

# Subhakamana Store Management System

## Overall System Report

Customer Website - Admin ERP - Cashier POS - Production Architecture

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| Project Name   | Subhakamana Store Management System                                               |
| Business Type  | Nepali fashion and clothing store                                                 |
| System Type    | Front-end e-commerce, ERP, POS, inventory, reporting, and demonstration prototype |
| Prepared For   | Project submission, presentation, and local demonstration                         |
| Report Date    | July 13, 2026                                                                     |
| Current Status | Strong local prototype with production-readiness documentation                    |

This PDF summarizes the complete Subhakamana Store system, including the customer website, full Admin ERP, focused Cashier POS, local launch files, reports, production architecture, limitations, and future backend plan.

## 1. Executive Summary

The Subhakamana Store Management System is a polished local front-end prototype for a Nepali clothing and fashion store. It demonstrates how the business can manage online shopping, product onboarding, inventory, POS billing, orders, delivery, reports, documentation, and production planning from one organized system.

The system is now divided into three practical working areas: Admin ERP for full store management, Cashier POS for fast billing with fewer distractions, and Customer Website for the online storefront. The current build runs locally without a domain and uses browser localStorage for prototype persistence.

### System Division

| Area             | Purpose                                                                                                                 | Primary Users                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Admin ERP        | Full management for products, inventory, suppliers, purchases, orders, delivery, reports, settings, and manuals.        | Owner, admin, inventory staff |
| Cashier POS      | Fast counter billing with product search/barcode input, cart, payment method, bill number, and stock reduction concept. | Cashier, sales staff          |
| Customer Website | Online storefront with homepage, products, details, cart, checkout, trust sections, and order confirmation concept.     | Customers                     |

## 2. Main System Areas

### Customer Website

The customer website presents Subhakamana Store as a warm, elegant fashion shop. It includes a polished homepage, product cards, search and filters, product detail pages, cart, checkout, delivery or in-store selection, member/friends/family phone discount area, payment options, trust badges, and responsive layout.

### Admin ERP

The Admin ERP is the main staff dashboard. It includes Dashboard, Inventory, Products, POS, Orders, Delivery, Suppliers, Purchases, Customers, Reports, Business Controls, E-commerce Preview, User Manual, and Settings.

### Cashier POS

The Cashier POS is optimized for in-store billing. It opens in a focused mode, hides unnecessary admin distractions, and keeps POS, sales orders, and payments close together for daily counter work.

### Role Access Concept

The prototype demonstrates role-based access for Super Admin, Admin, Inventory Staff, and Sales & Delivery Staff. In production, these roles must be enforced through secure backend authentication and permission checks.

## 3. Products, Inventory, and Sales Workflow

### Product Entry

- Basic information: product name, SKU, barcode, category, subcategory, and brand.
- Pricing: cost price, selling price, original price, and discount.
- Inventory: opening stock, reorder level, stock status, and available stock logic.
- Variants: size, color, variant SKU, and stock quantity.
- Supplier, product description, website visibility, new arrival, featured flag, image upload, phone photo onboarding, and website preview.

### Supplier to Sale Flow

Supplier -> Purchase Order -> Receive Stock -> Create/Find Product -> Create Variants -> Generate SKU + Barcode -> Print Labels -> Update Inventory -> Take Photos -> Upload Product Images -> Website Preview -> Online Ready -> Publish -> POS/Online Sale -> Stock Reduction -> Reports/Returns/Audit.

### Data Integrity

- Duplicate SKU and barcode prevention.
- Opening stock movement record when a product is created.
- Stock movement records for receiving, sales, and manual adjustments.
- Demo labels for sample data and created-by labels for role-created records.

## 4. Technology and Data Storage

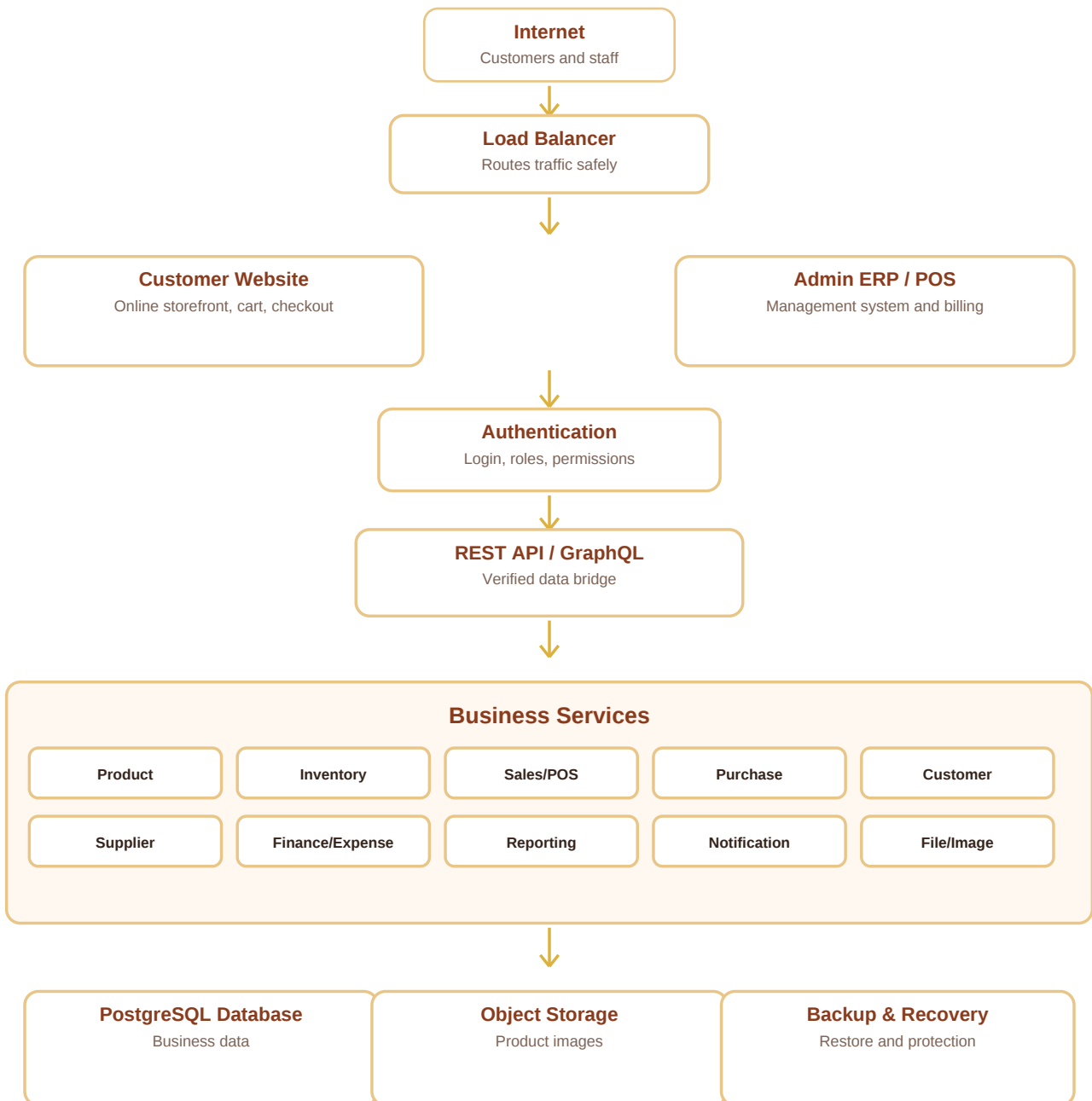
| Layer          | Current Prototype                                   | Future Production Direction                                                     |
|----------------|-----------------------------------------------------|---------------------------------------------------------------------------------|
| Frontend       | HTML, CSS, and vanilla JavaScript.                  | Can remain as the UI or later move to a modern app framework if required.       |
| Data Storage   | Browser localStorage for local demo persistence.    | PostgreSQL database for shared, secure business data.                           |
| Backend API    | Not live in the current front-end prototype.        | REST API or GraphQL with server-side validation and permissions.                |
| Authentication | Demo role/PIN concept.                              | Secure login, password hashing, sessions/tokens, RBAC, and audit logs.          |
| Images         | Local files and browser-selected previews.          | Object storage such as AWS S3, Cloudinary, or Firebase Storage.                 |
| Payments       | Prototype payment options and QR placement concept. | Server-verified eSewa, Khalti, bank transfer, COD, refunds, and reconciliation. |

### Important Current Limitation

The system is a strong front-end prototype. It is suitable for local demonstration and academic/project presentation, but real business use needs a backend API, secure authentication, shared database, payment verification, image storage, notifications, and backup/recovery.

## 5. Recommended Production Architecture

The future production system should connect the Customer Website and Admin ERP/POS to the same secure backend. This keeps products, prices, stock, images, orders, payments, and reports synchronized across every channel.



## 6. Documentation and Local Package

| File or Folder                            | Purpose                                                                                                                        |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| open-subhakamana-store-system.html        | Main local launcher for Admin ERP, Cashier POS, Customer Website, reports, manuals, and production-readiness files.            |
| subhakamana-store-ecommerce.html          | Customer-facing online shopping website.                                                                                       |
| subhakamana-store-final-erp-platform.html | Admin ERP, Cashier POS, inventory, reports, manual, and settings.                                                              |
| OPEN_ME_SUBHAKAMANA_STORE_COMPLETE_SYSTEM | Marked complete demonstration folder with synced major files.                                                                  |
| production-readiness/                     | Architecture, database schema, API contract, security/RBAC plan, testing/deployment plan, migration plan, and database module. |
| subhakamana-use-case-diagram.svg          | UML use case diagram for Admin and Customer workflows.                                                                         |

### How to Run Locally

- 1 Open the marked complete folder.
- 2 Double-click OPEN\_THIS\_FIRST.command or open open-subhakamana-store-system.html directly.
- 3 Use the launcher to open Admin ERP, Cashier POS, Customer Website, manuals, and reports.
- 4 For POS mode, use the Cashier POS button to open the focused billing workspace.

## 7. Testing, Limitations, and Future Work

### Testing and Validation Completed

- Major HTML and JavaScript files were checked for script parsing errors.
- Duplicate ID checks were performed on major HTML files.
- Local launcher links were checked for missing files.
- Main output files were synced into the marked complete folder after updates.
- Core sections were reviewed for presentation: website, ERP, POS, products, inventory, orders, delivery, reports, and manual.

### Production Features Still Needed

- Real backend API and PostgreSQL connection.
- Secure authentication and admin security.
- Real payment gateway verification for eSewa, Khalti, bank transfer, COD, and refunds.
- Automatic inventory synchronization across devices.
- Cloud product image storage, real notifications, courier integration, monitoring, and backups.

### Future Enhancements

- Connect the documented PostgreSQL/Prisma database module.
- Implement REST API or GraphQL service layer.
- Add secure role-based login and audit logs.
- Integrate payment gateways, delivery partners, object storage, and notification services.
- Create Android and iPhone apps that read from the same backend.

## 8. Final Conclusion

The Subhakamana Store Management System is now a clear and professional front-end prototype that demonstrates a complete retail workflow for a Nepali fashion store. It separates the system into Admin ERP, Cashier POS, and Customer Website, making the project easier to understand, present, and use.

The current version is ready for local demonstration and project submission. The next major stage is production backend development: secure login, API services, PostgreSQL database, image storage, payment verification, delivery integration, notifications, backup, and recovery.

|                  |                                                                    |
|------------------|--------------------------------------------------------------------|
| Final Status     | Presentation-ready local prototype with production-readiness plan. |
| Best Entry File  | outputs/open-subhakamana-store-system.html                         |
| Complete Package | outputs/OPEN_ME_SUBHAKAMANA_STORE_COMPLETE_SYSTEM                  |
| PDF Report       | outputs/subhakamana-store-overall-system-report.pdf                |