

Project Phases:-

Phase 1: Planning & Setup

- Requirement study
- Decide tech stack
- Create Git repository and start version control
- Database schema design (Users, Customers, Invoices, Inventory, Payments, Tax, Reports, etc.)
- Create UI/UX wireframes (Figma / Whimsical / Miro)

Phase 2: Authentication & Roles

Modules: Login, Registration, JWT Authentication, Roles & Permissions

Features: Super Admin, Manager, Accountant, Sales Executive, Inventory User'

Tools:-

- Backend: Django REST Framework (or FastAPI)
- Authentication: JWT / OAuth2
- Database: PostgreSQL / MySQL
- Frontend: React.js / Next.js
- Security: bcrypt/argon2 password hashing

Phase 3: Core Modules Development

- **Dashboard:** Sales, invoices, stock, and payments summary
- **User Management:** Add/View users with roles
- **Customer & Vendor Management:** Add/Edit customers & vendors, outstanding balances, payment terms, GST/PAN, etc.

Phase 4: Inventory & Payments

- **Inventory Management:** Products, categories, stock alerts, barcode scanner integration
- **Payments Module:** Payment tracking, refunds, UPI/Bank/Card integration, payment status & reports
- **Invoice & Billing:** Create invoices, recurring invoices, debit/credit notes, delivery challans, POS billing

Phase 5: Compliance & Finance

- **Tax & Compliance:** GST Config, HSN/SAC, GSTR-1, GSTR-3B exports
- **Expense Management:** Recurring + one-time expenses
- **Reports & Analytics:** Sales, purchase, Profit/Loss, tax summary, inventory valuation

Phase 6: Quotation & Notifications

- **Quotation & Estimates :-** Convert to invoice
- **Notifications:** WhatsApp, Email, SMS, UPI QR Code

Phase 7: Final Touch

- UI Layout: Sidebar, header, dark/light mode
- Profile Management: User profile, password reset
- Testing: Unit testing (pytest, Django test), Integration testing (Postman)
- Deployment:
- Backend: Docker + AWS / DigitalOcean / Railway
- Frontend: Vercel / Netlify
- Database: RDS / Cloud SQL

❖ Tools & Tech Stack (Recommended)

- ◆ Frontend: React.js / Next.js + TailwindCSS (fast UI design).
- ◆ Backend: Django REST Framework (robust & secure for finance).
- ◆ Database: PostgreSQL (best for relational data).
- ◆ Authentication: JWT (via DRF SimpleJWT).
- ◆ Payments: Razorpay / Stripe / Paytm API.
- ◆ Notifications: Twilio (SMS/WhatsApp), SendGrid (Email).
- ◆ Reporting: Django Admin + custom reports (PDF via ReportLab, Excel via Pandas/OpenPyXL).
- ◆ Version Control: GitHub / GitLab.
- ◆ Project Management: Jira / Trello.