# Project Documentation: NMSS SMART

**1. Project Title & Team**

- **Project Title:** NMSS SMART (School Management System)
- **Tagline:** Automating Education with a code.
- **Team Name:** Team NMSS
- **Team Members:**
    - Prasiddha Subedi : [Role, Front End, Project Coordinator]
    - Rishav Gautam : [Role, UI/UX Designer, Front End]]
    - Suyog Devkota : [Role, Backend/Flask Dev]

**2. Executive Summary**

NMSS SMART is a modern, full-stack School Management System built to **eliminate manual administrative tasks** in educational institutions. Our platform directly addresses the inefficiencies of manual attendance, grade reporting, and communication silos by providing an **integrated, automated digital ecosystem**. Leveraging the **MERN stack (MongoDB, Express.js, React, Node.js)** for a dynamic user interface and robust application layer, combined with **Flask** for efficient microservices, we deliver a seamless experience for administrators, teachers, students, and parents. NMSS SMART transforms school operations from paper-based workflows to real-time digital management.

**3. Introduction & Problem Statement**

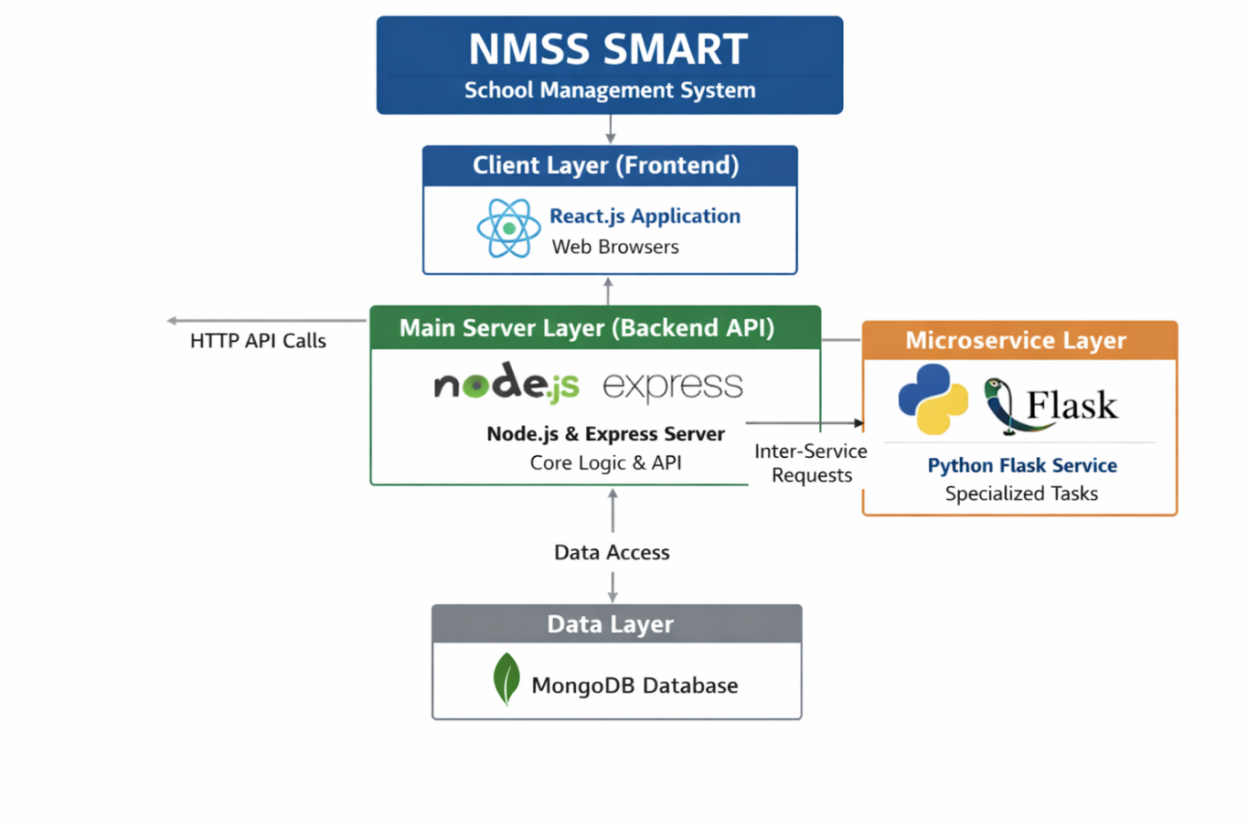
The administration of many schools relies on outdated, manual processes. This leads to:

- **Time-Consuming Attendance & Grading:** Teachers waste valuable teaching hours on manual entry, prone to errors and loss.
- **Inefficient Communication:** Notices and reports are delayed due to physical/hierarchical flow, affecting parental engagement.
- **Manual Routine Management:** Manual timetable creation on paper/spreadsheets is time-consuming, prone to overlaps, and difficult to modify or communicate to stakeholders.
- Our solution, **NMSS SMART**: consolidates these functions into a single, secure, and user-friendly platform, freeing educators to focus on education.

**4. Solution & Implementation**

**A. Overall Approach:** We built a role-based web application with dedicated dashboards for Admins, Teachers, Students, and Parents. Each module automates a key manual process.

**B. System Architecture:**




1. Frontend (React): Serves the dynamic user interface.
2. Backend (Node.js/Express): Handles core business logic, authentication, and API routing.
3. Microservice (Flask): Dedicated to specific heavy tasks (e.g., Routine Creator).
4. Database (MongoDB): A NoSQL database for flexible storage of user profiles, grades, attendance, and notices.

**C. Technology Stack:**

- **Frontend:** React, CSS
- **Backend:** Node.js, Express.js, Mongoose (ODM)
- **Microservice:** Python, Flask
- **Database:** MongoDB
- **Tools & Deployment:** Git/GitHub

**D. Key Features & Modules:**

- **Automated Attendance:** QR-code or one-click attendance marking with real-time logs.
- **Digital Gradebook & Report Card Generator:** Teachers input grades; system auto-calculates percentages and generates printable PDF reports (Flask service).

- **Centralized Notice Board:** Instant push of alerts/notices to specific roles (e.g., all parents, grade 10 students).
- **Role-Specific Dashboards:** Personalized views for all stakeholders.
- **One-Click Attendance & Grading:** QR-code or list-based attendance marking with automated logs and a digital gradebook that calculates percentages and generates PDF report cards.

**5. Future Work**

- Implement a **mobile app** using React Native for parent/student access.
- Integrate **data analytics** (using Flask/Pandas) for visualizing student performance trends.
- Add **AI-powered features** like attendance fraud detection or predictive grade analysis.

**6. How to Run This Project**

**Prerequisites:** Node.js, npm, Python/pip, MongoDB Atlas account

**7. References & Resources**

- MongoDB Documentation
- React Documentation
- Flask Documentation