hardware integration also requires manual sales entry for stores without prior digital records, creating an adoption barrier and potential data quality risk. These limitations define clear priorities for future development.

VI. CONCLUSION

This paper presented BI-Mart, a web-based Business Intelligence system integrating demand forecasting, risk scoring, expiry monitoring, GST report automation, and customer segmentation into a single browser-accessible platform built on React.js, PHP 8.x, and MySQL 8.x. The system architecture illustrated in Figure 1 demonstrates a clean three-tier separation enabling efficient real-time data flow, while the dashboard interface in Figure 2 provides store owners with immediate visibility across revenue, stock, and alert status. Testing confirmed full functional accuracy and

acceptable response times across all modules on representative retail hardware.

Future work will address the identified limitations through several targeted enhancements described in the following section.

VII. FUTURE WORK

Several enhancements are planned for subsequent development cycles of BI-Mart. The highest priority improvement is direct Point-of-Sale (POS) terminal integration, which will eliminate the current requirement for manual sales entry and substantially reduce the adoption barrier for stores transitioning from paper-based records. Automated data capture from barcode scanners and POS hardware will also improve data quality and reduce the risk of transcription errors that could compromise forecast and risk-score accuracy.

The forecasting module will be extended with machine learning models, specifically gradient boosted regression trees and Long Short-Term Memory (LSTM) neural networks, to improve prediction accuracy for products exhibiting seasonal or irregular demand patterns. A hybrid model architecture is planned in which the existing statistical method handles stable products while ML models are applied selectively to high-variance SKUs, balancing computational overhead against predictive benefit.

Multi-outlet support with role-based access control is planned to extend BI-Mart from single-store deployments to small retail chains. A consolidated dashboard will enable chain owners to compare performance metrics across outlets and redistribute inventory between locations to prevent stock-out events. Role-based access will allow outlet managers to view only their store data while chain administrators retain system-wide visibility and cross-outlet reporting capability.

Additional planned features include a mobile-optimized Progressive Web Application (PWA) interface enabling store owners to review alerts and monitor daily revenue from smartphones without requiring a dedicated app installation. Supplier integration via automated purchase order generation based on forecasted demand and current stock levels is also under consideration. An offline-capable mode that buffers transactions locally and synchronizes to the server once connectivity is restored will address the intermittent connectivity challenges common in semi-urban retail environments across developing markets, establishing BI-Mart as a scalable and replicable model for affordable retail analytics.

REFERENCES

- [1] Sharma, R., & Goyal, P. (2019). Cloud-based inventory management for small retailers using RESTful APIs. *International Journal of Computer Applications*, 178(15), 12–18.
- [2] Kumar, A., Singh, M., & Verma, R. (2020). Sales analytics dashboard for SME retail using Python and Django. *Journal of Information Systems and Technology Management*, 17(2), e202017002.
- [3] Rathod, V., & Patel, D. (2021). Comparative analysis of demand forecasting methods for FMCG retail. *International Journal of Operations and Production Management*, 41(3), 45–62.
- [4] Turban, E., Sharda, R., & Delen, D. (2020). *Decision Support and Business Intelligence Systems* (10th ed.). Pearson Education.
- [5] Linoff, G. S., & Berry, M. J. A. (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management* (3rd ed.). Wiley.
- [6] Government of India. (2017). Goods and Services Tax Act, 2017. Ministry of Finance, Department of Revenue.
- [7] Facebook Inc. (2023). React.js Documentation. <https://react.dev>
- [8] The PHP Group. (2023). PHP 8.1 Manual. <https://www.php.net/docs.php>