iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85354

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

MedProcure: An AI-Enabled Hospital Procurement Management System with Multi-Level Approval Workflow

Sabarinathan R S[1], Udhaya Kumar M G[2], Sathya Narayanan K S[3], Sangamithra S[4]
[1,2,3]*Student,* [4]*Assistant Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai, Tamil Nadu*

Abstract: ***Hospital procurement involves purchasing medicines, equipment and consumables while keeping stock available, spending controlled and every decision accountable. In many hospitals this process still relies on manual requests, informal approvals and periodic stock checks. That leads to delayed purchases, understocking of critical items, overstocking that wastes budget, and poor traceability. To address these problems, MedProcure, a web-based Hospital Procurement Management System, has been developed. It manages the full procurement lifecycle: purchase request, configurable multi-level approval, purchase order, supplier response, goods receipt and inventory update. The system has a secure role-based access control mechanism, master data management for items, suppliers, manufacturers, categories, units and departments, and an approval matrix that routes each request by department, category and amount. The platform is built with Angular for the user interface and ASP.NET Core Web API for backend processing. Data is stored in a MySQL database accessed through Entity Framework Core, and JWT authentication protects the system. To make procurement more intelligent, the system is being extended with machine learning modules. An inventory understock and overstock analysis module uses XGBoost to classify stock status and recommend procurement actions. A seasonal demand prediction module uses the Temporal Fusion Transformer, a time-series model that handles seasonality and shows which variables influenced a forecast. A smart purchase request review module uses LightGBM to compare a requested quantity against current stock and recent consumption and then recommend an appropriate quantity. The AI modules only recommend, and the human approver keeps the final decision. Together, these features aim to reduce stockouts and excess inventory, speed up approvals and support better decisions in hospital supply management.***

Index Terms: ***Hospital Procurement, Multi-Level Approval Workflow, Inventory Management, Role-Based Access Control, ASP.NET Core, Angular, XGBoost, Temporal Fusion Transformer, LightGBM, Decision Support***

I. INTRODUCTION

Hospitals depend on a continuous supply of medicines, medical equipment, and consumables. Procurement is the function that keeps this supply available while controlling expenditure and keeping every decision accountable. In many hospitals, however, procurement still relies on manual purchase requests, informal approvals, and periodic stock checks. Such practices lead to delayed purchases, understocking of critical items, overstocking that ties up budget, and weak traceability of who requested, approved, and received which item.

A web-based procurement management system addresses these problems by bringing the full lifecycle — purchase request, approval, purchase order, supplier response, goods receipt, and inventory update — onto a single platform with role-based access and a recorded decision trail. Workflow digitization alone, however, does not tell a hospital how much to order or when. Approvers must still judge a requested quantity without a quantitative view of current stock, recent consumption, and seasonal demand. A practical system must therefore combine workflow control with data-driven decision support.

Machine learning can provide this support. Gradient-boosted decision trees such as XGBoost and LightGBM perform well on structured tabular data [3], [4], while the Temporal Fusion Transformer (TFT) forecasts multi-horizon time series with seasonality and indicates which input variables influenced a forecast [5]. In a hospital setting, such recommendations must remain advisory: the human approver, not the model, stays accountable for the final purchasing decision.

This paper proposes MedProcure, a web-based Hospital Procurement Management System built with Angular for the user interface and ASP.NET Core Web API for backend processing.

©IJRASET: All Rights are Reserved

Data is stored in MySQL through Entity Framework Core (EF Core), and JWT authentication protects the system. MedProcure provides role-based access control, master data management for items, suppliers, manufacturers, categories, units, and departments, and an approval matrix that routes each request by department, category, and amount. The system is being extended with three machine learning modules — XGBoost-based stock analysis, TFT-based seasonal demand prediction, and LightGBM-based purchase request review — that only recommend, leaving the final decision with the human approver.

The major contributions of this work are as follows:

1) An integrated web-based hospital procurement platform that covers the complete lifecycle from purchase request to inventory update, protected by JWT authentication and role-based access control.

2) A master data management layer for items, suppliers, manufacturers, categories, units, and departments that keeps procurement records consistent across all transactions.

3) A configurable multi-level approval matrix that routes each purchase request by department, category, and amount.

4) A layered architecture of Angular, ASP.NET Core Web API, EF Core, and MySQL, following a Controller–Service–Repository pattern for maintainability and extension.

5) A human-in-the-loop AI decision-support design that uses XGBoost for stock status analysis, TFT for seasonal demand prediction, and LightGBM for purchase request review, with the approver retaining the final decision.

II. RELATED WORK

Research on hospital inventory has largely concentrated on forecasting demand and deciding replenishment quantities. The following studies are the most relevant to the machine learning modules and the decision-support approach adopted in MedProcure.

Fernandez et al. [1] applied a data-based model predictive decision support system to drug inventory management in the pharmacy of a mid-size hospital in Spain.

The system gave pharmacy staff the quantities needed to meet clinical demand together with the risk of stockout for different time horizons, and it was tested over four months against the pharmacy's usual policy. This supports the approach taken in MedProcure, where data-driven recommendations assist hospital staff rather than replace their judgment.

Petchrompo et al. [2] proposed a two-step framework for vaccine planning in a hospital travel clinic. XGBoost and LightGBM were used for daily demand forecasting, and a simulation then evaluated the trade-off between stockout risk and inventory efficiency. They report that XGBoost performed better for seasonal or volatile demand, whereas LightGBM performed well on smoother demand profiles. This is relevant to MedProcure, which applies gradient boosting to stock analysis and request review.

Chen and Guestrin [3] introduced XGBoost, a scalable tree-boosting system built around a regularized learning objective, which MedProcure uses to classify stock status. Ke et al. [4] proposed LightGBM, which speeds up gradient boosting on large datasets through gradient-based one-side sampling and exclusive feature bundling; MedProcure uses it for purchase request review. Lim et al. [5] proposed the Temporal Fusion Transformer, an attention-based architecture for multi-horizon forecasting that combines static and time-varying inputs and provides interpretable insight into variable importance, which suits the seasonal demand prediction module.

However, existing studies mainly address forecasting or replenishment for a single class of items and are largely separate from the procurement workflow itself. An integrated approach that combines purchase request management, configurable approvals, supplier ordering, goods receipt, and inventory tracking with advisory machine learning modules is comparatively limited. MedProcure addresses this gap by embedding recommendation modules inside a complete, auditable procurement workflow.

III. PROPOSED SYSTEM

MedProcure digitizes hospital procurement from the first purchase request to the final inventory update, addressing the delays, stock imbalances, and lack of traceability of manual procurement. It integrates role-based access, master data management, a configurable approval matrix, a transactional procurement workflow, and advisory machine learning modules into one unified platform.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

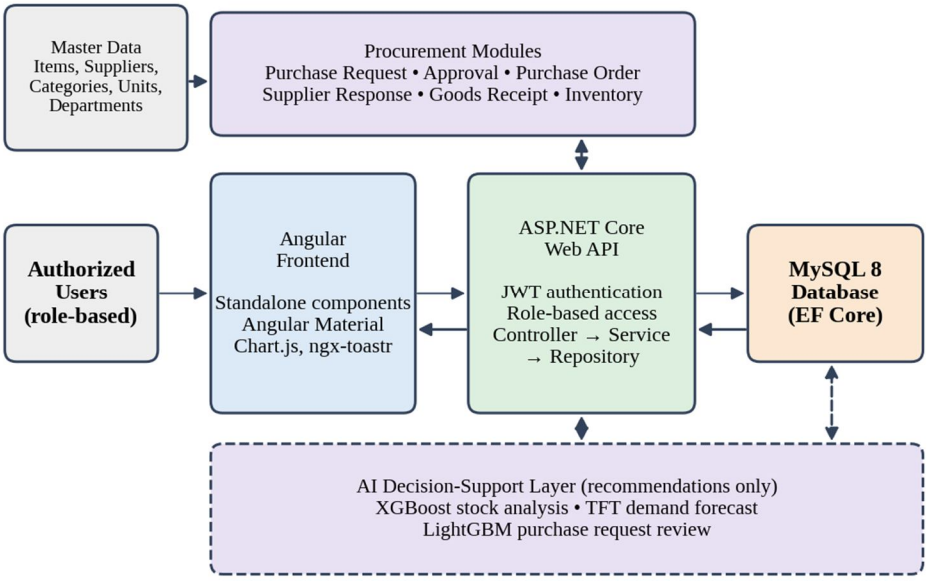


Fig. 1. Overall System Architecture of MedProcure.

As shown in Fig. 1, authorized users interact with an Angular frontend, which communicates with the ASP.NET Core Web API over REST calls secured with JWT. The Web API authenticates each request, checks the role of the user, and passes the request through services that apply business rules. Repositories using EF Core store and retrieve data from the MySQL database, which holds the master data, procurement transactions, approval matrix, and stock records.

A purchase request raised by a department is routed through the approval matrix according to its department, category, and amount. Once approved, it is converted into a purchase order for a supplier; the supplier's response and the subsequent goods receipt are recorded, and inventory is updated accordingly. Every step is stored in the database, giving hospital management a traceable record of each procurement decision. An AI decision-support layer, which is being added to the platform, analyzes stock levels, consumption history, and requested quantities. It performs stock status analysis, seasonal demand prediction, and purchase request review, and presents its recommendations to the approver. The layer only recommends; the human approver keeps the final decision.

IV. METHODOLOGY

The methodology of MedProcure follows the real procurement lifecycle of a hospital. It combines master data management, a configurable approval workflow, supplier ordering, goods receipt with inventory update, and three machine learning modules that support, but do not replace, the human approver.

A. Master Data Management

Items, suppliers, manufacturers, categories, units, departments, and roles are maintained as master data. Every procurement transaction refers to these records instead of free text, which keeps item names, units, and supplier details consistent and makes later analysis of stock and consumption reliable.

B. Purchase Request Creation

A department raises a purchase request for the items it needs, specifying the required quantities. The request is stored with its department, category, amount, and approval status, and it enters the approval workflow instead of being handled through informal communication.

C. Configurable Multi-Level Approval

Administrators configure an approval matrix in which each rule is defined by department, item category, and amount. When a request is submitted, the system matches it against the matrix to determine its approval path, and the request then moves through the configured approval levels. Each decision is recorded so that approvals remain accountable and traceable.

D. Purchase Order and Supplier Response

An approved request is converted into a purchase order addressed to a supplier from the master data. The supplier's response to the order is recorded in the system, so the hospital can follow the status of every order without separate tracking.

E. Goods Receipt and Inventory Update

When ordered goods arrive, a goods receipt is recorded against the purchase order and the inventory is updated. The updated stock, together with consumption history, forms the data on which the machine learning modules operate.

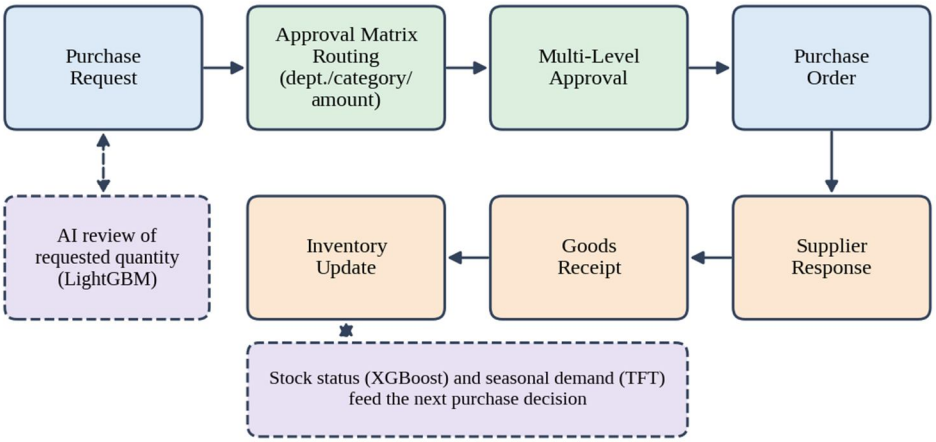


Fig. 2. Procurement Lifecycle and AI Decision-Support Points in MedProcure.

Fig. 2 shows the lifecycle. The dashed boxes mark where the AI modules assist: the LightGBM module reviews a request before approval, while the XGBoost and TFT modules use stock and demand data to inform the next purchase decision.

F. Inventory Understock and Overstock Analysis Using XGBoost

The first AI module classifies the stock status of each item, for example as understocked or overstocked, and recommends a procurement action such as ordering more or deferring a purchase. XGBoost [3] builds an ensemble of regularized decision trees by minimizing the objective

$$L = \Sigma\, l(\hat{y}_i, y_i) + \Sigma\, \Omega(f_k), \quad \Omega(f) = \gamma T + \tfrac{1}{2}\lambda\|w\|^2 \quad (1)$$

where l is a differentiable loss between the prediction $\hat{y}_i$ and the target y_i, f_k is the k-th tree, T is the number of leaves, w holds the leaf weights, and γ and λ are regularization parameters that limit model complexity.

G. Seasonal Demand Prediction Using Temporal Fusion Transformer

The second module forecasts demand across several future periods while accounting for seasonality, which is important for hospital items whose usage changes with the season. The Temporal Fusion Transformer [5] produces quantile forecasts and is trained with the quantile loss

$$QL(y, \hat{y}, q) = q \cdot max(y - \hat{y}, 0) + (1 - q) \cdot max(\hat{y} - y, 0) \quad (2)$$

where y is the observed demand, $\hat{y}$ is the predicted value for quantile q. Because TFT uses variable selection and attention mechanisms, it also shows which variables influenced a forecast, allowing the approver to understand why a particular demand level is predicted.

H. Smart Purchase Request Review Using LightGBM

The third module reviews a purchase request before approval. It compares the requested quantity with the current stock and recent consumption of the item and recommends an appropriate quantity. LightGBM [4] is used for this task because its gradient-boosting implementation trains quickly on tabular data. Duplicate purchase requests are handled separately by rule-based checks rather than by a learned model.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

I. Human-in-the-Loop Decision Control

All three modules only recommend. Their outputs are shown to the approver as advice alongside the request, and the approver may accept, modify, or reject the recommendation. This design keeps accountability with the human decision-maker, which is essential in hospital supply management.

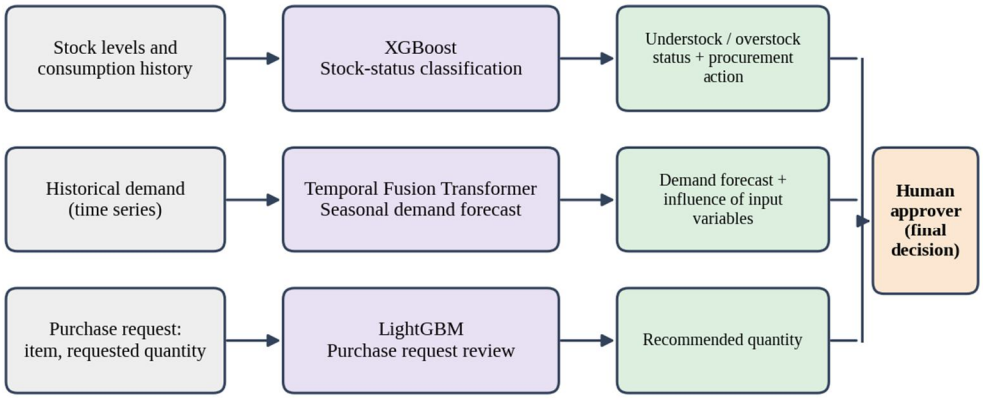


Fig. 3. AI Decision-Support Modules and Human-in-the-Loop Control.

V. SYSTEM ARCHITECTURE AND DESIGN

A. System Architecture

MedProcure follows a layered web architecture consisting of the presentation, API, business, data access, and database layers, with an AI decision-support layer attached to the backend. The Angular frontend handles user interaction, the ASP.NET Core Web API handles authentication, authorization, and request processing, and MySQL provides persistent storage. Separating the layers keeps the system modular and allows new modules, including the AI modules, to be added without changing the existing ones.

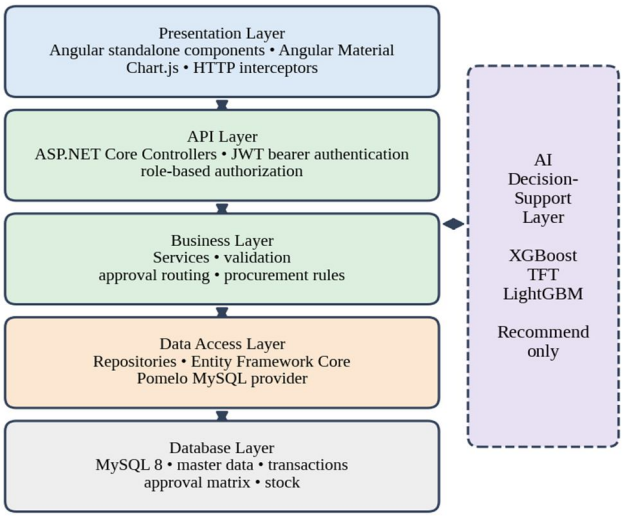


Fig. 4. Layered Architecture and Component Interaction of MedProcure.

B. Frontend Design

The frontend is developed with Angular using standalone components, Angular Material for the interface, ngx-toastr for notifications, and Chart.js for charts. Each module is split into a list component and a details component and uses route-based navigation of the form /module, /module/add, and /module/edit/:id. Add and edit forms open as inline panels within the list page instead of modals, which keeps the user in context. The sidebar navigation is loaded from the backend API.

C. Backend Design

The backend is an ASP.NET Core Web API running on .NET 10 and organized as Controller, Service, and Repository layers. Controllers receive requests and return responses, services apply business rules and validation, and repositories handle data access through EF Core. All responses use a common ApiResponse wrapper that carries the data and status, so the frontend can handle every endpoint in a uniform way.

D. Database Design

MySQL 8 stores the master data, procurement transactions, approval matrix, and stock records. Data is accessed through EF Core with the Pomelo MySQL provider, which maps the entities to tables and manages their relationships. Table I lists the master data entities and their roles in procurement.

Table I. Master Data Entities In Medprocure

Entity	Purpose in procurement
Item	Medicines, equipment, and consumables that can be requested and ordered
Supplier	Vendors to whom purchase orders are addressed
Manufacturer	Producers of the items
Category	Groups items; used as one of the keys of the approval matrix
Unit	Unit in which an item is requested and stocked
Department	Requesting departments; used as one of the keys of the approval matrix
Role	Defines the access rights of users

E. Authentication and Role-Based Access Control

MedProcure uses JWT bearer authentication [6] and role-based access control. After login, the Angular application attaches the token to every request through an HTTP interceptor, and the backend verifies the token and the role of the user before granting access to a protected endpoint. A second interceptor manages the loading state of the interface.

F. Approval Matrix Design

The approval matrix is the core of the workflow design. Each rule combines a department, an item category, and an amount, and determines the approval levels that a matching request must pass. Because the matrix is stored as data and managed through the Approval Configuration module, the hospital can change its approval policy without modifying the application code.

G. AI Module Design

The AI decision-support layer contains three modules, summarized in Table II. Each module takes data that the procurement workflow already produces — stock levels, consumption history, and request details — and returns a recommendation to the approver.

Table II. AI Decision-Support Modules

Module	Model	Task and output
Inventory analysis	XGBoost	Classifies stock status (understock / overstock) and recommends a procurement action
Demand prediction	Temporal Fusion Transformer	Forecasts seasonal demand and shows which variables influenced the forecast
Request review	LightGBM	Compares requested quantity with current stock and recent consumption and recommends a quantity

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

VI. IMPLEMENTATION

A. Development Environment

MedProcure is implemented as a client–server web application. The frontend uses Angular, the backend uses ASP.NET Core Web API on .NET 10, and data is stored in MySQL 8 through EF Core with the Pomelo provider. JWT bearer tokens secure the communication between the two sides.

B. Frontend Implementation

Each module is implemented as a pair of standalone components, one for the list and one for the details, with the routes described in Section V-B. Dropdown data such as departments and categories is loaded in parallel using RxJS forkJoin, and search inputs use a 250 ms debounce to avoid unnecessary processing. Lists use paginated data, and change detection is set to OnPush in the optimized modules to reduce re-rendering. Success and error messages are shown through ngx-toastr.

C. Backend Implementation

Controllers expose REST endpoints for each module, services contain the business rules and validation, and repositories perform database operations through EF Core. Validation failures are returned in the response and shown to the user as notifications, so invalid data is stopped before it reaches the database.

D. Database Implementation

EF Core maps the entities to MySQL tables and manages the relationships between master data, procurement transactions, and the approval matrix. List endpoints support pagination so that large tables remain responsive.

E. Authentication and Role Management

The login endpoint issues a JWT after successful authentication. The Role module, implemented as separate list and details components, manages the roles to which users belong, and the backend authorizes each request according to the role carried in the token.

F. Master Data Modules

The Manufacturer, Supplier, Category, Unit, Department, Role, and Item modules are implemented with a common pattern: a searchable, paginated list, an inline panel for adding and editing records, and validation of the entered data. Reusing one pattern across the modules makes the interface consistent and the code easier to maintain.

F. Approval Configuration Implementation

The Approval Configuration module lets administrators define approval rules. It consists of a list of existing rules and an inline add panel in which the department and category are selected from dropdowns loaded in parallel and the amount is entered. The backend validates each rule before it is saved, and any validation message is presented to the administrator.

G. Procurement Workflow Implementation

The procurement transactions follow the lifecycle of Fig. 2. A purchase request is matched against the approval matrix, approved through its configured levels, converted into a purchase order, answered by the supplier, and finally received into stock. Each transition is stored in the database, so the current status and history of every request can be retrieved.

H. AI Module Integration

The three machine learning modules are being integrated as recommendation services of the backend. Their results are intended to be shown to the approver during review, while the approval action itself remains a human step.

VII. RESULTS AND DISCUSSION

The implemented MedProcure system provides secure login, role-based access, master data management, configurable approval routing, and the procurement workflow from purchase request to inventory update. The following figures show the main screens of the application.

©IJRASET: All Rights are Reserved

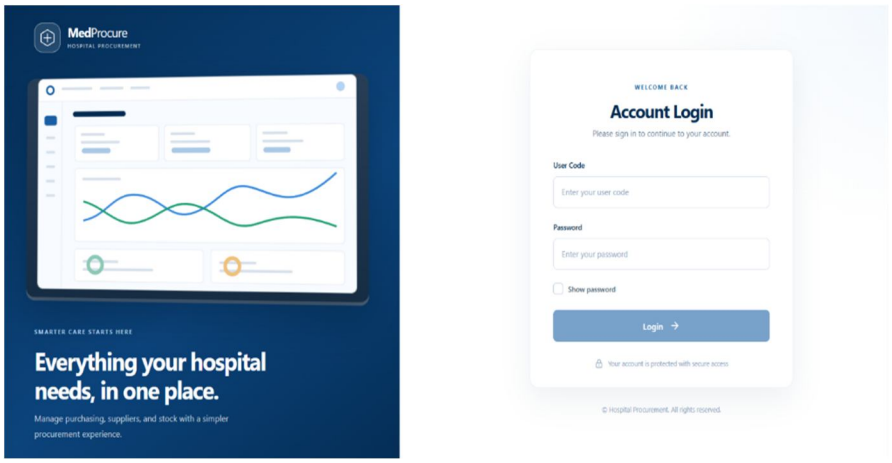


Fig. 5. Login Page and Role-Based Dashboard.

Fig. 5 shows the login page and the dashboard shown after authentication. Access to each part of the application depends on the role of the signed-in user.

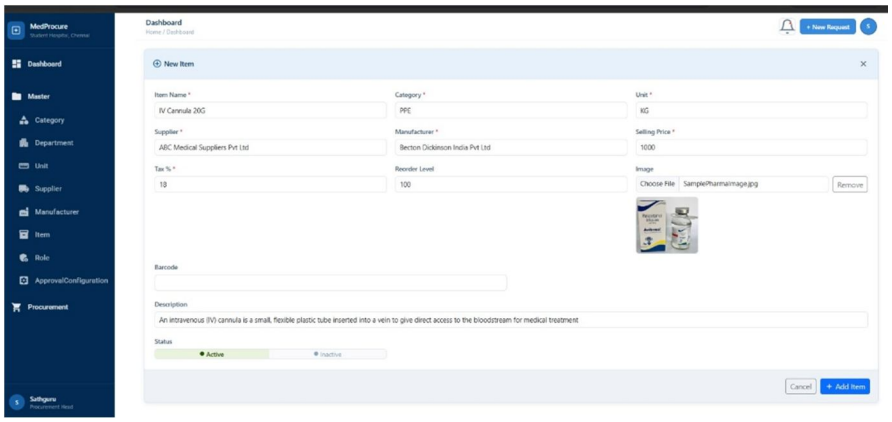


Fig. 6. Master Data Management Module.

Fig. 6 shows a master data module with its searchable list and inline add panel. All master data modules share this layout, so users work with the same interface throughout.

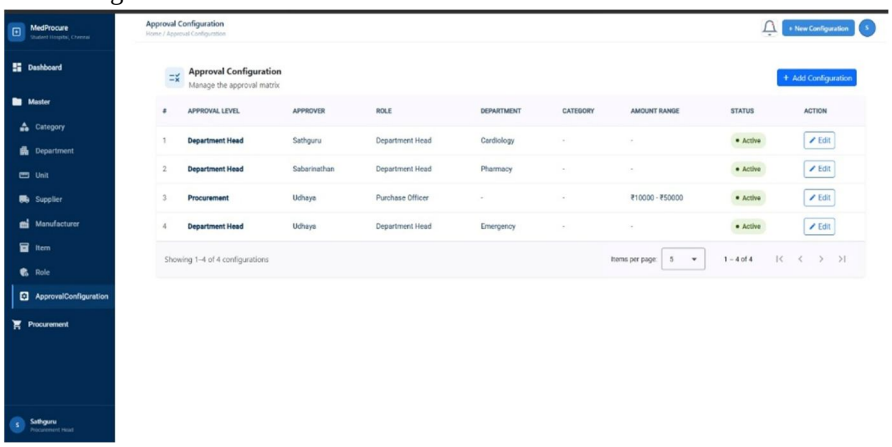


Fig. 7. Approval Configuration Module.

Fig. 7 shows the Approval Configuration module, where approval rules are defined by department, category, and amount. A change in the matrix changes how later requests are routed, without any change to the application code.

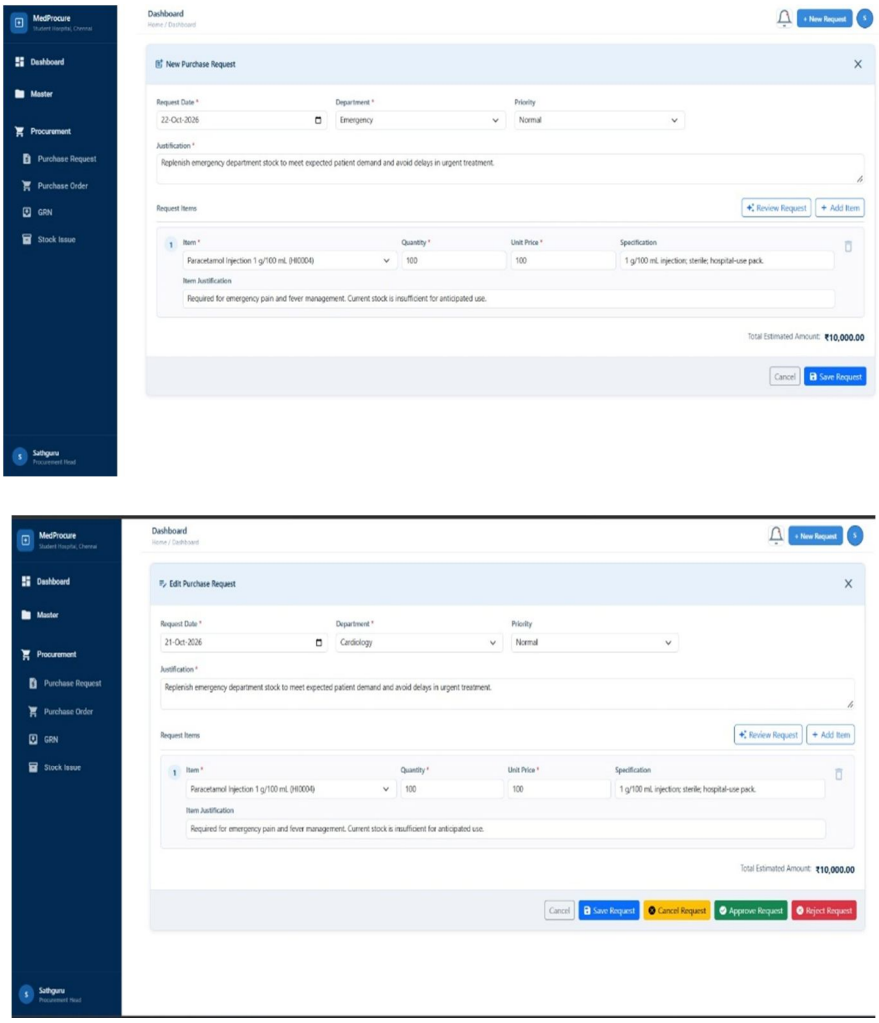
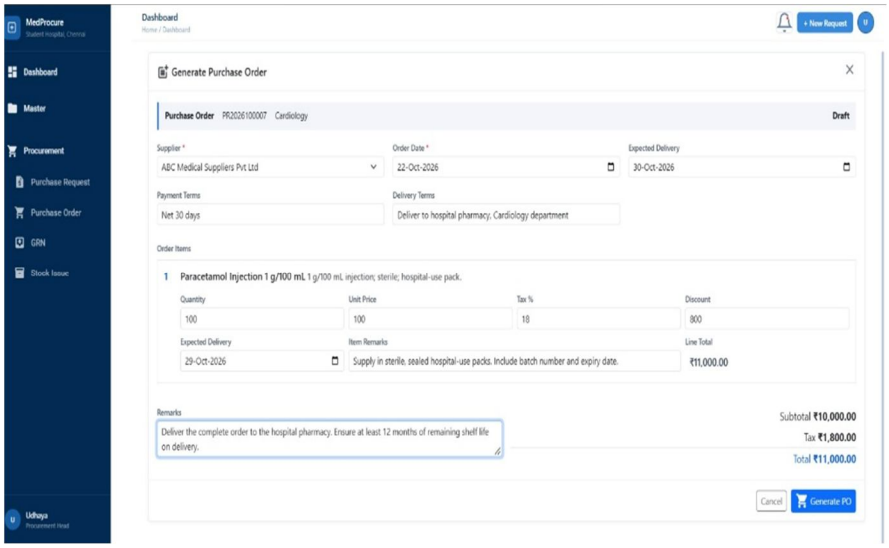


Fig. 8. Purchase Request Creation and Approval.

Fig. 8 shows the creation of a purchase request and its approval. The request follows the approval path defined by the matrix, and each decision is recorded.



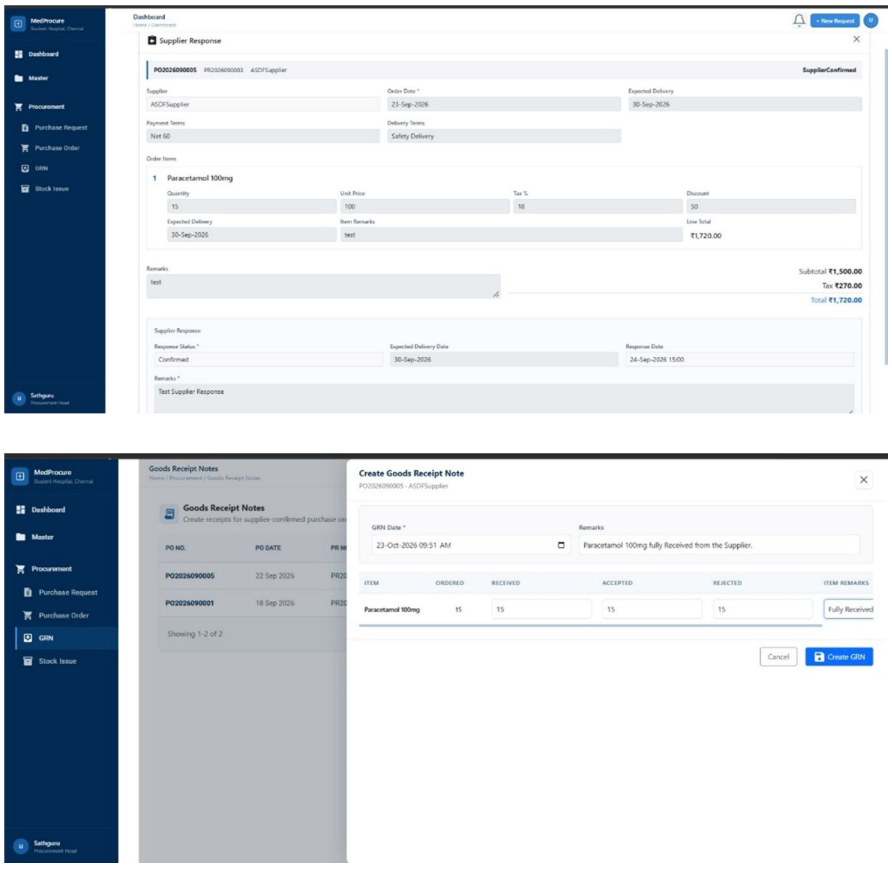


Fig. 9. Purchase Order, Supplier Response, and Goods Receipt.

Fig. 9 shows the purchase order, the supplier's response, and the goods receipt that updates inventory. This completes the lifecycle from request to stock update within one system.

Overall, the results show that MedProcure replaces manual requests, informal approvals, and periodic stock checks with a single workflow in which every request, approval, order, and receipt is recorded. The configurable approval matrix removes the need to hard-code approval rules, and the shared module pattern keeps the interface consistent and the system straightforward to extend.

The three AI modules are still being integrated, so this paper does not report their quantitative performance. Their evaluation on hospital stock and consumption data — for example classification accuracy for stock status, forecast error for demand prediction, and the deviation of recommended quantities from approved quantities — is planned as the next stage of the work.

VIII. CONCLUSION AND FUTURE WORK

MedProcure presents a web-based Hospital Procurement Management System that manages the full procurement lifecycle — purchase request, configurable multi-level approval, purchase order, supplier response, goods receipt, and inventory update — on a single platform. Its Angular frontend, ASP.NET Core Web API, MySQL database with EF Core, JWT authentication, and role-based access control provide a secure and maintainable foundation, while the approval matrix routes each request by department, category, and amount. The planned XGBoost, Temporal Fusion Transformer, and LightGBM modules extend the system with stock status analysis, seasonal demand prediction, and purchase request review, with the human approver retaining the final decision. Future work will complete the integration of these modules and evaluate them on hospital data to measure their effect on stockouts, excess inventory, and approval time.

REFERENCES

[1] M. I. Fernandez, P. Chanfreut, I. Jurado, and J. M. Maestre, "A Data-Based Model Predictive Decision Support System for Inventory Management in Hospitals," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 6, pp. 2227–2236, 2021, doi: 10.1109/JBHI.2020.3039692.

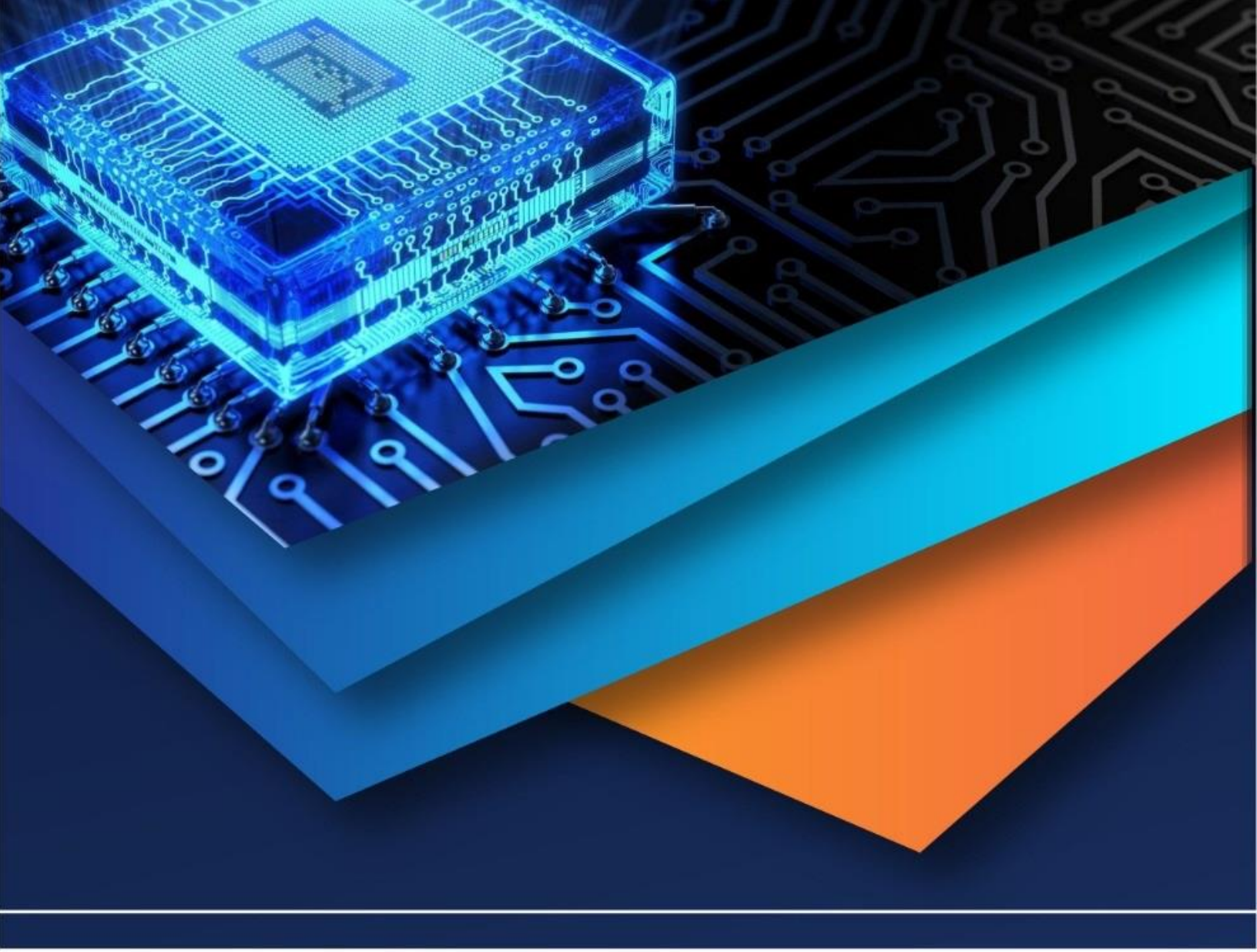
[2] S. Petchrompo *et al.*, "A Data-Driven Framework for Vaccine Demand Forecasting and Inventory Simulation in a Hospital Travel Clinic," *IEEE Access*, vol. 14, 2026, doi: 10.1109/ACCESS.2026.3685781.

[3] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[4] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, Long Beach, CA, USA, 2017, pp. 3146–3154.

[5] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[6] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, May 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)