Product Requirements Document (PRD)

Project Title: Multischool ID Card System (MERN Stack Redevelopment)
Version: 1.0
Date: November 2025

1. Overview

The **Multischool ID Card System** is a modern, multi-tenant web application built using the **MERN Stack (MongoDB, Express.js, React.js, Node.js)**.
It enables educational institutions to efficiently manage student, teacher, and class data, and generate **customized ID cards** using flexible templates.
The platform supports **multi-session management**, **bulk data operations**, **role-based permissions**, and a **responsive interface** for desktop and mobile users.

2. System Architecture & Technology Stack

Layer	Technology / Framework	Description
Frontend	React.js, Material-UI / Bootstrap	Responsive and dynamic user interface
Backend	Node.js + Express.js	RESTful API with modular route structure
Database	MongoDB	Schema-flexible storage for multi-tenant school data
File Storage	AWS S3 /Cloud Storage/ Local	Storage for photos, templates, and attachments
Authentication	JWT + Google OAuth 2.0	Secure multi-role authentication and authorization
PDF Generation	pdf-lib / Puppeteer	Server-side dynamic ID card generation
Bulk Operations	XLSX.js, Multer	Bulk import/export for student and teacher data

3. Roles & Permissions

Role	Scope	Key Permissions
Superadmin	All schools	Manage all schools, users, templates, sessions, notices, logs, bulk operations
Schooladmin	Assigned school	Manage school-specific data (students, teachers, classes, templates, notices)
Teacher	Assigned class/school	Manage students within assigned class, update profile, generate cards if permitted

4. Key Features & Modules

4.1 User Management & Authentication

- JWT-based secure authentication
- Email/password login with encrypted storage
- Password reset via secure email link
- Google OAuth 2.0 SSO login
- Profile management and password change

4.2 Dashboard & Role-Based Navigation

- **Superadmin:** System-wide statistics, user/session/template management
- **Schooladmin:** School-level insights, quick access to data and operations
- **Teacher:** Class-based view of students and ID cards

4.3 Template Management for ID Cards

- Create/edit/delete templates with drag-and-drop or form editor
- Dynamic data binding with student/teacher fields
- Real-time preview with sample data
- Assign templates to specific sessions, classes, or roles

4.4 Multi-Session Support

- Create/manage academic sessions per school
- Assign students, teachers, and classes to sessions
- Session-based card generation and archival

4.5 Student & Teacher Data Management

- CRUD with data validation (e.g., 10-digit mobile, 12-digit Aadhaar)
- **Class Freeze** to lock class data for given sessions
- Teacher assignment to single class (one teacher per class)
- Bulk XLSX import/export for students & teachers
- Bulk deletion and promotion workflows
- Bulk image upload mapped by admission number or ID

4.6 Photo Handling

- Client-side compression before upload
- Server-side optimization and validation
- Bulk photo import with mapping
- Efficient cloud-based image storage and retrieval

4.7 Notices & Communication

- CRUD notices with attachments (PDF/JPG/PNG/Links)
- Role-based visibility on dashboards
- Notifications and alerts for important updates

4.8 Login Activity & Security Logs

- Log every login attempt with timestamp, IP, role, and school
- Viewable by Superadmin for audits
- Configurable login permissions for Schooladmins and Teachers

---

5. Data Model (Conceptual)

Collection	Fields
Users	{ id, username, name, email, role, schoolId, passwordHash, status, createdAt }
Teachers	{ id, name, mobile, email, schoolId, photoUrl, classId, password, status, createdAt }
Students	{ id, admissionNo, name, dob, classId, fatherName, motherName, mobile, address, aadhaar, photoUrl, sessionId, lastUpdated }
Classes	{ id, schoolId, className, frozen, sessionId }
Sessions	{ id, schoolId, sessionName, startDate, endDate, activeStatus }
Templates	{ id, schoolId, type, layoutConfig, dataTags, createdAt, updatedAt }
Notices	{ id, schoolId, content, noticeDate, attachmentUrl, attachmentType, status }
LoginLogs	{ id, username, role, schoolId, ipAddress, timestamp }
AllowedLogins	{ schoolId, allowSchoolAdmin, allowTeacher }

---

6. User Flows

6.1 Superadmin Flow

1. Login via email/password or Google OAuth
2. Access system-wide dashboard
3. Manage users and assign schools
4. Create/manage academic sessions
5. Customize and assign ID templates
6. Configure login permissions
7. Freeze/unfreeze classes
8. Perform bulk imports/exports/promotions
9. Monitor login logs
10. Generate ID cards for schools/classes/sessions
11. Manage global notices

6.2 Schooladmin Flow

1. Login via credentials or Google OAuth
2. Access school dashboard
3. Manage sessions, classes, students, teachers
4. Handle XLSX imports, bulk image uploads
5. Assign templates and generate ID cards
6. Post/manage school-specific notices
7. Promote students between sessions
8. Update profile and password

6.3 Teacher Flow

1. Login via email/mobile or Google OAuth
2. View class overview and student list
3. Add/edit student info (if class not frozen)
4. Generate student or personal ID cards (if allowed)
5. Update personal profile
6. Request password reset

7. Technical & Operational Specifications

Aspect	Specification
Security	HTTPS, JWT with expiry, RBAC
Validation	Client & server-side (all forms and uploads)
Error Handling	Friendly error messages & fallback options
API Documentation	Swagger / OpenAPI 3.0
Performance	Pagination, async bulk ops, optimized image delivery
Storage	Separate image storage for students & teachers
Notifications	Email-based password reset & alert options
Responsiveness	Fully responsive across devices
Testing	Unit + integration tests for key modules

8. Future Enhancements (Optional Scope)

- SMS/Email notification module
- Mobile app integration using React Native
- Role-based analytics dashboard (attendance, performance, etc.)
- QR-based ID card verification
- Multi-language support

9. Deliverables

- Fully functional MERN-based web application
- API documentation
- Deployment setup documentation (AWS/Render/Vercel)
- Source code repository with version control