



Task Title

Roadmap creation for School Management Software

Author's Name: Debajyoti Banerjee
Employee ID: CTX/205956
Assigned By: Akhilesh Yadav (Tech Lead)
Company Name: Code Talent X
Document Version: 1.0





Table Of Contents

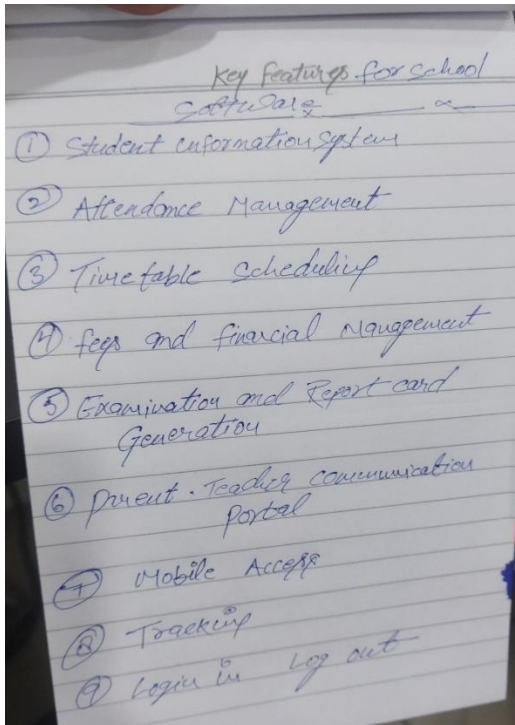
Serial No:	Content Headers	Page No
1	Problem Statement & Realization	2-5
2	Abstract	6
3	Introduction	7
4	Benefits & drawback	8
5	Feasibility Study	9-10
5	Approach to solve the problem statement	10-13
6	System Requirement Analysis	13-14
7	Technology Stack Overview	15
9	Charts & Road Maps	16-18
10	Future Scope & Conclusion	19





Problem Statement

Given:



Page | 2

Key data extracted from the given picture:

Downloading Source: Microsoft Teams platform

Key Features for School Software (Heading):

1. **Student Information System**
2. **Attendance Management**
3. **Timetable Scheduling**
4. **Fees and Financial Management**
5. **Examination and Report Card Generation**
6. **Parent. Teacher communication portal**
7. **Mobile Access**
8. **Tracking**
9. **Log-in and Log-out**

Problem: Create/Make a software that manages school Activities.





Statement provided: That software should handle: Student information system, attendance management, Timetable Scheduling, fees and financial management, Examination and Report Card generation, parent and teacher communication portal, mobile access, tracking, login in and log out.

Problem Realisation and System Understanding Statement

The owner of the school envisions a comprehensive, modular School Management Software (SMS) that serves as a centralised platform to manage all academic, administrative, financial, and communication-related operations across the school ecosystem. The system should be scalable, secure, and cloud-accessible, with responsive support for both desktop and mobile devices.

Page | 3

1. Student Information System

The system must store and manage complete student profiles, including:

- Personal information: Name, Roll No., Registration No., Blood Group, Contact Info, Date of Admission, Pass-out Year
- Academic details: Class, Assigned Subjects (core and optional), Head Teacher, Examination records, Report Cards
- Attendance records (daily, exam-specific), Leave status
- Fee details: Amount, Mode (cash/online), Status, Scholarship info
- Guardian information and communication logs
- Complaint or feedback submissions linked to teachers or system issues

2. Teacher and Employee Management

Teachers and staff are employees and must have:

- Profiles with joining/resignation date, employee ID, designation (head teacher, guest teacher, etc.)
- Class and subject assignments
- Attendance and leave records
- Feedback received from students/parents
- Salary, allowance, and financial tracking
- Library access and book issuance history
- Communication portal access for class and student interaction

3. Principal and Administrative Access

The principal acts as both an employee and administrator with extended privileges:

- View and manage all student, teacher, non-teaching staff, and system data
- Approve leaves, scholarships, fee waivers, and class upgrades
- Monitor financial transactions and receive alerts on suspicious activities
- Authorize infrastructure or salary-related expenses

4. Library System (Optional Integration)

Includes:

- Books categorized by subject and availability
- Issuance/return management for both staff and students
- Book return enforcement during employee resignations
- Librarian management as a non-teaching staff member

5. Attendance and Tracking System

- Attendance is integrated with the timetable and the RFID/OTP-based login system
- Logs student/staff entry/exit with timestamps
- Mobile access via registered email/phone for attendance and tracking
- Logs leave approvals for students, teachers, and staff

6. Leave Management System

- Role-based approval process:





- Student leaves: Approved by teacher or principal
- Staff leaves: Approved by the department head or principal
- Linked to attendance and salary records

7. Fee and Financial Management

- Fee payment via cash or payment gateway (e.g., Razorpay, GPay)
- Fee tracking per student ID, with status updates and receipt generation
- Principal/admin can approve scholarships or fee waivers
- Finance dashboard for:
 - Salary distribution
 - Infrastructure upgrades
 - Maintenance, repairs, and advanced handling
- Auto-alerts for unauthorized transactions

8. Examination and Report Card System

- Exams scheduled per class-subject with timetable integration
- Exam fees linked to the student fee system
- **Report card generation showing:**
 - Subject-wise marks
 - Attendance
 - Percentage calculation using credit points
 - Pass/fail status based on predefined criteria
- Reports downloadable by parents via student ID

9. Class and Timetable Management

- Class entities maintain:
 - Assigned subjects
 - Head teacher
 - Year/session
 - Linked students and timetable slots
- Timetable integrated with attendance and exam schedules

10. Authentication and Role-Based Access

- Four main login options:
 - Parent login via student ID
 - Teacher/Employee login via employee ID
 - Admin login with full dashboard control
 - Principal logs in with elevated privileges
- OTP-based secure login with full login/logout logs

11. Parent-Teacher Communication Portal

- Parents can communicate with teachers via the portal
- Can submit complaints or feedback against specific teachers
- Access student report cards, attendance, leave status, and more

12. Mobile and Cloud Access

- The entire system must be responsive and cloud-hosted
- Accessible from desktops, laptops, tablets, and smartphones
- Ensures real-time access and updates across all modules

Note:

The realization of this School Management Software demands a modular, integrated architecture covering all aspects of school functioning—from academic records to finances, library, HR, communication, and parental involvement. The system must be robust enough to handle CRUD operations across all fields, enforce access control, and support automation where





possible, ensuring data consistency, transparency, and an enhanced user experience. Continuous customization and upgrade paths must also be embedded for future-proofing the system across planning, prototyping, development, testing, and deployment stages.





Abstract

This proposal outlines the design and development of a comprehensive, modular, and scalable School Management Software (SMS) aimed at streamlining and automating all core academic, administrative, and financial operations within an educational institution. The system is envisioned as a cloud-enabled, mobile-responsive platform that ensures secure, role-based access to students, teachers, administrators, principals, and parents.

Page | 6

The proposed SMS will encompass multiple integrated modules, including but not limited to:

- Student Information System: Captures and maintains detailed student records such as personal info, registration number, class/subject allocations, guardian contact, attendance logs, and academic history.
- Attendance Management System: Offers real-time tracking through RFID tags and OTP-based authentication, with leave approval workflows and logging tied to timetables.
- Timetable Scheduling: Automates class and teacher scheduling across academic years and subjects.
- Fees and Financial Management: Integrates digital payment options (e.g., GPay, Razorpay), manages student fee records, scholarships, and administrative expenses, with financial tracking and audit notifications for anomalies.
- Examination & Report Card Module: Manages exam scheduling, marks entry, and automated report generation based on configurable grade criteria and attendance-based validations.
- Parent-Teacher Communication Portal: Allows real-time, secure messaging and feedback between parents and teachers using student IDs as the bridge.
- Library System (Optional): Tracks book inventory, allocations, due dates, and returns for students and teaching staff, especially during staff resignations.
- Leave Management: Supports digital leave requests and approval chains for students, teachers, and non-teaching staff.
- Tracking System: Enables real-time status checks on attendance, system access, financial activity, and mobile access history.

Special roles are defined for teachers, principals, and administrative heads, each with controlled privileges to view, input, or modify data. The principal, for instance, can manage class promotions, salary disbursements, and financial oversight, while teachers can handle subject-wise attendance and mark entry.

The proposed software ensures interoperability, data security, centralized management, and customizability to reflect school-specific policies and workflows. By replacing fragmented, paper-based systems, this platform will significantly enhance institutional transparency, efficiency, and parent engagement, laying the foundation for data-driven decision-making and modern digital education governance.





Introduction

The rising complexity of modern educational institutions has highlighted the urgent need for a centralized, intelligent, and scalable School Management Software (SMS) capable of addressing diverse administrative, academic, and communication needs. The school owner envisions a robust digital solution that integrates student management, staff administration, academic scheduling, fee processing, examination handling, and communication portals into a unified, secure platform with cloud and mobile accessibility. This SMS must offer role-based access tailored to students, teachers, administrators, principals, and parents, ensuring efficiency, transparency, and streamlined operations.

Page | 7

Realizing this vision, the system must comprehensively handle student data—including personal information, academic records, attendance (daily and exam-specific), guardian contacts, complaint logs, RFID tracking, exam performance, and library access—while also supporting login/logout tracking and mobile compatibility. Teachers and staff require role-based functionalities encompassing employment details, attendance, library transactions, and two-way communication with students and parents. Principals serve as privileged stakeholders overseeing student promotions, financial audits, leave approvals, and policy enforcement. A modular library system should maintain book inventories, monitor issuance, and enforce due/return policies, especially during staff transitions.

The leave management system must facilitate workflow-based approvals for students, teachers, and non-teaching staff. An integrated tracking module should monitor real-time attendance (via RFID or OTP), system usage, financial anomalies, and login activity. Secure parent-teacher communication is enabled through portals linked to student IDs, offering access to academic progress, attendance, fee status, and complaint submission. The platform's cloud-hosted, mobile-first architecture ensures anytime-anywhere accessibility, fortified with OTP-based multi-role authentication.

Attendance logs must synchronize with timetables and examination records, while fee management should support both offline and digital payments (e.g., GPay, Razorpay), scholarship handling, and modular linkage with the central finance system. Examination modules must manage subject-wise scheduling, mark entry (restricted to teachers), and automated report card generation, with only administrators and principals authorized to promote students. Classes are managed with head teacher assignments, subject mappings, and academic year progression. Role-based dashboards and strict login/logout management further safeguard system integrity.

Altogether, this software's architecture must reflect scalability, interoperability, and customization across its frontend, backend, middleware, and cloud layers. By addressing these multifaceted requirements, the proposed SMS directly resolves the institutional challenge of fragmented, paper-based systems, paving the way for data-driven decision-making, enhanced accountability, and seamless digital governance in educational environments.





Benefits & Drawbacks

Benefits:

1. **Centralized Management:** Integrates academic, administrative, financial, and communication processes into one cohesive system.
2. **Role-Based Access:** Ensures security and data integrity by restricting functionality based on user roles (students, teachers, admin, etc.).
3. **Real-Time Tracking:** RFID and OTP-based attendance provide real-time visibility of student and staff presence.
4. **Cloud & Mobile Accessibility:** Enables anytime-anywhere access across devices, improving stakeholder engagement.
5. **Automated Report Cards & Exams:** Reduces manual workload and errors through automated mark calculations and grade reports.
6. **Digital Payments & Finance Management:** Offers secure, traceable fee transactions and tracks expenses and anomalies.
7. **Parent-Teacher Communication:** Enhances transparency and accountability with secure messaging, feedback, and updates.
8. **Leave and Complaint Management:** Streamlines approvals and logging with audit trails for transparency.
9. **Library System Integration:** Facilitates book tracking, issuance, and return management, especially during staff transitions.
10. **Scalability and Customization:** Supports school-specific policies, workflows, and future expansion needs.
11. **Promotes Data-Driven Decisions:** Consolidated data insights help administrators make informed decisions.

Drawbacks:

1. **High Initial Development Cost:** Building a full-scale system with all modules may require significant upfront investment.
2. **Complex Implementation:** Integration across diverse modules (attendance, finance, academics) requires meticulous planning and skilled personnel.
3. **Training Requirement:** Teachers, staff, and parents may need training to effectively use the system.
4. **Dependence on Internet Connectivity:** Cloud-hosted and mobile-responsive features require reliable internet access, which may be limited in rural areas.
5. **Maintenance Overhead:** Regular updates, bug fixes, and support will require dedicated technical resources.
6. **Privacy & Security Concerns:** Handling sensitive data (student records, salaries, payment info) demands strong encryption and compliance with data protection laws.
7. **Device Compatibility Issues:** Ensuring seamless performance across all screen sizes and operating systems can be challenging.
8. **Resistance to Change:** Stakeholders accustomed to manual systems may initially resist digital adoption.
9. **Bug or System Failure Risks:** A critical failure could disrupt multiple school operations unless proper backups and recovery systems are in place.





Feasibility Study

What is a Feasibility Study?

A feasibility study is an evaluation and analysis of a proposed project or system to determine if it is technically feasible, economically viable, legally permissible, and operationally achievable within the constraints of time, resources, and technology.

— IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

Page | 9

When is Feasibility Study Done in SDLC?

The **Feasibility Study** is performed during the **first or pre-requirement phase** of the **SDLC**, often referred to as the **Preliminary Investigation** or **Planning Phase**.

This is **before requirement gathering and analysis** begins, and helps decide whether the project should move forward.

Types Of Feasibility Assessed:

Type	Description
Technical Feasibility	Can the system be developed using existing hardware, software, and team skills?
Economic Feasibility	Is the project cost-effective? Do benefits outweigh the investment?
Operational Feasibility	Will the solution work in the real-world environment? Are users willing to adopt it?
Legal Feasibility	Does the system comply with laws, policies, and regulations (e.g., data privacy, licensing)?
Schedule Feasibility	Can the project be completed within the required timeframe and deadlines?

1. Technical Feasibility

Objective: Determine whether the system can be developed with current technology, tools, and resources.

- **Platform:** Web and Mobile (Responsive UI)
- **Technology Stack:**
 - Backend: Java (Spring Boot)
 - Frontend: HTML5, CSS3, JavaScript, Bootstrap
 - Database: MySQL
 - Cloud Deployment: AWS / Vercel
 - Authentication: OTP, Role-based access
 - Hardware: RFID readers, mobile phones, web devices
- **Assessment:**
 - Technologies are mature, widely used, and supported.
 - Developers are familiar with the stack (Java, MySQL).
 - Cloud hosting ensures scalability.
 - Integration with RFID and OTP systems is feasible.

2. Economic Feasibility (Cost-Benefit Analysis)

Objective: Evaluate if benefits outweigh the estimated costs.

- **Estimated Costs:**
 - Development Team: ₹5–7 Lakhs (initial)
 - Infrastructure/Cloud: ₹15,000–25,000 annually
 - RFID and Hardware Devices: ₹50,000 one-time
 - Maintenance & Updates: ₹10,000/month
- **Expected Benefits:**
 - Elimination of paper-based workflows (₹1–2 Lakhs/year saved)
 - Improved productivity and reduced manual errors





- Faster communication with parents and stakeholders
- Improved financial and academic transparency
- Scalable to more branches in future

3. Operational Feasibility

Objective: Check if the system will function effectively in the actual environment.

- **Stakeholders' Readiness:**
 - Admins and teachers are willing to use digital platforms
 - Parents are familiar with mobile apps and online portals
 - Librarians, finance, and HR departments are aligned with the change
- **User Training & Adoption:**
 - Minimal training required due to intuitive design
 - Support documentation and help desk planned

Page | 10

4. Legal Feasibility

Objective: Ensure compliance with laws and regulations.

- **Compliance Considerations:**
 - Student data privacy and protection (align with India's Digital Personal Data Protection Act, 2023)
 - Financial transaction compliance (integration with UPI, Razorpay, GPay is permissible)
 - Proper licenses used for third-party tools, open-source components

5. Schedule Feasibility

Objective: Can the system be built within the required timeframe?

- **Estimated Timeline:**
 - Requirements Analysis: 2 weeks
 - UI/UX Prototyping: 1 week
 - Development: 12–16 weeks
 - Testing: 2 weeks
 - Deployment: 1 week
 - Training & Go-Live: 1 week
- **Total Duration:** ~5 months
- **Project Timeline Considerations:**
 - Agile/iterative sprints for modular releases
 - Prioritized core modules (Student Info, Attendance, Fees) in early versions

Trial-Approach to solve the problem statement

The school owner envisions a robust, centralized **School Management Software (SMS)** solution capable of managing multiple interconnected systems, each handling distinct aspects of school operations. The core objective is to digitize and integrate student management, staff administration, academic scheduling, fee and financial processing, library access, examination handling, and communication portals into a unified platform with secure, role-based access. Based on the key features mentioned, the following system-level realizations and functional expectations are identified:

1. Student Management

The system must store comprehensive student data including:

- Personal details (name, roll number, registration number, blood group, date of admission, pass-out year)
- Academic info (class, subjects, optional subjects, head teacher assignment)
- Attendance status (daily, class-wise, exam-specific)
- Contact and guardian information





- Complaint/feedback submission
- Leave status (with approval workflow)
- Library access logs
- RFID/mobility tracking
- Exam performance and report cards
- Login/logout history
- Mobile access compatibility for students and guardians

2. Teacher and Staff Management

Teachers, as employees, should have:

- Assigned roles (head teacher, guest teacher)
- Employment details (joining/resignation dates, salary, allowances, leave records)
- Attendance logs and complaint feedback from students/parents
- Library book allocations (with return verification during resignation)
- Communication privileges with students and parents via portal

Non-teaching staff (e.g., librarian) also require:

- Role-specific details
- Leave and attendance tracking
- Interaction limited to relevant systems (e.g., library)

3. Principal and Administrative Stakeholders

The principal serves as:

- An employee with administrative access
- A stakeholder who can view and manage student/staff data
- A decision-maker in approvals (class promotions, scholarships, financial anomalies, salary disbursements, etc.)

4. Library System (Optional)

The optional library system should:

- Track books by subject and availability
- Manage allocations to students and staff
- Enforce return rules based on resignation or due dates
- Enable the librarian to manage book issuance and logging

5. Leave Management System

A workflow-based system for:

- Teachers to raise leave requests for approval by principals or department heads
- Students' leaves to be approved by their assigned teachers or principal
- Non-teaching staff to follow a similar leave workflow

6. Tracking and Monitoring

An integrated tracking system should:

- Record attendance through RFID or OTP logins
- Monitor user login/logout activities on web and mobile portals
- Track financial activities, flagging unauthorized transactions
- Maintain access logs via phone number or email-based login
- Provide real-time monitoring for all stakeholders

7. Parent-Teacher Communication Portal

A secure portal must enable:

- Parents/guardians to log in via student ID
- Communication with assigned teachers





- Raising complaints or feedback against specific teachers
- Access to student report cards, attendance, fee status, and leave history

8. Mobile and Cloud Access

The platform should be:

- Mobile-responsive and compatible with devices of varying screen sizes
- Cloud-hosted to allow anytime-anywhere access
- Secure with OTP-based multi-role login (Parent, Admin, Teacher, Principal)

9. Attendance Management

Attendance must:

- Be recorded via RFID or digital methods
- Be linked to timetables and leave policies
- Require teacher/principal approval
- Be integrated with examination systems to validate student presence

10. Fees and Financial Management

- Students pay fees via cash or online (Razorpay, GPay, etc.)
- Fee status is tracked against student IDs
- Scholarships or waivers can be approved by administrators
- Finance module handles infrastructure upgrades, salaries, advances, repairs, etc.
- Financial transactions are monitored; anomalies flagged for board review
- Student fee payments are indirectly tied to the finance system to allow modular flexibility

11. Examination and Report Card System

- Subjects carry predefined credit points (e.g., 100 per subject)
- Exams are scheduled by class and subject with timetables and fees
- Report cards combine marks, attendance, and subject credits
- A pass/fail decision is made using percentage formula:
 - $\text{Percentage} = (\text{Total Marks Obtained} / (\text{Total Subjects} \times \text{Credit Points})) \times 100$
- Only admin and principal can promote students
- Teachers can enter but not edit marks
- Parents can view/download report cards via portal

12. Class Management

- Each class has multiple students and subjects
- Each class is assigned a head teacher
- Academic year mapping and upgrades are handled by privileged roles

13. Login and Access Management

- Login/logout tracked across all users
- Role-specific dashboards:
 - Parents (via student ID)
 - Teachers/employees (via employee ID)
 - Admins and Principals with elevated privileges
- OTP-based authentication ensures session security

Conclusion

The system's design and architecture should cater to the requirements and workflows detailed above. Customization and scalability must be built into the frontend, backend, middleware, and cloud network layers. Each component—planning, prototyping, development, testing, and deployment—must be executed iteratively to ensure reliability, security, and future





extensibility. The realization of these features collectively addresses the problem statement while laying a strong foundation for digital transformation in school management.

System Requirement Analysis

1. **Objective:** To design and develop a secure, cloud-hosted, mobile-compatible School Management Software that digitizes and centralizes the management of student, staff, academic, financial, and administrative operations within an educational institution.

Page | 13

2. **Stake Holder Requirement:**

Roles	Responsibilities
School Owner	Monitor all systems, approve policies, view financials
Principal	Manage students/staff, approve leaves, view analytics
Teachers	Enter attendance and marks, communicate with parents
Librarian	Issue and track books, maintain book records
Students	View schedule, attendance, results, communicate
Parents	Track student progress, raise queries/complaints

3. **Functional Requirements:**

3.1 Student Management

- Add/edit/delete student profile with academic and personal details
- Record attendance (daily/class-wise/RFID-based)
- Upload and view report cards
- Apply for leave; track status
- Access library records, complaints, login/logout history

3.2 Staff and Teacher Management

- Manage employee roles and leave records
- View attendance logs and complaints
- Assign teachers to classes and subjects

3.3 Class & Timetable Management

- Create/modify class sections and academic schedules
- Map subjects to classes
- Assign head teachers

3.4 Fees & Financial Management

- Process payments via cash/online (Razorpay, GPay, etc.)
- Track fee dues, fines, scholarships
- Generate financial reports and dashboards

3.5 Library Management (Optional Module)

- Track book availability and category-wise mapping
- Issue/return books with deadlines
- Block clearance upon staff/student resignation

3.6 Leave and Complaint Management

- Submit and approve leave requests (students/staff)
- Lodge complaints or feedback
- Approval workflows by authority level

3.7 Parent-Teacher Communication

- Parents log in via student ID
- Two-way messaging
- View child's reports, fees, attendance





3.8 Authentication & Role-Based Access

- OTP/email/phone-based login
- Dashboards vary by role (Admin, Student, Teacher, Parent)
- Logging of all login/logout sessions

4. Non- Functional Requirements:

4.1 Performance Requirements

- Support 1000+ concurrent users
- Response time under 2 seconds for major transactions

4.2 Security Requirements

- OTP-based multi-factor authentication
- Encrypted data storage (especially for financials)
- Role-based data access and activity logging

4.3 Usability Requirements

- Intuitive web/mobile interface
- Responsive design across devices

4.4 Availability & Reliability

- 99.9% uptime with cloud hosting
- Daily automated backups
- Failover support for critical operations

4.5 Maintainability & Scalability

- Modular microservices for easy upgrades
- Scalable cloud-based infrastructure

5. Assumptions and Constraints:

- The school will provide existing student/staff data for migration
- Internet connection is available at all user endpoints
- RFID system and payment gateway APIs are integrated

6. Expected Deliverables

- Web-based and mobile-accessible SMS platform
- Admin dashboard with analytics and user management
- Role-based login systems
- Complete CRUD modules for all key entities (students, staff, classes, exams, etc.)
- Documentation and training manual





Technology Stack Overview

To ensure scalability, maintainability, and ease of integration, the School Management Software (SMS) is designed using a robust and modern technology stack. The stack is divided into the following major categories:

Front-End Technologies   	<p>These technologies are used to build the User Interface (UI) that is accessible through the web and mobile browsers:</p> <ul style="list-style-type: none"> • Thymeleaf: A server-side Java template engine for rendering HTML views in Spring Boot. • Bootstrap: A responsive CSS framework for fast and consistent web UI design. • React (Optional): A client-side JavaScript library for building dynamic, component-based interfaces, suitable for mobile-responsiveness and single-page applications (SPA).
Backend Technologies  	<p>These technologies form the core of the system's business logic, service orchestration, and data processing:</p> <ul style="list-style-type: none"> • Java (Spring Boot): The primary backend framework for creating robust and secure RESTful web services. • JPA (Java Persistence API)/Hibernate: ORM tools for managing database operations and reducing boilerplate SQL.
Database 	<p>Persistent storage of all application data including students, staff, attendance, finance, and library records:</p> <ul style="list-style-type: none"> • MySQL: A reliable, open-source relational database that supports ACID transactions, indexing, normalization, and data integrity.
APIs and Integration Layer 	<ul style="list-style-type: none"> • RESTful APIs (via Spring Controllers): All internal and external communication is conducted via REST APIs, enabling seamless integration with mobile apps, third-party services, and web front-ends.
DevOps and Automation   	<p>To streamline development, testing, and deployment:</p> <ul style="list-style-type: none"> • GitHub Actions: Automate builds, tests, and deployment workflows. • Jenkins: Optional CI server for more advanced continuous integration pipelines. • Docker: Containerization of the backend services to simplify deployment and maintain consistency across environments.
Cloud Hosting and Deployment   	<p>These platforms are chosen for affordability (including generous free tiers), scalability, and ease of deployment:</p> <ul style="list-style-type: none"> • Railway: For quick deployment of backend and databases. • Render: Full-stack application hosting with automatic Git deployment. • Firebase (for Mobile/Authentication/Push Notifications): Firebase can be used for handling mobile access, push notifications, and optional OTP-based authentication.





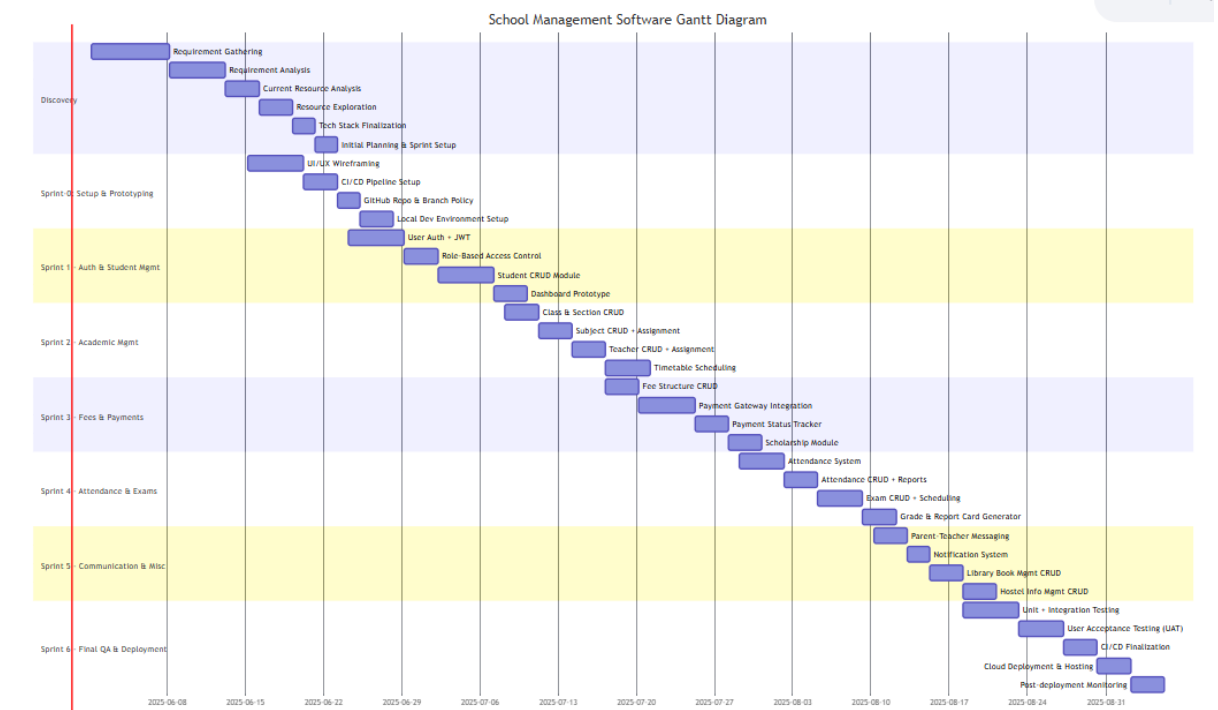
Charts & Road Maps

System Architecture Layers:

Frontend Clients Web Portal (Admin, Teachers, Parents) Mobile App (Students, Parents, Staff)
API Gateway / REST Layer Spring Boot REST APIs Authentication & Authorization (JWT, OAuth2)
Application Logic Student Management Module Timetable Scheduler Fee & Financial Handler Attendance Tracker Report Card Generator Leave/Complaint Handler
Database Layer MySQL / PostgreSQL (Normalized Schema) Redis (Optional: for cache/session/token store)
Infrastructure / Deployment Cloud: Vercel / AWS EC2 / GCP CI/CD: GitHub Actions / Jenkins Security: SSL, Firewalls, OAuth2, Role-based ACLs

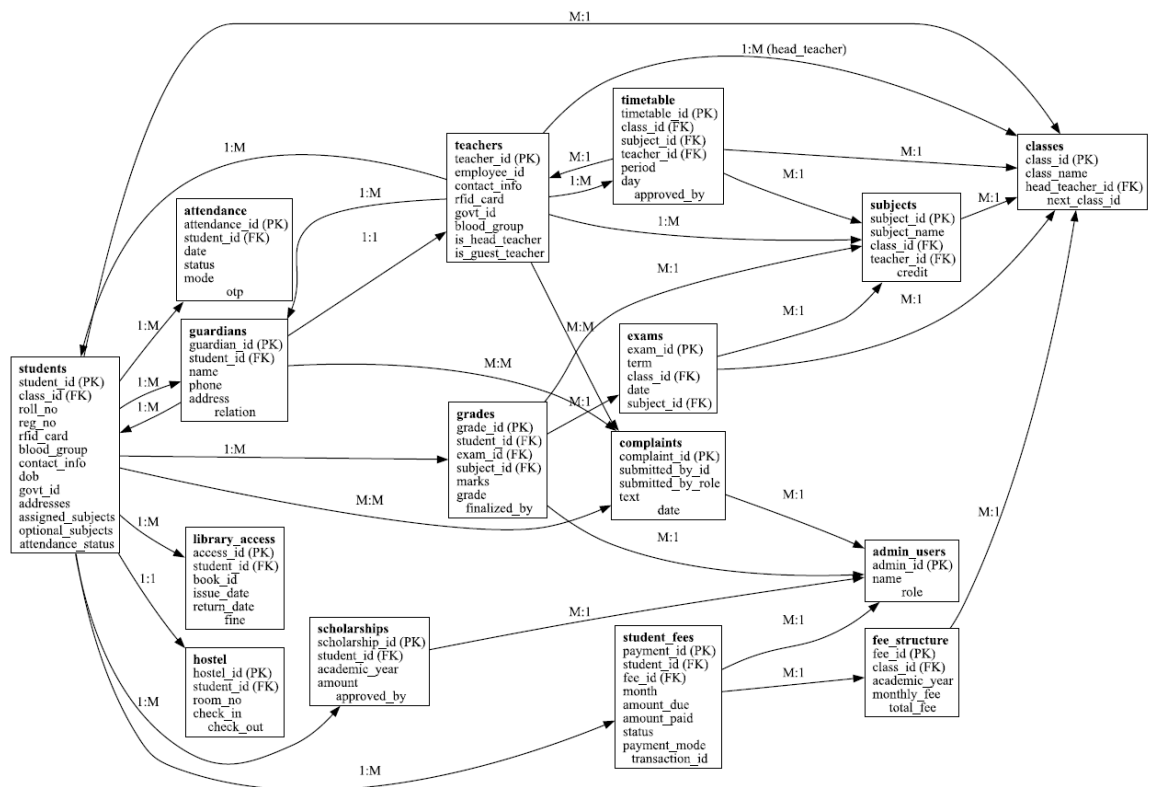
Page | 16

Gantt Chart:



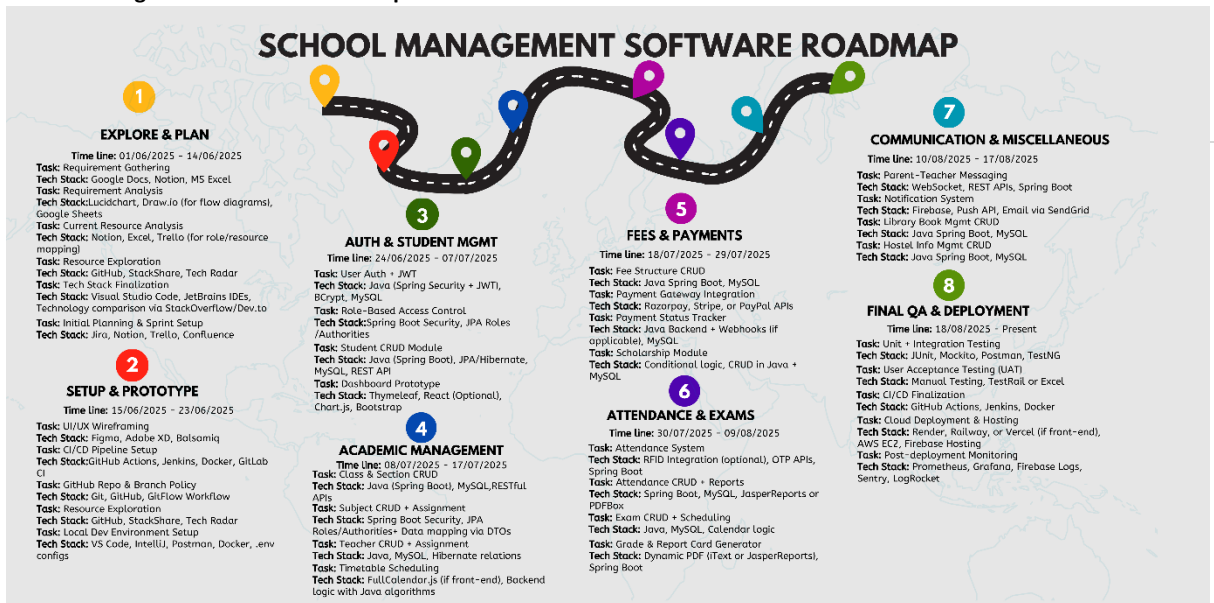


Entity-Relationship Diagram:





School Management Software Roadmap:



18





Future Scope:

The future scope of the School Management Software (SMS) includes integrating AI-powered predictive analytics to track student performance and optimize resources; leveraging biometric and IoT technologies for smart attendance and classroom automation; embedding Learning Management System (LMS) features for online learning, assessments, and parental oversight; and enabling GPS-based transport tracking. It will support multilingual access and inclusive design, automated cloud backups, real-time notifications via SMS, email, and app alerts, and gamified learning with badges and certificates. Administrators will benefit from dynamic role management, custom workflows, enhanced dashboards, and advanced analytics tools like Power BI. Seamless API integrations with payment gateways and government portals, a smart library system with barcode support, and visual documentation (DFD, CFD, UML) will bolster usability. Future versions will support multi-branch school management, blockchain-based digital certificates, comprehensive asset tracking, AI chatbots, voice command support, and enhanced cybersecurity with MFA and secure authentication protocols, ensuring a scalable, intelligent, and inclusive education ecosystem.

Page | 19

Conclusion

The proposed School Management Software (SMS) offers a comprehensive and scalable solution tailored to meet the evolving administrative and academic needs of educational institutions. Developed using a robust technology stack—comprising Java (Spring Boot), MySQL, RESTful APIs, and optionally enhanced by modern front-end frameworks such as Thymeleaf or React—the system ensures high performance, security, and usability. It consolidates key school operations including student information management, attendance tracking, fee and examination processing, timetable scheduling, library services, and communication into a centralized platform. The system's modular architecture allows for seamless customization and expansion, making it suitable for institutions of varying sizes, from single campuses to multi-branch organizations. Key benefits include streamlined workflows, real-time data accessibility, reduced administrative overhead, and improved stakeholder engagement. Its versatility enables cross-platform access, mobile responsiveness, and integration with third-party services. Looking forward, the future scope encompasses intelligent enhancements such as AI-based analytics, biometric and IoT integration, blockchain for certificate verification, dynamic role management, and Learning Management System (LMS) support. These additions will further strengthen its capacity to support smart education delivery, data-driven decision-making, and operational transparency. In essence, this School Management Software lays a strong digital foundation for educational excellence, fostering a more organized, informed, and responsive academic ecosystem.

