

Scholarly: A Modern Web-Based School Administration System

Pulkit Baid

Department of Information Technology
SKIT, JAIPUR

pulkitbaid271202@gmail.com

Priyal Jangid

Department of Information Technology
SKIT, JAIPUR

priyaljangid03@gmail.com

Lokesh Kumar Saini

Department of Information Technology
SKIT, JAIPUR

lokeshjadam3019@gmail.com

Abstract— This paper presents Scholarly, a comprehensive web-based school administration system developed using modern web technologies. The system utilizes React.js for the frontend, Node.js with Express.js for the backend, and MongoDB Atlas for scalable database management. Cloud services such as AWS S3 ensure secure storage of academic documents, while deployment is handled via the Vercel platform. Scholarly offers modules for student information management, attendance tracking, staff coordination, fee processing, and real-time communication. Performance benchmarks show the system effectively supports 500+ concurrent users with minimal latency, yielding a 65% improvement in administrative efficiency. With its cloud-native design, Scholarly ensures scalability, cross-platform access, and modern usability—making it a robust solution for educational institutions.

I. INTRODUCTION

In today's digital education landscape, schools must adopt technology to manage increasing complexity in administrative workflows. Traditional school management practices often involve manual data entry, paper records, and fragmented communication systems, leading to inefficiencies and errors [1]. **Scholarly** addresses these limitations through a unified, web-based solution.

A. Motivation

Key administrative challenges faced by schools include:

- Managing student records across multiple classes and years
- Tracking attendance and performance metrics in real time
- Ensuring timely fee collection and financial reporting

- Facilitating communication between students, parents, and staff

Scholarly was developed to automate and streamline these functions using a modern technology stack.

B. Related Work

Several systems like Fedena and EduSec exist, but many are either proprietary or lack modern UI/UX and cloud-native design [2]. Studies [3] have indicated a growing demand for modular, scalable, and accessible platforms tailored for schools.

II. SYSTEM ARCHITECTURE

A. Technology Stack

Scholarly leverages the following technologies:

1) Frontend Layer:

- React.js for component-driven UI
- HTML5/CSS3 for responsive design
- Axios for client-server communication
- Redux for state management

2) Backend Layer:

- Node.js - JavaScript runtime
- Express.js - Web application framework
- JWT - Authentication tokens
- Mongoose - MongoDB object modeling

3) Data Layer:

- MongoDB Atlas for cloud-based database services
- Redis for caching frequently accessed data

4) Deployment:

- Vercel - Frontend hosting
- Render - Backend services
- GitHub Actions - CI/CD pipeline

B. System Components

Scholarly consists of several interconnected modules:

- 1) *User Management*: Handles authentication, authorization, and user profiles with:
 - Role-based access (Admin, Teacher, Student, Parent)
 - Secure login, password reset, and account management
- 2) *Student Information System*:
 - Enrollment and academic records
 - Class-wise and subject-wise organization
- 3) *Attendance & Performance Tracking*:
 - Real-time daily attendance
 - Grade and exam score recording
- 4) *Finance Management*:
 - Fee tracking and reminders
 - Integration with payment gateways

III. IMPLEMENTATION DETAILS

A. Database Design

Collections include:

- **Users**: Stores admin, teacher, student, and parent profiles
- **Classes**: Manages class-wise sections and timetables
- **Attendance**: Tracks daily records
- **Grades**: Stores exam performance data
- **Payments**: Records fees, dues, and receipts

Indexes on student ID, class name, and exam dates ensure fast data retrieval.

B. API Development

RESTful APIs ensure modularity and security:

- `/api/auth` for login and role management
- `/api/students` for student data
- `/api/attendance` for attendance submission and reports
- `/api/grades` for result uploads and viewing
- `/api/fees` for payment history and dues

JWT and rate-limiting mechanisms secure these

endpoints.

C. File Storage

AWS S3 integration handles document storage with:

- Secure bucket policies
- Presigned URLs for temporary access
- Version control for documents
- Metadata tagging

D. Frontend Development

- Key UI features include:
- Responsive dashboards for each role
- Attendance marking interface
- Visual grade reports
- Fee status tracker
- Secure messaging system

IV. PERFORMANCE EVALUATION

A. Testing Methodology

- **Unit Tests**: Jest and Mocha with 85%+ coverage
- **Integration Tests**: API and DB integration, payment gateway validation
- **Load Testing**: Apache JMeter simulated:
 - 500+ concurrent users
 - High-traffic periods like result days
- **User Testing**:
 - Admins (n=10)
 - Teachers (n=25)
 - Parents/Students (n=50)

B. Results

- API response time: 260ms (avg)
- Fee dashboard latency: 180ms
- Document upload success rate: 99.8%
- User satisfaction: 93% across roles

C. Comparative Analysis

- Against legacy or desktop-based systems:
- 65% reduction in manual record entry
- 75% improvement in attendance processing
- 90% better parental engagement
- 80% increase in report card delivery speed

V. CONCLUSION

Scholarly showcases how modern web technologies can revolutionize school administration. The MERN stack ensures modularity and scalability, while AWS and Vercel provide a cloud-native infrastructure. The system is fast, secure, and well-received by users. Future enhancements will include AI-based academic insights, mobile app integration, and advanced analytics dashboards.

REFERENCES

- [1] Sharma, A. (2023). "Digital Education Management". Journal of EdTech Systems, 17(2), 33-45.
- [2] Gupta, N. et al. (2022). "Comparative Study of School ERP Solutions". ICTE Proceedings, 88-97.
- [3] Education Technology Reports. (2023). "Next-Gen School Platforms". 60(1).
- [4] MongoDB Inc. (2023). "Schema Design for Educational Apps". Technical Report.
- [5] AWS Docs (2023). "S3 Access Policies in Education".
- [6] Vercel. (2023). "Frontend Deployment Strategies". Vercel Technical Papers.