GenERP BD

Complete Technical & Product Documentation

Generalised Cloud ERP / SaaS Platform — Bangladesh

Version 3.0 | Final MVP Planning Document | 2026

Stack	Timeline	Payment Model	Language	Document Sections
Laravel 12 + FilamentPHP v4 + MySQL 8 + Redis	20 Weeks — Solo Dev + AI-Assisted	Manual verification (bKash/Nagad/Rocket/Cheque)	Bangla (primary) + English bilingual	25 sections — fully specified

1. Project Overview

1.1 What is GenERP BD?

GenERP BD is a generalised cloud-based ERP SaaS platform built specifically for the Bangladeshi market. It serves any kind of business — from a 1-person freelancer to a 1,000-employee RMG factory — using a single flexible application that each business configures to match their own workflows, vocabulary, policies, and operations. One codebase, infinite configurations, zero custom development per industry.

1.2 Target Business Types

Business Type	Primary Use of GenERP BD
Retail shops & supermarkets	POS, inventory, daily sales, supplier management, expiry tracking
Pharmacies & drug stores	Batch/expiry, POS, Mushak VAT compliance, patient records, drug license tracking
Wholesale & trading companies	Bulk pricing, LC management, credit terms, supplier ledger, TDS on payments
Manufacturing & RMG factories	Production, buyer/style/LC fields, wage board payroll, BOM (post-MVP)
Restaurants & food businesses	Table-based POS, menu inventory, shift management, daily covers report
Service businesses (IT, agencies, clinics)	Client billing, project invoicing, service catalogue, retainer tracking
Solo entrepreneurs & freelancers	Income/expense, client invoicing, simplified dashboard
NGOs & non-profit organisations	Donor management, project costing, fund utilisation, grant tracking
Government offices & institutions	Multi-level purchase approvals, budget heads, expense compliance
Schools & educational institutions	Student fee management, grade/section tracking, teacher payroll
E-commerce businesses	Order management, inventory tracking (courier integration post-MVP)

1.3 Key Differentiators

Differentiator	Description
One login, many businesses	Single account manages unlimited companies/branches with per-company roles

Entity aliasing	Customers → Patients/Students/Donors — every entity renamed to match business vocabulary
Dynamic custom fields	Any field on any entity, per company, no developer needed
Configurable workflows	Each company defines their own approval chains and status flows
Bangladesh compliance	Mushak VAT forms, NBR invoice format, TDS/VDS, wage board payroll, BD income tax slabs
Configurable tax engine	All BD VAT rates, compound tax, TDS/VDS — fully configurable
Manual payment verification	Admin verifies bKash/Nagad/Rocket/Cheque transaction IDs — no gateway dependency
Configurable number sequences	Custom invoice/PO/voucher formats with reset periods and branch codes
Service mode	Non-inventory businesses get a service catalogue — no warehouse complexity
Bangla-first bilingual UI	Full Bangla default with Bengali numerals, Noto Sans Bengali font throughout
Freemium model	3 companies free forever — upgrade naturally as business grows
Enterprise-grade security	SQL injection proof, DDoS resistant, encrypted at rest and in transit, full audit trail

2. Industry & International Standards Compliance

This section defines how GenERP BD aligns with international ERP standards, Bangladesh-specific regulatory requirements, and industry-specific compliance needs. Meeting these standards is what separates a professional ERP from a basic accounting app.

2.1 International ERP Standards

Standard / Framework	Relevance to GenERP BD	Implementation in MVP
Double-entry accounting principle	Every financial transaction has equal debit and credit entries — the foundation of all accounting standards	Auto journal entries from all modules maintain double-entry integrity. Chart of accounts follows standard structure.
GAAP-aligned reporting	Generally Accepted Accounting Principles — P&L, Balance Sheet, Cash Flow follow internationally recognised formats	Reports structured to GAAP format. Post-MVP: IAS/IFRS alignment for enterprise clients.
Role-Based Access Control (RBAC)	ISO 27001-aligned access management — users only access what their role permits	Full RBAC via Spatie Permission, per-company roles, field-level and action-level permissions.
Audit trail (non-repudiation)	ISO 27001 requirement — every action traceable to a specific user at a specific time	Full audit log with old/new value diffs, immutable records, monthly partitioning.
Data encryption standards	AES-256 for data at rest, TLS 1.3 for data in transit — international minimum	HTTPS enforced (TLS 1.3), database encryption at rest on VPS, encrypted backups.
SOC 2 principles (Type 1 alignment)	Security, Availability, Processing Integrity, Confidentiality, Privacy	MVP achieves alignment — full SOC 2 Type 2 audit is post-revenue target for enterprise sales.
GDPR principles (data privacy)	Right to data access, right to deletion, data minimisation, consent — increasingly expected globally	Data export per user, account deletion workflow, no unnecessary data collection. Full GDPR post-MVP.
Multi-tenancy isolation	Critical SaaS standard — one tenant's data must never be accessible to another	company_id global scope on all queries, architectural isolation not just policy.
API REST standards (OpenAPI 3.0)	Consistent, versioned, documented API for integrations	REST API v1 with OpenAPI documentation, consistent response envelope, semantic versioning.

Idempotent financial operations	Same financial operation submitted twice must not double-post	Idempotency keys on all payment and invoice creation endpoints — duplicate detection enforced.

2.2 Bangladesh-Specific Regulatory Compliance

Regulation / Standard	Authority	GenERP BD Compliance
VAT & Supplementary Duty Act 2012	National Board of Revenue (NBR)	Full Mushak form generation (6.1, 6.2, 6.3, 6.6, 9.1), all BD VAT rates, NBR-compliant invoice format, VAT BIN on all documents
Income Tax Ordinance 1984	NBR	BD income tax slab auto-calculation on payroll, TDS on supplier payments (Section 52), TDS certificate generation
Bangladesh Labour Act 2006 (amended 2018)	Ministry of Labour & Employment	Configurable leave types per BD law (annual 18 days, sick 14 days, casual 10 days), overtime at double rate, festival bonus calculation
Minimum Wage Board Gazette	Minimum Wage Board	Wage board grade support for RMG, minimum wage enforcement, per-grade salary structure
Company Act 1994	Registrar of Joint Stock Companies	Company registration number field in company settings — printed on official documents
Bangladesh Bank guidelines (MFS)	Bangladesh Bank	bKash/Nagad/Rocket transaction ID verification field — admin records and verifies manually, no gateway dependency
NBR e-filing requirements	NBR	Monthly VAT return data exportable in NBR-compatible Excel format for manual e-filing on NBR portal
Financial Reporting Act 2015	Financial Reporting Council	Financial reports structured per FRC guidelines for large company compliance
Data Privacy (Draft Bill 2022 — pending)	Ministry of Posts, Telecom & IT	Privacy-by-design architecture, user data export, account deletion workflow — ready for when Bill passes

2.3 Industry-Specific Standards

Industry	Specific Standard / Requirement	GenERP BD Support
Pharmacy / Drug	Drug license number mandatory on all purchase records, batch tracking for regulatory traceability, FIFO/FEFO stock management	Drug license field on supplier profiles, batch + expiry tracking, FEFO enforcement configurable
Food & Restaurant	BSTI (Bangladesh Standards and Testing	Expiry tracking on all food inventory, configurable alert thresholds, batch traceability report

	Institution) compliance fields, expiry/batch on food inputs	
RMG / Garments	BGMEA/BKMEA compliance, wage board adherence, buyer compliance audit readiness	Wage board grade system, per-buyer reporting, full audit trail on production and payroll records
Healthcare / Clinic	Patient record confidentiality, treatment history, prescription tracking (post-MVP)	Patient entity alias, appointment-to-invoice workflow, role-restricted access to patient records
NGO / Non-Profit	Donor fund segregation, project-based expense tracking, NGO Affairs Bureau reporting	Fund/project code on all expenses, donor ledger, project utilisation report exportable
Education	BANBEIS (Bangladesh Bureau of Educational Information and Statistics) compatible student data	Student entity alias, grade/section custom fields, fee head configuration, attendance for students (post-MVP)
Government	CPTU (Central Procurement Technical Unit) procurement rules, multi-level approval, budget head tracking	Configurable multi-level purchase approval workflow, budget head on expenses, full audit trail

2.4 What GenERP BD Covers vs Comparable Products

Feature Area	Tally ERP	SAP B1	QuickBooks	GenERP BD MVP
Bangladesh VAT / Mushak	Limited	With customisation	No	Full — built-in
Bangla UI	No	No	No	Yes — primary language
Multi-company one login	Separate files	Separate instances	Separate accounts	Yes — native
Dynamic custom fields	No	Limited	No	Yes — all entities
Configurable workflows	No	Yes (complex)	No	Yes — visual builder
Wage board payroll (BD)	No	No	No	Yes — built-in
Manual payment verification (BD MFS)	No	No	No	Yes — admin verification
Freemium / cloud SaaS	No (desktop)	No (on-premise)	Yes	Yes
Entity aliasing	No	No	No	Yes — per business type
BD income tax auto-calc	No	No	No	Yes — slab-based

Price (entry level)	~30,000 BDT one-time	Very expensive	USD-based	Free (3 companies)

3. Business Model

3.1 Subscription Plans

Feature	Free	Pro (~3,500 BDT/mo)	Enterprise (~10,000+ BDT/mo)
Companies	Up to 3	Unlimited	Unlimited + custom domain
Users per company	3	Unlimited	Unlimited + advanced roles
Modules	Core only	All MVP modules	All + API access
Products / SKUs	500 max	Unlimited	Unlimited
Custom fields per entity	5	Unlimited	Unlimited
Workflow statuses	3 per doc type	Unlimited	Unlimited + automation
Dashboard widgets	5	Unlimited	Unlimited
Reports	Basic only	Full + scheduled exports	Drill-down + custom builder
PDF documents	Watermarked	Clean branded bilingual	Custom branding per company
Storage	1 GB	Unlimited	Unlimited + dedicated
Audit log	30 days	Full history	Full + SIEM export
Notifications	In-app only	In-app + email	In-app + email + SMS
Mushak VAT forms	Basic VAT calc only	Full Mushak 6.1/6.2/6.3/6.6	Full + NBR export
Entity aliasing	Fixed names	Fully configurable	Fully configurable
Number sequences	Fixed format	Fully configurable	Fully configurable
Notification templates	System default	Customisable	Fully customisable + multilingual
Data import	Products only	All entities	All entities + migration support
API access	None (blocked)	Full REST API	Full + webhooks + higher limits
Support	Community / forum	Priority email	Dedicated account manager + phone

3.2 Manual Payment Verification Model

GenERP BD does not use payment gateways. All subscription payments are verified manually by the GenERP BD admin team. This approach is simpler to build, avoids gateway fees and technical dependencies, and is familiar to Bangladeshi businesses accustomed to bKash/Nagad manual transfers.

Supported Payment Methods:

- bKash (personal or merchant number)
- Nagad
- Rocket
- Bank cheque (account number + cheque number)
- NPSB / bank transfer (transaction reference)

Payment Verification Workflow:

1. User goes to Settings > Subscription > Upgrade Plan — sees pricing and payment instructions
2. User selects plan (Pro or Enterprise) and billing cycle (monthly/yearly)
3. App shows GenERP BD's bKash/Nagad/bank account details with the amount to send
4. User makes payment via their preferred method outside the app
5. User returns to app and submits: payment method, transaction ID / cheque number, amount, date, screenshot (optional)
6. Payment request record created with status: Pending Verification
7. GenERP BD admin receives in-app alert and email notification of new payment request
8. Admin logs into superadmin panel, verifies the transaction ID against bKash/Nagad/bank records
9. Admin marks payment as Verified — plan instantly activated, user receives confirmation email and in-app notification
10. If transaction ID is invalid or amount is wrong, admin marks as Rejected with reason — user notified to resubmit

Payment Request Data Model:

Column	Type	Description
id	BIGINT	Primary key
company_id	BIGINT	Which company is paying
requested_plan_id	BIGINT	Which plan they want
billing_cycle	ENUM	monthly / yearly
amount_bdt	DECIMAL	Amount they claim to have sent
payment_method	ENUM	bkash / nagad / rocket / bank_transfer / cheque
transaction_id	VARCHAR	Transaction ID / cheque number / bank reference
payment_date	DATE	Date payment was made
screenshot_path	VARCHAR	Optional uploaded screenshot of payment receipt

status	ENUM	pending / verified / rejected
rejection_reason	TEXT	Admin note if rejected
verified_by	BIGINT	Superadmin user ID who verified
verified_at	TIMESTAMP	When verification was done
created_at	TIMESTAMP	When request was submitted

SLA & Admin Workflow:

- Target verification time: within 24 hours on business days
- Admin dashboard shows: pending payment requests, count, oldest pending
- Admin can add internal notes on each payment request (e.g. "verified against bKash statement 12 Jan")
- All payment verifications are audit logged with admin user ID and timestamp
- Monthly payment report for the GenERP BD team: revenue collected, by method, by plan
- Future upgrade path: SSLCommerz / bKash gateway can be added later — the manual system is replaced, not restructured

4. Technical Architecture

4.1 Tech Stack

Layer	Technology	Reason
Backend	Laravel 12 (monolithic)	AI-generation friendly, built-in queues/events/notifications/API, excellent security defaults
Database	MySQL 8 (UTF8mb4_unicode_ci)	Full Bangla support, JSON columns, partitioning, mature and reliable
Cache / Queues	Redis + Laravel Horizon	Subscription context caching, async jobs, rate limit counters
Admin UI	FilamentPHP v4	Beautiful responsive UI, datatables, forms, charts — ideal for solo dev
Charts	Filament Charts + ApexCharts	Native integration, rich visualisations
Real-time	Laravel Reverb + Echo	Official WebSockets, zero extra cost, native Laravel
Auth + RBAC	Laravel Sanctum + Spatie Permission	API tokens + per-company role/permission
Multi-tenancy	stancl/tenancy (shared DB + company_id)	Scalable, Filament compatible
Testing	Pest + Laravel Dusk	Fast, AI-generation friendly test suites
Bilingual	Laravel Localization (lang/bn + lang/en)	Full Bangla/English with per-company preference
Fonts	Noto Sans Bengali (Google CDN)	Best Bangla web font
PDF	DomPDF with Bangla font embedding	Bangla text in invoices, payslips, Mushak forms
Security	Laravel built-ins + custom middleware stack	See Section 6 for full security architecture
Error monitoring	Sentry	Real-time error tracking from day one
AI dev tools	Cursor.sh + Claude 4 + GitHub Copilot	Best combo for Laravel + Filament — see Section 9 for rules
Hosting (MVP)	XeonBD / Alpha Net VPS (BD-based)	Low latency, affordable, local
CDN	Cloudflare (free tier)	DDoS protection, static asset caching, WAF

4.2 Multi-Tenancy Model

Shared database with company_id scoping. Every business data table has a company_id column. A global Eloquent scope automatically filters all queries by the active company. Architectural isolation — not just policy. A bug in application code cannot expose cross-company data because the query literally cannot return rows from another company.

4.3 Core Database Tables (Foundation)

Table	Purpose
users	Central accounts — one per person, used across all companies
companies	Each registered business with settings, plan, branding, entity alias config
branches	Outlets / locations under a company
company_user	Pivot: users ↔ companies with per-company roles
plans	Plan definitions with limits and features as JSON
subscriptions	Active subscription per company with status, expiry, grace period
payment_requests	Manual payment submissions pending admin verification
subscription_invoices	Billing history — every verified payment generates a record
usage_counters	Real-time metered usage per company per metric
entity_aliases	Company-specific entity renaming (Customer → Patient etc.)
custom_field_definitions	User-defined fields per entity per company
custom_field_values	Stored values for custom fields
custom_field_indexes	MySQL generated columns for indexed custom field filtering
number_sequences	Configurable numbering rules per document type per company
workflow_definitions	Status machine configs per document type per company
workflow_transitions	Allowed transitions, required roles, automation triggers
tax_groups	Named tax configurations per company
notification_templates	Company-specific notification content per event
audit_logs	Full activity log — immutable, partitioned by month
alert_rules	User-configured trigger conditions and notification targets
failed_login_attempts	Tracks failed logins per IP and per user for brute force protection
api_tokens	Sanctum tokens with scope, rate limit, last used tracking
security_events	Critical security events: failed 2FA, suspicious IP, mass export

5. The Dynamic Configuration Layer

The architectural heart of GenERP BD. One codebase, any business type, zero custom development.

5.1 Business Type Templates

Business Type	Entity Aliases	Key Preset Custom Fields	Modules Enabled
Retail Shop	Customer, Product, Supplier	Barcode, SKU, selling unit	Inventory, POS, Sales, Purchases, Expenses, Reports
Pharmacy	Patient, Medicine, Drug Supplier	Batch no., expiry date, generic name, drug category, storage temp	Inventory, POS, Sales, Purchases, Batch/Expiry, Reports
Wholesale / Trading	Client, Item, Vendor	LC number, credit days, bulk price tier	Inventory, Sales, Purchases, Accounting, Reports
Manufacturing / RMG	Buyer, Style/Item, RM Supplier	Style no., buyer, LC no., shipment date, season, production line	Inventory, Sales, Purchases, HR, Payroll, Reports
Restaurant / Food	Guest, Menu Item, Food Supplier	Table no., cover count, shift, food category	POS, Inventory, Sales, Expenses, Reports
Service Business	Client, Service, Vendor	Project name, deliverable date, retainer type	CRM, Sales, Expenses, HR, Reports
Solo Entrepreneur	Client, Service, Vendor	Project name, payment terms	Income/Expense, Invoicing, Basic Reports (Simplified Mode)
NGO / Non-Profit	Donor, Programme, Partner	Donor type, project code, grant ref, fund head	Donors, Projects, Expenses, HR, Payroll, Reports
Government / Institution	Department, Item, Contractor	Budget head, sanction no., requisition no.	Purchases, Expenses, HR, Payroll, Reports
School / Education	Student, Course/Fee Head, Vendor	Grade, section, guardian, admission no., roll no.	Fees, HR, Payroll, Reports

5.2 Entity Aliasing System

Every core entity name is configurable per company using the __entity() helper. Navigation labels, form field labels, PDF documents, notification messages, report titles — all respect the alias. A pharmacy sees "Patient" everywhere the system would say "Customer." Implementation: entity_aliases table keyed by company_id + entity_key, with fallback to business type default, then system default.

5.3 Dynamic Custom Fields Engine

Field Type	Example Use Case	Performance Strategy
Text (short)	Generic drug name, Style no., LC reference	VARCHAR column — indexed if marked filterable via generated column
Text (long)	Product description, supplier notes	TEXT column — not indexed
Number	Carton qty, commission %, weight KG	DECIMAL column — indexed if filterable
Date	Expiry date, shipment date, warranty expiry	DATE column — indexed always (date fields are commonly filtered)
Dropdown (single)	Drug category, fabric type, buyer country	VARCHAR column + options JSON in definition
Multi-select	Certifications, applicable sizes, markets	JSON column
Boolean	Controlled substance, VAT applicable, bonded	TINYINT(1) — indexed
File upload	Drug license, LC document, NID scan	File path VARCHAR — file in storage
URL	Supplier catalogue, product spec sheet	VARCHAR column
Formula (calculated)	Commission = amount × rate	Computed on render, not stored — prevents stale data

Performance: MySQL generated columns with indexes are created automatically for any custom field marked "filterable." This gives native-speed filtering at any scale without per-company schema changes.

5.4 Dynamic Workflow Engine

Each company configures their own state machine per document type. Configurable elements: statuses, allowed transitions, per-role permissions per transition, single/parallel/sequential approval, automation actions on transition (notify, update stock, create journal entry), rejection routing, optional SLA timers (post-MVP).

Document types with configurable workflows: Purchase Orders, Sales Orders/Invoices, Expense Claims, Leave Requests, Payroll Runs, Credit Notes/Returns, Salary Advances, Stock Transfers, Payment Requests.

5.5 Dynamic Dashboard, Reports, Alerts & Simplified Mode

- Dashboard: widget-based grid, per-company configuration, real-time via Reverb. Widgets include: Sales, Purchases, Cash Balance, Low Stock, Expiry Alerts, Pending Approvals, Attendance, Receivables, Payables, Best Sellers, VAT Due.
- Report Builder: choose entity, select fields (including custom fields), group/aggregate, filter, visualise, save as template, schedule with email delivery, export PDF/Excel (Pro+).
- Alert Rules: any field + any condition + any channel (in-app/email/SMS) + any target role + repeat behaviour — fully configurable per company.
- Simplified Mode: for solo entrepreneurs and micro-businesses — stripped navigation (Dashboard, Income, Expenses, Clients, Invoices), shorter forms, no warehouse/HR/POS visible. Same codebase, UI configuration only.

6. Security Architecture

Security is not an afterthought in GenERP BD. It is a first-class architectural requirement. Every layer of the application — database, API, file system, network, authentication — has explicit security controls. This section defines them completely so they are built correctly from day one, not retrofitted later.

6.1 Authentication Security

Password Security:

- Passwords hashed with bcrypt (cost factor 12) — Laravel default, never stored in plain text
- Minimum password policy: 8 characters, at least one uppercase, one number, one special character — enforced at registration and password change
- Password history: last 5 passwords cannot be reused — prevents cycling back to old passwords
- Secure password reset: time-limited tokens (expire in 60 minutes), single-use, invalidated on use

Session Security:

- Session cookies: HttpOnly (not accessible to JavaScript), Secure (HTTPS only), SameSite=Strict (CSRF protection)
- Session lifetime: 120 minutes idle timeout — auto logout with session invalidation
- Session regeneration on every login — prevents session fixation attacks
- Concurrent session limit: configurable per company (default: 3 active sessions per user) — new login beyond limit invalidates oldest session
- Session data encrypted in Redis using Laravel's built-in session encryption

Brute Force Protection:

- Login rate limiting: 5 failed attempts per IP per 15 minutes — then lockout with CAPTCHA challenge
- Account lockout: 10 failed attempts on a specific account — account locked for 30 minutes, owner notified by email
- Failed login tracking in failed_login_attempts table: IP, user agent, timestamp, target user
- Geographic anomaly detection (post-MVP): alert if login from unusual country

Two-Factor Authentication (2FA):

- TOTP-based 2FA (Google Authenticator, Authy compatible) — available to all users, mandatory for owner/admin roles on Enterprise plan
- Backup codes: 10 single-use codes generated on 2FA setup — stored hashed, shown once
- 2FA bypass detection: all 2FA events logged as security_events
- Recovery flow: lost 2FA device requires manual identity verification by GenERP BD support (Enterprise) or account recovery email chain (Pro/Free)

6.2 SQL Injection Prevention

SQL injection is the most common critical vulnerability in web applications. GenERP BD eliminates it architecturally — not just by policy.

Primary Defence — Eloquent ORM Only:

- All database queries use Laravel's Eloquent ORM with parameterised bindings. Raw SQL is categorically forbidden in application code — enforced by AI development rules (Section 9) and code review.
- Eloquent automatically parameterises all values: User::where('email', $email) compiles to SELECT * FROM users WHERE email = ? with $email as a bound parameter — never interpolated into the SQL string.
- No string concatenation in database queries — ever. No $db->query("SELECT * FROM users WHERE id = ".$id). This is a hard rule enforced by Cursor AI rules.

Secondary Defence — Query Builder Safety:

- When using DB::table() query builder (rarely), all values must use ? placeholders or named bindings. Whereraw() is forbidden. Cursor AI rule flags any use of whereRaw() and requests review.
- Column names from user input: if a user can influence a column name (e.g. in a sort parameter), a whitelist validation is applied before use. Never: DB::orderBy($request->sort). Always: DB::orderBy(in_array($request->sort, $allowedColumns) ? $request->sort : 'id').

Tertiary Defence — Database User Permissions:

- The application database user has only SELECT, INSERT, UPDATE, DELETE on the application database — no DROP, CREATE, ALTER, TRUNCATE, or FILE permissions.
- Even if SQL injection bypasses ORM (theoretically impossible with pure Eloquent), the database user cannot drop tables, create admin accounts, or read the file system.

Defence in Depth — Input Validation:

- All user input validated using Laravel Form Requests before reaching any database layer.
- Integer IDs: validated as integer and existence-checked via ->exists() rule — not just cast.
- Enum fields: validated against explicit array of allowed values.
- Free-text fields: max length enforced, HTML stripped via strip_tags() before storage.

6.3 Cross-Site Scripting (XSS) Prevention

- Laravel's Blade templating auto-escapes all output using {{ $variable }} — renders as HTML entities, not executable JavaScript. Unescaped output {!! !!} is forbidden in templates unless explicitly review-approved for trusted internal content.

- Content Security Policy (CSP) header: restricts which scripts can execute — blocks inline scripts and unauthorised external script sources. Set via middleware on all responses.
- Filament v4 escapes all user-provided content in its components by default.
- Stored XSS prevention: all text stored in the database is HTML-stripped on input (strip_tags()) and escaped on output. No raw HTML from user input is ever rendered.
- DOM-based XSS: Alpine.js (used by Filament) context-escapes all dynamic content — no innerHTML with user data.
- HttpOnly session cookies: even if an XSS payload executed, it cannot read the session cookie.

6.4 Cross-Site Request Forgery (CSRF) Protection

- Laravel CSRF middleware enabled on all POST, PUT, PATCH, DELETE routes — every state-changing request requires a valid CSRF token.
- CSRF tokens are per-session, cryptographically random, validated server-side.
- SameSite=Strict on session cookies: browser will not send cookies on cross-origin requests, providing a second layer of CSRF protection.
- API routes: CSRF not applicable — protected by Sanctum token authentication instead.

6.5 DDoS & Rate Limiting Protection

Network Layer (Cloudflare — Free Tier):

- All traffic routed through Cloudflare — acts as a reverse proxy and absorbs volumetric DDoS attacks before they reach the server.
- Cloudflare's anycast network distributes attack traffic across their global infrastructure.
- Cloudflare WAF (Web Application Firewall) on free tier blocks known attack patterns (SQLi, XSS, OWASP Top 10) at the network edge.
- Rate limiting at Cloudflare level: block IPs sending >1,000 requests per minute to any endpoint.
- Challenge mode for suspicious traffic: Cloudflare issues CAPTCHA challenge to flagged IPs before allowing through.

Application Layer (Laravel Rate Limiting):

- Login endpoint: 5 attempts per IP per 15 minutes (throttle:5,15)
- Password reset: 3 requests per IP per hour
- API endpoints: 60 requests per minute per token (Pro), 200 per minute (Enterprise), blocked (Free)
- File upload endpoint: 10 uploads per user per minute
- Payment request submission: 3 per company per hour (prevents spam)
- Report generation: 10 report jobs per company per hour (prevents CPU exhaustion)
- All rate limit counters stored in Redis — atomic increments, automatic expiry

Slow HTTP Attack Protection (nginx config):

- client_body_timeout 10s — disconnect clients that take too long to send the request body
- client_header_timeout 10s — disconnect slow header attacks

- keepalive_timeout 30s — release idle connections
- limit_req_zone and limit_req in nginx for additional server-level rate limiting before Laravel is even hit

6.6 Multi-Tenant Data Isolation

A bug that exposes one company's data to another company is one of the most catastrophic possible failures for a SaaS product. GenERP BD prevents this architecturally, not just by policy.

Global Eloquent Scope:

- Every model that holds company data implements CompanyScope — a global Eloquent scope that automatically appends WHERE company_id = ? to every query.
- This scope is applied at the model level — it cannot be forgotten in a controller or service because it is part of the model definition itself.
- The active company_id is set once in SetActiveTenantMiddleware and stored in a request-scoped singleton — it cannot be tampered with by user input during the request lifecycle.

API Multi-Tenancy:

- API tokens are tied to a specific company_id at creation time. The token identifies both user and company. It is not possible to use a valid token to access a different company's data.
- Even if a user is a member of multiple companies, each company requires its own API token.

Isolation Testing:

- Automated Pest tests in the test suite specifically test cross-company isolation: create two companies, create data in company A, assert that authenticated requests as company B return zero results for company A's data.
- These tests run on every commit via GitHub Actions CI/CD. A regression in tenant isolation will be caught before it reaches staging.

6.7 Authorisation & Access Control

- Every Filament resource, every API endpoint, and every action has an explicit Policy class that checks: is the user authenticated? does the user belong to the active company? does the user's role have permission for this action?
- Laravel's Gate and Policy system used throughout — never manual role checks like if($user->role === "admin").
- Spatie Permission: roles (owner, admin, manager, accountant, hr_manager, pos_cashier, staff) and granular permissions (invoices.create, invoices.edit, invoices.delete, reports.export) — both checked per-company.
- Sensitive operations (delete, bulk action, payroll approve, audit log view) require explicit permission — not just a general admin role.

- Field-level visibility: sensitive fields (employee salary details, customer credit limit) hidden from roles without explicit permission — not just greyed out, actually not sent to the browser.

6.8 File Upload Security

- File type validation: whitelist of allowed MIME types checked against the actual file signature (not just the extension). Allowed: images (jpg, png, webp), documents (pdf, xlsx, docx, csv). Executables, PHP, JS, SVG — rejected.
- File size limits: 5 MB per file, 50 MB total per upload session — enforced at both Nginx and Laravel level.
- File storage: uploaded files stored in /storage/app/private/ — never in a web-accessible public directory. Files served via a controller that checks authorisation before streaming.
- File names: original file name discarded. Files stored with a UUID-based name (e.g. 3f8a1b2c-4d5e.pdf) — prevents path traversal and name collision attacks.
- Virus scanning: ClamAV integration on the upload job — files scanned before being made available. Infected files quarantined and admin alerted.

6.9 Data Encryption

Layer	What is Encrypted	Method
Data in transit	All HTTP traffic between client and server	TLS 1.3 enforced. TLS 1.0 and 1.1 disabled. HSTS header with 1-year max-age.
Session data	Session stored in Redis	Laravel session encryption (AES-256-CBC) — even if Redis is compromised, sessions are unreadable
Sensitive database fields	API tokens, 2FA secrets, payment transaction IDs in storage	Laravel's Encrypted casts on sensitive model attributes — AES-256 encryption via app key
Database backups	Daily database dump files	GPG encrypted before upload to backup storage — decryption key held only by server admin
Environment secrets	App key, database password, API keys	Stored in .env file, not in codebase. .env in .gitignore always. Never logged.
File uploads	Documents uploaded by users	Stored in private directory. File names are UUIDs. No encryption at rest of files in MVP — post-MVP for healthcare/enterprise data.

6.10 Security Headers

Every HTTP response from GenERP BD includes the following security headers, set by a SecurityHeadersMiddleware applied globally:

Header	Value	Purpose
Strict-Transport-Security	max-age=31536000; includeSubDomains; preload	Forces HTTPS for 1 year — prevents SSL stripping attacks

Content-Security-Policy	default-src 'self'; script-src 'self' (allowlisted CDNs); style-src 'self' fonts.googleapis.com	Prevents XSS and data injection attacks
X-Frame-Options	DENY	Prevents clickjacking — GenERP BD pages cannot be embedded in iframes
X-Content-Type-Options	nosniff	Prevents MIME type sniffing attacks
Referrer-Policy	strict-origin-when-cross-origin	Limits referrer information sent to external sites
Permissions-Policy	camera=(), microphone=(), geolocation=()	Denies access to device hardware APIs not needed by the app
X-XSS-Protection	1; mode=block	Legacy browser XSS filter — defence in depth

6.11 Dependency & Supply Chain Security

- Composer packages: composer audit run on every CI/CD build — fails the build if any known vulnerability is found in a dependency.
- npm packages: npm audit run on every CI/CD build.
- Automated dependency updates: Dependabot configured to auto-open PRs for security patches — reviewed and merged within 24 hours.
- Pinned dependency versions in composer.lock and package-lock.json — no surprise version changes on deploy.
- Minimum external packages principle: every third-party package is evaluated for necessity. More packages = larger attack surface.

6.12 Infrastructure Security

- SSH access to VPS: key-based authentication only — password SSH login disabled on server.
- SSH port: changed from default 22 to a non-standard port — reduces automated scan noise.
- Fail2ban: automatically bans IPs with repeated failed SSH login attempts.
- Firewall (UFW): only ports 80 (HTTP → redirected to HTTPS), 443 (HTTPS), and the custom SSH port are open. All other ports blocked.
- Database: MySQL listens on localhost only — never exposed to the internet. Database access requires SSH tunnel for admin work.
- Redis: listens on localhost only, password protected, no public exposure.
- Server updates: unattended-upgrades configured for automatic security patch application on the OS.
- Principle of least privilege: application runs as a dedicated non-root system user with only the permissions it needs.

6.13 Backup & Disaster Recovery

- Daily automated database backup: mysqldump run at 2 AM, GPG encrypted, uploaded to off-server storage (separate VPS or object storage).

- Daily file storage backup: uploaded user files backed up to off-server storage.
- Retention: 30 daily backups, 12 monthly backups retained.
- Backup restoration test: monthly — restore backup to staging environment, verify data integrity.
- Recovery Time Objective (RTO): target 4 hours from failure to restoration.
- Recovery Point Objective (RPO): maximum 24 hours data loss (daily backup cycle).
- Monitoring: uptime monitoring (UptimeRobot or similar) with SMS/email alert if server down for >2 minutes.

6.14 Security Audit & Penetration Testing

- Pre-launch: manual penetration test against OWASP Top 10 vulnerabilities — run by developer using OWASP ZAP against staging environment.
- Post-launch (Month 3): professional penetration test by a BD-based security firm — budget from early revenue.
- Ongoing: Sentry error monitoring alerts on suspicious error patterns (e.g. mass 403s from one IP, unusual error spikes).
- Security event log: critical events (failed 2FA, mass export, unusual login) recorded in security_events table and reviewable by superadmin.
- Responsible disclosure: a security@generp.bd email address with a clear responsible disclosure policy published on the website from day one.

7. Bangladesh Regulatory Compliance

7.1 VAT & Mushak Framework

Mushak Form	Name	When Generated
Mushak 6.3	VAT Tax Invoice (Challan)	On every VAT sale to a VAT-registered buyer — NBR-compliant format
Mushak 6.6	Credit Note / Debit Note	On sales return or price adjustment after VAT invoice
Mushak 9.1	Treasury Challan	When depositing VAT collected to government treasury
Mushak 6.1	Purchase Input Tax Register	Monthly record of all VATable purchases with input tax
Mushak 6.2	Sales Output Tax Register	Monthly record of all VATable sales with output tax
Monthly VAT Return	Output VAT minus Input VAT = Net VAT Payable	Exportable in NBR-compatible Excel format for online filing

BD VAT Rate	Applies To
15%	Standard — most goods and services
10%	Selected services (audit, consultancy)
7.5%	Selected goods and services (contractors)
5%	Selected goods (handloom, cottage industry)
2%	Truncated rate for certain traders
0% (Zero-rated)	Exports — VAT at 0%, input VAT refundable
Exempt	Essential goods, specific medicine categories

7.2 TDS & VDS (Withholding Tax)

- TDS (Advance Income Tax): deducted from supplier payments per Income Tax Ordinance 1984 Section 52. Rate per supplier category. Auto-calculated on payment entry. Monthly remittance report for NBR.
- VDS (VAT Deducted at Source): deducted from service supplier payments per VAT Act 2012 Section 49. Payer remits VAT directly to NBR.
- TDS certificate generated for supplier on request. All TDS/VDS deductions audit logged.

7.3 Payroll Compliance

- Bangladesh Labour Act 2006: annual leave (18 days), sick leave (14 days), casual leave (10 days) — configurable leave types with BD law defaults.

- Wage Board: grade-based minimum wage for RMG factories. Minimum wage enforcement warning. Overtime at double rate.
- Festival bonus: 2 bonuses/year (Eid-ul-Fitr, Eid-ul-Adha) at one month's basic — auto-calculated.
- Provident Fund: configurable % deduction and employer contribution. PF statement per employee.
- Income tax slabs: stored as configurable data, updatable by superadmin annually without code deployment.

8. Subscription Enforcement Architecture

8.1 Data Model

Table	Key Columns	Purpose
plans	id, name, slug, price_bdt, billing_cycle, features (JSON), limits (JSON)	Plan definitions — changes require only DB update, no code deployment
subscriptions	id, company_id, plan_id, status (active/grace/expired/suspended), started_at, expires_at, grace_until	One active subscription per company — status drives all enforcement
payment_requests	id, company_id, plan_id, amount_bdt, payment_method, transaction_id, status (pending/verified/rejected), verified_by, verified_at	Manual payment verification flow — see Section 3.2
subscription_invoices	id, company_id, amount_bdt, status, paid_at	Billing history — generated on payment verification
usage_counters	id, company_id, metric, current_value	Real-time metered usage — Redis-backed atomic operations

8.2 Middleware Stack

Order	Middleware	Action
1	AuthenticateMiddleware	Verify user is logged in and session valid
2	SetActiveTenantMiddleware	Set company_id scope, load subscription from Redis into request context
3	CheckSubscriptionStatusMiddleware	If expired past grace_until → redirect to /subscription/expired. If grace → warning banner. If suspended → support page.
4	EnforceModuleAccessMiddleware	Check if requested module slug is in plan modules array. If not → 403 with upgrade prompt.
5	CheckFeatureFlagMiddleware	For feature-level actions (export, API, Mushak forms, entity aliasing): check boolean in plan features JSON. If false → 403 with upgrade prompt.

Zero extra DB queries per request for subscription checking — subscription context is cached in Redis and loaded once in middleware step 2.

8.3 Limit Boundary Behaviour

Stage	Threshold	UX
Warning	80% of any quota	Yellow banner on relevant page: "400 of 500 products used. Upgrade for unlimited."
Soft block	100% of quota	Create button greyed with lock icon → upgrade modal on click. Existing data fully accessible.
Grace period	Subscription expired	7 days full access. Red banner. Daily escalating reminder emails.
Hard block	Grace period ended	Read-only mode. All data visible. No new records. Full-screen upgrade prompt. Data never deleted.

8.4 Free Tier Permanent Restrictions

Enforced at route/middleware level — not just UI. Cannot be bypassed by any client-side manipulation:

- PDF invoices always watermarked
- Report export (PDF/Excel) routes return 403
- Scheduled report delivery blocked
- API routes return 403 for all Free company requests
- Entity aliasing configuration page blocked
- Custom number sequences blocked
- Mushak form generation blocked
- Notification template customisation blocked
- Audit log auto-pruned to 30 days by nightly job
- Dashboard widget add button hidden after 5 (and endpoint blocked)

9. Tax Configuration Engine

Tax rates and categories are defined per company and never hardcoded. A VAT-exempt company sees no tax fields anywhere. A VAT-registered company gets full Mushak compliance. A VAT-registered company that is also a TDS deductor gets both systems.

9.1 Tax Groups

Example Tax Group	Rate	Compound?	Mushak Generated
Standard VAT	15%	No	Yes — Mushak 6.3
Reduced VAT (Contractor)	7.5%	No	Yes
Export Zero-Rate	0%	No	Yes — zero-rated
VAT Exempt	Exempt	No	No
Supplementary Duty + VAT	SD 45% then VAT 15% on (value+SD)	Yes	Yes — compound challan
TDS — Contractor (Section 52)	5% TDS deducted on payment	N/A (payment-side)	No — deducted on payment
VDS — Service	15% VDS on service payment	N/A (payment-side)	No — remitted to NBR

9.2 VAT-Exempt Companies

Single toggle in company settings: "VAT Registered: Yes/No." When No: tax groups hidden, Mushak forms disabled, invoices show no VAT fields. Clean and uncluttered for non-VAT businesses (most small shops and freelancers).

10. Number Sequencing Engine

Every document has a configurable number format per company, per document type, per branch. Atomic MySQL transactions prevent duplicate numbers under concurrent creation.

Config Field	Description	Example
Prefix	Fixed text at start	INV-, PO-, RX-, EXP-
Suffix	Fixed text at end	-DHA (Dhaka branch code)
Number length	Minimum digits with zero-padding	5 → 00001
Start value	Starting counter (for mid-year migrations)	1001, 5000
Reset period	When counter resets: never / yearly / monthly	Yearly → INV-2026-00001 → INV-2027-00001
Include year	Embed current year	Yes → INV-2026-00001
Include month	Embed current month	Yes → INV-2026-01-00001
Include branch code	Embed branch identifier	Yes → INV-DHA-2026-00001
Preview	Live format preview as configured	INV-DHA-2026-00001

NBR Rule: VAT invoice numbers (Mushak 6.3) must be strictly sequential with no gaps. Deleted invoices cannot be renumbered. The engine enforces this — sequence config is locked for VAT invoice document types after first use.

11. Inter-Branch Operations

11.1 Branch Structure

Company → Branches → Warehouses. Branch users are restricted to their assigned branches. Company owner and accountant see all branches. Branch Manager sees only their branch.

11.2 Stock Transfers

- Stock Transfer document: source branch/warehouse → destination branch/warehouse
- Configurable workflow: Created → In Transit → Received (or Rejected)
- Stock deducted from source on "In Transit." Added to destination on "Received."
- Partial receipts: receive 80 of 100 — remaining 20 stay In Transit
- Full audit trail on all inter-branch movements

11.3 Branch Reporting

Report	Description
Company consolidated	Totals across all branches — sales, purchases, inventory, P&L
Branch-level	Same reports filtered to single branch — branch manager view
Branch comparison	Side-by-side: Branch A vs B vs C on key metrics
Inter-branch transfers	All stock movements between branches in a period
Branch P&L	Revenue and costs per branch — requires branch-level expense tagging

12. Data Migration & Import Strategy

Entity	Import Method	Validation
Products / Services	Excel template — core + custom fields	Duplicate SKU check, unknown category/warehouse flagged
Opening stock	Excel template — product, warehouse, qty, cost	After products imported. Creates opening stock balance.
Customers	Excel template — contact, credit terms, opening balance	Opening balance creates AR journal entry
Suppliers	Excel template — contact, payment terms, opening balance, TDS rate	Opening balance creates AP journal entry
Employees	Excel template — personal info, department, salary structure	Salary components must match company structure
Opening account balances	Excel template — account, debit/credit, date	Sets starting trial balance for migrating businesses
Historical sales	Excel template (optional)	Imported as closed/paid — for reporting continuity only
Attendance (historical)	CSV from biometric machine	Standard biometric CSV: date, employee ID, time in/out

All imports: user downloads template → fills data → uploads → system validates row-by-row → shows error report (red rows with specific errors, green rows ready) → user confirms → background job processes → completion notification with summary.

13. API Architecture

API access available on Pro and Enterprise plans. REST JSON API, versioned (/api/v1/), Sanctum token auth, per-company tokens, rate limited per plan. All API calls logged in audit log.

13.1 Core Endpoints (MVP)

Resource	Endpoints	Notes
Products	/products (GET, POST), /products/{id} (GET, PUT, DELETE)	Includes custom fields
Customers	/customers (GET, POST), /customers/{id} (GET, PUT), /customers/{id}/ledger (GET)	Entity alias respected
Suppliers	/suppliers (GET, POST), /suppliers/{id} (GET, PUT)	Includes TDS rate
Invoices	/invoices (GET, POST), /invoices/{id} (GET), /invoices/{id}/pdf (GET)	POST triggers workflow
Purchases	/purchases (GET, POST), /purchases/{id} (GET, PUT)	Status transition via PUT
Expenses	/expenses (GET, POST), /expenses/{id} (GET)	
Stock	/stock/movements (GET), /stock/transfer (POST)	Real-time stock per warehouse
Employees	/employees (GET, POST), /employees/{id} (GET, PUT)	Salary excluded unless HR scope
Reports	/reports/{slug} (GET)	JSON report data
Webhooks (Enterprise)	/webhooks (GET, POST, DELETE)	invoice.created, payment.received, stock.low etc.

13.2 API Security

- Tokens scoped: read-only / read-write / module-specific (e.g. only stock and products)
- Tokens revocable instantly — revocation propagated via Redis cache invalidation
- All API calls logged: endpoint, method, status code, response time, timestamp
- Suspicious API usage (mass data export, rapid sequential calls) triggers security_events alert
- Rate limiting: 1,000/hr (Pro), 10,000/hr (Enterprise), 0 (Free — hard blocked)
- Idempotency keys on POST endpoints — prevents double-posting from network retries

14. Notification Template System

Channel	When Used
In-app toast (Reverb)	Real-time: low stock, new approval, payment verified
Notification centre	Persistent inbox — all alerts, cleared manually or after 30 days
Email (Mailgun/SES)	Invoices to customers, approval requests, payslips, subscription alerts, payment verification
SMS (post-MVP)	Critical: overdue invoice, salary processed, low stock

Template variables available in all templates: {{ company_name }}, {{ user_name }}, {{ entity_name }} (aliased), {{ document_number }}, {{ amount }}, {{ due_date }}, {{ quantity }}, {{ action_url }}, {{ sender_name }}, {{ days }}, {{ plan_name }}.

Both Bangla and English versions of every template — sent in recipient's preferred language. Pro+ companies customise templates (WYSIWYG for email, plain text for in-app/SMS). Free companies use system defaults.

15. Audit Log Design

15.1 What is Logged

Category	Actions Logged
Authentication	Login (success+failed), logout, password change, 2FA events, token created/revoked
Company/settings	Settings changed, plan change, module toggle, user role changed, user invited/removed
Financial documents	Invoice/PO/expense created/edited/deleted/approved/rejected, payments recorded, journal entries
Inventory	Stock received, issued, adjusted, transferred, product created/edited/deleted
HR/Payroll	Employee changes, payroll run created/approved/processed, leave approved/rejected
Configuration	Custom field changes, workflow edits, number sequences, alert rules, notification templates
Data operations	Import run (entity, count, errors), export (report, filters, format), bulk actions
API access	Every API call: token, endpoint, method, status, timestamp
Security events	Failed 2FA, suspicious login, mass export, repeated failed auth

15.2 Schema & Performance

Column	Type	Notes
id	BIGINT	Primary key
company_id	BIGINT	Tenant scope — indexed
user_id	BIGINT	Who acted (null for system/scheduled)
action	VARCHAR	created, updated, deleted, approved, login, export etc.
entity_type	VARCHAR	Invoice, Product, Employee, PurchaseOrder etc.
entity_id	BIGINT	ID of affected record
entity_label	VARCHAR	Human label: "Invoice INV-2026-00123" — for display
old_values	JSON	Field snapshot before action (null for creates)
new_values	JSON	Field snapshot after action (null for deletes)
ip_address	VARCHAR	Client IP
created_at	TIMESTAMP	When action occurred — partition key

Performance: audit_logs table partitioned by month (MySQL). Writes are async via queued jobs — user-facing operations never slowed by audit logging. Indexes on company_id, entity_type, user_id, created_at.

16. MVP Module List

#	Module	Free	Pro+	Service Mode
1	Foundation & Multi-Company	Core	Custom domain, white-label	Yes
2	Products & Inventory	Core (500 SKU limit)	Unlimited, bulk import	Yes — Service Catalogue
3	Purchasing & Suppliers	Core	Advanced import	Yes — service purchases
4	Sales & Invoicing	Core	Clean PDFs, Mushak, scheduled reminders	Yes — service invoicing
5	Expenses	Core	Full custom fields, budget heads	Yes
6	Basic Accounting & Ledger	Core	Full history, export	Yes
7	HR & Payroll	Core	Advanced salary, wage board tools	Yes
8	POS (Point of Sale)	Core	Full session management	No — inventory dependent
9	Batch & Expiry	Core	Full traceability report	No — inventory dependent
10	CRM (Customers)	—	Pro+ only	Yes
11	Reports & Dashboard	Basic	Full builder, exports, scheduled	Yes

Service Catalogue Mode: when a company has no physical inventory (agencies, clinics, consultancies) the Inventory module becomes a service catalogue — no stock movements, warehouses, or barcodes. Just services with prices, added to invoices.

17. Bilingual Implementation

Component	Implementation
Translation files	resources/lang/bn/ + resources/lang/en/ — all visible strings translation-keyed. Zero hardcoded text.
Default	Bangla (bn) — detected from browser Accept-Language on first visit
Per-company	Company Settings → preferred language — overrides system default
Per-user	User profile → language preference — overrides company default
Entity aliases	__entity() helper respects language setting
Filament	Filament v4 built-in translation + custom Bangla string files
Font	Noto Sans Bengali via Google CDN — applied in Filament theme CSS
Database	UTF8mb4_unicode_ci — full Bangla storage and sorting
Bangla numerals	bangla_number() helper: 123 → □□□ — toggled per company in settings
Dates	Carbon with Bangla month names — toggled per company: □□ □□□□□□□□□□ □□□□
PDF	DomPDF with Noto Sans Bengali embedded — all PDFs render Bangla correctly
Emails	Bangla and English template versions — sent in recipient's preferred language
RTL readiness	CSS structure RTL-compatible for future Arabic/Urdu support (Bangla is LTR)

AI-generated Bangla translations (from en.json via Claude 4) are approximately 85% accurate. A native Bangla speaker must review and correct all AI-generated strings before launch. Budget this review time explicitly.

18. AI-Assisted Development — Rules & Standards

These rules exist because AI coding tools (Cursor, Claude, Copilot) generate code that is syntactically correct but may miss security requirements, BD-specific business rules, bilingual requirements, or multi-tenancy constraints. These rules are non-negotiable and must be set as Cursor rules from day one. They apply to every file generated, every line reviewed.

18.1 Cursor AI Rules (Set in .cursor/rules)

Create a file at .cursor/rules in the project root containing the following rules. Cursor will apply these to every code generation session automatically.

RULE 1 — Multi-Tenancy (Highest Priority):

- Every Eloquent model that holds company data MUST have protected $fillable containing company_id AND MUST use the CompanyScope global scope via the booted() method or HasCompanyScope trait.
- NEVER generate a model without checking if it needs company_id. Ask explicitly: "Does this entity belong to a specific company? If yes, add company_id and CompanyScope."
- NEVER generate a controller or service that sets or overrides company_id from user input. company_id is always read from Auth::user()->active_company_id, never from $request.
- Every Filament resource MUST have getEloquentQuery() overridden to return Model::query() (which triggers the global scope) — never Model::all() or Model::withoutGlobalScope().

RULE 2 — SQL Injection Prevention (Zero Exceptions):

- NEVER generate raw SQL queries. NEVER use DB::statement() with user input. NEVER use whereRaw() with user input. NEVER concatenate user input into any query string.
- ALWAYS use Eloquent ORM methods: ->where(), ->whereIn(), ->find(), ->firstOrFail().
- If sorting or ordering by a user-provided column name, ALWAYS validate against a whitelist first: $allowed = ['name','created_at']; $col = in_array($request->sort, $allowed) ? $request->sort : 'created_at';
- ALWAYS use Form Requests for validation — never $request->input() directly in controller logic without prior validation.

RULE 3 — Bilingual (Every String):

- NEVER hardcode any visible string in PHP, Blade, or Filament. ALWAYS use __("key") or trans("key") for every label, placeholder, message, button text, error message, notification message, and PDF content.
- ALWAYS add the key to BOTH resources/lang/en/module.php AND resources/lang/bn/module.php simultaneously. Never add a key to one without the other.
- For entity names, ALWAYS use __entity("Customer") not __("Customer") — this respects the entity aliasing system.

- For numbers in output (reports, invoices), ALWAYS use the bangla_number() helper when rendering to PDF or invoice views.

RULE 4 — Security Headers & Validation:

- NEVER trust user input. ALWAYS validate in a Form Request class before any processing.
- For file uploads: ALWAYS validate MIME type (using 'mimes:' rule), file size ('max:'), and store with Storage::put() using a UUID filename. NEVER use the original filename.
- For IDs in URLs (route model binding): ALWAYS ensure the model uses CompanyScope so a user cannot access another company's record by guessing an ID.
- ALWAYS use $this->authorize() in controllers and Filament actions to check the Policy before the operation.

RULE 5 — Subscription Enforcement:

- NEVER add a new module route or Filament panel resource without tagging it with the correct module middleware: middleware(['module:inventory']).
- NEVER add a new premium feature without adding a feature flag check: if(!$company->plan->features['feature_name']) abort(403).
- NEVER increment a usage counter manually — always use the UsageCounterService::increment('metric') method which handles Redis atomics and MySQL sync correctly.

RULE 6 — Bangladesh VAT & Tax:

- NEVER calculate tax as a simple percentage on the invoice total. ALWAYS calculate per line item based on the product's tax_group.
- NEVER hardcode a tax rate (like 0.15 for 15% VAT). ALWAYS read the rate from the tax_group record.
- For compound taxes (Supplementary Duty + VAT): SD is calculated first, then VAT is applied on (item_value + SD_amount). Never apply both on item_value.
- For Mushak 6.3 generation: ALWAYS use the MushakInvoiceService — never generate the Mushak PDF inline in a controller.

RULE 7 — Testing (No Exceptions):

- EVERY new model, service, and API endpoint MUST have corresponding Pest tests generated in the same session. Never defer test writing.
- EVERY test that creates company data MUST create two companies and assert that company B cannot access company A's data (tenant isolation test).
- EVERY financial calculation (VAT, TDS, payroll, commission) MUST have a unit test with at least 3 known input/output pairs.
- EVERY test MUST use RefreshDatabase or database transactions — never leave test data in the database.

RULE 8 — Filament v4 Standards:

- ALWAYS use Filament v4 components. NEVER use Filament v2 or v3 syntax.

- ALWAYS define getTableColumns(), getFormSchema(), getTableFilters(), getTableActions() — never leave these empty.
- For all Filament tables: ALWAYS add ->searchable() to name/title columns, ->sortable() to date and amount columns.
- For all Filament forms: ALWAYS group fields logically using Section components. Never a flat list of fields.
- ALWAYS add ->required() or ->nullable() explicitly on every form field — never rely on database default.

RULE 9 — Custom Fields Integration:

- When generating a Filament resource for any major entity (Product, Customer, Invoice, Employee, etc.): ALWAYS include the renderCustomFields() method that appends dynamic custom fields to the form schema.
- ALWAYS include the custom field columns in the table via renderCustomFieldColumns() method.
- NEVER hardcode custom field names in any form, table, filter, or report — they must always be read from custom_field_definitions.

RULE 10 — Code Quality:

- ALWAYS use Service classes for business logic. Controllers are thin — they receive a request, call a service, return a response.
- ALWAYS use Action classes for single-responsibility operations (CreateInvoiceAction, ProcessPayrollAction).
- ALWAYS use Job classes for long-running operations: report generation, payroll run, bulk import, PDF generation, email sending.
- NEVER put business logic in a Blade template or Filament component directly.
- ALWAYS add a doc block comment to every Service and Action class explaining what it does, what it expects, and what it returns.
- ALWAYS use PHP 8.2+ features: typed properties, readonly classes, enums for status fields, match expressions instead of switch.

18.2 Daily AI Workflow

11. Write a precise architecture prompt: entity name, relationships, BD-specific business rules, security requirements, bilingual keys needed
12. Paste the Cursor rules reminder at the top of the prompt: "Apply all rules in .cursor/rules — especially multi-tenancy, SQL injection prevention, and bilingual requirements"
13. AI generates: migration, model, Form Request, Service, Filament resource, Pest tests
14. Review each generated file against the rules checklist (Section 18.3)
15. Fix any violations — BD compliance edge cases that AI misses are the most common
16. Run tests: ./vendor/bin/pest — all must pass before committing
17. Commit to Git → GitHub Actions runs CI/CD → auto-deploy to staging → manual verification

18.3 Pre-Commit Review Checklist

Check every AI-generated file against this list before committing:

Check	Question	AI Error Rate
Tenant scope	Does every new model have company_id and CompanyScope?	High — AI often forgets
No raw SQL	Is there any raw query, whereRaw(), or string concatenation in queries?	Medium
Form Request	Is all user input validated via Form Request before use?	Medium
Translation keys	Is every visible string using __() or __entity()?	High — AI loves to hardcode strings
Both lang files	Is the key added to both en/ and bn/ files?	Very High — AI adds to en only
Tax calculation	Is tax calculated per line item from tax group, not as flat rate on total?	High
Policy check	Does every controller action call $this->authorize()?	Medium
File upload	Is file stored with UUID name, MIME validated, size limited?	High
Usage counter	Is any new resource creation going through UsageCounterService?	Medium
Module tag	Is the new route tagged with module middleware?	Medium
Tests	Are Pest tests generated including tenant isolation test?	Very High — AI defers tests

18.4 Prompt Templates for Common Tasks

New Module Prompt Template:

> *Build a [MODULE_NAME] module for GenERP BD. Apply all rules in .cursor/rules. Requirements: (1) The [EntityName] model must have company_id with CompanyScope. (2) All strings in PHP and Blade must use __("module.key") — add to both lang/en/module.php and lang/bn/module.php. (3) Use [EntityAlias] as the entity display name via __entity(). (4) Tax calculation: per line item from tax_group, not flat percentage. (5) Generate: migration, model, CompanyScope trait application, Form Request with all validation rules, [EntityName]Service with full business logic, Filament resource with form, table, filters, actions. (6) Generate Pest tests: basic CRUD, tenant isolation (company A cannot see company B data), VAT calculation with 3 known pairs. Filament: v4 only. No hardcoded strings. No raw SQL.*

New API Endpoint Prompt Template:

Add a REST API endpoint for [resource] to GenERP BD API v1. Apply all .cursor/rules. Requirements: (1) Route in api/v1.php with Sanctum auth middleware and module middleware. (2) Controller uses $request->user()->active_company for company scope — never trusts request body for company_id. (3) Full Form Request validation. (4) Idempotency key support on POST. (5) Response uses ApiResponseHelper with standard envelope: {success, data, message, errors, meta}. (6) Rate limiting applied. (7) Action logged to audit_log. (8) Generate Pest API tests: successful case, validation failure case, unauthorized case, wrong company case (403).

New Report Prompt Template:

Build a [REPORT_NAME] report for GenERP BD. Apply all .cursor/rules. Requirements: (1) Report runs as a queued job (implements ShouldQueue) — never inline in a request. (2) Query uses Eloquent with company scope — no raw SQL. (3) Results cached in Redis for 10 minutes keyed by company_id + filters hash. (4) Supports export to PDF (DomPDF with Bangla font) and Excel (via PhpSpreadsheet). (5) Export is Pro+ only — check feature flag before generating. (6) Column headers use __("reports.column_name") with both lang files. (7) Numbers formatted via bangla_number() helper for PDF/invoice output. (8) Generate Pest test for calculation accuracy with 2 known data sets.

19. Development Plan (20 Weeks)

Solo developer. AI generates 70-80% of code. Developer writes architecture prompts, applies the pre-commit checklist, corrects BD-specific business rules AI misses, and handles judgment-heavy decisions.

Phase 0: Environment & Architecture (Week 1)

Most critical phase. Every decision here is permanent. Do not rush.

- Install Laravel 12, FilamentPHP v4, MySQL 8, Redis, Laravel Horizon, Laravel Reverb
- Configure stancl/tenancy with shared DB + company_id global Eloquent scope
- Configure Spatie Permission for per-company RBAC
- Configure UTF8mb4_unicode_ci on all tables
- Set up bilingual scaffolding: lang/bn + lang/en + __() + __entity() helpers
- Install Noto Sans Bengali via Google CDN — apply in Filament theme
- Configure DomPDF with Bangla font embedding
- Set up Pest + GitHub Actions CI/CD with security checks (composer audit, npm audit)
- Configure Sentry error monitoring
- Configure Cloudflare (free) for DDoS protection and WAF
- Implement SecurityHeadersMiddleware (all headers in Section 6.10)
- Configure fail2ban and UFW firewall on VPS
- Set Cursor rules file (.cursor/rules) with all 10 rules from Section 18.1
- Create clean folder structure: App/Models, App/Services, App/Actions, App/Jobs, App/Policies, App/Http/Requests, App/Http/Middleware
- Create CompanyScope trait and HasCompanyScope interface — used by every company-scoped model
- Create UsageCounterService — all resource creation/deletion routes through here
- Create ApiResponseHelper — all API responses use this

Deliverable: Running app, security foundations in place, all dev rules configured. Nothing else.

Phase 1: Multi-Company + Subscription (Weeks 2–3)

- Registration, login (with brute force protection), email verification, password reset
- Company creation wizard: business type template → entity aliases applied → first branch → number sequences
- Company switcher in Filament navigation
- Branch management
- Plans table, subscriptions model, payment_requests model with manual verification workflow
- Payment request submission UI and admin verification panel
- Full middleware stack: tenant scope, subscription status, module access, feature flags

- Usage counters with Redis-backed atomic operations
- Freemium enforcement: 3 company cap, user limits, module blocking
- Spatie roles and permissions per company
- User invitation system
- Company settings: logo, VAT toggle, BIN, language, Bengali numerals, invoice template
- Number sequencing engine with configuration UI
- Entity aliasing system + __entity() helper active throughout
- Superadmin panel: company management, payment verification, subscription management

Deliverable: Multi-company UX end to end. Payment request submitted and verified by admin.

Phase 2: Dynamic Fields + Workflow Engine (Weeks 4–5)

- Custom field definitions UI with all 10 field types
- Filament dynamic form renderer
- Custom field values storage + generated column indexes for filterable fields
- Custom field columns in list views
- Visual workflow builder with automation runner
- Business type template loader applying preset custom fields + workflows on company creation

Deliverable: Company admin adds custom fields and builds document workflows — no code needed.

Phase 3: Products & Inventory (Weeks 6–7)

- Categories, units, products with variants and custom fields
- Service Catalogue mode
- Multi-warehouse stock management and movements
- Opening stock entry for migrating businesses
- Low stock thresholds → alert rules engine
- Weighted average cost inventory valuation
- Bulk import via Excel with row-level validation report

Deliverable: Full inventory with custom fields, service mode, real-time stock.

Phase 4: Purchasing & Suppliers (Week 8)

- Supplier profiles with TDS rate, credit terms, custom fields
- Purchase orders with configurable workflow and number sequencing
- Goods received (partial), stock auto-update, TDS/VDS auto-calculation on payment
- Purchase returns, supplier ledger and ageing report

Deliverable: Full purchase cycle with TDS compliance.

Phase 5: Sales & Invoicing + Tax Engine (Weeks 9–10)

- Customers with custom fields, credit terms, entity alias

- Quotations → sales orders → invoices with BD VAT per line item (tax group based)
- Mushak 6.3 auto-generation for VAT-registered companies (Pro+)
- Sales returns, credit notes (Mushak 6.6)
- Customer payments: partial, split payment types
- Invoice PDF: watermarked (Free), clean bilingual branded (Pro+)
- Tax groups configuration UI, all BD VAT rates, compound tax (SD+VAT), zero-rated
- Monthly VAT liability report, Mushak 6.2 and 6.1 generation
- Monthly TDS/VDS remittance report

Deliverable: Full sales cycle with complete BD tax compliance.

Phase 6: Expenses + Basic Accounting (Weeks 11–12)

- Expense categories, entry, receipt attachment, approval workflow, petty cash, budget heads
- Chart of accounts (pre-configured by business type template)
- Auto journal entries from all modules
- AR/AP summaries, cash/bank tracking, P&L, balance sheet, manual journal

Deliverable: Business owner sees P&L, cash position, receivables/payables in real time.

Phase 7: HR & Payroll (Weeks 13–14)

- Employee profiles, departments, designations, custom fields
- Attendance (manual + biometric CSV import)
- Leave types and configurable approval workflow
- Configurable salary structure builder
- Monthly payroll: full auto-calculation, BD income tax slabs, wage board compliance, festival bonus, PF
- Payslip PDF (bilingual), bank payment list export

Deliverable: Complete BD-compliant monthly payroll cycle.

Phase 8: Inter-Branch + POS + Batch/Expiry + CRM (Weeks 15–16)

- Branch-level access control, stock transfers with workflow, branch reporting
- Dedicated POS screen — fast-billing, split payment, receipt (PDF + thermal), session management
- Batch number and expiry tracking, FEFO logic, expiry alerts, traceability report
- CRM module (Pro+): enhanced customer profiles, follow-up reminders, segmentation

Phase 9: Notifications + Alert Rules (Week 17)

- Alert rules engine with full configuration UI
- Real-time in-app notifications (Reverb + Echo)
- Bilingual notification template system with variable substitution
- Email notifications with bilingual template rendering

- Notification preferences per user
- Scheduled report delivery via queued jobs

Phase 10: API + Data Migration Tools (Week 18)

- REST API v1 with all core endpoints, Sanctum token management with scopes
- API security: rate limiting, idempotency keys, audit logging of all API calls
- Webhook system (Enterprise): event subscriptions, delivery with retry, webhook log
- API documentation (auto-generated)
- Excel import templates for all major entities with row-level validation
- Background job import processing, import history log, opening balance import

Phase 11: Audit Log (Week 19)

- audit_logs with monthly partitioning and async writes
- Audit log UI: filtering, search, old/new value diff view
- Export to Excel (Pro+), SIEM forward (Enterprise)
- Nightly pruning job for Free plan (30-day retention)

Phase 12: Testing, Security Review & Polish (Week 19 continued)

- Pest coverage: all API endpoints, financial calculations (VAT, TDS, payroll), workflow transitions, tenant isolation
- OWASP ZAP penetration test against staging — fix all findings
- Database indexing review — all FKs, filter columns, generated custom field columns
- Redis caching for dashboard widgets, reports, subscription context
- Query optimisation: zero N+1 queries
- Full Bangla translation review by native speaker
- Bengali numeral and Bangla date formatting tested on all PDFs
- Security headers verified with securityheaders.com
- Mobile responsiveness check on all Filament resources
- User-friendly error messages for all validation and system errors

Phase 13: Pre-Launch (Week 20)

- Demo data: retail shop, pharmacy with Mushak VAT, 50-person company with full HR/payroll
- 5-minute onboarding wizard: business type → company details → first branch → key settings → done
- Public landing page: features, pricing, demo video embeds, free signup CTA
- Bangla demo videos for YouTube (retail, pharmacy, RMG/HR, solo entrepreneur, service)
- VPS deployment: HTTPS, domain, Sentry, Cloudflare, fail2ban, UFW configured
- Beta launch to 10–20 users across business types with structured feedback

Deliverable: Production-ready, security-hardened, BD-compliant ERP ready for real businesses.

20. Timeline Summary

Phase	Focus	Weeks	Deliverable
0	Environment, Architecture, Security Foundations, Cursor Rules	1	Secure base, all dev rules active
1	Multi-Company + Manual Payment + Subscription Engine	2–3	Company UX, payment request verification
2	Dynamic Custom Fields + Workflow Engine	4–5	Configuration layer working on all entities
3	Products & Inventory	6–7	Full inventory with service mode
4	Purchasing & Suppliers	8	Full purchase cycle with TDS/VDS
5	Sales & Invoicing + Tax Engine	9–10	BD VAT + Mushak compliance
6	Expenses + Basic Accounting	11–12	P&L, balance sheet, AR/AP
7	HR & Payroll (BD compliant)	13–14	Full payroll with wage board and income tax
8	Inter-Branch + POS + Batch/Expiry + CRM	15–16	Multi-branch, fast POS, expiry tracking
9	Notifications + Alert Rules	17	Real-time bilingual alerts
10	API + Data Migration	18	REST API v1, import tools
11	Audit Log	19	Full immutable audit trail
12	Testing, Security Audit, Polish	19 cont.	95%+ test coverage, OWASP pen test passed
13	Pre-Launch	20	Production deploy, beta users

21. Post-MVP Roadmap

Everything below is out of scope for the 20-week MVP. Priority determined by user feedback and revenue after launch.

21.1 Immediate Post-Launch (Months 1–3)

- SMS notification gateway (SSL Wireless / bDBL)
- WhatsApp Business notifications
- Full double-entry accounting with trial balance
- Biometric device direct integration (ZKTeco API)
- Multi-currency support
- Gratuity calculation (BD Labour Act)
- SSLCommerz / bKash payment gateway (replace manual verification)
- Advanced POS: table management for restaurants

21.2 Revenue-Dependent (Months 3–12)

- Manufacturing / Production module: BOM, job cards, MRP
- E-commerce: Shopify, WooCommerce, Daraz, Chaldal integrations
- Advanced CRM: pipeline, deal tracking, rep targets
- Asset Management: depreciation, maintenance schedule
- Project Management: costing, milestones, client billing
- NBR online VAT return filing integration
- White-labeling for resellers
- SOC 2 Type 2 audit (for enterprise sales)

21.3 AI Features (Post-Revenue Investment Required)

AI features require GPU compute investment. Only after sufficient MRR. Not promised to any customer at launch.

- Sales forecasting and demand prediction
- Anomaly detection: suspicious expenses, unusual stock loss
- Natural language report generation
- AI payroll discrepancy detection
- Business health score and AI recommendations dashboard

22. Hosting & Infrastructure

22.1 MVP Hosting

Component	Solution	Est. Monthly Cost
Web + App Server	XeonBD / Alpha Net VPS (BD-based, low latency)	1,500–3,000 BDT
Database	MySQL 8 on same VPS initially	Included
Cache / Queue	Redis on same VPS	Included
CDN / DDoS / WAF	Cloudflare free tier	Free
File Storage	Local VPS → Wasabi/DO Spaces when >20 GB	Included → ~500 BDT/mo
Email	Mailgun or AWS SES	500–1,000 BDT/mo
SSL	Let's Encrypt (auto-renewing)	Free
Error Monitoring	Sentry free tier	Free
Uptime Monitor	UptimeRobot free tier	Free
Domain	.com or .com.bd	1,000–2,000 BDT/year

22.2 Scale Path

As customers grow: (1) Separate database server. (2) Managed Redis cluster. (3) Object storage for all uploads. (4) Read replicas for reporting. (5) Dedicated servers for Enterprise clients with data sovereignty. (6) SOC 2 audit when targeting large enterprise. (7) Replace manual payment verification with SSLCommerz/bKash gateway once revenue justifies the integration cost.

23. Launch Strategy

23.1 Beta (End of Week 20)

10–20 beta users across business types: 2 retail shops, 1 pharmacy, 1 RMG/manufacturing, 1 service business, 1 solo entrepreneur. Free Pro plan for 3 months in exchange for structured feedback and bug reports. Fix all critical issues before public launch.

23.2 Public Launch

- Landing page: features, pricing, demo videos, free signup, live chat
- Bangla YouTube videos: retail, pharmacy, HR/payroll, solo, service — one per type
- Facebook ads targeting BD business owners (primary SME acquisition channel)
- BD tech influencer and business YouTuber outreach
- Facebook group presence: BD business, accounting, ERP communities
- Product Hunt (English) for developer/startup audience

23.3 Conversion Strategy

Free tier limits create natural upgrade moments: 3-company cap, 3-user-per-company cap, 500-SKU limit, watermarked PDFs, blocked exports, no Mushak forms. A growing business hits at least one limit within 3–6 months of active use. Every limit boundary shows an upgrade prompt — not a wall, an invitation with clear benefit explanation.

24. Security Incident Response Plan

What happens when something goes wrong. Having this plan before launch means you respond calmly and correctly, not in panic.

24.1 Severity Classification

Severity	Definition	Example	Response Time
Critical (P0)	Data breach, cross-tenant exposure, authentication bypass, active attack in progress	Company A can see Company B data, mass login bypass, database exfiltrated	Immediate — stop the breach within 1 hour, notify affected companies within 24 hours
High (P1)	Significant data exposure risk, major feature broken, payment system error	SQL injection found (not exploited), all invoice PDFs failing, wrong payroll calculation	Fix within 24 hours, communicate to affected users within 48 hours
Medium (P2)	Feature partially broken, performance degraded, minor data inconsistency	Report showing wrong totals for one company, slow dashboard for high-volume company	Fix within 72 hours
Low (P3)	Minor UI bug, cosmetic issue, non-critical feature broken	Bangla numeral not showing in one report, wrong colour on a button	Fix in next regular release

24.2 Critical Incident Response (P0)

18. Detect: Sentry alert, user report, or automated monitoring trigger
19. Contain: immediately revoke affected tokens / sessions, put affected company into maintenance mode if needed, enable Cloudflare "Under Attack" mode
20. Assess: determine scope — how many companies affected, what data exposed, how long
21. Fix: patch the vulnerability, deploy to staging first, test, deploy to production
22. Verify: confirm the vulnerability is closed, run tenant isolation tests
23. Notify: affected company owners notified by email within 24 hours with clear explanation of what happened, what was exposed, what was done, and what they should do
24. Post-mortem: written incident report within 72 hours — what happened, root cause, timeline, fix applied, prevention measures

24.3 Responsible Disclosure

security@generp.bd email published on website from day one with clear policy: report vulnerabilities confidentially, we will respond within 48 hours, fix within 30 days, credit the reporter publicly if they wish. No legal threats against good-faith researchers.

25. Pre-Launch Security & Standards Checklist

Run this checklist at the end of Phase 12 (testing/security review) before any beta user is given access.

Authentication & Access

- bcrypt cost factor 12 confirmed in config/hashing.php
- 2FA available and working for all users
- Brute force protection: 5 attempts → lockout, tested and verified
- Session cookies: HttpOnly, Secure, SameSite=Strict confirmed in browser devtools
- Password reset tokens expire in 60 minutes and are single-use — tested
- Role-based access: verified that each role cannot access endpoints outside their permissions

SQL Injection & Input Security

- Grep codebase for DB::statement, whereRaw, raw — zero occurrences
- All controllers use Form Request validation — no $request->input() without prior validation
- All file uploads use UUID filename and MIME type whitelist — tested with malicious files
- OWASP ZAP active scan run against staging — all high/medium findings resolved

Multi-Tenancy Isolation

- Automated tenant isolation Pest tests passing: company A cannot see company B data
- API tested: token for company A returns 403 for company B endpoints
- All new models confirmed to have CompanyScope and company_id

Security Headers & Network

- securityheaders.com scan on staging: A+ rating achieved
- SSL Labs scan on domain: A rating achieved (TLS 1.3, no weak ciphers)
- Cloudflare enabled and traffic routing through CDN confirmed
- UFW firewall: only ports 80, 443, custom SSH open — all others blocked
- fail2ban active and tested with repeated failed SSH attempts
- MySQL and Redis: localhost only — not exposed to internet

Bangladesh Compliance

- VAT calculation tested with known values: 15% standard, 7.5% contractor, 0% export
- Compound tax (SD + VAT) tested with known values
- Mushak 6.3 PDF generated and reviewed against NBR format requirements
- Mushak 6.2 monthly sales register generated and verified totals

- TDS deduction tested on supplier payment with known rate
- Payroll income tax calculation tested against BD slab table for 3 salary ranges

Backup & Monitoring

- Daily backup running and first backup successfully uploaded to off-server storage
- Backup restoration tested on staging — data integrity confirmed
- Sentry receiving errors from production environment
- UptimeRobot monitoring configured with email and SMS alert

AI Development Rules

- .cursor/rules file committed to repository and confirmed active in Cursor
- All 10 rules verified against existing codebase — no violations
- Pre-commit checklist used on last 3 feature commits — confirmed effective

---