

PHASE 1–9 ARCHITECTURE VERIFICATION REPORT

A. VERIFICATION SUMMARY

Verified State: Phase 9 Complete | 62 Tables | 47 Models | 7 Services | 47 Routes

B. WHAT PHASE 8 & 9 ACTUALLY BUILT

The repository contains significantly more than the original specification documented. Here is the verified truth:

Tables Built Beyond Original Spec (Phase 8 & 9 Additions)

Table Phase Status

student_goals	8	✓	EXISTS
goal_progress	8	✓	EXISTS
intervention_plans	8	✓	EXISTS
intervention_actions	8	✓	EXISTS
class_session_joins	9	✓	EXISTS
zoom_recordings	9	✓	EXISTS
zoom_webhooks	9	✓	EXISTS
teacher_invitations	9	✓	EXISTS
policy_acceptances	9	✓	EXISTS
credential_delivery_logs	9	✓	EXISTS
parent_action_items	9	✓	EXISTS
parent_action_item_logs	9	✓	EXISTS
notifications	9	✓	EXISTS
notification_preferences	9	✓	EXISTS
teacher_compensation_settings	9	✓	EXISTS

Services Built

Service Purpose

OpsCalculator Phase 7 — OPS scoring engine

ZoomVideoSDKService Phase 9 — Zoom Video SDK JWT token generation

AutoAttendanceService Phase 9 — Auto-attendance from join events
TeacherInvitationService Phase 9 — Invite-only teacher onboarding
CredentialGeneratorService Phase 9 — Student ID and credential generation
SessionNotificationService Phase 9 — Session-related notifications
TxtConnectService Phase 9 — SMS gateway integration

Critical Observations

teacher_compensation_settings already exists. This is a Phase 11 table that was pre-built in Phase 9. This is correct and expected — it was needed for the payroll-aware architecture.

notifications table exists — Laravel's built-in polymorphic notifications table is migrated and has 16 rows of data already. The system is already sending notifications.

notification_preferences exists — Per-user per-channel preferences are already architected.

Paystack is NOT installed. Composer output shows no Paystack package. This must be added before Phase 10 begins.

Firebase JWT is installed (firebase/php-jwt 7.1.0) — Already used by ZoomVideoSDKService.

DomPDF is installed (barryvdh/laravel-dompdf 3.1.2) — Ready for Phase 10 receipts and Phase 13 certificates.

Excel is installed (maatwebsite/excel 3.1.69) — Ready for Phase 10–15 exports.

C. GAPS BETWEEN SPECIFICATION AND ACTUAL REPOSITORY

Gap 1 — CONFIRMED MISSING: All Phase 10–15 Finance & Payroll Tables

The following tables from the approved specification do not exist:

text

fee_categories MISSING

fee_structures MISSING

student_fee_assignments MISSING

invoices MISSING

invoice_items MISSING

payments MISSING

payment_receipts MISSING

payroll_periods MISSING

payroll_calculations MISSING

payroll_transactions MISSING

payroll_adjustments MISSING

payroll_audit_logs MISSING

Gap 2 — CONFIRMED MISSING: All Phase 12 Communications Tables

text

conversations MISSING

conversation_participants MISSING

messages MISSING

message_attachments MISSING

announcements MISSING

announcement_reads MISSING

email_logs MISSING

sms_logs MISSING

Gap 3 — CONFIRMED MISSING: All Phase 13–15 Tables

text

certificates MISSING

audit_logs MISSING (Spatie activity_log exists but custom audit_logs not built)

audit_log_details MISSING

analytics_snapshots MISSING

fact_student_performance MISSING

fact_attendance MISSING

fact_finance MISSING

student_risk_scores MISSING

student_recommendations MISSING

teacher_performance_metrics MISSING

teacher_reports MISSING

Gap 4 — CONFIRMED MISSING: Supporting Tables Referenced in Spec

text

user_dashboard_preferences MISSING

Gap 5 — CONFIRMED MISSING: Paystack Payment Package

The approved payment provider is Paystack. It is not installed. Required before Phase 10 can process any payment.

Gap 6 — CONFIRMED GAP: Assessment Question Types

The current quiz_questions table supports:

multiple_choice

true_false

short_answer

numeric

The approved business decisions require additional types:

multi_select — MISSING

essay — MISSING

matching — MISSING

fill_in_blank — MISSING

file_upload — MISSING

Gap 7 — CONFIRMED GAP: Assessment Attempt Rules

The current exams table has no attempt control columns. The approved business decisions require:

attempt_limit_type (single / unlimited / fixed / teacher_override)

max_attempts

allow_teacher_override

Gap 8 — CONFIRMED GAP: Assessment Grading Rules

The current grading system uses a fixed letter-grade scale. The approved business decisions require configurable grading schemes:

grading_type (percentage / letter_grade / numeric_score / custom_scale)

custom_scale_config (JSON for custom scales)

Gap 9 — CONFIRMED GAP: Primary Guardian Flag

The parent_student pivot table links parents to students but has no is_primary_guardian column. The approved business decision requires:

One primary guardian per student

Invoices go to primary guardian

Consent policies belong to primary guardian

Payment responsibility belongs to primary guardian

Gap 10 — CONFIRMED GAP: Assessment Feedback Rules

No feedback configuration exists on assessments. Required:

show_correct_answers (boolean)

show_feedback_after (enum: immediately / after_due / after_graded / never)

feedback_visibility (enum: student / parent / both)

D. ARCHITECTURE UPDATES FOR ALL APPROVED BUSINESS DECISIONS

D1 — Parent Self-Registration & Primary Guardian Model

Required database change: parent_student pivot table

SQL

ALTER TABLE parent_student

ADD COLUMN is_primary_guardian BOOLEAN NOT NULL DEFAULT FALSE,

ADD COLUMN relationship VARCHAR(50) NULL,

ADD COLUMN linked_at TIMESTAMP NULL,

ADD COLUMN linked_by BIGINT UNSIGNED NULL;

Business Rule Enforcement:

Only ONE is_primary_guardian = true allowed per student

Invoices always foreign-key to the primary guardian's user_id

policy_acceptances already exists — links to parent registration

D2 — Student Cannot Self-Register / Credential Generation

Already architected correctly:

credential_delivery_logs table exists

CredentialGeneratorService exists

Parent registration routes exist at /register/parent and /register/add-ward

Public registration route does NOT expose student self-registration 

Required addition: Student ID format column needs verification. The spec says RP-2026-XXXXX format — need to confirm this is in users.student_id field (added via alter_users_add_student_fields migration).

D3 — Teacher Invitation-Only Registration

Already architected correctly:

teacher_invitations table exists 

TeacherInvitationService exists 

Routes exist: POST admin/teachers/invite, GET invite/{token}, POST invite/{token}/register 

Spatie Media Library wired for CV uploads 

D4 — Paystack Integration (Card / Mobile Money / Bank Transfer / USSD)

Required: Install Paystack package and build payment tables

Bash

composer require unicodeveloper/laravel-paystack

Required new columns on payments table (when built):

PHP

// payment_method enum must support all Paystack channels

'card'

'mobile_money' // MTN, Vodafone, AirtelTigo (Ghana)

'bank_transfer' // Paystack bank transfer

'ussd' // USSD payment via Paystack

'bank' // Direct bank

// Paystack-specific tracking columns

\$table->string('paystack_reference')->nullable()->unique();

```

$table->string('paystack_access_code')->nullable();
$table->string('paystack_authorization_code')->nullable();
$table->string('paystack_channel')->nullable();
$table->json('paystack_metadata')->nullable();
$table->string('currency', 3)->default('GHS'); // Ghana Cedis
$table->string('gateway_response')->nullable();
$table->timestamp('gateway_paid_at')->nullable();

```

D5 — Assessment Question Types (Extended)

Required: Migration to extend quiz_questions question_type enum

The current question_type enum needs extending:

PHP

// Current

```
['multiple_choice', 'true_false', 'short_answer', 'numeric']
```

// Required after business decisions

```
['multiple_choice', 'multi_select', 'true_false', 'short_answer',
'essay', 'matching', 'fill_in_blank', 'file_upload', 'numeric']
```

New table required: quiz_question_matching_pairs

PHP

```

$table->id();

$table->foreignId('question_id')->constrained('quiz_questions')->cascadeOnDelete();

$table->string('left_item');

$table->string('right_item');

$table->integer('order_index')->default(0);

$table->timestamps();

```

New columns required on quiz_questions:

PHP

```

$table->string('file_upload_accepted_types')->nullable(); // for file_upload type

$table->integer('max_file_size_kb')->nullable(); // for file_upload type

```

```
$table->boolean('is_multi_select')->default(false); // clarifies multi_select behavior
```

```
$table->json('fill_in_blank_answers')->nullable(); // for fill_in_blank type
```

D6 — Assessment Attempt Rules

Required: Migration to add attempt control to exams table

PHP

```
$table->enum('attempt_limit_type', [  
'single', 'unlimited', 'fixed', 'teacher_override'  
)->default('single');
```

```
$table->unsignedTinyInteger('max_attempts')->default(1);
```

```
$table->boolean('allow_teacher_attempt_override')->default(false);
```

New table required: exam_attempts

PHP

```
$table->id();
```

```
$table->foreignId('exam_id')->constrained()->cascadeOnDelete();
```

```
$table->foreignId('student_id')->constrained('users')->cascadeOnDelete();
```

```
$table->unsignedTinyInteger('attempt_number')->default(1);
```

```
$table->enum('status', ['in_progress', 'submitted', 'graded', 'voided'])->  
>default('in_progress');
```

```
$table->timestamp('started_at')->nullable();
```

```
$table->timestamp('submitted_at')->nullable();
```

```
$table->decimal('score', 8, 2)->nullable();
```

```
$table->boolean('is_teacher_override')->default(false);
```

```
$table->foreignId('override_by')->nullable()->constrained('users')->nullOnDelete();
```

```
$table->text('override_reason')->nullable();
```

```
$table->timestamps();
```

// Unique per attempt number

```
$table->unique(['exam_id', 'student_id', 'attempt_number']);
```

D7 — Assessment Grading Rules

Required: Migration to add grading configuration to both exams and assignments tables

PHP

// On both exams and assignments tables

```
$table->enum('grading_type', [  
    'percentage', 'letter_grade', 'numeric_score', 'custom_scale'  
)->default('percentage');  
  
$table->json('custom_scale_config')->nullable();  
  
// Example custom_scale_config:  
// [{"label": "Excellent", "min": 90, "max": 100, "color": "#22C55E"},  
// {"label": "Good", "min": 70, "max": 89, "color": "#F59E0B"},  
// {"label": "Fail", "min": 0, "max": 69, "color": "#DC2626"}]
```

D8 — Assessment Feedback Rules

Required: New columns on exams and assignments tables

PHP

```
$table->boolean('show_correct_answers')->default(false);  
  
$table->enum('show_feedback_after', [  
    'immediately', 'after_due', 'after_graded', 'never'  
)->default('after_graded');  
  
$table->enum('feedback_visibility', [  
    'student', 'parent', 'both'  
)->default('both');
```

D9 — Zoom Video SDK (Already Partially Implemented)

Verified as implemented:

class_session_joins table exists 

zoom_recordings table exists 

zoom_webhooks table exists 

ZoomVideoSDKService exists 

AutoAttendanceService exists ✓

Classroom routes exist ✓

ClassroomController exists ✓

Gaps remaining:

No ZOOM_VIDEO_SDK_KEY and ZOOM_VIDEO_SDK_SECRET documented in .env.example

Webhook signature verification needs audit

Auto-attendance threshold configuration not in school_settings

D10 — Notification Preferences (Already Implemented)

Verified as implemented:

notification_preferences table exists ✓

notifications table exists with 16 rows ✓

NotificationPreference model exists ✓

Notification model exists ✓

Gap: notification_preferences needs channel columns verified. Need to confirm it has in_app, email, sms, push columns.

E. REQUIRED DATABASE CHANGES BEFORE PHASE 10

The following migrations must be created and run before Phase 10 development begins.

MIGRATION SET E1 — Pre-Phase 10 Required Migrations

E1-A: Add is_primary_guardian to parent_student

Bash

```
php artisan make:migration add_primary_guardian_to_parent_student_table
```

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```

{
public function up(): void
{
Schema::table('parent_student', function (Blueprint $table) {
$table->boolean('is_primary_guardian')->default(false)->after('student_id');
$table->string('relationship', 50)->nullable()->after('is_primary_guardian');
$table->timestamp('linked_at')->nullable()->after('relationship');
$table->foreignId('linked_by')->nullable()
->constrained('users')
->onDelete()
->after('linked_at');
});
}

public function down(): void
{
Schema::table('parent_student', function (Blueprint $table) {
$table->dropForeign(['linked_by']);
$table->dropColumn([
'is_primary_guardian',
'relationship',
'linked_at',
'linked_by',
]);
});
}
};

```

E1-B: Extend quiz_questions question_type for all approved types

Bash

php artisan make:migration extend_question_type_enum_in_quiz_questions_table

PHP

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

use Illuminate\Support\Facades\DB;

return new class extends Migration

{

public function up(): void

{

// MySQL: modify enum to add new types

DB::statement(")

ALTER TABLE quiz_questions

MODIFY COLUMN question_type ENUM(

'multiple_choice',

'multi_select',

'true_false',

'short_answer',

'essay',

'matching',

'fill_in_blank',

'file_upload',

'numeric'

) NOT NULL DEFAULT 'multiple_choice'

");

Schema::table('quiz_questions', function (Blueprint \$table) {

\$table->string('file_upload_accepted_types', 255)->nullable()

```

->after('is_active')

->comment('Comma-separated MIME types for file_upload questions');

$table->unsignedInteger('max_file_size_kb')->nullable()

->after('file_upload_accepted_types');

$table->json('fill_in_blank_answers')->nullable()

->after('max_file_size_kb')

->comment('Array of accepted answers for fill_in_blank type');

});

}

public function down(): void
{
    DB::statement("
    ALTER TABLE quiz_questions
    MODIFY COLUMN question_type ENUM(
    'multiple_choice',
    'true_false',
    'short_answer',
    'numeric'
    ) NOT NULL DEFAULT 'multiple_choice'
    ");

    Schema::table('quiz_questions', function (Blueprint $table) {
        $table->dropColumn([
            'file_upload_accepted_types',
            'max_file_size_kb',
            'fill_in_blank_answers',
        ]);
    });
}

```

```
};
```

E1-C: Create quiz_question_matching_pairs table

Bash

```
php artisan make:migration create_quiz_question_matching_pairs_table
```

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
    public function up(): void
```

```
    {
```

```
        Schema::create('quiz_question_matching_pairs', function (Blueprint $table) {
```

```
            $table->id();
```

```
            $table->foreignId('question_id')
```

```
                ->constrained('quiz_questions')
```

```
                ->cascadeOnDelete();
```

```
            $table->text('left_item');
```

```
            $table->text('right_item');
```

```
            $table->unsignedSmallInteger('order_index')->default(0);
```

```
            $table->timestamps();
```

```
            $table->index('question_id');
```

```
        });
```

```
    }
```

```
    public function down(): void
```

```
    {
```

```
        Schema::dropIfExists('quiz_question_matching_pairs');
```

```
}
```

```
};
```

E1-D: Add attempt rules to exams table

Bash

```
php artisan make:migration add_attempt_rules_to_exams_table
```

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
public function up(): void
```

```
{
```

```
Schema::table('exams', function (Blueprint $table) {
```

```
$table->enum('attempt_limit_type', [
```

```
'single', 'unlimited', 'fixed', 'teacher_override'
```

```
])->default('single')->after('duration_minutes');
```

```
$table->unsignedTinyInteger('max_attempts')
```

```
->default(1)
```

```
->after('attempt_limit_type');
```

```
$table->boolean('allow_teacher_attempt_override')
```

```
->default(false)
```

```
->after('max_attempts');
```

```
});
```

```
}
```

```
public function down(): void
```

```
{
```

```

Schema::table('exams', function (Blueprint $table) {
    $table->dropColumn([
        'attempt_limit_type',
        'max_attempts',
        'allow_teacher_attempt_override',
    ]);
});
}
};

```

E1-E: Create exam_attempts table

Bash

```
php artisan make:migration create_exam_attempts_table
```

PHP

```
<?php
```

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('exam_attempts', function (Blueprint $table) {
            $table->id();
            $table->foreignId('exam_id')
                ->constrained()
                ->cascadeOnDelete();
            $table->foreignId('student_id')
                ->constrained('users')

```

```

->cascadeOnDelete();

$table->unsignedTinyInteger('attempt_number')->default(1);

$table->enum('status', [
    'in_progress', 'submitted', 'graded', 'voided'
])->default('in_progress');

$table->timestamp('started_at')->nullable();

$table->timestamp('submitted_at')->nullable();

$table->decimal('score', 8, 2)->nullable();

$table->decimal('percentage', 5, 2)->nullable();

$table->string('grade_letter', 3)->nullable();

$table->boolean('is_passing')->nullable();

$table->boolean('is_teacher_override')->default(false);

$table->foreignId('override_by')
->nullable()
->constrained('users')
->onDelete('null');

$table->text('override_reason')->nullable();

$table->timestamps();

$table->unique(['exam_id', 'student_id', 'attempt_number']);

$table->index(['exam_id', 'student_id']);

$table->index('student_id');

});

}

public function down(): void
{
    Schema::dropIfExists('exam_attempts');
}

};

```

E1-F: Add grading rules to exams table

Bash

```
php artisan make:migration add_grading_rules_to_exams_table
```

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
public function up(): void
```

```
{
```

```
Schema::table('exams', function (Blueprint $table) {
```

```
$table->enum('grading_type', [
```

```
'percentage', 'letter_grade', 'numeric_score', 'custom_scale'
```

```
])->default('percentage')->after('allow_teacher_attempt_override');
```

```
$table->json('custom_scale_config')->nullable()
```

```
->after('grading_type');
```

```
$table->boolean('show_correct_answers')
```

```
->default(false)
```

```
->after('custom_scale_config');
```

```
$table->enum('show_feedback_after', [
```

```
'immediately', 'after_due', 'after_graded', 'never'
```

```
])->default('after_graded')->after('show_correct_answers');
```

```
$table->enum('feedback_visibility', [
```

```
'student', 'parent', 'both'
```

```
])->default('both')->after('show_feedback_after');
```

```
});
```

```

}

public function down(): void
{
    Schema::table('exams', function (Blueprint $table) {
        $table->dropColumn([
            'grading_type',
            'custom_scale_config',
            'show_correct_answers',
            'show_feedback_after',
            'feedback_visibility',
        ]);
    });
}
};

```

E1-G: Add grading rules to assignments table

Bash

```
php artisan make:migration add_grading_rules_to_assignments_table
```

PHP

```
<?php
```

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::table('assignments', function (Blueprint $table) {
            $table->enum('grading_type', [

```

```

'percentage', 'letter_grade', 'numeric_score', 'custom_scale'
])->default('percentage')->after('allow_late_submission');
$table->json('custom_scale_config')->nullable()
->after('grading_type');
$table->boolean('show_correct_answers')
->default(false)
->after('custom_scale_config');
$table->enum('show_feedback_after', [
'immediately', 'after_due', 'after_graded', 'never'
])->default('after_graded')->after('show_correct_answers');
$table->enum('feedback_visibility', [
'student', 'parent', 'both'
])->default('both')->after('show_feedback_after');
});
}

public function down(): void
{
    Schema::table('assignments', function (Blueprint $table) {
        $table->dropColumn([
            'grading_type',
            'custom_scale_config',
            'show_correct_answers',
            'show_feedback_after',
            'feedback_visibility',
        ]);
    });
}
};

```

E1-H: Add auto-attendance threshold to school_settings

Bash

```
php artisan make:migration add_attendance_config_to_school_settings_table
```

PHP

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
public function up(): void
```

```
{
```

```
Schema::table('school_settings', function (Blueprint $table) {
```

```
$table->unsignedSmallInteger('auto_attendance_threshold_minutes')
```

```
->default(10)
```

```
->after('branding_config')
```

```
->comment('Minutes after session start within which joining counts as Present');
```

```
$table->string('paystack_public_key')->nullable()-
```

```
>after('auto_attendance_threshold_minutes');
```

```
$table->string('paystack_secret_key')->nullable()->after('paystack_public_key');
```

```
$table->string('currency_code', 3)->default('GHS')->after('paystack_secret_key');
```

```
$table->string('currency_symbol', 5)->default('¢')->after('currency_code');
```

```
});
```

```
}
```

```
public function down(): void
```

```
{
```

```
Schema::table('school_settings', function (Blueprint $table) {
```

```
$table->dropColumn([
```

```

'auto_attendance_threshold_minutes',
'paystack_public_key',
'paystack_secret_key',
'currency_code',
'currency_symbol',
]);
});
}
};

```

E1-l: Create user_dashboard_preferences table

Bash

```
php artisan make:migration create_user_dashboard_preferences_table
```

PHP

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('user_dashboard_preferences', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')
                ->unique()
                ->constrained()
                ->cascadeOnDelete();
            $table->json('widget_order')->nullable()

```

```

->comment('Ordered array of widget keys');
$table->json('hidden_widgets')->nullable()
->comment('Array of widget keys to hide');
$table->string('theme', 20)->default('light')
->comment('light | dark | system');
$table->string('sidebar_state', 20)->default('expanded')
->comment('expanded | collapsed');
$table->json('extra_preferences')->nullable();
$table->timestamps();
});
}
public function down(): void
{
Schema::dropIfExists('user_dashboard_preferences');
}
};

```

COMMAND TO RUN ALL PRE-PHASE 10 MIGRATIONS

Bash

```
php artisan migrate
```

COMMAND TO INSTALL PAYSTACK

Bash

```
composer require unicodeveloper/laravel-paystack
```

Then publish the config:

Bash

```
php artisan vendor:publish --
provider="Unicodeveloper\Paystack\PaystackServiceProvider"
```

Add to .env:

env

PAYSTACK_PUBLIC_KEY=pk_live_XXXXXXXXXXXX

PAYSTACK_SECRET_KEY=sk_live_XXXXXXXXXXXX

PAYSTACK_PAYMENT_URL=https://api.paystack.co

MERCHANT_EMAIL=payments@rplearnig.com

F. UPDATED UX/UI HANDOVER REPORT FOR MINIMAX

RP LEARNING PORTAL — COMPLETE UX/UI HANDOVER REPORT

Version: 3.0 — Post Phase 9 Verified Architecture

Prepared for: MiniMax UX/UI Design Team

Backend Status: Phase 9 Complete | 62 Tables | 47 Models | 7 Services

SECTION F1 — VERIFIED TECHNICAL STACK (LOCKED — DO NOT CHANGE)

| Layer | Technology | Version |
|-------|------------|---------|
|-------|------------|---------|

| | | |
|---------|---------|---------|
| Backend | Laravel | 11.54.0 |
|---------|---------|---------|

| | | |
|---------|-----|--------|
| Runtime | PHP | 8.4.22 |
|---------|-----|--------|

| | | |
|----------|-------|--------|
| Database | MySQL | 8.0.46 |
|----------|-------|--------|

| | | |
|----------|---------------------|---|
| Frontend | Blade + Bootstrap 5 | — |
|----------|---------------------|---|

| | | |
|---------------|-----------|---|
| Interactivity | Alpine.js | — |
|---------------|-----------|---|

| | | |
|--------|------------|-----------|
| Charts | ApexCharts | Mandatory |
|--------|------------|-----------|

| | | |
|-----|--------|-------|
| PDF | DomPDF | 3.1.2 |
|-----|--------|-------|

| | | |
|---------|-------------------|--------|
| Exports | Maatwebsite Excel | 3.1.69 |
|---------|-------------------|--------|

| | | |
|------|----------------|-------|
| Auth | Laravel Breeze | 2.4.2 |
|------|----------------|-------|

| | | |
|-------------|-------------------|--------|
| Permissions | Spatie Permission | 6.25.0 |
|-------------|-------------------|--------|

| | | |
|-----------------|----------------------|---------|
| File Management | Spatie Media Library | 11.23.0 |
|-----------------|----------------------|---------|

| | | |
|--------------|---------------------|--------|
| Activity Log | Spatie Activity Log | 4.12.3 |
|--------------|---------------------|--------|

| | | |
|-------|----------------|--------------------------|
| Video | Zoom Video SDK | (firebase/php-jwt wired) |
|-------|----------------|--------------------------|

| | | |
|----------|----------|-------------------|
| Payments | Paystack | (to be installed) |
|----------|----------|-------------------|

| | | |
|-----|------------|----------------------------|
| SMS | TxtConnect | (TxtConnectService exists) |
|-----|------------|----------------------------|

| | | |
|-------|-----------------|---|
| Queue | Database driver | — |
|-------|-----------------|---|

| | | |
|-------|-----------------|---|
| Cache | Database driver | — |
|-------|-----------------|---|

Session Database driver —

ABSOLUTE RULES FOR UX TEAM:

No React. No Vue. No Livewire. Server-rendered Blade + Alpine.js only

All strings via __() i18n — zero hardcoded UI text

All charts via ApexCharts — no exceptions

Bootstrap 5 grid and utility classes only

SECTION F2 — VERIFIED COLOR SYSTEM (LOCKED)

CSS

--color-primary: #0B3A66; /* Deep Academic Blue — sidebars, headers */

--color-secondary: #1E5FA8; /* Royal Blue — actions, links */

--color-accent: #D4A017; /* Academic Gold — gamification, CTAs */

--color-background: #F5F7FA; /* Light Grey — page backgrounds */

--color-header-text: #0F172A; /* Slate 900 — H1–H6 */

--color-body-text: #1E293B; /* Slate 800 — body copy */

--color-success: #22C55E;

--color-warning: #F59E0B;

--color-danger: #DC2626;

--color-white: #FFFFFF;

--color-border: #E2E8F0;

--color-muted: #64748B;

Dark Mode: [data-theme="dark"] maps background to #0F172A, text to #F8F AFC

Currency: Ghana Cedis (¢) — GHS

SECTION F3 — VERIFIED ROLE ARCHITECTURE

| Role | Portal Name | Primary Responsibility |
|------|-------------|------------------------|
|------|-------------|------------------------|

| | | |
|-------------|---------------------|--------------------------------------|
| super-admin | Executive Dashboard | Platform oversight, bypass all gates |
|-------------|---------------------|--------------------------------------|

| | | |
|-------|---------------------|------------------------|
| admin | School Admin Portal | Full school operations |
|-------|---------------------|------------------------|

| | | |
|---------|-------------------|-------------------------------|
| teacher | Teacher Workspace | Teaching, grading, attendance |
|---------|-------------------|-------------------------------|

| | | |
|---------|----------------|--------------------------------|
| student | Student Portal | Learning, submitting, tracking |
|---------|----------------|--------------------------------|

parent Parent Command Center Monitor child, pay fees, communicate

Critical UX Rule: Each role has a completely separate portal. A teacher never sees admin screens. A parent never sees teacher management screens.

SECTION F4 — COMPLETE SCREEN INVENTORY (VERIFIED AGAINST PHASE 9)

F4.1 — Public / Registration Screens (NEW — Phase 9)

| Screen ID | Screen Name | Route | Actor | Description |
|-----------|----------------------------|--------------------------|---------|---|
| REG-01 | Registration Landing | /register | Public | Landing page with Parent Registration CTA |
| REG-02 | Parent Registration Form | /register/parent | Public | Parent creates account, accepts T&C |
| REG-03 | Add Ward Form | /register/add-ward | Parent | Parent registers child after own account created |
| REG-04 | Credential Display | (after ward creation) | Parent | Shows student ID + temp password with copy/download/email options |
| INV-01 | Teacher Invitation Landing | /invite/{token} | Teacher | Token-gated page for invited teachers |
| INV-02 | Teacher Profile Completion | /invite/{token}/register | Teacher | Complete profile + CV upload |
| AUTH-01 | Login | /login | All | Email + password |
| AUTH-02 | Forgot Password | /forgot-password | All | Password reset request |
| AUTH-03 | Reset Password | /reset-password/{token} | All | New password entry |
| AUTH-04 | Email Verification | /verify-email | All | Email verification notice |

F4.2 — Admin Screens

| Screen ID | Screen Name | Description |
|-----------|----------------------|-----------------------------|
| ADM-01 | Admin Dashboard | KPIs, alerts, quick actions |
| ADM-02 | Academic Years List | CRUD + activate |
| ADM-03 | Academic Year Detail | Manage terms within year |
| ADM-04 | Subjects List | CRUD subjects |
| ADM-05 | School Classes List | CRUD classes |

| | | |
|--------|-------------------------------|---|
| ADM-06 | Enrollment Manager | Enroll/withdraw students |
| ADM-07 | Teachers List | CRUD teachers + invite |
| ADM-08 | Teacher Detail | Assignments, availability, payroll, compensation settings |
| ADM-09 | Students List | CRUD students |
| ADM-10 | Student Detail | Full profile, OPS, teacher history |
| ADM-11 | Parents List | CRUD parents, link to students |
| ADM-12 | Parent Detail | Linked students, invoices, policy acceptances |
| ADM-13 | Timetable Manager | Create/edit timetable entries |
| ADM-14 | Session Manager | View/manage sessions |
| ADM-15 | Assignments List | All assignments |
| ADM-16 | Exams List | All exams |
| ADM-17 | Resource Library Admin | Manage categories + resources |
| ADM-18 | OPS Dashboard | School-wide performance |
| ADM-19 | Interventions List | All active interventions |
| ADM-20 | Fee Categories | CRUD fee categories |
| ADM-21 | Fee Structures | CRUD fee structures |
| ADM-22 | Invoices List | View/generate/manage invoices |
| ADM-23 | Invoice Detail | Items, payments, receipt |
| ADM-24 | Payments List | All payments |
| ADM-25 | Payroll List | Payroll periods |
| ADM-26 | Payroll Period Detail | Preview, trigger, view results |
| ADM-27 | Announcements | Create/manage announcements |
| ADM-28 | Audit Logs | Full audit trail |
| ADM-29 | School Settings | Branding, OPS weights, Paystack keys, currency |
| ADM-30 | Teacher Compensation Settings | Per-teacher admin fee configuration |
| ADM-31 | Zoom Settings | SDK configuration |

F4.3 — Teacher Screens

| Screen ID | Screen Name | Description |
|-----------|--------------------------|--|
| TCH-01 | Teacher Dashboard | KPIs, timetable, at-risk alerts, earnings |
| TCH-02 | My Timetable | Weekly calendar view |
| TCH-03 | Session List | Upcoming + history |
| TCH-04 | Session Detail | Start, end, log attendance, Zoom SDK integration |
| TCH-05 | Live Classroom | Zoom Video SDK embedded classroom |
| TCH-06 | Attendance Logger | Bulk mark per session |
| TCH-07 | My Students | List with OPS indicators |
| TCH-08 | Student Detail (Teacher) | OPS, attendance, submissions, goals |
| TCH-09 | Create Assignment | Form + question builder |
| TCH-10 | Assignment Detail | Submissions + grading queue |
| TCH-11 | Grade Submission | Score, feedback, file preview |
| TCH-12 | Create Exam | Form + question builder + attempt/grading rules |
| TCH-13 | Exam Detail | Results, grading, finalise |
| TCH-14 | Grade Exam Result | Score, finalise |
| TCH-15 | Question Review Queue | Short answer + essay review + override |
| TCH-16 | Question Builder | MCQ, Multi-Select, T/F, Short Answer, Essay, Matching, Fill-in-Blank, File Upload, Numeric |
| TCH-17 | Resource Library | Upload, manage resources |
| TCH-18 | Student Goals | Create + track goals |
| TCH-19 | Intervention Plans | Create + log actions |
| TCH-20 | Parent Action Items | Create + assign to parents |
| TCH-21 | My Availability | Set weekly availability |
| TCH-22 | My Payroll | Gross → admin fee → net history |
| TCH-23 | My KPI Score | Performance metrics |
| TCH-24 | Messages | Inbox, conversation, compose |
| TCH-25 | Invite Teacher | (Admin only via teacher portal) |
| TCH-26 | My Profile | Edit profile + CV |

F4.4 — Student Screens

| Screen ID | Screen Name | Description |
|-----------|--------------------------|---|
| STU-01 | Student Dashboard | OPS hero ring, streak, leaderboard, subject cards |
| STU-02 | OPS Breakdown | Component scores, weights explained |
| STU-03 | My Timetable | Weekly schedule |
| STU-04 | Session Detail | Join Zoom or see room |
| STU-05 | Live Classroom | Zoom Video SDK embedded (student view) |
| STU-06 | Assignments List | Pending, submitted, graded |
| STU-07 | Assignment Detail | View + submit file or answer questions |
| STU-08 | Exam List | Upcoming, in progress, completed |
| STU-09 | Exam Interface | Question-by-question with timer |
| STU-10 | Exam Result | Score, feedback, grade |
| STU-11 | Resource Library | Browse, download, view |
| STU-12 | My Attendance | Calendar heatmap, rate, by subject |
| STU-13 | My Goals | Goal progress bars |
| STU-14 | Achievements | Badges, tier progress, streak tracker |
| STU-15 | Leaderboard | Class + school ranking |
| STU-16 | Messages | Inbox |
| STU-17 | Notification Center | All notifications with read/unread state |
| STU-18 | Notification Preferences | Per-category channel opt-in/opt-out |

F4.5 — Parent Screens

| Screen ID | Screen Name | Description |
|-----------|-----------------------|---------------------------------|
| PAR-01 | Parent Command Center | All child data + child switcher |
| PAR-02 | Child OPS Detail | OPS breakdown + why this rank |
| PAR-03 | Child Attendance | Attendance record + heatmap |
| PAR-04 | Child Goals | Goal progress bars |
| PAR-05 | Teacher Remarks Feed | Chronological notes |
| PAR-06 | Action Items | List + mark complete |

| | | |
|--------|--------------------------|---|
| PAR-07 | Action Item Detail | Full action item + log history |
| PAR-08 | Invoices List | Outstanding + paid |
| PAR-09 | Invoice Detail | Line items + pay + download |
| PAR-10 | Payment Flow | Paystack payment (card/mobile money/bank transfer/USSD) |
| PAR-11 | Receipt Download | PDF download |
| PAR-12 | Messages | Message teachers |
| PAR-13 | Notification Preferences | Per-category channel opt-in/opt-out |
| PAR-14 | My Profile | Edit parent profile |
| PAR-15 | Policy Acceptances | View accepted policies + dates |
| PAR-16 | Credential Management | View/re-send student credentials |

SECTION F5 — COMPLETE WORKFLOW SPECIFICATIONS (UPDATED FOR PHASE 9)

F5.1 — Parent Self-Registration Workflow

text

[Public Landing: /register]

↓

Parent clicks "Register as Parent"

↓

[/register/parent] — Parent fills:

- Full name
- Email address
- Phone number
- Password
- Confirm password
- Accepts Terms of Service ← policy_acceptances record created
- Accepts Payment Responsibility ← policy_acceptances record created
- IP address + timestamp logged cryptographically

↓

Account created with role: parent

↓

[/register/add-ward] — Parent adds child:

- Child full name
- Date of birth
- Gender
- Class (select from active classes)

↓

CredentialGeneratorService runs:

- Generates student ID: RP-2026-XXXXX
- Generates temporary password
- Creates user with role: student
- Links parent to student via parent_student pivot
- Sets is_primary_guardian = TRUE
- Logs to credential_delivery_logs

↓

[Credential Display Screen]

- Shows: Student ID + Temporary Password
- [Copy Credentials] button
- [Download Credentials PDF] button
- [Email to Student] button

↓

Parent can add additional wards (repeat from /register/add-ward)

F5.2 — Teacher Invitation Workflow (Phase 9 Verified)

text

[Admin: ADM-07 Teachers List]

↓

Admin clicks [Invite Teacher]

↓

Admin enters teacher's email address

→ POST /admin/teachers/invite

→ TeacherInvitationService creates teacher_invitations record

→ Hashed token generated (72-hour expiry)

→ Invitation email sent

↓

Teacher receives email with link: /invite/{token}

↓

[/invite/{token}] — Teacher sees invitation landing

↓

Teacher clicks [Accept Invitation & Complete Profile]

↓

[/invite/{token}/register] — Teacher fills:

- Full name (pre-filled from invitation)

- Password

- Date of birth

- Phone number

- Qualifications

- Bio

- Upload CV (Spatie Media Library)

↓

Account created with role: teacher, status: pending

Token marked as used

↓

Admin receives notification → approves → status → active

F5.3 — Live Classroom Workflow (Zoom Video SDK — Phase 9)

text

[TCH-04: Session Detail]

↓

Teacher clicks [Start Session]

→ class_sessions.status → in_progress

→ class_sessions.started_at → now()

↓

[TCH-05: Live Classroom Screen]

→ ZoomVideoSDKService generates JWT token

→ Browser connects to Zoom Video SDK

→ Classroom renders IN-BROWSER (no external Zoom app)

↓

Students see [Join Session] on their dashboard

→ Students click [Join]

→ POST /classroom/{session}/join

→ class_session_joins record created with joined_at timestamp

→ AutoAttendanceService fires:

IF joined_at <= (started_at + threshold_minutes):

attendance.status = 'present'

ELSE:

attendance.status = 'late'

↓

Teacher can:

- [Start Recording] → POST /classroom/{session}/recording/start

- [Stop Recording] → POST /classroom/{session}/recording/stop

↓

Teacher clicks [End Session]

→ POST /classroom/{session}/end

→ class_sessions.status → completed

→ class_sessions.ended_at → now()

→ Zoom webhook fires back to /webhooks/zoom to confirm

F5.4 — Assessment Workflow (Updated with New Question Types & Attempt Rules)

text

[TCH-12: Create Exam]

↓

Teacher fills:

- Title, description, instructions
- Subject, class, term, session
- Max score, passing score
- Duration (minutes)
- Scheduled at
- Attempt Limit Type: [Single / Unlimited / Fixed / Teacher Override]
- Max Attempts (if Fixed): [number]
- Grading Type: [Percentage / Letter Grade / Numeric / Custom Scale]
- Custom Scale Config (if Custom Scale): [visual builder]
- Show Correct Answers: [toggle]
- Show Feedback After: [Immediately / After Due Date / After Graded / Never]
- Feedback Visible To: [Student / Parent / Both]

↓

[TCH-16: Question Builder — per question type]

|

└─ [Multiple Choice]

| → Question text → 2–6 options → mark ONE correct

|

└─ [Multi-Select]

| → Question text → 2–6 options → mark ONE OR MORE correct

|

└─ [True/False]

| → Question text → True/False auto-created → mark correct

|

└─ [Short Answer]

| → Question text → correct text answer → case sensitive toggle

|

└─ [Essay]

| → Question text → word limit (optional) → manual grading only

|

└─ [Matching]

| → Question text → add left/right pairs

| → quiz_question_matching_pairs table

|

└─ [Fill in the Blank]

| → Question text with [BLANK] markers

| → Acceptable answers array (JSON)

|

└─ [File Upload]

| → Question text

| → Accepted file types (PDF, DOC, JPG, etc.)

| → Max file size

|

└─ [Numeric]

→ Question text → correct number → tolerance

↓

[Publish] → students notified

↓

[STU-09: Exam Interface]

- Student sees attempt counter: "Attempt 1 of 3"
- Timer visible if duration set
- Questions displayed (UX decision: one per screen recommended for mobile)
- Per question: student answers → auto-graded immediately (MCQ/T-F/Numeric/Multi-Select/Fill-in-Blank)
- Essay/Short Answer/File Upload → flagged requires_review

↓

[Submit] → exam_attempts record created

- If attempt_limit_type = 'single' OR attempts exhausted → no more attempts
- If more attempts allowed → [Retry Exam] button shown

↓

[TCH-14: Grade Exam Result]

- Teacher reviews flagged answers
- Teacher finalises → exam_results.is_finalised = true
- OPS recalculated

↓

[STU-10: Exam Result]

- Feedback shown per show_feedback_after setting
- Correct answers shown per show_correct_answers setting
- Visible to parent per feedback_visibility setting

F5.5 — Parent Invoice Payment Workflow (Paystack — Phase 10)

text

[PAR-08: Invoices List]

- Outstanding Fees KPI shows amount + due date
- [Pay Now] button on each invoice

↓

[PAR-09: Invoice Detail]

- Invoice number: INV-2026-NNNN

→ Line item breakdown

→ Total amount in GHS (¢)

→ [Proceed to Payment]

↓

[PAR-10: Payment Flow]

→ Paystack payment modal OR redirect

→ Payment methods:

[Card] — Enter card details inline

[Mobile Money] — Enter MTN/Vodafone/AirtelTigo number

[Bank Transfer] — Paystack virtual account details shown

[USSD] — USSD code displayed per bank

↓

Paystack webhook fires to backend

→ payments record created

→ Invoice status → paid

→ payment_receipts generated as PDF (DomPDF)

↓

[PAR-11: Receipt Download]

→ Download PDF receipt

→ Email receipt sent automatically

SECTION F6 — COMPLETE KPI & DASHBOARD SPECIFICATIONS

F6.1 — Parent Command Center Dashboard

text

TOPBAR:

[Child Switcher Dropdown] → Shows all linked children

Avatar + Name + Class displayed

AJAX state switch — reloads child data without full page refresh

KPI CARDS (2×2 grid):

[Child OPS % + trend arrow] [Attendance % this term]

[Outstanding Fees + due date] [Uncompleted Action Items count]

WIDGETS (vertical stack):

1. Action Items List

→ Title, teacher name, due date

→ Status badge: [Pending] [In Progress] [Completed] [Overdue]

→ [Mark Complete] button

→ [View Detail] link

2. Teacher Remarks Feed

→ Chronological, subject tag, date

3. Goal Progress Bars

→ Each active goal: title + progress bar + target date

4. Invoice Quick-Pay

→ Outstanding invoice card with amount + [Pay Now] button

5. Upcoming Sessions

→ Next 3 sessions for child

6. Credential Management Widget (NEW — Phase 9)

→ Child name + Student ID

→ [Copy Password] [Resend Credentials] [Download PDF]

F6.2 — Student Dashboard

text

HERO SECTION (full width):

[OPS Score Ring — large circular display, color-coded zones]

[Class Rank: #X of Y]

[Streak: 🔥 X days]

[Current Tier: Bronze/Silver/Gold/Platinum/Diamond badge]

SUBJECT CARDS (grid — one per enrolled subject):

Subject name + color

Score % + trend arrow

Last grade

Color: green (>80%) / amber (60–79%) / red (<60%)

WIDGETS:

Upcoming Deadlines (assignments + exams by date)

Recent Badges Earned (last 3)

Points to Next Tier (progress bar)

Leaderboard Position (mini leaderboard)

OPS BREAKDOWN (expandable panel):

"Why is my score X%?"

Component breakdown:

Exams: 30% weight | Your score: X% | Contribution: Y pts

Assignments: 20% weight | Your score: X% | Contribution: Y pts

Attendance: 25% weight | Your score: X% | Contribution: Y pts

Participation: 15% weight | Your score: X% | Contribution: Y pts

Behavior: 10% weight | Your score: X% | Contribution: Y pts

SECTION F7 — GAMIFICATION SPECIFICATIONS (VERIFIED)

| Tier | Min Points | Color | Icon |
|------|------------|-------|------|
|------|------------|-------|------|

| | | | |
|--------|---|---------|---|
| Bronze | 0 | #CD7F32 |  |
|--------|---|---------|---|

| | | | |
|--------|-----|---------|---|
| Silver | 500 | #C0C0C0 |  |
|--------|-----|---------|---|

| | | | |
|------|-------|---------|---|
| Gold | 1,500 | #D4A017 |  |
|------|-------|---------|---|

| | | | |
|----------|-------|---------|---|
| Platinum | 3,000 | #E5E4E2 |  |
|----------|-------|---------|---|

| | | | |
|---------|-------|---------|---|
| Diamond | 6,000 | #B9F2FF |  |
|---------|-------|---------|---|

Streak Rules (Verified in Code):

Increments on consecutive calendar days only

Breaks if gap > 1 calendar day (resets to 1)

Longest streak preserved historically — never decreases

Bonus points at milestones: 7 / 14 / 30 / 60 / 100 days

Bonus formula: $5 \times \text{streak_days}$ points at milestone

Achievement Milestone Keys (Verified):

first_submission

first_exam_pass

streak_7 / streak_14 / streak_30

perfect_attendance

gold_tier

diamond_tier

SECTION F8 — BLADE COMPONENT LIBRARY (REQUIRED)

All views must use x-rp-* prefixed Blade components:

| Component | Purpose |
|-----------|---------|
|-----------|---------|

| | |
|-------------|---------------------|
| <x-rp-card> | Base card container |
|-------------|---------------------|

| | |
|-----------------|--|
| <x-rp-kpi-card> | KPI display with trend arrow + sparkline |
|-----------------|--|

| | |
|----------------------|--|
| <x-rp-badge type=""> | Status tags: success/warning/danger/info |
|----------------------|--|

| | |
|--------------|--|
| <x-rp-table> | Responsive table with sort/filter/pagination |
|--------------|--|

| | |
|-----------------|----------------------------|
| <x-rp-timeline> | Vertical activity timeline |
|-----------------|----------------------------|

| | |
|-----------------|--------------------------|
| <x-rp-calendar> | Calendar with scheduling |
|-----------------|--------------------------|

| | |
|--------------------|---|
| <x-rp-empty-state> | Empty state: illustration + message + CTA |
|--------------------|---|

| | |
|-----------------|-----------------------------|
| <x-rp-skeleton> | CSS loading skeleton blocks |
|-----------------|-----------------------------|

| | |
|-----------------|-------------------------|
| <x-rp-ops-ring> | Circular OPS score ring |
|-----------------|-------------------------|

| | |
|---------------------|---------------------------|
| <x-rp-streak-badge> | Flame icon + streak count |
|---------------------|---------------------------|

| | |
|-------------------|---|
| <x-rp-tier-badge> | Bronze/Silver/Gold/Platinum/Diamond badge |
|-------------------|---|

| | |
|---------------------|---|
| <x-rp-session-card> | Session info: subject, time, mode, status |
|---------------------|---|

| | |
|----------------------------|---------------------------------|
| <x-rp-attendance-calendar> | Monthly heatmap attendance view |
|----------------------------|---------------------------------|

| | |
|------------------|---------------------------------------|
| <x-rp-goal-card> | Goal title, progress bar, target date |
|------------------|---------------------------------------|

| | |
|---------------------|--|
| <x-rp-invoice-card> | Invoice number, amount, status, pay button |
|---------------------|--|

| | |
|-------------------------|-------------------------------|
| <x-rp-question-builder> | Question editor (all 9 types) |
|-------------------------|-------------------------------|

| | |
|--------------------------|---|
| <x-rp-child-switcher> | Parent child switcher dropdown |
| <x-rp-classroom> | Zoom Video SDK embedded classroom container |
| <x-rp-credential-card> | Student credential display with copy/download |
| <x-rp-paystack-button> | Paystack payment trigger button |
| <x-rp-notification-bell> | Topbar notification bell with unread count |

SECTION F9 — NOTIFICATION SYSTEM SPECIFICATIONS

The following notifications are verified to have infrastructure in place (notifications table, notification_preferences table, NotificationPreference model).

Notification Categories & Channels

| Category | In-App | Email | SMS | Push | |
|----------------------|--------|-------|-----|------|---|
| Academic Updates | ✓ | ✓ | ✗ | ✓ | |
| Financial / Invoices | ✓ | ✓ | ✓ | ✓ | |
| Attendance Warnings | | ✓ | ✗ | ✓ | ✓ |
| Payroll / Earnings | ✓ | ✓ | ✗ | ✗ | |
| System Announcements | | ✓ | ✓ | ✗ | ✓ |

Required Notification Classes (Phase 10–15)

Notification Trigger Recipients

| | | |
|---------------------|-----------------------------|---------------------------|
| InvoiceGenerated | Invoice created | Parent (primary guardian) |
| InvoicePaid | Payment confirmed | Parent + Admin |
| InvoiceOverdue | Due date passed | Parent (primary guardian) |
| PaymentReceiptReady | Receipt generated | Parent |
| PayrollProcessed | Payroll finalised | Teacher |
| AssignmentPublished | Assignment published | Students in class |
| AssignmentGraded | Submission graded | Student + Parent |
| ExamScheduled | Exam published | Students in class |
| ExamResultFinalised | Exam finalised | Student + Parent |
| AttendanceWarning | Attendance < 80% | Parent (primary guardian) |
| ActionItemAssigned | Teacher creates action item | Parent |

| | | |
|-----------------------|------------------------------|--------------------|
| ActionItemOverdue | Action item past due date | Parent |
| CertificateIssued | Certificate generated | Student + Parent |
| StudentGoalAchieved | Goal status = achieved | Student + Parent |
| MilestoneEarned | Achievement milestone earned | Student |
| SessionStarting | 15 minutes before session | Students + Parents |
| TeacherInvitationSent | Invitation created | Teacher (email) |

SECTION F10 — COMPLETE FORM INVENTORY (UPDATED)

F10.1 — Registration Forms (NEW — Phase 9)

Parent Registration Form (REG-02)

text

Fields:

Full Name (text, required)

Email Address (email, required, unique)

Phone Number (tel, required)

Password (password, min 8)

Confirm Password (password, match)

☐ I accept the Terms of Service (required)

☐ I accept Payment Responsibility (required)

Validation:

Email must be unique

Password min 8 characters

Both checkboxes required to proceed

On Submit:

policy_acceptances records created (2 records)

IP address logged

User created with role: parent, status: active

Add Ward Form (REG-03)

text

Fields:

Child's Full Name (text, required)

Date of Birth (date, required)

Gender (select: male/female/other)

Class to Enroll (select from active school_classes)

On Submit:

CredentialGeneratorService generates RP-YYYY-XXXXX

Temporary password generated

Student user created, role: student

parent_student pivot: is_primary_guardian = TRUE

credential_delivery_logs record created

Redirect to credential display screen

Credential Display (REG-04)

text

Display:

Student Name

Student ID: RP-2026-XXXXX (large, copyable)

Temporary Password: xxxxxxxx (large, copyable with eye toggle)

Actions:

[ Copy Credentials] — copies both to clipboard

[ Download PDF] — generates credential PDF via DomPDF

[ Email to Student] — sends credentials to entered email

[+ Add Another Ward] — returns to REG-03

[Go to Dashboard] — enters parent portal

Warning banner:

"Please save these credentials. The temporary password will expire on first login."

F10.2 — Assessment Forms (UPDATED — New Question Types & Rules)

Create/Edit Exam Form (TCH-12) — Updated Fields

text

New fields added to exam creation:

Attempt Rules Section:

Attempt Limit Type (radio):

- ☐ Single Attempt (default)
- ☐ Unlimited Attempts
- ☐ Fixed Number: [____] attempts
- ☐ Teacher Override (teacher can reset per student)

Allow Teacher Override (toggle) — shown for all types

Grading Section:

Grading Type (select):

- ☐ Percentage (default)
- ☐ Letter Grade (A+/A/B+/B/C+/C/D+/D/F)
- ☐ Numeric Score (raw number)
- ☐ Custom Scale (opens scale builder)

Custom Scale Builder (shown if Custom Scale selected):

Add Tier: Label | Min % | Max % | Color

[+ Add Tier]

Feedback Section:

Show Correct Answers (toggle)

Show Feedback After (select):

- ☐ Immediately after submission
- ☐ After due date passes
- ☐ After teacher grades (default)
- ☐ Never

Feedback Visible To (select):

- ☐ Student only (default)
- ☐ Parent only

☐ Both student and parent

Question Builder — All 9 Types (TCH-16)

text

TYPE SELECTOR (tabs or dropdown):

[Multiple Choice]

Question Text (rich textarea)

Points (number)

Options (add/remove rows):

☐ Option text | [Mark Correct] | [Delete]

☐ Option text | [Mark Correct] | [Delete]

[+ Add Option]

Explanation (optional textarea)

[Multi-Select]

Question Text (rich textarea)

Points (number)

Options (add/remove rows):

☐ Option text | [Delete]

☐ Option text | [Delete]

[+ Add Option]

Minimum correct selections (number)

Partial credit (toggle)

Explanation (optional textarea)

[True/False]

Question Text (rich textarea)

Points (number)

Correct Answer (radio): ☐ True ☐ False

Explanation (optional textarea)

[Short Answer]

Question Text (rich textarea)

Points (number)

Correct Answer (text input)

Case Sensitive (toggle)

Explanation (optional textarea)

[Essay]

Question Text (rich textarea)

Points (number)

Word Limit (number, optional)

Marking Criteria (textarea, optional — shown to teacher when grading)

NOTE: Always manual grading — no auto-grade

[Matching]

Question Text (rich textarea)

Points (number)

Match Pairs (add/remove rows):

Left Item [text input] → Right Item [text input] | [Delete]

Left Item [text input] → Right Item [text input] | [Delete]

[+ Add Pair]

NOTE: Right items are shuffled on student interface

[Fill in the Blank]

Question Text (use [BLANK] marker where blanks go)

Preview (live render of question with blank boxes)

Points (number)

Acceptable Answers (add multiple accepted answers per blank):

Blank 1: [answer] [+ Add Alternative]

Blank 2: [answer] [+ Add Alternative]

Case Sensitive (toggle)

[File Upload]

Question Text / Instructions (rich textarea)

Points (number)

Accepted File Types (checkboxes):

☐ PDF ☐ DOC/DOCX ☐ JPG/PNG ☐ ZIP ☐ Other

Max File Size: [____] MB

NOTE: Always manual grading — teacher downloads and reviews

[Numeric]

Question Text (rich textarea)

Points (number)

Correct Answer (number input)

Tolerance (number — ± acceptable range)

Explanation (optional textarea)

F10.3 — Payment Form (PAR-10 — Paystack)

text

Invoice Summary:

Invoice Number: INV-2026-NNNN

Student: [Name]

Items: [fee breakdown list]

Total: ₦[amount]

Payment Method (tabs):

[Card]

Paystack inline popup OR redirect

Card number, expiry, CVV

[Mobile Money]

Provider (select): MTN / Vodafone Cash / AirtelTigo

Mobile number (tel input)

[Send Payment Request] → customer confirms on phone

[Bank Transfer]

Paystack generates virtual account

Bank: [Bank Name]

Account Number: [generated number]

Amount: ₦[exact amount]

[I have made the transfer] → polling starts

[USSD]

Select Bank (dropdown)

USSD Code displayed: *737*[amount]*[reference]#

Instructions shown per bank

[I have completed the USSD payment]

On Success:

Success screen: "Payment Confirmed  "

[Download Receipt PDF]

[Return to Dashboard]

SECTION F11 — ACCESSIBILITY & PERFORMANCE

WCAG 2.1 AA Requirements:

Minimum contrast 4.5:1 normal text, 3:1 large text

Visible focus rings: 2px solid #1E5FA8

All images have alt text; icons have aria-labels

Every form input has associated <label>

Error messages via aria-describedby

Skip navigation link at page top

All functionality keyboard accessible

Touch targets minimum 44×44px on mobile

Performance Targets:

Mobile LCP < 2.5s, initial load < 3s

Desktop load < 2s

Chart Standards:

Chart Type Use Case

Area (soft curve) OPS multi-term trend

Heatmap Attendance calendar

Dual Line Financial: realized vs projected

Stacked Bar Enrollment by class level

Radial Gauge Teacher KPI score

Waterfall Payroll: Gross → Admin Fee → Net

Donut Score component breakdowns

Funnel Payment conversion (admin)

G. PHASE 10–15 ARCHITECTURE & IMPLEMENTATION PLAN

PHASE 10 — FINANCIAL MANAGEMENT

Scope: Fee structures → Invoice generation → Paystack payments → PDF receipts

Phase 10 Database Architecture

SQL

-- 7 new tables required

fee_categories

fee_structures

student_fee_assignments

invoices

invoice_items

payments

payment_receipts

Phase 10 Migration — fee_categories

Bash

php artisan make:migration create_fee_categories_table

PHP

```
Schema::create('fee_categories', function (Blueprint $table) {
```

```
    $table->id();
```

```

$table->string('name', 100)->unique();
$table->text('description')->nullable();
$table->string('color', 7)->default('#0B3A66');
$table->boolean('is_active')->default(true);
$table->unsignedSmallInteger('sort_order')->default(0);
$table->timestamps();
$table->softDeletes();
});

```

Phase 10 Migration — fee_structures

PHP

```

Schema::create('fee_structures', function (Blueprint $table) {
    $table->id();
    $table->foreignId('fee_category_id')
        ->constrained()
        ->restrictOnDelete();
    $table->foreignId('academic_year_id')
        ->nullable()
        ->constrained()
        ->nullOnDelete();
    $table->string('name', 150);
    $table->decimal('amount', 10, 2);
    $table->string('currency', 3)->default('GHS');
    $table->enum('frequency', [
        'once', 'termly', 'monthly', 'annually'
    ]->default('termly'));
    $table->date('due_date')->nullable();
    $table->text('description')->nullable();
    $table->boolean('is_active')->default(true);

```

```
$table->timestamps();  
$table->softDeletes();  
$table->index(['fee_category_id', 'is_active']);  
$table->index('academic_year_id');  
});
```

Phase 10 Migration — student_fee_assignments

PHP

```
Schema::create('student_fee_assignments', function (Blueprint $table) {  
    $table->id();  
    $table->foreignId('student_id')  
        ->constrained('users')  
        ->cascadeOnDelete();  
    $table->foreignId('fee_structure_id')  
        ->constrained()  
        ->restrictOnDelete();  
    $table->foreignId('academic_year_id')  
        ->constrained()  
        ->restrictOnDelete();  
    $table->enum('status', [  
        'active', 'waived', 'cancelled'  
    ]->default('active'));  
    $table->text('waiver_reason')->nullable();  
    $table->foreignId('assigned_by')  
        ->constrained('users')  
        ->restrictOnDelete();  
    $table->timestamps();  
    $table->softDeletes();  
    $table->unique(['student_id', 'fee_structure_id', 'academic_year_id']);
```

```
$table->index('student_id');  
$table->index('fee_structure_id');  
});
```

Phase 10 Migration — invoices

PHP

```
Schema::create('invoices', function (Blueprint $table) {  
    $table->id();  
    $table->foreignId('student_id')  
        ->constrained('users')  
        ->restrictOnDelete();  
    $table->foreignId('parent_id') // Always primary guardian  
        ->constrained('users')  
        ->restrictOnDelete();  
    $table->foreignId('academic_year_id')  
        ->constrained()  
        ->restrictOnDelete();  
    $table->string('invoice_number', 20)->unique(); // INV-2026-NNNN  
    $table->decimal('subtotal', 10, 2)->default(0);  
    $table->decimal('discount_amount', 10, 2)->default(0);  
    $table->decimal('total_amount', 10, 2);  
    $table->string('currency', 3)->default('GHS');  
    $table->enum('status', [  
        'draft', 'sent', 'partially_paid', 'paid', 'overdue', 'cancelled', 'refunded'  
    ])->default('draft');  
    $table->date('due_date');  
    $table->text('notes')->nullable();  
    $table->timestamp('sent_at')->nullable();  
    $table->timestamp('paid_at')->nullable();  
});
```

```

$table->foreignId('created_by')
->constrained('users')
->restrictOnDelete();
$table->json('metadata')->nullable();
$table->timestamps();
$table->softDeletes();
$table->index(['student_id', 'status']);
$table->index(['parent_id', 'status']);
$table->index('status');
$table->index('due_date');
});

```

Phase 10 Migration — invoice_items

PHP

```

Schema::create('invoice_items', function (Blueprint $table) {
    $table->id();
    $table->foreignId('invoice_id')
->constrained()
->cascadeOnDelete();
    $table->foreignId('fee_category_id')
->nullable()
->constrained()
->nullOnDelete();
    $table->string('description', 255);
    $table->decimal('unit_amount', 10, 2);
    $table->unsignedSmallInteger('quantity')->default(1);
    $table->decimal('discount_amount', 10, 2)->default(0);
    $table->decimal('line_total', 10, 2);
    $table->unsignedSmallInteger('sort_order')->default(0);

```

```
$table->timestamps();
```

```
$table->index('invoice_id');
```

```
});
```

Phase 10 Migration — payments

PHP

```
Schema::create('payments', function (Blueprint $table) {
```

```
$table->id();
```

```
$table->foreignId('invoice_id')
```

```
->constrained()
```

```
->restrictOnDelete();
```

```
$table->foreignId('parent_id')
```

```
->constrained('users')
```

```
->restrictOnDelete();
```

```
$table->decimal('amount', 10, 2);
```

```
$table->string('currency', 3)->default('GHS');
```

```
$table->enum('payment_method', [
```

```
'card', 'mobile_money', 'bank_transfer', 'ussd', 'bank', 'cash', 'cheque', 'waiver'
```

```
]);
```

```
$table->enum('status', [
```

```
'pending', 'processing', 'successful', 'failed', 'cancelled', 'refunded'
```

```
])->default('pending');
```

```
// Paystack fields
```

```
$table->string('paystack_reference', 100)->nullable()->unique();
```

```
$table->string('paystack_access_code', 100)->nullable();
```

```
$table->string('paystack_authorization_code', 100)->nullable();
```

```
$table->string('paystack_channel', 50)->nullable();
```

```
$table->string('gateway_response', 255)->nullable();
```

```
$table->timestamp('gateway_paid_at')->nullable();
```

```

$table->json('paystack_metadata')->nullable();

// Internal reference

$table->string('reference_number', 100)->nullable();

$table->text('notes')->nullable();

$table->foreignId('recorded_by')

->nullable()

->constrained('users')

->onDelete();

$table->timestamp('paid_at')->nullable();

$table->timestamps();

$table->softDeletes();

$table->index(['invoice_id', 'status']);

$table->index('parent_id');

$table->index('paystack_reference');

});

```

Phase 10 Migration — payment_receipts

PHP

```

Schema::create('payment_receipts', function (Blueprint $table) {

    $table->id();

    $table->foreignId('payment_id')

    ->constrained()

    ->restrictOnDelete();

    $table->string('receipt_number', 20)->unique(); // REC-2026-NNNN

    $table->string('pdf_path', 500)->nullable();

    $table->boolean('emailed')->default(false);

    $table->timestamp('emailed_at')->nullable();

    $table->timestamp('generated_at')->useCurrent();

    $table->timestamps();

```

```
$table->index('payment_id');  
});
```

Phase 10 Services

Bash

```
php artisan make:service InvoiceGeneratorService
```

```
php artisan make:service PaystackService
```

```
php artisan make:service ReceiptGeneratorService
```

InvoiceGeneratorService — Key Methods:

PHP

```
generateInvoiceNumber(): string // INV-2026-NNNN
```

```
generateForStudent(User $student, AcademicYear $year): Invoice
```

```
addItemFromFeeAssignments(Invoice $invoice): void
```

```
sendInvoice(Invoice $invoice): void
```

```
markOverdueInvoices(): int // Scheduled job
```

```
getPrimaryGuardian(User $student): User // From parent_student pivot
```

PaystackService — Key Methods:

PHP

```
initializePayment(Invoice $invoice, string $channel): array
```

```
verifyPayment(string $reference): array
```

```
handleWebhook(array $payload): void
```

```
getAuthorizationUrl(array $data): string
```

ReceiptGeneratorService — Key Methods:

PHP

```
generate(Payment $payment): PaymentReceipt
```

```
generateReceiptNumber(): string // REC-2026-NNNN
```

```
generatePDF(PaymentReceipt $receipt): string
```

```
emailReceipt(PaymentReceipt $receipt): void
```

Phase 10 Jobs

Bash

php artisan make:job MarkOverdueInvoicesJob

php artisan make:job SendInvoiceReminderJob

php artisan make:job GenerateReceiptJob

php artisan make:job ProcessPaystackWebhookJob

Phase 10 Events & Listeners

Bash

php artisan make:event InvoiceCreated

php artisan make:event InvoiceSent

php artisan make:event PaymentSuccessful

php artisan make:event PaymentFailed

php artisan make:event InvoiceOverdue

php artisan make:listener SendInvoiceNotification

php artisan make:listener GeneratePaymentReceipt

php artisan make:listener SendPaymentConfirmation

php artisan make:listener NotifyParentOfOverdue

Phase 10 Models

Bash

php artisan make:model FeeCategory

php artisan make:model FeeStructure

php artisan make:model StudentFeeAssignment

php artisan make:model Invoice

php artisan make:model InvoiceItem

php artisan make:model Payment

php artisan make:model PaymentReceipt

Phase 10 Policies

Bash

php artisan make:policy InvoicePolicy --model=Invoice

```
php artisan make:policy PaymentPolicy --model=Payment
```

```
php artisan make:policy FeeStructurePolicy --model=FeeStructure
```

InvoicePolicy rules:

PHP

viewAny(): admin, super-admin

view(): admin | parent (own child's invoices only via primary guardian check)

create(): admin

update(): admin (only draft status)

delete(): admin (only draft status)

pay(): parent (own invoices only)

download(): admin | parent (own)

Phase 10 Controllers

Bash

```
php artisan make:controller Finance/FeeCategoryController --resource
```

```
php artisan make:controller Finance/FeeStructureController --resource
```

```
php artisan make:controller Finance/InvoiceController --resource
```

```
php artisan make:controller Finance/PaymentController
```

```
php artisan make:controller Finance/PaystackController
```

```
php artisan make:controller Finance/ReceiptController
```

```
php artisan make:controller Parent/InvoiceController
```

Phase 10 Routes

PHP

```
// Admin Finance Routes
```

```
Route::prefix('admin/finance')->middleware(['auth', 'verified', 'role:admin|super-admin'])->group(function () {
```

```
Route::resource('fee-categories', Finance\FeeCategoryController::class);
```

```
Route::resource('fee-structures', Finance\FeeStructureController::class);
```

```
Route::resource('invoices', Finance\InvoiceController::class);
```

```

Route::post('invoices/{invoice}/send', [Finance\InvoiceController::class, 'send'])
->name('admin.invoices.send');

Route::post('invoices/{invoice}/cancel', [Finance\InvoiceController::class, 'cancel'])
->name('admin.invoices.cancel');

Route::get('invoices/{invoice}/receipt', [Finance\ReceiptController::class, 'download'])
->name('admin.invoices.receipt');

Route::get('payments', [Finance\PaymentController::class, 'index'])
->name('admin.payments.index');

Route::post('payments/record', [Finance\PaymentController::class, 'recordManual'])
->name('admin.payments.record');

});

```

// Parent Payment Routes

```

Route::prefix('parent/finance')->middleware(['auth', 'verified', 'role:parent'])-
>group(function () {

Route::get('invoices', [Parent\InvoiceController::class, 'index'])
->name('parent.invoices.index');

Route::get('invoices/{invoice}', [Parent\InvoiceController::class, 'show'])
->name('parent.invoices.show');

Route::post('invoices/{invoice}/pay', [Finance\PaystackController::class, 'initialize'])
->name('parent.invoices.pay');

Route::get('invoices/{invoice}/receipt', [Finance\ReceiptController::class,
'downloadParent'])
->name('parent.invoices.receipt');

});

```

// Paystack Webhooks (no auth — webhook signature verification instead)

```

Route::post('webhooks/paystack', [Finance\PaystackController::class, 'webhook'])
->name('webhooks.paystack');

Route::get('payment/callback', [Finance\PaystackController::class, 'callback'])

```

```
->name('payment.callback');
```

Phase 10 Notifications

Bash

```
php artisan make:notification InvoiceGeneratedNotification
```

```
php artisan make:notification InvoicePaidNotification
```

```
php artisan make:notification InvoiceOverdueNotification
```

```
php artisan make:notification PaymentReceiptNotification
```

Phase 10 Views

text

```
resources/views/admin/finance/
```

```
fee-categories/
```

```
index.blade.php
```

```
create.blade.php
```

```
edit.blade.php
```

```
fee-structures/
```

```
index.blade.php
```

```
create.blade.php
```

```
edit.blade.php
```

```
invoices/
```

```
index.blade.php
```

```
create.blade.php
```

```
show.blade.php
```

```
payments/
```

```
index.blade.php
```

```
record.blade.php
```

```
resources/views/parent/finance/
```

```
invoices/
```

```
index.blade.php
```

show.blade.php

pay.blade.php

receipts/

download.blade.php

resources/views/pdf/

invoice.blade.php

receipt.blade.php

Phase 10 Scheduled Jobs

PHP

// In routes/console.php or app/Console/Kernel.php

Schedule::job(new MarkOverdueInvoicesJob)->daily()->at('00:05');

Schedule::job(new SendInvoiceReminderJob)->daily()->at('09:00');

PHASE 11 — PAYROLL & COMPENSATION

Scope: Teacher payroll periods → Gross calculation → Admin fee deduction → Net earnings → Audit trail

Critical Business Rule (Immutable): Admin fee percentage is captured AT TIME OF PAYROLL EXECUTION and stored in payroll_calculations.admin_fee_percentage_applied. Historical payrolls ALWAYS show the rate that was applied, never the current rate.

Phase 11 Migrations

Bash

php artisan make:migration create_payroll_periods_table

php artisan make:migration create_payroll_calculations_table

php artisan make:migration create_payroll_transactions_table

php artisan make:migration create_payroll_adjustments_table

php artisan make:migration create_payroll_audit_logs_table

payroll_periods:

PHP

Schema::create('payroll_periods', function (Blueprint \$table) {

```

$table->id();

$table->string('name', 100); // "June 2026"

$table->date('start_date');

$table->date('end_date');

$table->enum('status', [
    'draft', 'preview', 'processing', 'completed', 'cancelled'
])->default('draft');

$table->foreignId('created_by')
    ->constrained('users')
    ->restrictOnDelete();

$table->foreignId('processed_by')
    ->nullable()
    ->constrained('users')
    ->nullOnDelete();

$table->timestamp('processed_at')->nullable();

$table->text('notes')->nullable();

$table->timestamps();

$table->softDeletes();

$table->index('status');

$table->index(['start_date', 'end_date']);

});

payroll_calculations:

PHP

Schema::create('payroll_calculations', function (Blueprint $table) {

    $table->id();

    $table->foreignId('payroll_period_id')
        ->constrained()
        ->restrictOnDelete();

```

```

$table->foreignId('teacher_id')
->constrained('users')
->restrictOnDelete();

$table->unsignedSmallInteger('session_count')->default(0);

$table->unsignedSmallInteger('private_session_count')->default(0);

$table->decimal('session_gross', 10, 2)->default(0);

$table->decimal('private_session_gross', 10, 2)->default(0);

$table->decimal('gross_earnings', 10, 2);

$table->boolean('admin_fee_enabled')->default(true);

// CRITICAL: Captured at execution time — never changes

$table->decimal('admin_fee_percentage_applied', 5, 2)->default(0);

$table->decimal('admin_fee_amount', 10, 2)->default(0);

$table->decimal('adjustments_total', 10, 2)->default(0);

$table->decimal('net_earnings', 10, 2);

$table->string('currency', 3)->default('GHS');

$table->enum('status', [
'draft', 'approved', 'paid', 'disputed'
])->default('draft');

$table->timestamp('calculated_at')->useCurrent();

$table->json('session_breakdown')->nullable(); // JSON array of session details

$table->timestamps();

$table->unique(['payroll_period_id', 'teacher_id']);

$table->index(['teacher_id', 'status']);

});

payroll_transactions:

PHP

Schema::create('payroll_transactions', function (Blueprint $table) {
$table->id();

```

```

$table->foreignId('payroll_calculation_id')
->constrained()
->restrictOnDelete();
$table->enum('type', ['payment', 'adjustment', 'reversal'])->default('payment');
$table->decimal('amount', 10, 2);
$table->string('reference', 100)->nullable();
$table->string('payment_method', 50)->nullable();
$table->text('notes')->nullable();
$table->foreignId('processed_by')
->constrained('users')
->restrictOnDelete();
$table->timestamp('processed_at')->useCurrent();
$table->timestamps();
$table->index('payroll_calculation_id');
});

```

payroll_adjustments:

PHP

```

Schema::create('payroll_adjustments', function (Blueprint $table) {
    $table->id();
    $table->foreignId('payroll_calculation_id')
->constrained()
->restrictOnDelete();
    $table->enum('type', ['addition', 'deduction'])->default('addition');
    $table->string('reason', 255);
    $table->decimal('amount', 10, 2);
    $table->foreignId('created_by')
->constrained('users')
->restrictOnDelete();

```

```

$table->timestamps();

$table->index('payroll_calculation_id');

});

payroll_audit_logs:

PHP

Schema::create('payroll_audit_logs', function (Blueprint $table) {

    $table->id();

    $table->foreignId('teacher_id')

    ->constrained('users')

    ->cascadeOnDelete();

    $table->string('field_changed', 100);

    $table->text('before_value')->nullable();

    $table->text('after_value')->nullable();

    $table->string('reason', 255)->nullable();

    $table->foreignId('changed_by')

    ->constrained('users')

    ->restrictOnDelete();

    $table->timestamp('changed_at')->useCurrent();

    $table->index(['teacher_id', 'changed_at']);

});

```

Phase 11 Services

Bash

```
php artisan make:service PayrollCalculatorService
```

PayrollCalculatorService — Key Methods:

PHP

```
preview(PayrollPeriod $period): Collection // Preview all teachers
```

```
calculate(Teacher $teacher, PayrollPeriod $period): PayrollCalculation
```

```
calculateGrossFromSessions(Teacher $teacher, PayrollPeriod $period): array
```

applyAdminFee(PayrollCalculation \$calc): void // Reads CURRENT fee, stores it

processPayroll(PayrollPeriod \$period): void // Trigger all calculations

generatePayslip(PayrollCalculation \$calc): string // PDF via DomPDF

Admin Fee Audit — Critical Implementation:

PHP

```
public function updateAdminFeePercentage(
    TeacherCompensationSetting $setting,
    float $newPercentage,
    User $changedBy
): void {
    // ALWAYS log the change before updating
    PayrollAuditLog::create([
        'teacher_id' => $setting->teacher_id,
        'field_changed' => 'admin_fee_percentage',
        'before_value' => (string) $setting->admin_fee_percentage,
        'after_value' => (string) $newPercentage,
        'reason' => 'Admin fee percentage updated',
        'changed_by' => $changedBy->id,
        'changed_at' => now(),
    ]);
    $setting->update(['admin_fee_percentage' => $newPercentage]);
}
```

PHASE 12 — COMMUNICATIONS

Scope: 2-way messaging → Announcements → Email logs → SMS logs → Notification center

Phase 12 Key Migrations

Bash

```
php artisan make:migration create_conversations_table
```

```
php artisan make:migration create_conversation_participants_table
```

```
php artisan make:migration create_messages_table
```

```
php artisan make:migration create_message_attachments_table
```

```
php artisan make:migration create_announcements_table
```

```
php artisan make:migration create_announcement_reads_table
```

```
php artisan make:migration create_email_logs_table
```

```
php artisan make:migration create_sms_logs_table
```

conversations:

PHP

```
Schema::create('conversations', function (Blueprint $table) {  
    $table->id();  
    $table->enum('type', ['direct', 'group'])->default('direct');  
    $table->string('title', 150)->nullable(); // For group conversations  
    $table->foreignId('created_by')  
        ->constrained('users')  
        ->restrictOnDelete();  
    $table->timestamp('last_message_at')->nullable();  
    $table->timestamps();  
    $table->softDeletes();  
    $table->index('last_message_at');  
    $table->index('created_by');  
});
```

messages:

PHP

```
Schema::create('messages', function (Blueprint $table) {  
    $table->id();  
    $table->foreignId('conversation_id')  
        ->constrained()
```

```

->cascadeOnDelete();

$table->foreignId('sender_id')
->constrained('users')
->restrictOnDelete();

$table->longText('body');

$table->boolean('is_admin_visible')->default(true);

$table->timestamp('sent_at')->useCurrent();

$table->timestamp('read_at')->nullable();

$table->timestamps();

$table->softDeletes();

$table->index(['conversation_id', 'sent_at']);

$table->index('sender_id');

});

```

announcements:

PHP

```

Schema::create('announcements', function (Blueprint $table) {

    $table->id();

    $table->string('title', 255);

    $table->longText('body');

    $table->enum('target_role', [
        'all', 'admin', 'teacher', 'student', 'parent'
    ]->default('all'));

    $table->boolean('is_urgent')->default(false); // Shows in system banner

    $table->boolean('is_published')->default(false);

    $table->foreignId('created_by')
->constrained('users')
->restrictOnDelete();

    $table->timestamp('published_at')->nullable();

```

```

$table->timestamp('expires_at')->nullable();

$table->timestamps();

$table->softDeletes();

$table->index(['target_role', 'is_published']);

$table->index('published_at');

});

```

Phase 12 Notification Classes

All approved notification types built and wired to notification_preferences table for per-user per-channel control.

PHASE 13 — CERTIFICATES

Scope: PDF certificate generation → Templates → Automated triggers → Download

Phase 13 Key Migration

PHP

```

Schema::create('certificate_templates', function (Blueprint $table) {

    $table->id();

    $table->string('name', 100);

    $table->enum('type', [

        'completion', 'achievement', 'attendance', 'milestone', 'custom'

    ]);

    $table->string('background_image_path')->nullable();

    $table->json('layout_config'); // Position, fonts, colors

    $table->boolean('is_active')->default(true);

    $table->boolean('is_default')->default(false);

    $table->timestamps();

});

Schema::create('certificates', function (Blueprint $table) {

    $table->id();

    $table->foreignId('student_id')

```

```

->constrained('users')
->cascadeOnDelete();
$table->foreignId('template_id')
->nullable()
->constrained('certificate_templates')
->nullOnDelete();
$table->enum('type', [
'completion', 'achievement', 'attendance', 'milestone', 'custom'
]);
$table->string('title', 255);
$table->text('description')->nullable();
$table->foreignId('issued_by')
->constrained('users')
->restrictOnDelete();
$table->date('issued_date');
$table->string('pdf_path', 500)->nullable();
$table->string('verification_code', 50)->unique(); // For public verification
$table->boolean('is_shared')->default(false);
$table->json('metadata')->nullable();
$table->timestamps();
$table->softDeletes();
$table->index(['student_id', 'type']);
$table->index('verification_code');
});

```

PHASE 14 — ENTERPRISE AUDIT & COMPLIANCE

Scope: Write-only audit logs → Before/after tracking → Admin viewer → IP logging

Phase 14 Key Migrations

PHP

```

Schema::create('audit_logs', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id')
        ->nullable()
        ->constrained()
        ->onDelete();
    $table->string('event', 100); // created, updated, deleted, login, etc.
    $table->string('auditable_type', 100)->nullable();
    $table->unsignedBigInteger('auditable_id')->nullable();
    $table->string('module', 100)->nullable();
    $table->string('ip_address', 45)->nullable();
    $table->text('user_agent')->nullable();
    $table->json('old_values')->nullable();
    $table->json('new_values')->nullable();
    $table->json('tags')->nullable();
    $table->timestamp('created_at')->useCurrent();
    // NO updated_at — write-only, immutable
    $table->index(['auditable_type', 'auditable_id']);
    $table->index(['user_id', 'created_at']);
    $table->index('event');
    $table->index('module');
});

```

Audit Implementation: Wire Spatie Activity Log (already installed and migrated) as the primary audit engine. Custom audit_logs table handles business-critical audits that require more granular before/after tracking.

PHASE 15 — BI & AI READINESS

Scope: Fact tables → Analytics snapshots → Student risk scoring → AI recommendations
→ Executive BI

Phase 15 Key Migrations

PHP

// Fact Tables for BI

```
Schema::create('fact_student_performance', function (Blueprint $table) {  
    $table->id();  
  
    $table->foreignId('student_id')->constrained('users')->cascadeOnDelete();  
    $table->foreignId('academic_year_id')->constrained()->cascadeOnDelete();  
    $table->foreignId('subject_id')->constrained()->cascadeOnDelete();  
  
    $table->string('metric', 100); // exam_avg, assignment_avg, attendance_rate  
    $table->decimal('value', 8, 2);  
  
    $table->timestamp('recorded_at')->useCurrent();  
  
    $table->index(['student_id', 'academic_year_id']);  
    $table->index(['subject_id', 'metric']);  
});
```

```
Schema::create('student_risk_scores', function (Blueprint $table) {  
    $table->id();  
  
    $table->foreignId('student_id')->constrained('users')->cascadeOnDelete();  
    $table->foreignId('academic_year_id')->constrained()->cascadeOnDelete();  
    $table->enum('risk_level', ['low', 'medium', 'high', 'critical'])->default('low');  
    $table->decimal('risk_score', 5, 2)->default(0);  
  
    $table->json('risk_factors'); // {"attendance": 0.4, "ops_trend": 0.3, ...}  
    $table->json('recommendations')->nullable();  
  
    $table->timestamp('calculated_at')->useCurrent();  
  
    $table->timestamps();  
  
    $table->index(['student_id', 'risk_level']);  
    $table->index('risk_level');  
});
```

```
Schema::create('teacher_performance_metrics', function (Blueprint $table) {  
    $table->id();
```

```

$table->foreignId('teacher_id')->constrained('users')->cascadeOnDelete();

$table->date('period_start');

$table->date('period_end');

$table->unsignedSmallInteger('sessions_conducted')->default(0);

$table->unsignedSmallInteger('assignments_graded')->default(0);

$table->unsignedSmallInteger('attendance_records_logged')->default(0);

$table->unsignedSmallInteger('resources_uploaded')->default(0);

$table->decimal('student_avg_ops_growth', 5, 2)->default(0);

$table->decimal('kpi_score', 5, 2)->default(0);

$table->json('breakdown')->nullable();

$table->timestamp('calculated_at')->useCurrent();

$table->timestamps();

$table->index(['teacher_id', 'period_start']);

});

```

COMPLETE IMPLEMENTATION SEQUENCE

Pre-Phase 10 (Run Now)

Bash

Step 1: Run all pre-Phase 10 migrations

```

php artisan make:migration add_primary_guardian_to_parent_student_table

php artisan make:migration extend_question_type_enum_in_quiz_questions_table

php artisan make:migration create_quiz_question_matching_pairs_table

php artisan make:migration add_attempt_rules_to_exams_table

php artisan make:migration create_exam_attempts_table

php artisan make:migration add_grading_rules_to_exams_table

php artisan make:migration add_grading_rules_to_assignments_table

php artisan make:migration add_attendance_config_to_school_settings_table

php artisan make:migration create_user_dashboard_preferences_table

```

Step 2: Install Paystack

```
composer require unicodeveloper/laravel-paystack
```

```
php artisan vendor:publish --  
provider="Unicodeveloper\Paystack\PaystackServiceProvider"
```

```
# Step 3: Run migrations
```

```
php artisan migrate
```

```
# Step 4: Commit pre-Phase 10 foundation
```

```
git add -A
```

```
git commit -m "chore(pre-phase-10): add primary guardian, question types, attempt  
rules, grading rules, dashboard prefs"
```

```
git push origin develop
```

```
Phase 10 (Finance)
```

```
Bash
```

```
git checkout -b feature/phase-10-finance
```

```
# Build all Phase 10 migrations, models, services, controllers, views, tests
```

```
git commit -m "feat(phase-10): complete financial management — invoices, Paystack  
payments, receipts"
```

```
git push origin feature/phase-10-finance
```

```
Phase 11 (Payroll)
```

```
Bash
```

```
git checkout -b feature/phase-11-payroll
```

```
# Build payroll period management, PayrollCalculatorService, admin fee snapshot  
system
```

```
git commit -m "feat(phase-11): complete payroll & compensation with dynamic admin  
fee audit"
```

```
Phase 12 (Communications)
```

```
Bash
```

```
git checkout -b feature/phase-12-communications
```

```
# Build messaging, announcements, notification system wiring
```

```
git commit -m "feat(phase-12): complete communications — messaging,  
announcements, notifications"
```

Phase 13 (Certificates)

Bash

```
git checkout -b feature/phase-13-certificates
```

```
# Build certificate templates, PDF generation, verification codes
```

```
git commit -m "feat(phase-13): complete certificate system with PDF generation"
```

Phase 14 (Audit & Compliance)

Bash

```
git checkout -b feature/phase-14-audit
```

```
# Wire Spatie Activity Log, build custom audit_logs, admin viewer
```

```
git commit -m "feat(phase-14): complete enterprise audit & compliance"
```

Phase 15 (BI & AI)

Bash

```
git checkout -b feature/phase-15-bi-ai
```

```
# Build fact tables, risk scoring, teacher metrics, executive BI feeds
```

```
git commit -m "feat(phase-15): complete BI & AI readiness — fact tables, risk scores, teacher KPIs"
```

SUMMARY OF REQUIRED ACTIONS BEFORE PHASE 10 CODE BEGINS

| Priority | Action Command |
|----------|----------------|
|----------|----------------|

- | | | |
|------------|---|--|
| ● CRITICAL | Add is_primary_guardian to parent_student | Migration E1-A |
| ● CRITICAL | Install Paystack package | composer require unicodeveloper/laravel-paystack |
| ● CRITICAL | Extend question_type enum (5 new types) | Migration E1-B |
| ● HIGH | Create quiz_question_matching_pairs table | Migration E1-C |
| ● HIGH | Add attempt rules to exams | Migration E1-D |
| ● HIGH | Create exam_attempts table | Migration E1-E |
| ● HIGH | Add grading rules to exams | Migration E1-F |
| ● HIGH | Add grading rules to assignments | Migration E1-G |
| ● HIGH | Add Paystack config to school_settings | Migration E1-H |

● MEDIUM Create user_dashboard_preferences table Migration E1-I