

UDYOG MANAGER

The complete product vision

Billing, customer records and business visibility for merchants.

Web application · Mobile application · Chatbot

Product plan · 14 October 2026 · udyogmanager.com

PRODUCT PURPOSE

Keep everyday business records together

Create bills, track customer dues, record purchases and expenses, and prepare information for accountant review.

The goal is a clear view of what was sold, what was collected and what remains unpaid.

Planned product. The welcome page and concept materials are built; the application is not yet launched.

CUSTOMER PROBLEM

Scattered records make daily work harder

Bills, customer balances and expenses can live in separate notebooks, messages and spreadsheets.

Our hypothesis: merchants will value fast billing backed by dependable records and simple usage pricing.

Validate demand through merchant interviews and a pilot; no market size or traction is claimed.

MERCHANT COVERAGE

One foundation across business categories

Shared needs: customers, products or services, invoices, payments, expenses and reports.

Specialist modules: retail variants, wholesale pricing, restaurant orders, pharmacy batches and repair job cards.

All-category coverage is the long-term vision. Specialized workflows are released in stages.

USER ROLES

Access follows each person's responsibility

Owners manage the business. Managers handle daily operations. Billing staff create bills and record authorized payments.

Accountants review records and exports. Platform support uses restricted, audited access.

Permissions are enforced by the backend for every business, document and export.

MERCHANT ONBOARDING

Prepare the business for its first bill

Verify account → Select category → Enter business and tax details →
Configure invoice series → Add customers and items.

Import opening records with validation, review the credit policy, then recharge and issue the first bill.

Walk-in sales should not require unnecessary customer details.

BILLING WORKFLOW

Draft, review, finalize and share

Select a customer, add products or services, review quantities, discounts and tax totals, then confirm issuance.

Django assigns the official number, posts the required records and consumes exactly one credit.

PDF generation and sharing happen after the saved financial transaction.

PAYMENTS AND DUES

Keep collections separate from sales

Record cash, UPI and bank receipts, including partial payments and customer advances.

Allocate receipts to invoices, show outstanding balances and ageing, and reverse mistakes with an audit trail.

A manually entered UPI receipt is not automatically a provider-verified payment.

PURCHASES AND INVENTORY

Record the cost of doing business

Capture supplier bills, operating expenses, attachments, stock receipts and stock adjustments.

For goods, track inventory movements and a defined costing method so profit reflects the cost of items sold.

Avoid recording the same purchase twice as both a purchase and an expense.

RETURNS AND CORRECTIONS

Preserve the original record

Use linked adjustments and controlled cancellation rules for issued bills.
Drafts remain editable.

Treat a financial adjustment, a physical stock return and a cash refund as related but distinct events.

Issued financial records must not be silently overwritten or deleted.

CREDIT PRICING

₹1 per finalized bill

One successfully issued bill consumes one credit. Customer creation, drafts and payment updates are free.

Reprinting or downloading the same bill does not consume another credit. Existing data remains accessible at zero balance.

Tax treatment, recharge minimums, credit expiry and refund terms still need approval.

CREDIT INTEGRITY

A retry must never become a second charge

Commit invoice issuance and the credit debit together. Prevent concurrent users from spending the same final credit.

Verify recharge success on the server. Deduplicate payment-provider events and reconcile missed notifications.

A failed PDF job retries against the same issued bill, without another deduction.

BUSINESS REPORTS

Sales, collections, dues and profit

Provide sales and purchase registers, customer statements, expense summaries, stock views and financial reports.

Profit requires complete costs, expenses and opening balances. Show missing information and provisional results.

Summary totals should reconcile to the underlying accounting journal.

ACCOUNTING EXAMPLE

A sale is not the same as cash received

Illustration: ₹1,000 sale + ₹180 tax = ₹1,180 invoice. A ₹500 receipt leaves ₹680 outstanding.

With assigned goods cost of ₹600, gross profit is ₹400 before other expenses. Issuance uses one platform credit.

The 18% tax rate is an arithmetic example only, not a recommended classification.

Prepare records for accountant review

Plan tax-component summaries, relevant HSN/SAC summaries, adjustment registers and GSTR-1 preparation data.

Add purchase reconciliation against imported GSTR-2B data and CA-reviewed GSTR-3B working summaries in stages.

Direct filing and e-invoicing are separate integrations. Current rules and report mappings require CA validation.

WEB AND MOBILE

The same business, wherever work happens

Next.js provides the web dashboard. React Native with Expo is proposed for Android and iOS.

Prioritize quick entry, customer lookup, sharing, readable reports and clear synchronization status.

Initial online finalization is proposed. Offline issuance needs numbering and credit-reservation design.

CHATBOT

Ask questions and prepare bills

Example requests: “Who has unpaid bills?” “How much did I collect today?”
“Prepare a bill for this customer.”

FastAPI resolves intent and calls authorized Django APIs. The merchant reviews and confirms financial changes.

Backend calculations remain authoritative. AI availability must not block ordinary billing.

SYSTEM ARCHITECTURE

Django is the business authority

Next.js web and mobile clients use the Django API. FastAPI handles conversation and invokes the same business services.

Proposed foundation: PostgreSQL, background workers, Redis and protected object storage for documents.

Start with a modular backend. Framework and provider versions are chosen during implementation.

DATA AND API DESIGN

Make every financial event traceable

Core records: business, membership, customer, item, invoice, payment, purchase, stock movement and journal entry.

Credit ledger, recharge orders, provider events and audit events support charging and reconciliation.

Versioned APIs, tenant-scoped access, exact money values, uniqueness constraints and idempotent writes.

SECURITY AND OPERATIONS

Protect access and recover reliably

Enforce business isolation and staff permissions. Protect credentials, exports, uploads and stored records.

Maintain audit history, backups, restore exercises, monitoring and incident runbooks.

Security requirements are planned controls, not completed certifications.

ACCEPTANCE TESTING

Test the failures that affect merchants

Repeated taps, network timeouts, concurrent credit use, failed PDF jobs, duplicate webhooks and partial-payment reversals.

Also test cross-business access, stale chatbot confirmations, financial-year numbering and backup restoration.

CA-reviewed examples validate supported tax calculations and accounting entries.

DELIVERY ROADMAP

Build the foundation, pilot, then expand

Discovery → Core billing and credits → Purchases and reports → Mobile and bounded chatbot → Merchant pilot.

Add specialist categories and external fiscal integrations after their workflows are validated.

The specification's 16-week scenario is indicative, dependent on scope and team capacity.

COMMERCIAL SCENARIOS

Measure usage and contribution

At 300 bills per merchant each month: 100 merchants generate ₹30,000 of usage value; 500 generate ₹150,000.

Account for payment fees, hosting, support, AI, taxes, refunds and promotional credits before assessing profitability.

Illustrative arithmetic, not a forecast. Recharge receipts and recognized revenue can differ.

LAUNCH READINESS

Close the decisions before paid use

Approve recharge and refund terms, launch categories, languages, offline scope, support coverage and mobile release order.

Recruit pilot merchants and a CA; validate billing, reporting, tenant isolation and recovery.

Domain purchased: udyogmanager.com. Public domain connection remains to be completed.

UDYOG MANAGER

The next milestone is a working merchant pilot

Prove a complete journey: onboard → recharge → issue a bill → record payment → review reports.

Then expand the product using evidence from everyday merchant use.

Supporting detail: complete product report and development specification.