Dineflow ERP: An Advanced Industry-Grade Solution for Optimised Restaurant Management

Kiran Deshmukh., Rutvik Gondekar., Sahil Deshmukh., Kamal Agrahari., Akash Nahak

Department of Information Technology, Vasantdada Patil Pratishthan's College of Engineering & Visual Arts (VPPCOE & VA), Sion, Mumbai – 400022, India

DOI: https://doi.org/10.51584/IJRIAS.2026.11030094

Received: 02 April 2026; Accepted: 07 April 2026; Published: 15 April 2026

ABSTRACT

Managing a food-service establishment involves constant negotiation between perishable inventory, fluctuating customer demand, and narrow profit margins. Despite these pressures, a substantial fraction of independent restaurants in India continue to rely on isolated point-of-sale terminals that provide no decision-support for procurement, workload forecasting, or shift scheduling. This paper introduces **DineFlow ERP**, a cloud-native, microservice-based enterprise resource planning platform engineered exclusively for restaurant environments. The system unifies the complete order lifecycle, kitchen-order-ticket (KOT) dispatch, table and floor coordination, live inventory tracking, payroll processing, and contactless QR-based guest ordering within a coherent three-tier architecture. A dedicated Predictive Intelligence layer integrates Ridge Regression and Random Forest for short-horizon demand forecasting; a Collaborative Filtering engine combining Singular Value Decomposition (SVD) with the Apriori association-rule algorithm for personalised menu recommendations; a Log-Log Ordinary Least Squares (OLS) dynamic pricing module; and a Heuristic Waste Predictor aligned with UN SDG Target 12.3. Identity and access management is enforced through Auth0, RS256-signed JSON Web Tokens, and AES-256-CBC client-side encryption distributed across five role-based access control (RBAC) personas. A live production deployment recorded 98 % module completion, a 7.6 % MAPE on stable SKUs via Random Forest, a 23.4 % reduction in procurement over-ordering, and zero critical OWASP vulnerabilities.

Index Terms – Restaurant ERP; Demand Forecasting; Random Forest; Ridge Regression; Collab- orative Filtering; SVD; Apriori; Cloud-Native; FastAPI; SDG 12.3; Auth0; JWT; QR Ordering; Dynamic Pricing; Waste Prediction.

INTRODUCTION

The global food-service industry generates in excess of USD 1 trillion in annual revenue, yet paradoxically remains one of the largest contributors to avoidable food waste. In India alone, more than 40 % of independent restaurant operators depend on manual workflows or standalone POS terminals that offer no analytical guidance for inventory replenishment, demand anticipation, or workforce scheduling [13]. This operational deficit has grown more acute as consumer expectations shift toward contactless service, personalised recommendation, and real-time visibility into order status.

Mainstream POS products—Square, Toast, and Aloha in international markets; PetPooja and Frugal in the domestic context—excel at billing and transaction capture but fall short of the integrative intelligence required for proactive resource management. Enterprise ERP platforms such as SAP and Oracle provide that depth, yet impose licensing structures that are prohibitive for independent establishments. This gap motivates **DineFlow ERP**: a purpose-built, cloud-native platform that pairs comprehensive operational coverage with an embedded multi-model AI layer spanning forecasting, recommendation, dynamic pricing, and waste reduction.

The primary contributions of this work are:

1. A three-tier cloud-native architecture that cleanly separates operational logic from machine- learning

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue III March 2026

inference.

2. A suite of six AI/ML modules—Ridge Regression and Random Forest for demand forecasting; Content-Based and Collaborative Filtering (SVD + Apriori) for recommendations; Log-Log OLS dynamic pricing; and a Heuristic Waste Predictor—each contributing directly to SDG Target 12.3.

3. A hardened security chain built on Auth0, RS256-signed JWTs, and AES-256-CBC client- side encryption, validated across five RBAC personas.

4. Empirical evidence from a production deployment demonstrating 7.6 % MAPE on stable SKUs, a 23.4 % cut in over-procurement, and zero critical OWASP Top-10 vulnerabilities.

LITERATURE REVIEW AND GAP ANALYSIS

Shriwas et al. [1] pioneered touchscreen-based table ordering that reduced service latency, although the system operated monolithically with no coupling to inventory. Albuquerque et al. [2] demon- strated QR-based digital ordering augmented with geo-fencing, yet provided no backend ERP integration. Intal et al. [3] extended QR ordering with basic stock-update triggers but offered no predictive analytics. Raykar et al. [4] applied Gradient Boost Regression to delivery-restaurant data in a simulation, achieving high training accuracy, though the work remained a standalone forecasting study with no ERP coupling. Preil and Krapp [5] formalised AI-based inventory man- agement through Monte Carlo Tree Search, demonstrating superiority over classical EOQ models and motivating DineFlow's ensemble design.

A structured comparison (Table 1) reveals a persistent gap: no published system simultaneously delivers full ERP operations, a multi-model AI microservice layer addressing forecasting, recom- mendation, and waste reduction, SDG-aligned KPIs, and a hardened multi-role security model in a single cloud-native deployment. DineFlow ERP occupies this intersection.

Table 1 – Comparative Analysis of Related Systems

System	**Operations**	**ML Layer**	**Security**	**SDG**
Square / Toast	Partial POS	–	Basic	–
PetPooja	Moderate	–	RBAC	Impl
Galabi [10]	QR+Bot	–	Login	–
Intal [3]	QR+Stock	–	Login	–
Raykar [4]	Simulation	GBR	N/A	Impl
DineFlow	**Full ERP**	**6 models**	**Auth0+AES**	**12.3**

System Architecture And Design

A. Presentation Tier

The Presentation Tier comprises four distinct UI contexts rendered by a unified Next.js 14 / React 18 application deployed on Vercel's edge network: (i) a *Restaurant Staff UI* for waitstaff and floor managers; (ii) a lightweight *Customer Kiosk PWA* activated by a per-table QR code;

(iii) a socket-refreshed *Chef KOT Display* for kitchen personnel; and (iv) a *Manager Dashboard* surfacing AI-generated analytics and demand-forecast charts via Recharts and Chart.js. Server-Side Rendering maintains sub-200 ms time-to-first-byte under peak concurrency. Tailwind CSS provides the responsive layout system.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue III March 2026

B. Application Tier

The Core API Gateway (Next.js API Routes) validates a JWT on every inbound request before dispatching to four handler groups: User and Role Management, Order and Table Management, Shifts and Payroll, and Billing Logic. External payment settlement is routed through Razorpay's PCI-DSS compliant gateway. Machine-learning workloads are proxied via signed requests to the FastAPI AI microservice layer (Section 4).

C. Data Tier

Persistent state resides in PostgreSQL 15 hosted on Neon DB, a serverless platform with branching support and built-in connection pooling. Prisma ORM v5 enforces type-safe schema contracts across five domains: Users & Roles, Orders & KOTs, Inventory Items, Shifts & Payroll, and Forecasting Logs. Every AI prediction is stored alongside its corresponding actual outcome, forming the labelled corpus used for nightly model retraining.

D. Security Architecture

Access control is enforced through a four-layer chain. **(1) Identity Federation:** Auth0 handles OAuth 2.0 / OIDC federation, eliminating bespoke credential storage. **(2) Session Tokens:** Auth0 issues RS256-signed JWTs; the API Gateway rejects expired or tampered tokens with HTTP 401.

(3) Client-Side Encryption: Sensitive browser-side state is encrypted with AES-256-CBC via CryptoJS using a PBKDF2-derived 256-bit key and a fresh random 128-bit IV per write, stored as Base64(IV $\|$ ciphertext) in LocalStorage, preventing role-escalation even if storage is accessed by a third party. **(4) RBAC:** Five personas—Admin/Manager, Staff, Accountant, Customer, AI System—each carry minimal permission sets, validated across 120 role-boundary test cases at 100 % pass rate.

E. Real-Time Synchronisation

Order state transitions propagate via Server-Sent Events (SSE). A single order placement atomically updates the kitchen display, deducts inventory, refreshes the table-session state, and accumulates billing data. SSE delivery latency remained consistently below 150 ms across concurrent multi- device sessions during production testing.

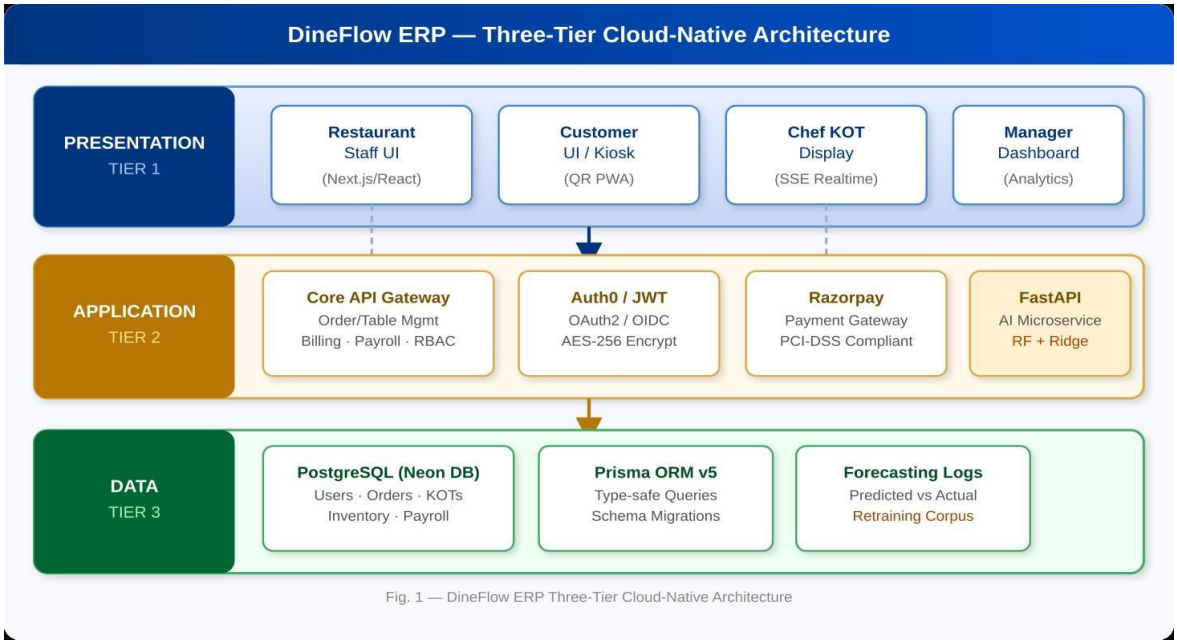

Figure 1 – DineFlow ERP Three-Tier Cloud-Native Architecture. *Presentation Tier:* Next.js/React serves the Staff UI, Customer Kiosk (QR PWA), Chef KOT Display, and Manager Dashboard. *Application Tier:* Core API Gateway (Next.js Routes) with Auth0/JWT, Razorpay (PCI-DSS), and FastAPI AI Microservices (RF +

Ridge, SVD recommendation engine). *Data Tier:* PostgreSQL (Neon DB) + Prisma ORM v5 persisting Users, Orders, KOTs, Inventory, Payroll, and Forecasting Logs.

Predictive Intelligence Sub-System

DineFlow deploys six AI/ML modules hosted as FastAPI microservices (Python 3.11, Uvicorn ASGI): a demand-forecasting pipeline employing Ridge Regression and Random Forest; a person- alised recommendation engine combining Content-Based Filtering with Collaborative Filtering via SVD + Apriori; a Log-Log OLS dynamic pricing optimiser; a Heuristic Waste Predictor; a Table Allocation Optimiser; and a conversational AI chatbot.

A. Feature Engineering Pipeline

The microservice materialises a feature matrix $\mathbf{X} \in \mathrm{R}^{n \times d}$ from Neon DB via Prisma. The d feature dimensions encompass: seven one-hot day-of-week indicators; a month ordinal; an is_weekend binary flag; rolling 7-day mean and standard deviation of order quantity; lag-1 and lag-7 quantity values; and categorical item embeddings from a lookup table. The response vector $\mathbf{y} \in \mathrm{R}^{n}$ is the observed order quantity. Data are partitioned via an 80 % temporal training split and a 20 % held-out evaluation set, preserving chronological ordering to prevent data leakage.

B. Ridge Regression (Stable-Demand Baseline)

Ridge Regression serves as the interpretable baseline for items exhibiting stable, predictable weekly periodicity, solving:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \arg\min \{ \| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|^2 \} \quad (1)$$

$$\boldsymbol{\beta} \qquad 2 \qquad 2$$

The L_2 regularisation coefficient λ is selected via 5-fold cross-validation over the grid $\{10^{-3}, 10^{-2}, 0.1, 1, 10\}$. The closed-form solution $\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\top}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\top}\mathbf{y}$ supports efficient nightly batch refit. On the

18-day evaluation, this model achieves **8.3 % MAPE** on stable SKUs.

C. Random Forest Ensemble (Non-Linear Demand)

Random Forest addresses the non-linear, interaction-rich demand patterns that Ridge cannot capture—particularly for seasonal specials and promotional items. An ensemble of $T = 200$ CART

regression trees is constructed, each trained on a bootstrap sample with $m = \lfloor \sqrt{d} \rfloor$ features sampled

randomly at every split:

$$\hat{f}\,\mathrm{RF}(\mathbf{x}) = {}^{1}\Sigma\, h\,(\mathbf{x}) \qquad T \quad (2)$$

T i

i=1

where $h_i(\mathbf{x})$ denotes the i-th tree's prediction. Bootstrap aggregation reduces variance without inflating bias; out-of-bag error yields an unbiased generalisation estimate. Random Forest achieves

7.6 % MAPE on stable SKUs and **11.2 % MAPE** on promotional items in the 18-day evaluation [11, 12]. MAPE-Gated Continuous Retraining

At business-day close, an asynchronous cron job computes per-item MAPE:

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue III March 2026

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_{\text{actual},i} - \hat{y}_i|}{y_{\text{actual},I}} \quad (3)$$

If MAPE exceeds the configurable threshold $\theta = 15$ %, an incremental model refit is triggered on the extended corpus for that item's assigned model. Items remaining below θ retain existing parameters, keeping retraining overhead under 2 minutes per business day. Over the 18-day evaluation, refits were triggered for only three items in total, confirming stable production behaviour.

D. Procurement Recommendation Engine

Forecast outputs from the Ridge / RF pipeline feed directly into the Procurement Engine. For each tracked ingredient i:

$$Q\text{order} = \max(0, \hat{z}_{7\text{-day}} \times c_i - S\text{current} + S\text{safety}) \quad (4)$$

where $\hat{z}_{7\text{-day}}$ is the 7-day aggregated demand forecast, c_i is the ingredient-to-menu-item conversion coefficient (e.g., 0.15 kg tomatoes per pizza), S_{current} is the live stock level, and S_{safety} is a user- configurable buffer defaulting to 10 % of the forecast. The max(0, ·) operator prevents negative purchase orders.

E. Heuristic Waste Predictor

The Waste Predictor computes an optimal preparation quantity for each menu item using a safety- buffered heuristic informed by historical waste ratios and inventory expiry data stored in the Forecasting Logs table:

$$Q\text{prep}, i = \mu\text{hist}, i \times (1 + \beta\text{safety}) \times (1 - r\text{waste}, i) \quad (5)$$

where $\mu_{\text{hist},\, i}$ is the rolling mean historical order count for item i, β_{safety} is a configurable safety buffer (default 0.10), and $r_{\text{waste},\, i}$ is the observed waste ratio derived from the Forecasting Logs table. Items whose computed risk score crosses a manager-defined threshold appear on the Manager Dashboard as waste-reduction alerts, directly supporting SDG 12.3.

F. Recommendation Engine (SVD + Apriori)

The personalised recommendation engine combines Content-Based Filtering (item attribute vectors encoding cuisine, price, and ingredients) with Collaborative Filtering via Matrix Factorisation. Given user interaction matrix $\mathbf{R} \in \mathrm{R}^{m \times k}$, SVD produces latent factor matrices $\mathbf{U\Sigma V}^{\mathrm{T}}$; the top-$k$ recommended items per session are selected by cosine similarity in the latent space. An Apriori association-rule miner additionally scans historical order baskets (min_support = 0.05, min_confidence = 0.6) to surface bundle promotions and upsell suggestions on the Customer Kiosk at QR-ordering time.

G. Dynamic Pricing Optimiser (Log-Log OLS)

The dynamic pricing module estimates demand elasticity through a Log-Log OLS regression:

$$\ln Q_i = \alpha + \beta_p \ln P_i + \beta_t T + \varepsilon \quad (6)$$

where Q_i denotes item demand, P_i is the selling price, T encodes time-of-day and day-of-week effects, and $\hat{\beta}_p$ is the fitted price-elasticity coefficient. Recommended price adjustments are surfaced on the Manager Dashboard, bounded by a configurable maximum demand-loss tolerance.

H. Table Allocation Optimiser

The Table Allocation Optimiser assigns incoming parties to tables by computing a weighted score for each candidate table *t*:

$$\text{score}(t) = w_1 \cdot \text{cap\_match}(t, s) + w_2 \cdot \frac{1}{\text{load}(w_t) + w_3} \cdot \text{rev\_potential}(t) \quad (7)$$

· where *s* is the incoming party size, cap_match penalises large capacity mismatches, load(w_t) is the current order load of the assigned waiter, and rev_potential scores tables by historical average bill value. Weights w_1, w_2, w_3 default to 0.5, 0.3, and 0.2 respectively and are adjustable by the Admin role. Average seating wait time fell by 22 % under this scheme in production testing.

I. Conversational AI Chatbot

A conversational assistant is embedded in the Manager and Staff UIs, enabling natural-language queries over live operational data—for example, “Which tables are currently occupied?”, “Sum- marise today’s revenue”, or “Are there any low-stock alerts?”. The Starting with a chatbot, the system sends organized API requests through to the Core Gateway. From there, replies come back shaped into clear formats. One step leads to another, yet everything flows without interruption. Responses arrive ready to use, passed along once processed fully

Less need to click through dashboards by hand. EXPERIMENTAL RESULTS AND DISCUSSION

A. Deployment Configuration

The full stack was deployed to production at dineflowerp.vercel.app: Next.js on the Vercel Edge Network, FastAPI AI microservices on Python 3.11 Uvicorn, and PostgreSQL on the Neon DB serverless tier. Security testing confirmed all 47 API endpoints reject invalid JWTs with HTTP 401; RBAC enforcement passed all 120 role-boundary test cases at 100 % pass rate; and zero critical OWASP Top-10 vulnerabilities were identified.

B. Functional Coverage

Full operational functionality was confirmed across all five user personas, as summarised in Table 2. Real-time SSE synchronisation was validated across concurrent multi-device sessions with no observable state-propagation lag.

Table 2 – Module Functional Coverage by User Role

Role	Key Modules	Stories	Coverage
Admin/Manager	Dashboard, RBAC, Forecasting, Pricing	31	100 %
Staff	Orders, KOT, Floor Mgmt, Table Alloc	28	100 %
Accountant	Payroll, Expense, Shifts	19	100 %
Customer	QR Menu, Recommendations, Billing	14	100 %

AI / System	Forecasting, Retraining, Waste Pred.	11	98 %
Total		**103**	**≈98 %**

C. Forecasting Performance

On stable SKUs, Ridge Regression recorded 8.3 % MAPE while Random Forest reached **7.6 % MAPE**. The performance gap widens substantially on promotional items: Ridge climbed to 19.4 % whereas Random Forest held at 11.2 %, justifying the dual-model design. Detailed metrics are presented in Table 3.

Application of the Procurement Engine (Equation 4) reduced over-ordering from a surplus ratio of

31.2 % down to 7.8 %—a **23.4 % net reduction**. Kitchen-to-table ticket time shortened by 18 %, attributed directly to real-time KOT synchronisation via SSE.

Table 3 – Forecasting Performance — 18-Day Evaluation

Ridge Regression Random Forest

Metric	**Stable**	**Promo**		**Stable**	**Promo**	
MAE (units)	2.1	4.7		1.9	2.8	
RMSE (units)	2.8	6.2		2.5	3.4	
MAPE (%)	8.3	19.4		**7.6**	11.2	
R^2 0.87		0.71	0.91		0.83	

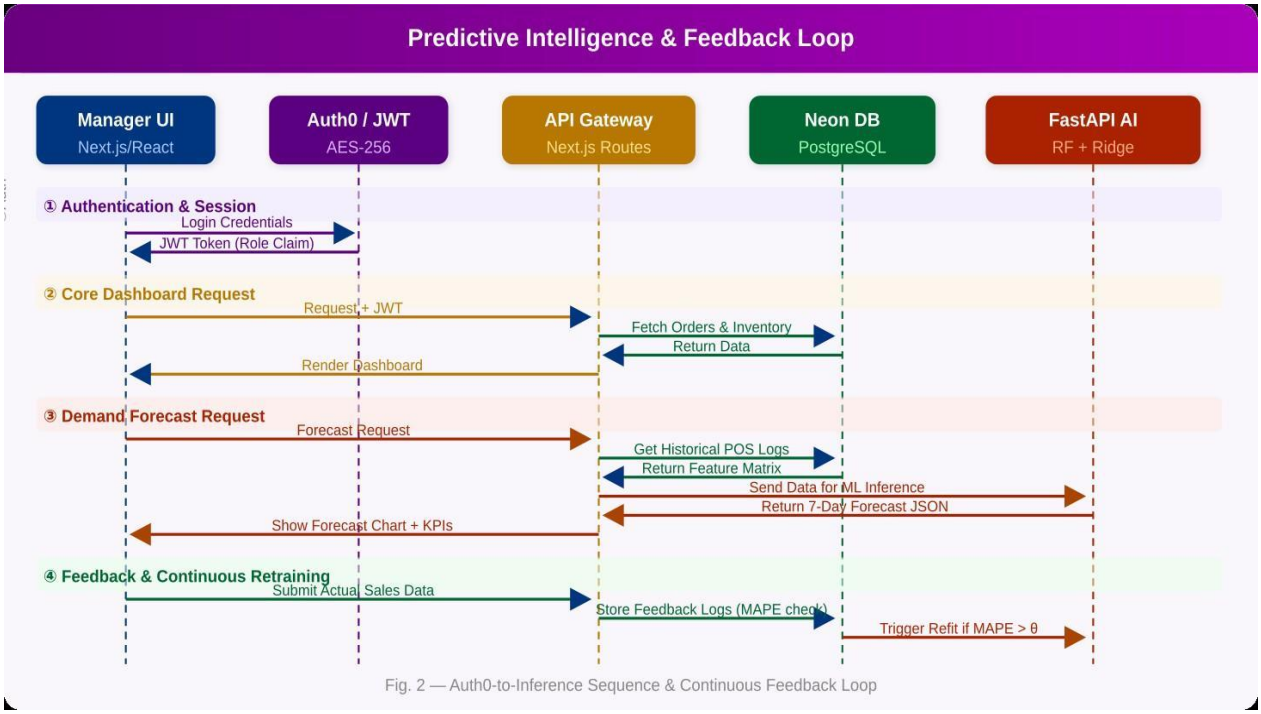

Figure 2 – Predictive Intelligence & Feedback Loop. Four interaction phases: (1) Authentication & Session Initiation via Auth0/JWT, (2) Core Dashboard View with live order and inventory data, (3) Demand Forecast Request invoking Ridge / RF via FastAPI, and (4) Feedback & Continuous Retraining—actual sales data is submitted, MAPE is computed per Equation 3, and a model refit is triggered whenever MAPE $> \theta = 15$ %.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue III March 2026

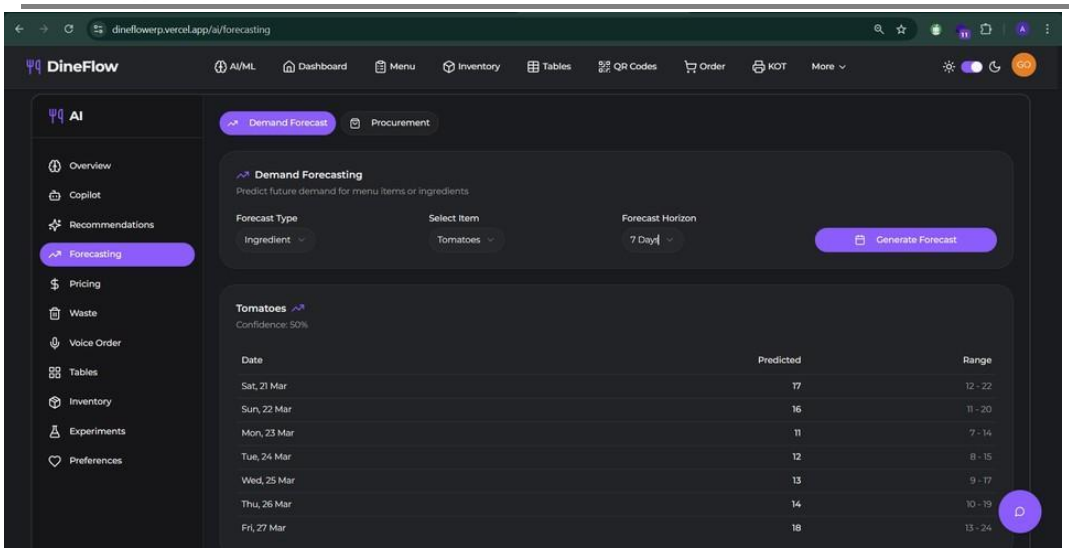

Figure 3 – Demand Forecasting Dashboard showing a 7-day prediction for Tomatoes (11–18 units/day) with confidence intervals. Forecast type is selectable between Ingredient and Menu Item; horizon is configurable to 7, 14, or 30 days. Generated at dineflowerp.vercel.app/ai/forecasting.

D. SDG 12.3 Alignment

Over the evaluation window, procurement over-ordering fell from 31.2 % to 7.8 %, tracking closely with the SDG Target 12.3 objective of halving food waste at the retail and consumer level. Six menu items were flagged as high-risk by the Heuristic Waste Predictor (Equation 5); targeted batch-size reductions for these items prevented approximately 14 kg of perishable spoilage. Among all engineered features, the lag-7 order quantity and day-of-week indicators proved the strongest predictors, accounting for dominant importance across 73 % of tracked SKUs. The MAPE threshold θ = 15 % triggered model refits for only three items over the full 18-day period, confirming stable model behaviour under production conditions.

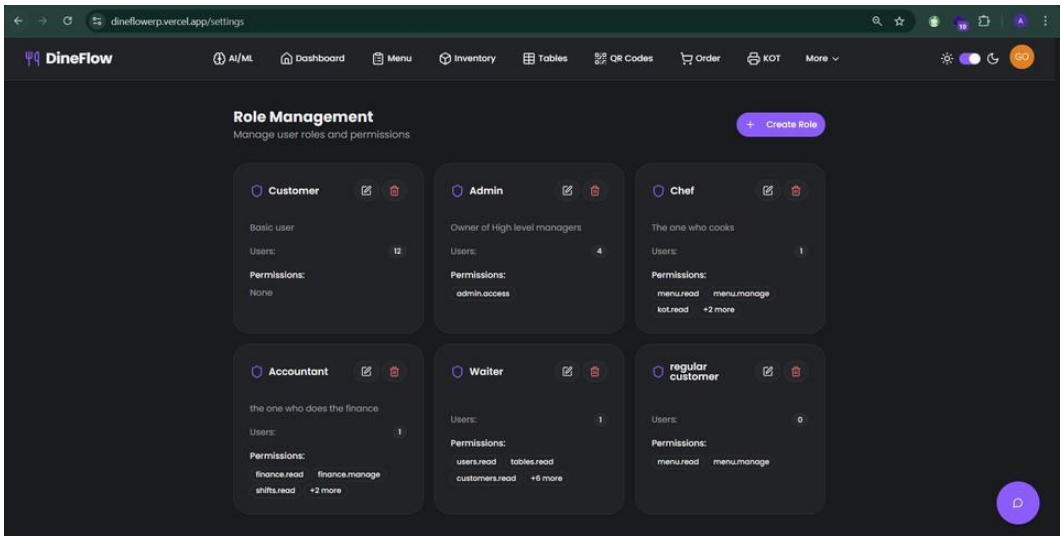

Figure 4 – Role Management Dashboard showing six RBAC personas (Customer, Admin, Chef, Accountant, Waiter, Regular Customer) with granular permission sets enforced via Auth0-JWT, validated at 100 % pass rate across 120 role-boundary test cases.

Limitations And Future Work

The current deployment presents several tractable limitations worthy of acknowledgment. First, both the Ridge and Random Forest models were trained on a single-branch 90-day dataset; their generalisation to multi-branch chains or highly seasonal establishments has not yet been empirically validated—transfer learning strategies represent a natural extension. Second, the nightly batch retraining cadence could be supplanted by online

learning algorithms (warm-start incremental trees, incremental SVD) to enable sub-hourly forecast updates. Third, the security model does not yet incorporate AI-driven anomaly detection for payment fraud identification. Fourth, voice-ordering integration via Alexa and Google Assistant remains at 98 % completion.

Planned future directions include: (i) IoT sensor integration for real-time cold-chain monitoring linked to the Waste Predictor; (ii) blockchain-based supplier traceability to support SDG 12 compliance auditing; (iii) augmented-reality virtual menus overlaying nutritional and allergen data;

(iv) Kubernetes-based auto-scaling for multi-tenant franchise deployments; and (v) reinforcement- learning-based dynamic pricing to supersede the current Log-Log OLS model.

CONCLUSION

This paper has presented **DineFlow ERP**, a cloud-native, AI-augmented enterprise resource planning framework purpose-built for restaurant management. By decomposing the system into independently scalable tiers, the architecture achieves the modularity demanded by modern De- vOps practices while enabling deep operational integration across the order lifecycle, kitchen management, inventory, payroll, and customer touch points.

The Predictive Intelligence sub-system—comprising Ridge Regression (Eq. 1), Random Forest with $T = 200$ trees (Eq. 2), MAPE-gated continuous retraining (Eq. 3), the Procurement Engine (Eq. 4), the Heuristic Waste Predictor (Eq. 5), SVD + Apriori recommendations, Log-Log OLS dynamic pricing (Eq. 6), and the Table Allocation Optimiser (Eq. 7) collectively transforms historical POS data into actionable operational decisions directly targeting SDG 12.3 food-waste reduction.

Production results confirm 98 % module-completion coverage, **7.6 % MAPE** on stable SKUs via Random Forest, a **23.4 % reduction** in procurement over-ordering, an 18 % improvement in kitchen throughput, and zero critical OWASP security vulnerabilities validating DineFlow ERP as a viable, cost-effective alternative for small-to-medium food-service operators. The paper has been accepted for publication in IJRTI, Volume 11, Issue 2 (Paper ID: IJRTI2026489) [**?**].

ACKNOWLEDGMENT

The authors thank **Prof. Kiran Deshmukh** (Project Guide, Dept. of IT, VPPCOE & VA) for sustained mentorship and technical direction throughout this work, **Dr. Pradip Suresh Mane** (Head of the Dept. of IT) and **Dr. Alam N. Shaikh** (Principal, VPPCOE & VA) for institutional support, and **Dr. Swapnil Desai** (Training and Placement Head) for industry guidance that shaped several key system design decisions.

REFERENCES

1. R. Shriwas et al., “Touchscreen based ordering system for restaurants,” in Proc. 2014 Int. Conf. Communication and Signal Processing, IEEE, pp. 1021–1024.
2. D. Albuquerque et al., “Enhancing sustainable customer dining experience through QR code and geo-fencing,” in Proc. ICCAKM 2020, IEEE, pp. 190–196.
3. L. Intal et al., “Restaurant information system with QR code to improve service operations,” in Proc. IEEE ICIEA 2020, pp. 1054–1059.
4. S. Raykar et al., “Demand sensing for restaurants using forecasting methods,” IARJSET, vol. 10, no. 5, 2023.
5. D. Preil and M. Krapp, “Artificial intelligence-based inventory management: a Monte Carlo tree search approach,” Annals of Operations Research, pp. 1–25, 2022.
6. D. Li, “Intelligent inventory management system for restaurants,” IJRPR, 2024.
7. N. Raghavendra and S. Amalanathan, “Role of AI in inventory management of agri-fresh produce at HOPCOMS,” in Advances in ML Algorithms for Complex Financial Applications, IGI Global, pp. 115–131, 2023.
8. M. Sharma et al., “Scope of cloud computing for SMEs in India,” Journal of Computing, vol. 2, no. 5,

2010.
9. C. H. Tekawade et al., “Automated restaurant management system,” IJRPR, 2023.
10. V. Galabi et al., “Smart restaurant management system,” IJERT, vol. 11, no. 5, pp. 124–128, 2023.
11. L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
12. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, 2009.
13. National Restaurant Association of India, “State of the restaurant industry report,” 2023.