

Functional Document for EEC

Educational ERP & LMS Platform

Project Name:	EEC
Document Type:	Functional / Technical Reference Document
Document Version:	2.0
Original EEC Start Date:	
EEC Completion Date:	
EEC Re-architecture Date:	
Document Date:	February 2026
Note:	This document only includes main features/modules.

1. Project Overview

EEC is the next-generation re-architecture of the original EEC (Educational ERP & LMS) platform. While EEC was built on Django/Python with a Flutter mobile app, EEC rebuilds the entire stack using Node.js (Express), React, and MongoDB - retaining all original business logic and adding new AI-powered learning features, multi-tenant institution management, and a modern responsive UI.

The platform serves educational institutions (schools and colleges) providing end-to-end management of academics, attendance, fee collection, human resources, communication, examinations, assignments, lesson plans, parent engagement, and AI-driven personalised learning.

2. Technology Stack

2.1 Backend

- Runtime / Language: Node.js (JavaScript)
- Framework: Express.js v5.1.0
- Database: MongoDB (via Mongoose v8.15.1)
- Authentication: JWT (jsonwebtoken v9.0.2), bcryptjs v3.0.2
- File Storage: Multer v2.0.2, Cloudinary v2.8.0
- Email: Nodemailer v6.9.13
- CSV / Bulk Import: csv-parser v3.2.0
- API Documentation: Swagger (swagger-autogen, swagger-ui-express)
- Rate Limiting: express-rate-limit v8.2.1
- Environment: dotenv v16.5.0, CORS v2.8.5

- HTTP Client (internal): Axios v1.13.2
- WebHooks: webhookSender utility

2.2 Frontend

- Framework: React v18.2.0
- Build Tool: Vite v6.3.5
- Routing: React Router DOM v7.6.2
- Styling: Tailwind CSS v4.1.10
- Charts & Visualisation: Chart.js v4.5.0, react-chartjs-2, Recharts v2.15.0
- Rich Text Editor: Quill v1.3.7
- PDF Export (client-side): jsPDF v3.0.2
- Excel Handling: xlsx v0.18.5
- Notifications: react-hot-toast, SweetAlert2 v11.26.3
- Icons: lucide-react v0.514.0
- 3D Graphics: three.js v0.171.0
- HTTP Client: Axios v1.13.2
- Deployment: Vercel (vercel.json SPA rewrites)

2.3 Original EEC Stack (for reference)

- Backend: Python, Django, Django REST Framework
- Auth: JWT (SimpleJWT)
- Real-time: Django Channels, Redis (WebSocket support)
- Database: PostgreSQL
- Frontend: React, Redux, Axios, Reactstrap, SCSS/CSS
- Mobile App: Flutter (Dart)
- Media: Cloudinary, Leaflet (mapping)
- Server: Ubuntu, AWS, Gunicorn, Daphne, Nginx

3. Project Statistics

Original EEC - DB tables:	347
Original EEC - Code files:	459
EEC - Backend models:	40
EEC - Backend route files:	39
EEC - Backend middleware:	8
EEC - Backend utility files:	9
EEC - Frontend component files:	206
EEC - Maintenance/migration scripts:	5
Total API endpoints (Swagger documented):	100+

4. User Roles & Access

Institution Head / Super Admin

Top-level multi-school management. Can manage all schools registered on the platform, create school accounts, and access aggregate reports across institutions.

School Admin

Per-school administrator. Manages academic setup, user creation (bulk or individual), fee structures, notices, reports, and overall school configuration.

Principal

Academic oversight and reporting. Access to analytics dashboards, staff management, student analytics, financial dashboards, and communication centre.

Teacher

Class and student management. Can mark attendance, create assignments, manage lesson plans, add practice questions, schedule PTMs, view timetables, and give student feedback.

Class Teacher

All teacher permissions plus additional class-specific responsibilities such as behaviour tracking and excuse letter approvals.

Student

Learner portal. Can view assignments, submit work, attempt exams and practice questions, view attendance and results, use AI learning tools, and chat with peers.

Parent

Child monitoring portal. Can view their child's attendance, results, fee status, class routine, and communicate via PTM portal and complaint management.

Staff / Accountant / Librarian

Role-specific access for support functions such as fee collection, library management, and HR-related operations. Optional roles configured per institution.

5. System Modules

5.1 Core / Academic Setup

- School Profile Management - School data (name, logo, address, contact) is stored and used for report cards, invoices, and email communication.
- Academic Year - Manages academic calendar years; all data is scoped to an academic year.
- User Management - Profile creation, password reset, and account status. Mobile number is the unique identifier/username. 256-bit password encryption. Every login is recorded.
- User Groups - Role-based user groups must be configured during installation.
- Class/Course - Manages the list of classes/courses. Class name and class code must be unique. Fee structure is linked at class level.

- Section - Acts as a room/division within a class batch. Each student is mapped to a section.
Mapping: Student -> Section -> Course Batch -> Course.
- Subject - Managed per course. Course + subject combination must be unique. Subjects hold chapters and study materials.
- Syllabus - Mapped as Syllabus -> Subject -> Batch -> Course.
- Timetable - Class timetable management; automated timetable generator utility available.
- Notice Management - Create and publish notices to targeted user groups.
- Holiday Management - Holidays can be created or deleted; affects attendance auto-generation.
- Complaint & Feedback Management - Manages complaints and feedback from all user types.

5.2 Student

- Student Database - Stores all student personal, academic, and parent data.
- User Account - Every student gets a system user account; mobile number is username.
- Unique Student Code - Auto-generated by the system on enrolment.
- Enrolment - Course/class, section, parent, and previous school data captured. Course, parent, and personal data are mandatory.
- Multi-Course Enrolment - One student can be enrolled in multiple courses simultaneously.
- Parent Linking - If parent already exists, entering their registered mobile number auto-links them.
- Fee Invoice - Automated fee invoice created per fee structure assigned during enrolment.
- Bulk Upload - Students can be bulk-imported via CSV/Excel template (with class, section, and parent details).
- Student Progress - Progress scores calculated per student, subject, batch, and course.
- Student Journal - Students can maintain personal journal entries.
- Wellbeing Tracking - Student wellbeing data tracked and accessible to teachers and admin.
- Behaviour Records - Behaviour incidents recorded and linked to student profile.
- Excuse Letters - Students can submit digital excuse letters for absences.

5.3 Teacher

- Teacher Database - Stores education, documents, and previous experience data.
- User Account - Teacher code is the username; unique teacher code required at registration.
- Employee Account - An employee record is automatically created for every new teacher.
- Subject Assignment - Teachers are assigned a set of subjects; all resources of assigned subjects are accessible.
- Teacher Timetable - Each teacher has a personal timetable. Change requests can be submitted to admin/management.
- Teacher Attendance - Separate attendance tracking for teaching staff.
- Teacher Feedback - Teachers can provide structured feedback on student performance.
- Teacher Expense - Expense recording for reimbursement.
- Teacher Leave - Leave requests managed within the HR module.
- Teacher Allocation - Subject and section allocation management.

5.4 Parent

- Parent Database - Stores all parent contact and profile data.
- Account Creation - Parent account created during student enrolment or independently via admin panel.
- Multiple Students - One parent account can be linked to multiple students.
- Access - Parents can view attendance reports, fee payment status, class routine, academic results, and PTM schedule.
- Complaint Management - Parents can raise complaints and track resolution.
- Fee Payment - Fee payment actions available from the parent portal.

5.5 Attendance

- Student Attendance - Auto-generated daily at 2 AM (configurable from super admin panel) with default status: absent, holiday, or Sunday. Status updated when marked by teacher.
- Employee Attendance - Same auto-generation process for all active employees.
- Time Analysis - System analyses attendance timing data and marks status accordingly.
- Payroll Integration - Attendance data feeds into monthly salary generation.
- Excuse Letters - Linked to attendance records for absence justification.

5.6 Examination

- Examination Database - New exams created and mapped to Subject -> Batch -> Course.
- Structure - Each exam has a set of subjects; each subject has questions, duration, and due date/time.
- Examiners - Each subject in the exam is assigned a set of examiner teachers.
- Question Types - Multiple question types supported (MCQ, FIB, MTF, TF, LAQ, SAQ).
- Live Activation - Exam becomes available to students only after admin/management publishes it.
- Immutability - No changes permitted once the exam is published.
- Student Attempt - Students attempt examinations subject-by-subject on the specified due date/time. One attempt only.
- No Back Navigation - Once started, students cannot go back. Exam auto-submits on duration expiry.
- Marking - Post-submission, answers are visible to examiners for review and marking.
- Report Card - Once all marking is complete, admin/management can publish the report card.
- Anti-Cheating - Certain browser actions are prevented during examination. AI-based anti-cheat integration is planned.

5.7 Assignments

- Assignment Creation - Teachers create assignments linked to class, section, and subject.
- Student Submission - Students can submit work (files, text) through their portal.
- Teacher Review - Teachers can view all submissions and provide marks and feedback.
- Status Tracking - Assignment status (pending, submitted, reviewed) tracked per student.

5.8 Lesson Plan

- Lesson Plan Database - Stores all lesson plan data; mapped to Course -> Batch -> Subject.
- Composition - A lesson plan can include one chapter, homework, tryout, quiz, assessment, and

retrieval. Chapter and teacher are mandatory.

- Teacher Visibility - Lesson plan visible only to the assigned teacher.
- Activation - Teacher can start the lesson on its scheduled due date.
- Completion Flow - Once teacher marks a lesson plan complete, associated homework, tryouts, and items appear in the student's section account.
- Student Submissions - Students can submit work against assigned lesson plan items.
- Feedback Loop - Teachers can review all student submissions and provide feedback.
- Post-Completion Survey - Student feedback/survey form delivered after lesson plan completion (planned feature).

5.9 Tryout & Practice Questions

- Tryout Database - Stores tryout/practice quiz data.
- Question Types - MCQ, FIB (Fill-in-the-Blank), MTF (Match-the-Following), TF (True/False), LAQ (Long Answer), SAQ (Short Answer).
- Auto-Marking - If a correct answer exists in the system, the system automatically marks responses as correct or incorrect.
- Teacher Override - Teachers can always change answer status manually.
- Practice Questions - AI-assisted practice question sets available to students.
- Practice Attempts - Student attempt history recorded for progress analysis.

5.10 Progress & Report Card

- Progress Algorithms - Module holds all algorithms to calculate student, subject, batch, and course progress.
- Background Calculation - Whenever related data changes, progress recalculation runs in background. Adequate server resources required.
- Cached Results - Calculation results stored in the database for instant access.
- Report Card - Exam result data stored and report card generated as PDF download.
- PDF Template - HTML design template used for PDF generation.

5.11 Fee Management

- Fee Structure - Fee structures configured per course/class by admin.
- Fee Invoice - Auto-generated on student enrolment based on assigned fee structure.
- Payment Recording - Fee payments tracked per invoice; payment history maintained.
- Parent Access - Parents can view outstanding invoices and payment history from their portal.
- Financial Dashboard - Principal and admin can view school-wide financial summaries.

5.12 Employee / HRMS

- Employee Database - Manages all non-teaching and teaching staff records.
- Department & Designation - Employees assigned to departments and designations.
- Multiple Employee Types - Teaching staff, administrative staff, support staff, etc.
- Attendance - Integrated with the Attendance module for daily tracking.
- Payroll - Automated monthly salary generation based on attendance.

- Pay Slips - Employees can access their pay slips through the portal.
- Leave Management - Leave requests and approvals managed in this module.
- Expense Management - Employee expense submissions tracked here.
- Resignation - Employee resignation workflow available.

5.13 Parent-Teacher Meeting (PTM)

- PTM Creation - Teachers create PTM sessions linked to specific students.
- Parent Notification - Auto-notification sent to mapped student's parent on PTM creation.
- Parent Actions - Parent can accept the meeting request or ask to reschedule.
- MOM Storage - Minutes of Meeting (MOM) can be stored for future reference.

5.14 Communication

- Notice Board - Notices created by admin visible to targeted user groups.
- Real-Time Chat - Users can chat in real time (requires friendship/connection).
- Abusive Language Filter - System automatically masks abusive words in chat.
- Chat Immutability - Chat messages cannot be deleted by users.
- Academic Alcove - Q&A / discussion forum for academic topics (AlcovePost, AlcoveComment models).
- Notifications - System-generated notifications for key events (PTM, assignments, results, etc.).
- Webhook Support - Webhook integration available for external system event notifications.

5.15 AI Learning Features

- AI Learning Dashboard - Personalised learning path dashboard for students.
- AI Problem Solver - Students can submit problems and receive AI-generated solutions.
- AI Quiz Generator - Automatically generates quiz questions on demand.
- AI Summary Generator - Generates summaries of study material.
- AI Textbook Chat - Students can interact with textbook content via AI chat interface.
- Flashcard Generator - AI-generated flashcards for revision.
- Mind Map Generator - Visual mind maps generated from topic inputs.
- Weak Student Identification - System identifies underperforming students and suggests targeted learning paths.
- Student AI Learning Route - NIF-specific AI learning route endpoints for personalised study.

5.16 Class Routine

- Student Daily Routine - Manages the daily class schedule for students.
- Mapping - Routine -> Section -> Course Batch -> Course.
- Availability - Routine visible to students according to enrolled course.
- Teacher Timetable - Separate per-teacher timetable management.
- Automated Generator - Timetable generator utility available for automated scheduling.

5.17 Events

- Event Management - Admin/management creates school events.

- Targeted Events - Events can be created for specific user types.
- Registration - Users can register to participate in events.
- Media Upload - Participants can upload images/videos; reviewed by admin/management.

5.18 Library

- Offline/Online Library - Manages physical and digital library resources.
- Book Orders & Payments - Users can order books and make payments.
- Due Date Tracking - System tracks all borrowed books and notifies librarian of overdue returns.

5.19 Handbook

- Handbook Management - Manages handbooks and related documents.
- Access Control - Available to admin and employees only.

5.20 Transport

- Vehicle & Driver Management - All vehicles and drivers managed in this module.
- Driver Accounts - System auto-creates a user and employee account for each new driver.
- Route Management - All vehicle routes managed with defined stops.
- Student-to-Route Mapping - Sets of students assigned to routes and stops.
- Real-Time Tracking - Admin/management can track vehicles on-route in real time.
- Drop Notifications - System notifies driver which students to drop at each stop.

5.21 Student Wellbeing

- Wellbeing Records - Student wellbeing data collected and tracked over time.
- Access - Visible to teachers and admin for pastoral care and early intervention.

5.22 Issue Tracking & Support

- Issue Module - Users can raise issues that are tracked through to resolution.
- Support Requests - Dedicated support request module for technical or administrative help.
- Audit Log - All significant system actions logged for auditing and accountability.

5.23 Personal Task Manager (TODO)

- Available to all user types.
- Users can create, manage, and track their personal daily tasks.

5.24 Games

- Interactive games integrated into the student portal for engagement and gamified learning.

6. Data Architecture & Key Mappings

Key entity relationship mappings in the system:

- Student -> Section -> Course Batch -> Course
- Subject -> Course

- Syllabus -> Subject -> Batch -> Course
- Routine -> Section -> Course Batch -> Course
- Lesson Plan -> Course -> Batch -> Subject
- Examination -> Subject -> Batch -> Course
- Teacher Timetable -> Teacher
- Fee Invoice -> Student -> Fee Structure -> Course
- Employee -> Department -> Designation
- Attendance (Student) -> Student -> Section -> Course
- Attendance (Employee) -> Employee -> Payroll
- Transport Route -> Stops -> Students

6.1 Backend Models (40 total)

User & Auth:	Admin, Principal, TeacherUser, StudentUser, ParentUser, StaffUser
Academic:	School, AcademicYear, Class, Section, Subject, Timetable
Operations:	Attendance, Assignment, Exam, ExamResult, Feedback, TeacherFeedback, LessonPlan, LessonPlanCompletion, PracticeQuestion, PracticeAttempt
Finance:	FeeStructure, FeeInvoice, FeePayment
Communication & Engagement:	Post, AlcoveComment, Notification, AuditLog
HR:	TeacherAllocation, TeacherAttendance, TeacherExpense, TeacherLeave
Student Tracking:	Behaviour, StudentProgress, StudentJournalEntry, Wellbeing
Other:	ExcuseLetter, Issue, ParentMeeting, SupportRequest

7. API Structure

The backend exposes a RESTful API documented via Swagger (available at /api-docs). Routes are grouped by domain:

Authentication:	adminRoutes, teacherRoute, staffRoutes, studentRoute, parentRoute, principalRoutes, superAdminRoutes, schoolRegistrationRoutes
Dashboard:	teacherDashboardRoutes, principalDashboardRoutes
Academic:	academicRoutes (years, classes, sections, subjects), subjectRoute
Operations:	attendanceRoutes, assignmentRoute, examRoute, feedbackRoute, behaviourRoute, progressRoute, timetableRoutes, lessonPlanRoutes
Finance:	feeRoutes
Communication:	alcoveRoute, notificationRoutes, meetingRoute
Admin / User Management:	adminUserManagement (bulk create, CSV import), reportRoutes, auditLogRoutes
AI Features:	aiLearningRoute, studentAILearningRoute, practiceRoutes
Student Features:	excuseLetterRoutes, issueRoutes, supportRoutes, wellbeingRoute, nifStudentRoutes
HR:	teacherAllocationRoutes
Infrastructure:	uploadRoutes, schoolRoutes

8. Security

- JWT-based stateless authentication - separate middleware per user role (adminAuth, authTeacher, authStudent, authParent, principalAuth, superAdminAuth, authAnyUser).
- bcryptjs password hashing (256-bit equivalent) - passwords never stored in plaintext.
- Password policy enforcement via passwordPolicy utility.
- API rate limiting via express-rate-limit to prevent brute-force attacks.
- Role-based route protection - each API endpoint is protected by role-specific middleware.
- Every user login activity is recorded for audit purposes.
- Abusive language filtration in real-time chat.
- Anti-cheating measures during online examinations (browser action restrictions; AI anti-cheat planned).
- Audit log records all significant system actions.
- Cloudinary secure media upload with signed URLs.
- CORS policy enforced on all API routes.
- Chat messages are immutable (cannot be deleted by users).
- Environment variables managed via .env (never committed to version control).

9. Deployment & Infrastructure

9.1 Frontend

- Hosted on Vercel.
- vercel.json configured with SPA fallback rewrites.
- Environment variables injected at build time via .env.

9.2 Backend

- Node.js Express server; entry point: index.js.
- MongoDB Atlas (cloud) or self-hosted MongoDB.
- Cloudinary for all media/file storage.
- Nodemailer for transactional email.
- Swagger UI available at /api-docs endpoint.

9.3 Original EEC Infrastructure (for reference)

- Ubuntu server on AWS.
- Gunicorn for Django WSGI serving.
- Daphne for Django ASGI (WebSocket) serving.
- Nginx as reverse proxy / load balancer.
- Redis for Django Channels (WebSocket message broker).

10. Planned & Future Features

- Question / tryout creation via file upload (bulk import).
- Teacher timetable change request workflow (admin approval).
- Post-lesson plan student feedback/survey form delivery.
- End-to-end chat encryption.
- AI anti-cheating integration for online examinations.
- 3D interactive content using three.js (infrastructure already in place).
- Games expansion for gamified learning.
- Transport module - full implementation with live GPS tracking.
- Library module - full offline/online library with payment integration.
- Event module - full participant media review workflow.
- Handbook module - full implementation.

Note: This document covers all main features and modules of EEC. Any further changes to the system will require an update to this document.