Here is the detailed summary of the work completed over the past week (Jan 19 - Jan 25, 2026) for the **E-Digitex School Management System**.

This log is structured by **Module**, detailing the specific **Files**, **Features**, and **Logic** implemented.

1. Academics Module

A. Departments (DepartmentController.php)

- **Purpose:** Organize academic divisions (e.g., Science Dept, Arts Dept) mainly for University/Mixed institutions.

- **Key Logic:**

  o Standard CRUD.

  o Linked to Institution.

  o Used as a higher-level grouping above Grade Levels/Programs in some university contexts.

- **Visibility:** Logic added to Sidebar to only show "Departments" if the institution type is 'university' or 'mixed'.

B. Grade Levels (GradeLevelController.php)

- **Logic:** Added logic to **Auto-Select/Hide** the "Education Cycle" field if the institution has a fixed type (e.g., a Primary School admin won't see "University" options).

2. Finance Module (Major Updates)

Goal: Establish a robust financial system handling fees, invoicing, student balances, budgets, and payments.

A. Fee Structure (FeeStructureController.php)

- **Purpose:** Define fees (Tuition, Bus, etc.) globally or per class.

- **Key Logic:**

  o **Global vs. Installment:** Introduced payment_mode ('global', 'installment').

  o **Validation:** Prevents deleting a Global Fee if child Installments exist. Prevents total installments from exceeding the Global Fee amount.

  o **Security:** Strict institution_id checks.

B. Invoicing (InvoiceController.php)

- **Purpose:** Generate bills for students based on Fee Structures.

- **Key Logic:**

  o **Bulk Generation:** Generates invoices for entire classes.

- o **Fee Caps:** Ensures invoices do not exceed the defined Global Fee cap for the year.

- o **Duplicate Check:** Prevents double-billing the same fee to the same student in a session.

- o **Unique Numbers:** Generates unique invoice numbers (INV-{Year}-{AdmNo}-{Time}).

C. Payments (PaymentController.php)

- **Purpose:** Record payments against invoices.

- **Key Logic:**

  - o **Security: Password Confirmation** is now mandatory before recording a payment.

  - o **Validation:** Payment cannot exceed the remaining balance of the invoice.

  - o **Status Update:** Automatically updates Invoice status (paid, partial, unpaid).

D. Student Balances (StudentBalanceController.php)

- **Purpose:** Provide a financial overview of students per class.

- **Features:**

  - o **Tabs:** Dynamic tabs for "Global Fee" and individual "Installments".

  - o **Stats:** Calculates Total Invoiced, Collected, and Outstanding per class.

  - o **Logic:** Tracks "Paid Students" vs. "Debtors" accurately.

E. Budgeting (BudgetController.php)

- **Purpose:** Manage school expenses and allocations.

- **Features:**

  - o **Period-Based:** Budgets are now allocated per period (e.g., "Q1 2025", "Jan 2026") rather than just annually.

  - o **Fund Requests:** Workflow for staff to request funds -> Admin approves/rejects.

  - o **Traceability:** Tracks allocated_amount vs spent_amount.

  - o **Flexibility:** Admins can now update budget allocations freely (removed strict "cannot reduce below spent" lock upon request).

3. Academic Reports Module

Goal: Generate academic documents (Bulletins, Transcripts) with institution-specific logic.

A. Report Generation (ReportController.php)

- **Purpose:** Generate PDF reports.

- **Key Logic:**

- **Institution Types:** Logic forks based on type (primary, secondary, university).

- **Bulk Printing:** Added ability to print bulletins for an entire class at once.

- **Validation:** Checks if the requested Period/Term is "Active" in settings before generating.

- **Ajax Pre-check:** Checks if data exists *before* opening the PDF tab to avoid blank pages.

B. Result Cards (ResultCardController.php)

- **Purpose:** View single student results.

- **Features:**

  - **Teacher Access:** Teachers can only view results for students they teach/are linked to.

  - **GPA/CGPA:** Added logic for credit-based GPA calculation for University context.

4. Student Enrollment (Workflow Overhaul)

Goal: Separate workflows for School vs. University and improve UX.

A. Standard Enrollment (StudentEnrollmentController.php)

- **Context:** Primary/Secondary schools.

- **Changes:**

  - **Bulk Enrollment:** Shifted from single-student entry to **Bulk Multi-Select**.

  - **Flow Update:** Select Class First -> Then Select Multiple Students.

  - **Filtering:** Student dropdown now correctly filters unenrolled students.

B. University Enrollment (UniversityEnrollmentController.php - NEW)

- **Context:** Universities/Higher Ed.

- **Features:**

  - **Separate Module:** Created a distinct controller and view set.

  - **Terminology:** Uses "Program" and "Level" instead of "Class".

  - **Routes:** Added specialized routes under /university/enrollments.

  - **Permissions:** Integrated specific permissions (university_enrollment.create, etc.).

5. System Configuration & Security

A. Roles & Permissions (RolePermissionSeeder.php)

- **Update:** Re-seeded permissions to include all new modules (University Enrollments, Budgets).

- **Scope:** Ensured university_enrollment.* permissions are created and assigned to Super Admin.

B. Sidebar & Navigation (sidebar.blade.php, sidebar.php)

- **Logic:** Added dynamic visibility for "University Enrollments" vs "Enrollments" based on the logged-in user's Institution Type.

- **Localization:** Fully localized sidebar items in sidebar.php.

6. File & Path Reference

Module	Controller	Model	Views Path
Academics (Depts)	DepartmentController	Department	departments/
Finance (Fees)	FeeStructureController	FeeStructure	finance/fees/
Finance (Invoice)	InvoiceController	Invoice	finance/invoices/
Finance (Budget)	BudgetController	Budget	finance/budgets/
Reports	ReportController	ExamRecord	reports/
Enrollment (Sch)	StudentEnrollmentController	StudentEnrollment	enrollments/
Enrollment (Uni)	UniversityEnrollmentController	StudentEnrollment	university/enrollments/
Grades	GradeLevelController	GradeLevel	grade_levels/

7. Pending / Next Steps

- **University Course Registration:** The "Enrollment" puts them in a Program. The next step is registering for specific *Courses* (Subjects) for the semester.

- **LMD Logic:** Implementing the specific calculation rules for Licence-Master-Doctorat grading (Credits, UE, EC).

This list covers all major architectural changes, logic implementations, and UI/UX improvements made in the last 7 days.