

# Project Changelog

This log details the structural layout enhancements, centralization of PDF engine binary paths, quality optimizations, fallback implementations, and the new Custom Landing Page Template System.

## 35. Super Admin Documentation Rewrite

Completely overhauled the Super Admin Panel documentation in the main `projectworlds_docs.html` file to transition from high-level summaries into detailed, actionable instructional guides.

### Documentation Enhancements

- **Step-by-Step Guides:** Replaced static feature grids with numbered, step-by-step instructional blocks ( `pw-step-list` ) across all 19 major Super Admin modules (including School Management, Subscription Plans, Orders, Landing Page Builder, and Platform Settings).
- **Capabilities Summary:** Restored a master capabilities grid on the top-level Super Admin Overview page, providing a fast index of all available modules before drilling down into the specific guides.
- **Gateways Deep-Dive:** Expanded the Payment & SMS Gateways section to explicitly document all supported third-party providers (Stripe, Razorpay, Twilio, MSG91, etc.) and added a dedicated setup guide for configuring the platform-level Meta/DLT Comms Wallet credentials.

## 34. Inventory & Stock Management — Full Module Overhaul

Transformed the inventory module from a thin name-and-quantity CRUD into a complete stock lifecycle system covering procurement, storage, issuance, and point-of-sale — delivered in three phases with a redesigned dashboard, demo data, an in-app guide, and an automated test suite. All records remain strictly per-school (tenant-scoped) with granular permissions.

### Phase 1 — Items, Stock Control & Dashboard

- **Expanded Item Model:** Items now carry SKU (auto-generated, unique per school), item type (consumable / saleable), unit of measure (predefined list + free-text "Other"), purchase & selling price, minimum-stock level, expiry date, storage location, barcode, photo, description, and active status.

- **Category, Supplier & Issuance Upgrades:** Categories gained type/description/icon; suppliers gained contact person, GSTIN, PAN, supplier type and active flag; item issuance became polymorphic — items can be issued to **staff, students, or departments** (previously staff-only) with an audit of who issued what.
- **Stock Adjustments:** A new audited adjustments screen for damage, expiry write-offs, loss, corrections, and opening balances — the controlled way to alter stock instead of hand-editing quantities.
- **Inventory Dashboard & Sub-Nav:** A new dashboard with stat cards (total items, low-stock, expired, stock value), an items-by-category chart, recent-activity timeline, and low-stock / expiring-soon alert tables — plus a module sub-navigation header matching the Academic module pattern.
- **Granular Permissions:** Replaced the single `inventory.manage` with `inventory.dashboard.view`, `inventory.items.manage`, `inventory.stock.manage`, `inventory.issues.manage`, and `inventory.suppliers.manage`. Backward compatible — existing roles/users keep access via a migration that grants the new permissions to anyone who had `inventory.manage`.

## Phase 2 — Saleable Items, POS & Vendor-on-Behalf Billing

- **Point of Sale:** A cart-based POS screen with type-to-search and **barcode scanner support** (keyboard-wedge), editable per-line quantity, price, discount and **GST %**, live totals, buyer selection (student / staff / walk-in), and multiple payment modes (cash / UPI / card / wallet / bank transfer). Sales decrement stock and generate a sequential, per-school invoice number.
- **Vendor-on-Behalf Billing (CBSE compliance):** Schools can issue an invoice under a **vendor's** name, GSTIN and address while acting as a collection agent — the printable invoice header switches to the vendor and notes "Collected by [School] on behalf of [Vendor]". A **Vendor Collections** report totals amounts owed to each vendor.
- **Printable Invoice, Sales History & Refunds:** A clean printable tax invoice, a searchable sales history, and a full-refund action that restores stock.

## Phase 3 — Procurement (Purchase Orders → Goods Receipt → Payments)

- **Purchase Orders:** Create draft POs with per-line cost and GST. POs require explicit **approval** (a dedicated `inventory.purchase.approve` permission) before any goods can be received.
- **Goods Received Notes (GRN):** Receiving against an approved PO records received/rejected quantities, increases stock, **refreshes the item's purchase cost to the latest PO cost**, and automatically advances the PO status (sent → partially received → received).
- **Supplier Payments:** Record payments against a PO (or general), with paid / balance tracking shown on each purchase order.

## Tooling, UI & Quality

- **Redesigned Dashboard UI:** Rebuilt the inventory dashboard with the same custom soft-card styling and colored icon badges as the Academic dashboard, a category doughnut chart, and an Operations summary (open POs, sales this month, suppliers).
- **In-App Module Guide:** A new "Guide" tab visually documents the full workflow (procurement → stock → issue → sale), with a tab-by-tab reference and a permissions table, so staff can understand the large module at a glance.
- **Demo Data Seeder:** `InventoryDemoSeeder` populates a school with realistic, varied data (low-stock, expiring and expired items, suppliers, issues, adjustments, sales including a vendor-on-behalf bill, purchase orders, a goods receipt and a supplier payment). Safe to re-run — it clears prior demo rows first.
- **Automated Tests:** Added a feature test suite covering SKU generation, issue/return stock movement, adjustment guards, sale total computation + refund, the PO approve→receive→cost/status flow, and supplier payments.

## 33. Academic Dashboard Enhancements & Mobile Optimization

Polished the Academic Dashboard for a more fluid, responsive, and geometrically balanced user experience across devices.

### Features added / Fixed

- **Scrollable Class Overview Widget:** Removed bulky pagination from the "Class & Section Overview" table. It now loads all classes in a single query with a sticky-header, vertically scrollable table for seamless browsing without page reloads.
- **Geometric Layout Alignment:** Permanently eliminated uneven visual gaps between the left table and right sidebar widgets. Applied strict fixed-height constraints (400px table / 250px timetable) that guarantee perfect alignment across columns without causing unnatural card stretching or internal white space.
- **Mobile Grid & Button Optimizations:** The four primary stat cards automatically arrange into a balanced 2x2 grid on mobile screens instead of stacking into a single tall column. The "Add New Class" header button smoothly collapses into a space-saving icon-only button on small devices.
- **Data Binding Fixes:** Corrected an issue where the "Student Count" column for classes displayed as zero due to a missing property binding. It now correctly maps the pre-calculated session counts to each row. Added missing external link icons to the Timetable Alerts widget header for consistency.

## 32. Data Explorer — Relationship Graph & Guard-Railed Editor

Extends the per-school Data Explorer with an interactive relationship graph, a carefully guard-railed manual editor (with a full audit trail), and a denser phpMyAdmin-style browsing experience.

### Features added

- **Relationship Graph (Cytoscape):** A per-school visual map of the database — every table that holds data is a node sized by row count and coloured by type, with foreign-key links drawn between them. Supports force / concentric / hierarchy / grid layouts, hover-to-highlight a table's relations, search-to-focus, and click-a-node to jump straight into browsing it. The library is vendored locally so it works fully offline.
- **Per-Record Relationship Map:** Each record's detail page now shows a focused ego-network graph — its parent records point in and its child records point out — built from data already on the page (no extra queries).
- **Guard-Railed Manual Editor:** Root admins can now insert, edit, and delete a single school's records directly. Editing is **read-only until a per-session "Enable edit mode" toggle** is switched on, and every change runs a **preview** → **confirm** → **commit** flow that shows an exact before/after diff. Updates and inserts require typing the table name plus a reason; **deletes additionally require the operator's password**.
- **Tenant-Safe Write Guarantees:** The target row is re-resolved within the school's scope on every operation; `school_id` and `id` are immutable; any changed or added foreign key must reference a row that exists **and belongs to the same school**; and values are validated for type, length and required-ness against the live schema. The audit-log table itself is protected from editing to preserve trail integrity.
- **Change History & Audit Log:** Every manual change is written to a dedicated log capturing the full before/after snapshot, the reason, the operator, IP and timestamp. A new **Edit History** screen lists all changes for a school with an expandable diff.
- **phpMyAdmin-Style Navigation & Dense UI:** Browsing and record pages gained a sticky, searchable **table-list sidebar** for jumping between tables without leaving the page, plus a far denser, isolated table style (compact rows, sticky headers, internal scrolling, aligned numeric counts) and a configurable rows-per-page control (25 / 50 / 100 / 200). Table counts are briefly cached so the navigation stays fast across a fleet of 1000+ schools.

## 31. Per-School Data Explorer (Read-Only)

A superadmin tool to browse and audit a single school's data inside the shared multi-tenant database — phpMyAdmin-style visibility, but tenant-scoped and guard-railed for a fleet of 1000+ schools. Reached via

**Manage** → **Data Explorer** on each school.

## Features added

- **Shared Tenant Data Map:** Extracted the school-scoping logic (tenant / dependent / belongs-to-parent table classification plus foreign-key-chain resolution) that powers Backup and Restore into a single `SchoolDataMap` service, so all three features agree on exactly which rows belong to a school. Backup output verified byte-identical after the refactor.
- **Scoped Dashboard:** Per-school landing page listing every table that holds data for that school, with scoped row counts, tenant/dependent badges, and a live table filter.
- **Tenant-Safe Table Browser:** Paginated, sortable, searchable view of any tenant or dependent table. Every query is forced through the scope resolver, so another school's rows can never appear. Includes one-click child-record drill-down.
- **Record Detail with Relations:** Single-record view that resolves foreign keys to human-readable labels (e.g. student names instead of raw IDs) and lists incoming child relations with live counts.
- **Audit-Log Foundation:** Added the `school_data_edits` table as the schema target for the upcoming guard-railed manual editor (captures before/after, reason, and operator for every change).

## 30. System Tooling & Gateway Integrations

### Features added

- **MSG91 SMS Gateway:** Integrated MSG91 as a supported SMS gateway, providing administrators with an additional robust provider for automated transactional notifications.
- **Send Test Mail Functionality:** Implemented a dynamic testing tool within the Superadmin Platform Settings. Administrators can now test SMTP configurations in real-time via a modal before finalizing the settings, ensuring reliable email delivery.
- **Library Barcode Generation:** Added a native "Download Barcodes" button to the Library Books management interface, allowing librarians to easily export and print barcode labels for inventory management.

## 29. Secure Integrated Document Viewer

Replaces native browser file tabs and downloads with an immersive, in-app Document Viewer modal. Integrated across Staff Profiles and the Parent Portal (Documents, Study Materials).

## Features added

- **Unified Alpine Component:** Created `x-document-viewer`, a robust Alpine.js component utilizing `pdf.js` to render PDFs and images natively within the application, maintaining user context.
- **Storage Path Refactoring & Security:** Fixed a critical bug in `StaffController@storeDocument` where staff documents were incorrectly uploaded to the private `local` disk but served via public URLs (causing 403 Forbidden errors). Uploads now target the `public` disk. Ran a background migration to move existing staff documents into the correct public storage structure.
- **Dynamic File Type Resolution:** Integrated an explicit `ext` property resolution handler. Handles URLs that obscure file extensions (e.g. AWS S3 signed URLs or dynamic routes) by reading the database `PATHINFO_EXTENSION` directly, ensuring the viewer always knows how to render the file.
- **Mobile-Optimized Layout:** Engineered a fully responsive flexbox header for the Document Viewer. Automatically stacks titles and close buttons on mobile devices while converting the zoom/pagination toolbar into a horizontally-scrollable area to prevent overflow and UI clipping.

## 28. Native Custom UI Google Translate

Replaces the default Google Translate toolbar with a completely custom AdminLTE-native language switcher dropdown, improving layout stability and performance.

## Features added

- **Custom UI Integration:** Added `language_switcher.blade.php` to the top navbar, hiding the bulky Google iframe and 40px layout shifts.
- **Instant Translation & Render Blocking Fix:** Google Translate is now initialized fully asynchronously after the DOM loads, completely removing the initial blank screen flicker.
- **Regional Native Fonts:** Automatically detects the selected language and dynamically injects specialized premium Google Fonts (like `Noto Sans Devanagari` and `Noto Sans Bengali`) strictly when needed, transforming the UI for regional users without bloating the English site.
- **Aggressive Cookie Reset:** Built a highly robust cookie destroyer loop to fix translation caching issues, ensuring users can natively revert back to English without being stuck in a translated DOM.
- **African Languages Supported:** Added Swahili, Hausa, Yoruba, and Igbo to the language dropdown options to support clients in Nigeria and Kenya out of the box.

## 27. Superadmin Communications Report

Adds a global platform-level report for the Superadmin to monitor all outgoing notification logs (SMS, WhatsApp, Push, Email).

### Features added

- **Communications Report:** Introduced `communicationsReport()` in `App\Http\Controllers\SuperAdmin\ReportController.php` which queries the `notification_deliverability_logs` table. Features efficient query clustering for KPI summary boxes (Total Sent, Total Failed, Breakdown by Channel) and uses `reorder()` to prevent MySQL strict mode (`ONLY_FULL_GROUP_BY`) conflicts.
- **Filtering & UI:** A new detailed AdminLTE view with extensive filters (by School, Channel, Status, Date Range). The data table includes an interactive modal to view raw message contents and failure reasons safely. Fixed summary card height mismatches using flexbox utilities.
- **CSV Exports:** Added `exportCommunications()` endpoint to safely chunk and stream the logs into a `.csv` file.

## 26. Superadmin Platform Default Notification Templates

Adds a fully-featured UI in the Superadmin dashboard to edit the "Platform Default" notification templates, allowing superadmins to define global fallback messages for new or unconfigured schools.

### Features added

- **Superadmin Template Controller** (`App\Http\Controllers\SuperAdmin\NotificationTemplateController.php`): Mirrors the school's template editing logic but specifically saves records with `school_id = null`, acting as the platform-level default. Added robust `DB::beginTransaction()` error handling to ensure atomic persistence and graceful 500 JSON API responses.
- **UI Integration** (`resources/views/superadmin/notification-types/index.blade.php`): Added an "Edit Defaults" button and ported the template editor modal from the school settings. Includes sequential numbering in the table for better UX.
- **Error Handling:** Backported the robust `try-catch` database transaction error handling into the `School\NotificationTemplateController` to ensure data consistency across both ends.

## 25. Dynamic Currency Refactoring

Eliminates all hardcoded instances of the Indian Rupee ( INR / ₹ ) across the application, replacing them with a fully dynamic currency resolution system suitable for international multi-tenant deployments.

### Key Changes

- **Currency Helper:** Introduced `platform_currency_code()` , `platform_currency_symbol()` , `school_currency_code()` , and `school_currency_symbol()` .
- **API & Payments:** The UPI Gateway, Parent App endpoints, and Comms Wallet APIs now dynamically resolve the school's active currency.
- **Dashboards & Views:** Superadmin analytics, school fee collections, and reporting tables render the dynamic currency symbol contextually instead of defaulting to India localization.
- **Onboarding:** Newly registered schools automatically inherit the Superadmin's platform default currency settings.

## 24. Editable Notification Templates — Phase 4 (WhatsApp Providers + DLT)

Adds opt-in WhatsApp **Official Meta Cloud (Graph) API** and **360dialog** approved-template sending, plus fuller **India SMS DLT** binding. Entirely additive and per-template opt-in: a channel only uses a provider when a school saves a WhatsApp template with `provider = meta_cloud` (or `360dialog`) **and** a `meta_template_name` . Every existing free-text WhatsApp send is byte-identical.

### WhatsApp provider routing

- **MetaWhatsAppGateway** ( `app/Services/Gateways/MetaWhatsAppGateway.php` ): `sendTemplate()` / `sendText()` against `graph.facebook.com/{api_version}/ {phone_number_id}/messages` . Pre-flight (config present, E.164 normalization), builds positional components from the template's `meta_components` map, masks recipients in logs, records `fbtrace_id` .
- **Error classification + retry** ( `TransientWhatsAppException` ): transient (HTTP 429/5xx, Meta `130429/80007/131056/131048` , connect/timeout) vs permanent (bad token, template paused, param mismatch, undeliverable). The gateway throws only on transient and never bubbles an uncaught error (no 500s).
- **360dialog** ( `ThreeSixtyDialogGateway::sendTemplate()` ): approved-template ( `type:template` ) sends with optional WABA namespace, alongside the existing free-text path. Also fixed a latent `$apiKey` type ( `string` → `?string` ).

- **WhatsAppChannel** now branches: a resolved Meta/360dialog template routes to the provider gateway (config read from `school.whatsapp_config['meta'] / ['threesixty_key']`); anything else uses the unchanged free-text SendWAPI path. If the chosen provider isn't configured, it falls back to free-text. Outcomes are written to `notification_deliverability_logs`.
- **No churn in the 12 wired notification classes:** the `RendersNotificationTemplates` trait exposes the resolved template to the channel via `whatsappRenderedTemplate()` / `whatsappTemplateData()` (render is memoized per channel), so provider routing needed zero edits to any `toWhatsApp()`.

## SMS DLT binding

- `CustomGateway` now also substitutes `[dlt_pe_id]` and `[sender_id]` URL placeholders (alongside the existing `[dlt_template_id]`).
- `SmsChannel` forwards `dlt_pe_id / sender_id`, auto-filled from the resolved SMS template (`smsRenderedTemplate()` hook) when `toSms()` doesn't supply them. Platform gateways ignore the extra arguments.

## Known follow-ups (not in this change)

- **Per-channel retry:** a transient Meta failure is logged, not rethrown — a queue retry re-runs the whole notification and would duplicate the other channels (SMS/push). True retry needs channel-isolated jobs.
- **Meta credential entry UI:** creds are read from `whatsapp_config`; a settings form to capture `phone_number_id / access_token /etc.` is still TODO.
- Meta delivery webhooks (Phase 5).

## Files

Action	File
NEW	<code>app/Exceptions/TransientWhatsAppException.php</code>
NEW	<code>app/Services/Gateways/MetaWhatsAppGateway.php</code>
MODIFIED	<code>app/Services/Gateways/ThreeSixtyDialogGateway.php</code>
MODIFIED	<code>app/Services/Gateways/CustomGateway.php</code>
MODIFIED	<code>app/Channels/WhatsAppChannel.php</code>
MODIFIED	<code>app/Channels/SmsChannel.php</code>
MODIFIED	<code>app/Traits/RendersNotificationTemplates.php</code>

## 23. Editable Notification Templates — Phase 2 Expansion (6 More Types)

Extends the editable-template system (sections 20–21) to the remaining parent-facing notification types, so each now appears in the "Customize Message" editor (SMS/WhatsApp/Push) with a legacy fallback identical to its old message.

### Low-risk — wired existing senders

- **live\_class\_start** ( **LiveClassReminderNotification** ): added the template trait + `templateData()` + per-channel `renderTemplate()` guards. `via()` now honours the per-type matrix toggles when a settings row exists, falling back to the previous global-switch behaviour when none does (no regression).
- **bus\_proximity\_alert**: refactored `SendProximityAlert` to dispatch a new `BusProximityNotification` through the standard channel stack instead of ad-hoc `Http / Mail` calls. This fixes the dead SMS placeholder (SMS now actually sends via the school gateway) and unifies WhatsApp through `whatsappChannel`.

### New classes + dispatch triggers

- **online\_exam\_publish** ( **OnlineExamPublishedNotification** ): dispatched from `School\Exam\OnlineExamController::store()` (online exams have no manual publish step — creation is the publish event) to active students of the exam's class/section in the current session, via the parent account.
- **library\_book\_issue** ( **LibraryBookIssuedNotification** ): dispatched from `School\Library\BookIssueController::store()` for student members, to the borrowing student's parent.
- **transport\_fees** ( **TransportFeesNotification** ): dispatched from `School\Transport\StudentTransportController::store()` when a transport fee is assigned (`fare > 0`), to the student's parent.

All new dispatches are fail-safe (wrapped so a notification error never breaks the underlying action) and fired after the DB transaction commits.

### Cleanup

- **Removed dead** `app/Channels/ParentFeePaid.php`: a non-autoloadable stub (file path  $\neq$  namespace), never dispatched, with an empty `toMail()`; it duplicated `fee_submission` ( `FeeSubmissionNotification` ), which already covers

parent fee-payment alerts. The `parent_fee_paid` type definition itself is left untouched.

## Registry & seeds

- `TemplateVariables` : registered variable catalogues for all 6 types.
- `NotificationTemplateSeeder` : added inactive platform-default SMS/WhatsApp/Push templates mirroring the legacy strings.
- `NotificationTypeSeeder` : added the `bus_proximity_alert` type (previously only seeded via migration) for consistency.

## Files

Action	File
MODIFIED	<code>app/Notifications/LiveClassReminderNotification.php</code>
NEW	<code>app/Notifications/BusProximityNotification.php</code>
NEW	<code>app/Notifications/OnlineExamPublishedNotification.php</code>
NEW	<code>app/Notifications/LibraryBookIssuedNotification.php</code>
NEW	<code>app/Notifications/TransportFeesNotification.php</code>
MODIFIED	<code>app/Jobs/SendProximityAlert.php</code>
MODIFIED	<code>app/Http/Controllers/School/Exam/OnlineExamController.php</code>
MODIFIED	<code>app/Http/Controllers/School/Library/BookIssueController.php</code>
MODIFIED	<code>app/Http/Controllers/School/Transport/StudentTransportController.php</code>
MODIFIED	<code>app/Support/Notifications/TemplateVariables.php</code>
MODIFIED	<code>database/seeder/NotificationTemplateSeeder.php</code>
MODIFIED	<code>database/seeder/NotificationTypeSeeder.php</code>
DELETED	<code>app/Channels/ParentFeePaid.php</code>

## 22. Parent Portal — Session-Change Crash Fix (Critical)

### Root Cause

- **currentSession Returns Null:** The `Student::currentSession()` relationship filters by `academic_session.is_current = true`. When the school admin switches the current academic session, any student not enrolled in the new session gets `null` from `currentSession`. Controllers then crash on `$student->currentSession->school_class_id` with Attempt to read property "school\_class\_id" on null (500 Internal Server Error).

### Fix Applied — Graceful Degradation

- **Controller Null Guards:** Added `if (!$student->currentSession)` early-return guards to 4 controllers. Instead of crashing, each controller now returns the view with safe empty defaults (empty collections, empty JSON arrays) and a `noCurrentSession` flag.
- **Blade Nullsafe Operators:** Replaced unsafe `$student->currentSession->schoolClass->name` chains with PHP 8.0 nullsafe `$student->currentSession?->schoolClass?->name ?? 'N/A'` in 2 Blade templates.
- **User-Facing Warning Banner:** Added a visible alert banner on the parent dashboard when `$noCurrentSession` is true, informing the parent that their child is not enrolled in the current academic session and to contact the school if unexpected.
- **Session-Independent Features Unaffected:** Fees, Library, Transport, Attendance History, Profile, and other pages that don't depend on `currentSession` continue working normally for unenrolled students.

### Files Changed

Action	File
MODIFIED	<code>app/Http/Controllers/Parent/DashboardController.php</code>
MODIFIED	<code>app/Http/Controllers/Parent/NoticeController.php</code>
MODIFIED	<code>app/Http/Controllers/Parent/TimetableController.php</code>
MODIFIED	<code>app/Http/Controllers/Parent/OnlineExamController.php</code>
MODIFIED	<code>resources/views/parent/dashboard.blade.php</code>
MODIFIED	<code>resources/views/parent/profile/edit.blade.php</code>

# 21. Fee Submission Notification — Duplicate WhatsApp/SMS Fix

## Root Cause Analysis

- **Double-Send on Same Phone Number:** The `FeeDepositObserver` dispatched `FeeSubmissionNotification` to both the parent `User` and the student `User`. The deduplication guard only compared User IDs (`$studentUser->id !== $parentUser->id`), but in most schools both accounts resolve to the **same phone number** (e.g. `father_phone`) via `routeNotificationForSms()` — resulting in the recipient receiving two identical WhatsApp/SMS messages per fee payment.
- **Contrast with Attendance:** The `StudentAttendanceObserver` only sends to `$student->parent` (single recipient), which is why attendance notifications correctly sent once.

## Fix Applied

- **Phone-Number-Aware Dedup ( `FeeDepositObserver.php` ):** The student `User` now only receives the notification if **both** their User ID and resolved phone number differ from the parent `User`. If both accounts resolve to the same `father_phone`, the duplicate send is skipped.

## Files Changed

Action	File
MODIFIED	<code>app/Observers/FeeDepositObserver.php</code>

# 20. Editable Notification Templates — Phases 0–3

## Phase 0 — DLT Pass-Through Fix

- **SMS DLT Template ID Forwarding ( `SmsChannel.php` ):** Fixed a silent bug where `SmsChannel` called the gateway's `send()` with only 2 arguments, causing the `[dlt_template_id]` URL placeholder in `CustomGateway` to always resolve to empty — non-compliant for India DLT (mandatory). The channel now forwards `dlt_template_id` from each notification's `toSms()` return array as the 3rd argument.

## Phase 1 — Foundation (Additive, Dormant)

- **Database Schema:** Created `notification_templates` migration with support for per-school overrides (`school_id` nullable — NULL = platform default), per-channel templates (`email|sms|whatsapp|push`), WhatsApp Meta Cloud fields (`provider`, `meta_template_name`, `meta_template_language`,

meta\_components JSON), and SMS DLT fields ( dlt\_template\_id , dlt\_pe\_id , sender\_id ). Unique constraint on ( school\_id , notification\_type\_id , channel , language ) .

- **Template Renderer ( TemplateRenderer.php )**: Resolve order: school template → platform-default template ( school\_id = NULL ) → null (legacy fallback). Substitutes {{variables}} in template body.
- **Rendered Template DTO ( RenderedTemplate.php )**: Immutable value object carrying body , subject , dlt\_template\_id , and Meta component builder for Phase 4.
- **Trait ( RendersNotificationTemplates.php )**: Mix-in for notification classes providing renderTemplate(\$channel) — returns a RenderedTemplate or null (legacy path).
- **Variable Registry ( TemplateVariables.php )**: Per-type variable catalogue with labels and sample values — powers editor variable chips and server-side validation.
- **Platform Default Seeder ( NotificationTemplateSeeder.php )**: Seeds platform defaults for exam\_result and fee\_submission (4 channels each), idempotent via updateOrCreate . Runs automatically in the migration's up() and is reversible in down() .

## Phase 2 — Wire Notification Classes (10 Types, 11 Classes)

- **Template-Driven Rendering**: Wired 10 notification types across 11 classes: exam\_result , fee\_submission , fee\_reminder\_due , student\_absent , student\_birthday , student\_punch\_out , homework\_assigned , behaviour\_incident , online\_admission\_submit , due\_slip\_generated . Each class received notificationTypeName() , templateData() , and renderTemplate() guards in toSms() / toWhatsApp() / toFcm() — with unchanged legacy fallback when no active template exists.
- **Email Deferred**: toMail() deliberately left on legacy across all classes (HTML mailable strategy pending).
- **Security Exclusion**: login\_credential notification excluded from the template system (contains sensitive OTP/password data).

## Phase 3 — School Editor UI

- **Template Editor Controller ( NotificationTemplateController.php )**: edit() returns per-channel content (school override, else platform default prefill) + variables as JSON; update() upserts school-owned rows. Validates unknown {{variables}} → 422, invalid Meta-components JSON → 422, and deletes override on empty body (reverts to legacy).
- **Routes**: school.settings.notifications.templates.edit|update under permission:settings.school.manage .
- **Settings Integration ( SettingController.php )**: notificationSettings() now passes \$templatableTypes from TemplateVariables::registeredTypes() so the "Customize Message" button only shows for wired types.
- **Blade Editor Modal ( notifications.blade.php )**: Tabbed editor (SMS / WhatsApp / Push) with clickable {{variable}} chips, live client-side preview, per-channel Active toggle, SMS DLT template ID field, and WhatsApp provider selector + Meta template-name/language/components fields (shown when provider is meta\_cloud or 360dialog ).

## Files Changed

Action	File
MODIFIED	app/Channels/SmsChannel.php
NEW	database/migrations/2026_05_31_120000_create_notification_templates_table.php
NEW	app/Models/NotificationTemplate.php
NEW	app/Services/Notifications/TemplateRenderer.php
NEW	app/Services/Notifications/RenderedTemplate.php
NEW	app/Traits/RenderNotificationTemplates.php
NEW	app/Support/Notifications/TemplateVariables.php
NEW	database/seeders/NotificationTemplateSeeder.php
MODIFIED	app/Notifications/ExamResultNotification.php
MODIFIED	app/Notifications/FeeSubmissionNotification.php
MODIFIED	app/Notifications/FeeDueNotification.php
MODIFIED	app/Notifications/StudentAbsentNotification.php
MODIFIED	app/Notifications/StudentBirthdayNotification.php
MODIFIED	app/Notifications/StudentPunchOutNotification.php
MODIFIED	app/Notifications/HomeworkAssignedNotification.php
MODIFIED	app/Notifications/BehaviourIncidentNotification.php
MODIFIED	app/Notifications/OnlineAdmissionSubmitNotification.php
MODIFIED	app/Notifications/DueSlipGeneratedNotification.php
MODIFIED	app/Notifications/StudentAttendanceNotification.php
NEW	app/Http/Controllers/School/NotificationTemplateController.php
MODIFIED	app/Http/Controllers/School/SettingController.php
MODIFIED	resources/views/school/settings/notifications.blade.php
MODIFIED	routes/school.php

# 19. Bulk Student Photo & Document Upload

## Architecture & Capabilities

- **ZIP + CSV Integration:** Developed two dedicated bulk upload modules—one for Student Photos and one for Student Documents. Schools upload a single ZIP archive containing all media files along with a lightweight CSV mapping file ( `photo_mapping.csv` or `document_mapping.csv` ), allowing hundreds of files to be ingested in one click.
- **Idempotent Background Processing:** Uploads are dispatched to background queue jobs ( `ProcessBulkStudentPhotos` and `ProcessBulkStudentDocuments` ). They track progress row-by-row, automatically caching database lookups for students, and safely skipping or overwriting existing records without duplicating data.
- **Progress & Error Tracking:** Integrated directly with the existing `ImportTask` model to provide real-time UI progress bars. To prevent memory exhaustion, error logs are automatically capped at 100 entries per job, keeping the database light even during massive failures.

## Robust Server Limit Handling

- **Dynamic Configuration Detection:** The UI now actively interrogates PHP's `upload_max_filesize` and `post_max_size` configuration. It natively restricts file inputs to the effective maximum size, ensuring that end-users cannot crash the server or experience silent failures.
- **Pre-Flight Size Validation:** Built server-side guards that intercept POST requests that exceed the configured size limit before they reach Laravel validation, providing a clean redirect and a descriptive error message instead of a generic server fault.
- **Magic Byte Security:** Implemented deep validation for photo uploads by checking the actual binary headers ("Magic Bytes") of files inside the ZIP, preventing malicious scripts disguised as images from being stored.

## Files Changed

Action	File
NEW	<code>app/Http/Controllers/School/Student/BulkStudentPhotoController.php</code>
NEW	<code>app/Http/Controllers/School/Student/BulkStudentDocumentController.php</code>
NEW	<code>app/Jobs/ProcessBulkStudentPhotos.php</code>
NEW	<code>app/Jobs/ProcessBulkStudentDocuments.php</code>
NEW	<code>resources/views/school/student/bulk-photo-upload/create.blade.php</code>
NEW	<code>resources/views/school/student/bulk-document-upload/create.blade.php</code>


Action	File
MODIFIED	routes/school.php
MODIFIED	resources/views/school/student/students/index.blade.php

## 18. Certificate Template Thumbnail Engine

### Core Architecture

- **Database Schema Expansion:** Added `thumbnail_path` to both `ready_templates` (Superadmin library) and `certificates` (School Admin templates), along with a `ready_template_id` foreign key on the `certificates` table to track template provenance.
- **Smart Cascading Inheritance:** Designed a sophisticated `getEffectiveThumbnailAttribute()` on the `Certificate` model. It checks for a custom thumbnail first, falls back to the parent `ReadyTemplate` thumbnail via relationship tracking, and defaults to an `iframe` preview if neither exists.
- **Zero-Downtime Migration:** Schema updates were made using nullable columns and `nullOnDelete()` constraints, ensuring zero disruptions for schools and 100% backward compatibility for older templates without thumbnails.

### UX & Performance Enhancements

- **Massive Render Optimization:** Replaced resource-heavy `srcdoc` iframes in the gallery grids with fast, static thumbnail `<img>` tags. This drastically reduces CPU overhead and DOM node counts when browsing large template libraries.
- **Click-to-Zoom Intuitive UI:** Scrapped clunky hover overlays and side buttons. The entire template image is now natively clickable with a `zoom-in` cursor, fading in a crisp white `fa-search-plus` icon perfectly centered in the preview area on hover.
- **Live Upload Preview & Override:** Added seamless AJAX-like image preview behavior to the file upload inputs on both Superadmin and School Admin edit pages. School Admins can instantly override an inherited Superadmin thumbnail with their own custom design.
- **UI Origin Badges:** Added contextual badges to the School Admin gallery:  `Original` preview when inheriting a Superadmin thumbnail, giving users clear visibility into the template's source.

### Files Changed

Action	File
NEW	database/migrations/2026_05_30_060802_add_thumbnail_support_to_templates.php

Action	File
NEW	resources/views/partials/_thumbnail_preview_modal.blade.php
MODIFIED	app/Models/ReadyTemplate.php
MODIFIED	app/Models/Certificate.php
MODIFIED	app/Http/Controllers/SuperAdmin/TemplateController.php
MODIFIED	app/Http/Controllers/School/Certificate/CertificateTemplateController.php
MODIFIED	resources/views/superadmin/templates/index.blade.php
MODIFIED	resources/views/superadmin/templates/create.blade.php
MODIFIED	resources/views/superadmin/templates/edit.blade.php
MODIFIED	resources/views/school/certificates/templates/index.blade.php
MODIFIED	resources/views/school/certificates/templates/create.blade.php
MODIFIED	resources/views/school/certificates/templates/edit.blade.php
MODIFIED	resources/views/school/certificates/templates/_template_library.blade.php
MODIFIED	resources/views/school/certificates/templates/_styles_and_scripts.blade.php

## 17. Topbar User Menu & Quick Nav Enhancements

### Dynamic Role-Based SVG Avatars

- **Fallback Avatars:** Added `adminlte_image()` and `generateSvgAvatar()` methods to the `User` model. When a user lacks a profile photo, the system now generates a beautiful, responsive SVG avatar featuring their initials on a gradient background that is distinctly color-coded based on their primary role (e.g., Red for Superadmin, Purple for School Admin, Blue for Teachers).

### Role-Aware Profile Routing

- **Intelligent Links:** Implemented `getProfileUrl()` and `getPasswordUrl()` on the `User` model. The user menu dropdown now dynamically routes the "My Profile" and "Change Password" links to the correct endpoints depending on whether the user is a Superadmin, Staff member, or Parent, avoiding 404s.

## Quick Nav Mega Menu

- **Data Parsing:** Developed a brand new `menu-item-quick-nav.blade.php` partial that intercepts the existing `$adminlte->menu('sidebar')` data. It parses the nested array structure, extracts top-level string headers and module submenus without requiring any additional database queries.
- **Masonry UI Grid:** Rendered the extracted modules into a dense, multi-column masonry layout (similar to professional SaaS app switchers).
- **Live Search:** Integrated a JavaScript-powered live search input inside the mega menu dropdown that instantly filters the nested module links.
- **Inline Cache-Busting CSS:** Injected structural CSS styles directly into the blade template to bypass aggressive browser CSS caching and guarantee immediate layout rendering for users.

## User Menu Cleanups

- **Removed Redundancy:** Removed the duplicate username text next to the avatar on the topbar trigger for a cleaner, modern look.

## Files Changed

Action	File
MODIFIED	<code>app/Models/User.php</code>
MODIFIED	<code>config/adminlte.php</code>
MODIFIED	<code>public/css/custom_v3.css</code>
MODIFIED	<code>resources/views/vendor/adminlte/partials/navbar/navbar.blade.php</code>
NEW	<code>resources/views/vendor/adminlte/partials/navbar/menu-item-dropdown-user-menu.blade.php</code>
NEW	<code>resources/views/vendor/adminlte/partials/navbar/menu-item-quick-nav.blade.php</code>

## 16. Hostel Dashboard Module

### Custom View & Controller

- **Hostel Dashboard:** Designed a brand new `Hostel Dashboard` page mapping exactly to UI designs, providing live statistical tracking for Total Hostels, Rooms, Occupied Beds, and Available Beds.
- **DashboardController:** Added a new controller dedicated strictly to collecting Hostel statistics, occupancy calculations (percentage based per-hostel), and recent allocations list without muddying existing controllers.

- **Dynamic Menu Injection:** Added a migration to inject the Hostel Dashboard menu item directly into the database's `master_menu_items` and assigned it to the `school_admin` role securely dynamically (without resetting seeded menu logic).

## Styling & Theming

- **Strict CSS Isolation:** Introduced CSS custom properties within `header.blade.php` applying a strict `.hst-` prefix. This completely isolated styling method guarantees zero class collisions across other dashboards (e.g. Transport or Library).
- **FontAwesome Optimization:** Refactored the dashboard to implement standard FontAwesome icons rather than Material Symbols, preventing icon-font rendering latency and UI glitches inside AdminLTE.
- **Universal Tab Headers:** Injected the dynamic isolated header and tabs ( `Dashboard` , `Student Allocation` , `Manage Rooms` , `Room Types` , `Manage Hostels` ) onto all index pages under the Hostel module for unified navigation.

## Files Changed

Action	File
NEW	<code>app/Http/Controllers/School/Hostel/DashboardController.php</code>
NEW	<code>database/migrations/2026_05_29_071617_add_hostel_dashboard_to_menu.php</code>
NEW	<code>resources/views/school/hostel/partials/header.blade.php</code>
NEW	<code>resources/views/school/hostel/dashboard.blade.php</code>
MODIFIED	<code>routes/school.php</code>
MODIFIED	<code>resources/views/school/hostel/hostels/index.blade.php</code>
MODIFIED	<code>resources/views/school/hostel/rooms/index.blade.php</code>
MODIFIED	<code>resources/views/school/hostel/room_types/index.blade.php</code>
MODIFIED	<code>resources/views/school/hostel/allocations/index.blade.php</code>

# 15. Transport Dashboard Module

## Custom View & Controller

- **Transport Dashboard:** Developed a comprehensive Transport Dashboard offering live tracking of total vehicles, active routes, daily commuters, and vehicle maintenance status.

- **DashboardController:** Added a dedicated `DashboardController` under the `Transport` namespace for isolated metric calculations.
- **Dynamic Menu Injection:** Created a migration ( `2026_05_29_060814_add_transport_dashboard_to_menu.php` ) to dynamically insert the `Transport Dashboard` into the sidebar for the `school_admin` role.

## Styling & Theming

- **Isolated CSS ( `.ts-` ): Encapsulated all `Transport dashboard` styling under a strict `.ts-` prefix in `partials/header.blade.php` to prevent global CSS bleed.**
- **Universal Tab Headers:** Integrated the new header with navigation tabs across all `Transport module index pages` (`Vehicles`, `Routes`, `Tracking`).

# 14. Library Dashboard Module

## Custom View & Controller

- **Library Dashboard:** Built a new `Library Dashboard` featuring a clean, zero-padding boundary layout for tracking total books, issued books, overdue returns, and available inventory.
- **DashboardController:** Added a dedicated `DashboardController` inside the `Library` namespace.
- **Dynamic Menu Injection:** Deployed a migration to inject the `Library Dashboard` into the `school_admin` menu.

## Styling & Theming

- **Isolated CSS ( `.lib-` ): Encapsulated `Library dashboard` styling under a strict `.lib-` prefix in `partials/header.blade.php`.**
- **Universal Tab Headers:** Integrated the `Library navigation tabs` (`Overview`, `Issue/Return Book`, `Categories`, `Manage Books`) consistently across all `Library module views`.

# 13. SuperAdmin Page CRUD Enhancements

## Validation & Slug Generation

- **Dedicated Form Requests:** Created `StorePageRequest` and `UpdatePageRequest` to replace inline `$request->validate()` calls, ensuring compliance with strict type and validation conventions ( `$request->validated()` ).

- **Pre-Validation Slug Generation:** Moved URL slug generation into `prepareForValidation()` inside the `FormRequests`. This fixes a critical bug where empty slugs would bypass the `unique:pages,slug` rule, generate a duplicate slug in the controller/model, and trigger a database Integrity constraint violation (500 Error).
- **Strict Return Types:** Added explicit `view` and `RedirectResponse` return types to all methods in `PageController.php`.

## UI Improvements

- **Validation Error Feedback:** Added global validation error display blocks ( `@if ($errors->any())` ) to the top of the form in both `create.blade.php` and `edit.blade.php`. Previously, validation failures would silently redirect back without showing the user what went wrong.

# 12. Fee Collection Audit & Analysis Enhancements

## Collector Audit Trail

- **Tracking Collections:** Added a `collected_by_user_id` field to the `fee_deposits` table to track exactly which admin or accountant processed each payment. Both UI ( `CollectFeeController` ) and backend services ( `FeePaymentService` ) now capture the authenticated user.
- **Daily Collection Report:** Enhanced the Daily Collection table to display a "Collected By" badge.
- **Collector-Wise Report:** Built a brand new audit report ( `school/reports/finance/collector-wise-report` ) that allows filtering collections by date range and specific staff member, complete with summary cards (cash/card/online) and Excel export capabilities.
- **Telegram Notifications:** "Collected By" name is now included in instant Telegram alerts to school admins.

## Collection Summaries & PDF Exports

- **Summary Tables:** Injected three compact summary tables (Collector-wise, Account-wise, and Payment Mode) at the bottom of the Daily Collection Report view.
- **PDF Export Sync:** Updated the `collection_pdf.blade.php` template to render the new "Collected By" column in the main table and include all three summary tables in the exported PDF layout.

## Fees Analysis Matrix Upgrades

- **Concession & Fine Columns:** Expanded the Fees Analysis Report matrix (grouped by Fee Heads or Months). Previously only showing Assigned/Paid/Due, it now breaks down `Concession` and `Fine` individually, ensuring `Due` amounts calculate correctly factoring in late fines.

- **Fee Heads Filter:** Added a multi-select "Fee Heads" dropdown to the analysis filter form. Users can now limit the massive matrix report to only show selected fee types (e.g., only "Admission Fee" and "Transport Fee"). All filters natively sync to the Excel Export function.

## 11. Login Page Logo Optimization

- **Constrained Logo Dimensions:** Fixed a bug where the `x-platform-logo` component and user-uploaded school logos were rendering at their native, giant resolutions on the login page due to conflicting Tailwind `w-auto` / `h-auto` classes.
- **Proportional Scaling:** Replaced fixed utility classes with `max-h-24 max-w-[280px]` and `object-contain` in `guest.blade.php` to ensure any logo fits perfectly above the login box without breaking the layout.
- **Component Fix:** Removed hardcoded SVG width/height from `platform-logo.blade.php` to allow parent layouts to control scaling.

### Files Changed

Action	File
MODIFIED	<code>resources/views/layouts/guest.blade.php</code>
MODIFIED	<code>resources/views/components/platform-logo.blade.php</code>

## 10. Compact Sidebar Width — Professional Panel Layout

- **Sidebar Width Reduced ( `custom.css` ):** Reduced the AdminLTE default sidebar from `250px` to `220px` , gaining `30px` of content area. All selectors scoped to `body:not(.sidebar-collapse)` so the collapsed/mini sidebar is unaffected.
- **Nav Link Density:** Tightened nav link padding ( `1rem` → `0.6rem` ) and reduced font size to `0.875rem` (`14px`) for a professional density matching panels like `aaPanel`.
- **Submenu, Icons, Headers:** Reduced icon width ( `1.6rem` → `1.2rem` ), submenu indent, and section header font size for proportional compaction.
- **Brand Area:** Logo reduced from `33px` → `28px` , brand text from `0.95rem` → `0.85rem` with `max-width: 120px` .
- **Search Box:** Tighter padding and smaller font for the sidebar search input.

## Files Changed

Action	File
MODIFIED	public/css/custom.css

## 9. Apply Discount — Individual Item Scoping

### Bug Fixes

- **Discount Applied to Entire Group Instead of Selected Items ( `ApplyDiscountController.php` ):** The "Apply Discount" button applied the discount to **all** items in a fee group, regardless of which individual fee items the user had checked. Now accepts optional `item_ids[]` from the form. When items are selected, only those items receive the discount; when no items are checked, it falls back to the full-group behavior (backward compatible).
- **Fixed-Amount Distribution Proportional to Selected Items:** When `item_ids` are provided, fixed-amount discounts are distributed proportionally among only the selected items' amounts, not the entire group total.
- **Status Bug ( `pending` → `unpaid` ):** The fallback status after applying a discount was set to `'pending'` which doesn't match the status enum used elsewhere ( `unpaid / partial / paid` ). Fixed to `'unpaid'` .
- **Discount Cap at Outstanding Balance:** Previously capped at `item->amount` ; now correctly caps at `amount - paid_amount` to prevent discount exceeding the remaining payable.

### UI Enhancements

- **Selected Items Info Banner:** The Apply Discount modal now shows a blue info banner indicating how many items are selected and their names, or a note that the discount will apply to all items if none are checked.

## Files Changed

Action	File
MODIFIED	app/Http/Controllers/School/Fees/ApplyDiscountController.php
MODIFIED	resources/views/school/fees/collect-fees/_student-ledger.blade.php

## 8. Fee Collection Discount Validation Fixes

### Bug Fixes

- **100% Discount Rejected ( collectFeeController.php )**: When applying a full discount equal to the fee amount (e.g. ₹5,000 discount on a ₹5,000 fee), the paying amount correctly auto-calculated to ₹0.00, but the backend filter ( `amount > 0` ) silently dropped the item — resulting in a "No payment amount selected" error. Fixed by also including items where `discount > 0` , so full-discount items are properly processed and marked as paid.
- **Discount Exceeds Fee Amount — No JS Restriction ( index.blade.php )**: The discount input field in the Collect Fees modal had no `max` attribute or JavaScript cap, allowing users to enter a discount greater than the fee amount due (e.g. ₹10,000 discount on a ₹5,000 fee). Added `max` attribute on the HTML input, and JS `input` handler that caps the discount value at the fee amount due in real-time.
- **Server-Side Discount Guard ( collectFeeController.php )**: Added a backend safety check that caps `discountAmount` at the outstanding balance before processing, protecting against tampered form submissions that bypass JS validation.

### Files Changed

Action	File
MODIFIED	app/Http/Controllers/School/Fees/CollectFeeController.php
MODIFIED	resources/views/school/fees/collect-fees/index.blade.php

## 7. Fee Reversal Bug Fix — Cross-Group Recalculation

### Bug Fix

- **Stale Group Summaries After Revert ( collectFeeController.php )**: When a single payment receipt covered fees across multiple fee groups (e.g. "Enrollment Fees" + "Uniform & Books"), the `revert()` method only recalculated the one `studentFee` group stored on the deposit record ( `$deposit->studentFee->recalculate()` ). The other groups retained stale `paid_amount` and `due_amount` values, causing the "Total Paid" and "Balance Due" pills on the Collect Fees page to display incorrect totals after reversal. Fixed by collecting all unique `student_fee_id` s from the deposit's items and recalculating every affected group — mirroring the same pattern already used in `store()` .

## Files Changed

Action	File
MODIFIED	app/Http/Controllers/School/Fees/CollectFeeController.php

## 6. Africa's Talking SMS Gateway Integration

- **New Gateway Service:** Added `AfricastalkingGateway.php` — a zero-dependency gateway class using Laravel's `Http` facade to call the Africa's Talking SMS API directly. Auto-detects sandbox vs production endpoint based on username. No additional composer packages required.
- **Admin UI Enhancement:** Updated the SuperAdmin SMS Gateway create and edit Blade views with dedicated Africa's Talking credential fields (Username, API Key, Sender ID/Shortcode) that dynamically show/hide when the gateway name is set to `Africastalking`.
- **Instructions Sidebar:** Added AT-specific setup guidance in the edit form sidebar, including sandbox testing instructions and links to the AT Dashboard.
- **Test Button on Index:** Added a "Test" button alongside "Edit" on the SMS Gateways listing page, allowing direct access to the test SMS page without going through the edit form first.
- **Seeder Migration:** Created `2026_05_28_000001_seed_africastalking_sms_gateway.php` to auto-insert the `Africastalking` gateway record into the `sms_gateways` table with empty credentials (ready for admin configuration).

## Bug Fixes

- **Decrypt Crash Fix ( `SmsGateway.php` ):** Changed the credentials accessor catch block from `DecryptException` to `\Throwable`. Previously, corrupted encrypted data would pass decryption but fail on `unserialize()`, throwing an uncaught `ErrorException` and crashing the edit page with a 500 error. Now gracefully falls back to an empty array.
- **Test Page Crash Fix ( `test.blade.php` ):** Added null-safe handling for `$smsGateway->credentials['key_id']` on the test page. Gateways with empty credentials no longer crash — they show "Not configured" instead.

## Files Changed

Action	File
NEW	app/Services/Gateways/AfricastalkingGateway.php
NEW	database/migrations/2026_05_28_000001_seed_africastalking_sms_gateway.php

Action	File
MODIFIED	app/Models/SmsGateway.php
MODIFIED	resources/views/superadmin/sms-gateways/create.blade.php
MODIFIED	resources/views/superadmin/sms-gateways/edit.blade.php
MODIFIED	resources/views/superadmin/sms-gateways/index.blade.php
MODIFIED	resources/views/superadmin/sms-gateways/test.blade.php

## 5. Custom Full HTML Landing Page System

- **Independent Template Engine:** Built a full HTML template system ( `landing_page_full_templates` table) allowing superadmins to bypass the section builder and use a completely custom HTML template for the main root landing page ( / ).
- **Advanced Code Editor:** Added a CodeMirror-powered editor (with Dracula theme and auto-formatting) under the Website > Full Templates menu. It includes a live responsive iframe preview with Desktop/Tablet/Mobile toggles and Ctrl+S save bindings.
- **Smart Mode Switching:** Integrated a global toggle in Setting ( `landing_page_mode` ) that instantly routes traffic between the dynamic section builder and the active full HTML template.
- **Safe Rendering Engine:** Replaced Laravel's `Blade::render()` with a robust `str_replace` and `preg_replace` implementation in `HomeController@index`. This ensures custom HTML featuring JSON-LD ( `@context` , `@type` ) or CSS ( `@media` , `@keyframes` ) does not trigger fatal Blade parse errors.
- **Quick Import Helper:** Added a one-click import button to automatically seed the database with the pre-built `devx/school-erp-landing.html` layout.

## Authentication Shortcodes for Custom HTML

- **Dynamic Variables:** Injected `{{ $authUrl }}` and `{{ $authText }}` variables into the template renderer, dynamically outputting `/dashboard` and "Dashboard" for authenticated users, and `/login` and "Sign In" for guests.
- **Conditional Blocks:** Implemented regex-based conditional rendering shortcodes. Wrapping HTML in `[if_logged_in]...[/if_logged_in]` or `[if_guest]...[/if_guest]` allows the raw HTML templates to contextually hide or show navigation buttons without needing any complex logic.

# 1. Fee Receipt PDF Layout & Scaling Adjustments

## Landscape Mode (Double Receipt Layout)

- **Table-based Column Constraints:** Converted the float-based and flex-based layout containers into a fixed HTML table ( `table-layout: fixed; width: 100%` ). This addresses the lack of CSS `box-sizing: border-box` support in DomPDF, ensuring that borders and padding do not extend columns beyond 100% of A4 landscape width, resolving right-edge clipping.
- **Logo Consistency:** Removed the duplicate logo on the right side of the Office Copy header to match the clean design of the Parent Copy.
- **Vertical Compaction:** Reduced the inner wrapper paddings ( `16px 20px` to `10px 12px` ), title bar sizing ( `13px` to `11px` ), student info tables ( `11px` to `10px` ), and other vertical spacings to prevent receipts with multiple fee heads from overflowing onto a second page.

## Portrait Mode (Stacked Layout)

- **Adaptive CSS Overrides:** Injected compact CSS overrides dynamically targeting the A4 portrait PDF container. Overrode hardcoded inline wrapper padding ( `padding: 20px 24px` to `8px 12px` ), margins, and default font-sizes ( `12px / 11px` to `9px` ) to guarantee that stacked copies fit cleanly on a single page.

# 2. Centralized Browsershot Binary Configuration

- **Dedicated Config File:** Created `config/browsershot.php` to centrally resolve Node.js, NPM, and Chrome headless binaries.
- **Environment Variable Support:** Appended matching environment variables ( `BROWSERSHOT_NODE_PATH` , `BROWSERSHOT_NPM_PATH` , `BROWSERSHOT_CHROME_PATH` ) to `.env` to enable clean cross-environment setup.
- **Reusable Trait Helper:** Added a shared helper method `configureBrowsershot(Browsershot $browsershot)` to the `HandlesPdfAssets` trait. All PDF controllers now pull configurations from this single function instead of duplicating hardcoded server paths.

# 3. High-Quality Certificate Preview Downloads

- **Orientation-Scoped PDF Options:** Added form-based POST buttons for "Download PDF (Portrait)" and "Download PDF (Landscape)" on the Generated Documents Preview page.
- **Browsershot with DomPDF Fallback:** Integrated high-quality Browsershot rendering into `GenerateDocumentController.php` with automatic fallback to Barryvdh DomPDF if Browsershot is unconfigured or encounters an error.

- **Base64 Asset Embedding:** Enabled the `HandlesPdfAssets` trait in the generator controller to convert all local image files and CSS background assets to base64 data URIs before PDF compilation.
- **Page-Break Control:** Restructured `pdf.blade.php` to dynamically enforce `page-break-after: always` when generating multi-student certificates.

## 4. Marksheet & Report Card PDF Downloads

- **Automatic PDF Engine Fallback:** Implemented Barryvdh DomPDF fallback inside the `download()` method of `MarkSheetController.php` . If Chrome or Node.js are missing, the controller silently falls back to DomPDF, maintaining a seamless download path.
- **Cleaned Preview UI:** Removed the blocking `$hasNode` checks in `show.blade.php` , allowing users on local Windows WAMP/XAMPP environments to always download marksheet PDFs utilizing the fallback engine.
- **Code Simplification:** Updated both the synchronous download controller and the asynchronous background queue job ( `GenerateBulkMarksheetJob.php` ) to use the centralized `configureBrowsershot` method.