

# **Software Requirements Specification**

for

## **Online School Management System**

**Version 1.4 approved**

**Prepared by Sumit Sharma**

**TMSL\_CSE**

**18.11.2024**

# Table of Contents

(to be generated automatically)

## Table of Contents

1	Introduction .....	3
1.1	Purpose/Objective .....	3
1.2	Document Conventions (Definition, Acronyms, Abbreviations) .....	3
1.3	Scope .....	3
1.4	References.....	4
2	History/Background Study (Sources of Domain Knowledge).....	4
2.1	Technical Literature .....	4
2.2	Existing Applications .....	4
2.3	Customer Survey .....	4
2.4	Expert Advice .....	10
2.5	Current/Future requirements .....	10
3	Overall Description.....	10
3.1	Product Functions .....	10
3.2	Functional Requirements .....	11
3.3	Non-Functional Requirements .....	14
3.4	User Characteristics .....	15
3.5	Design & Implementation Constrains .....	15
3.6	Assumptions & Dependencies .....	15
4	Interface Requirements.....	17
4.1	User Interfaces .....	17
4.2	Hardware Interfaces.....	17
4.3	Software Interfaces .....	17
4.4	Communication Interfaces .....	17
5	Conclusion.....	17

# 1 Introduction

## 1.1 Purpose/Objective

The Online School Management System (OSMS) aims to simplify and automate school operations such as admissions, academic scheduling, notice management, attendance tracking, and communication. It enhances efficiency and collaboration among administrators, teachers, students.

## 1.2 Document Conventions (Definition, Acronyms, Abbreviations)

### Definitions

- Online School Management System (OSMS): A digital platform for managing academic, administrative, and communication processes within an educational institution.
- Admin: User responsible for managing core functions such as creating classes, adding users (teachers, students), and configuring system settings.
- User Roles: Categories of system users, including Admin, Teacher, and Student, each with specific access and functionality.

### Acronyms

OSMS: Online School Management System

DB: Database

UI: User Interface

API: Application Programming Interface

### Abbreviations

Attd.: Attendance

Rec.: Record

Notif.: Notifications

Pwd.: Password

## 1.3 Scope

The OSMS will serve as a centralized platform to manage administrative, academic, and communication tasks efficiently. It is intended for schools of

varying sizes and will include modules for resource management, online classes, and report generation.

#### 1.4 References

- Academic journals on educational technologies.
- Case studies of existing school management systems.
- Technical manuals for web-based application development.

## 2 History/Background Study (Sources of Domain Knowledge)

### 2.1 Technical Literature

Research papers and books on modern educational management systems and cloud-based solutions.

### 2.2 Existing Applications

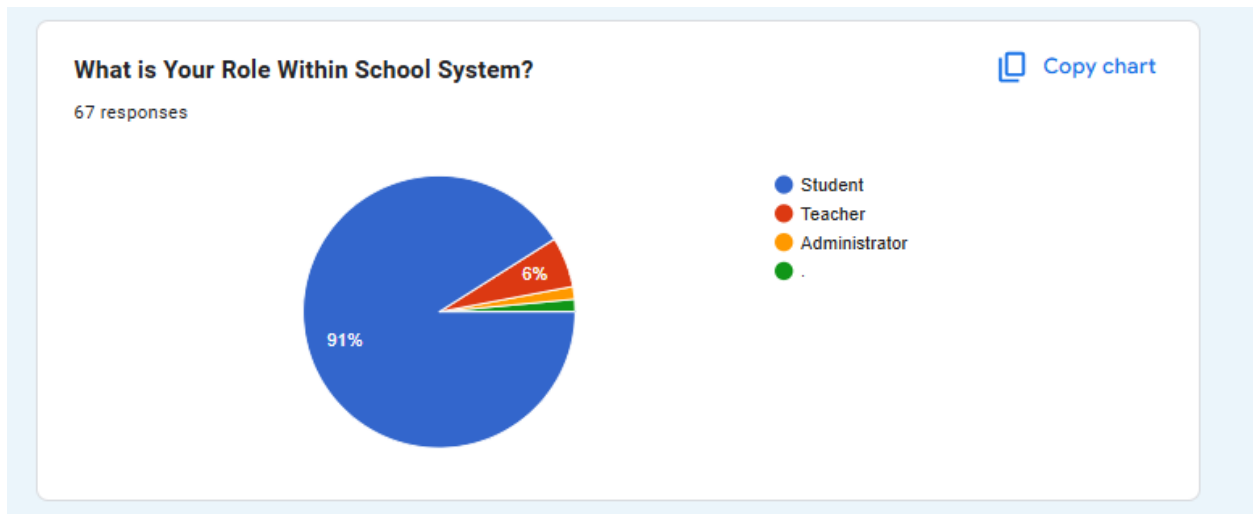
- [School Management System\(SMS\)](#)
- [eSkooly - Free Online School Management System](#)
- [Custom Software for Educational institutes| Zoho Creator](#)

### 2.3 Customer Survey

Feedback from school administrators, teachers, and students on existing challenges and desired features.

[Online School Management System\(Feedback Form\)](#)

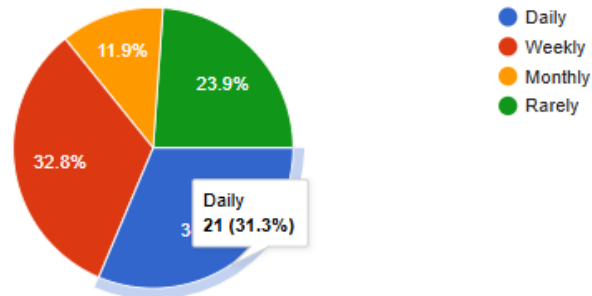
[Survey Report\(Responses\)](#)



### How frequently do you interact with the online school management system?

[Copy chart](#)

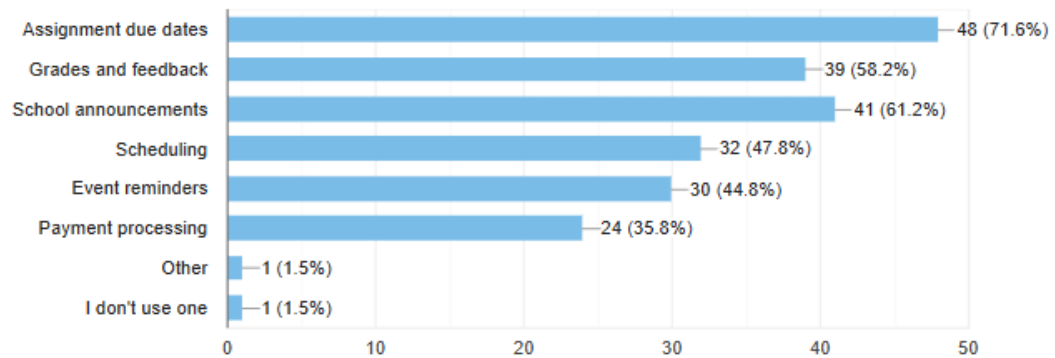
67 responses



### Which types of notifications would you like to receive?

[Copy chart](#)

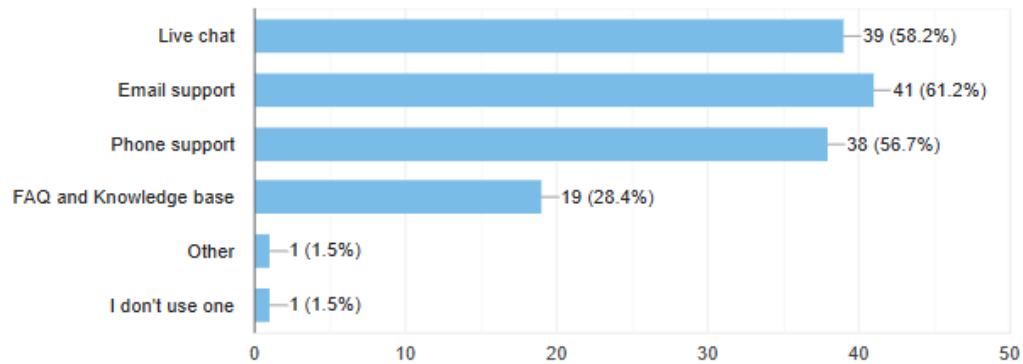
67 responses



### What support channels do you prefer when encountering issues with the system?

 [Copy chart](#)

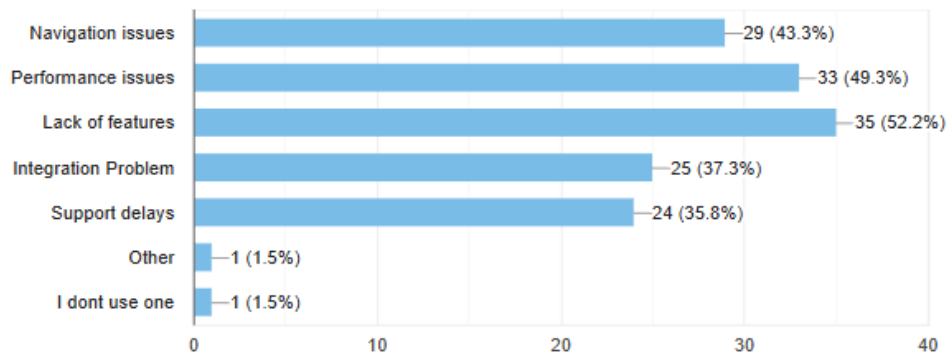
67 responses



### What challenges do you face with the current online school management system?

 [Copy chart](#)

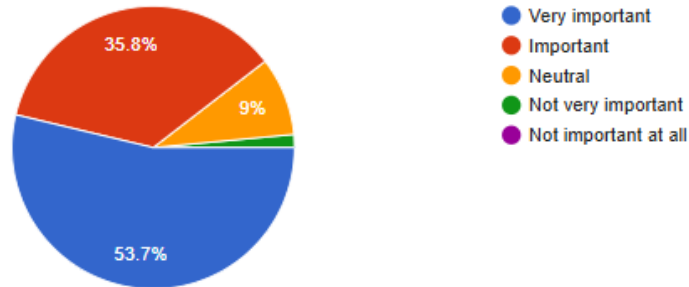
67 responses



### How important is real-time access to grades and assignments?

[Copy chart](#)

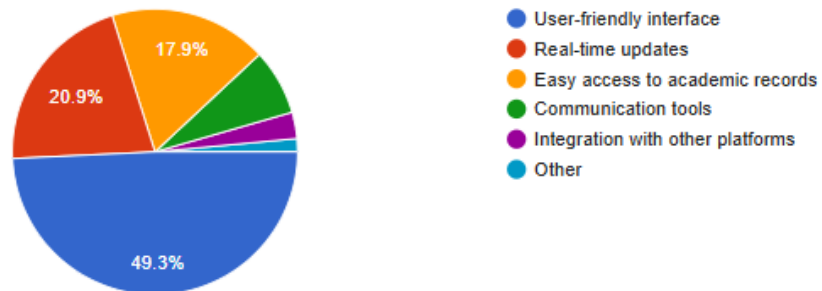
67 responses



### Which features are most important to you in the online school management system?

[Copy chart](#)

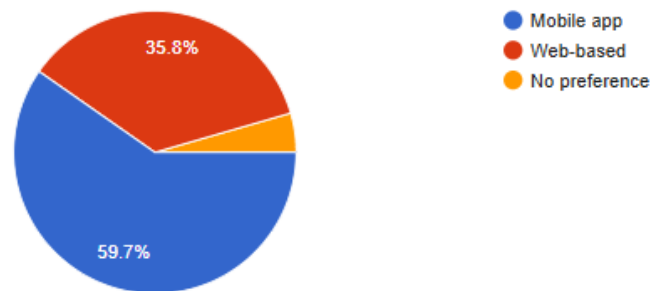
67 responses



### Would you prefer a mobile app or a web-based platform for managing school-related activities?

 [Copy chart](#)

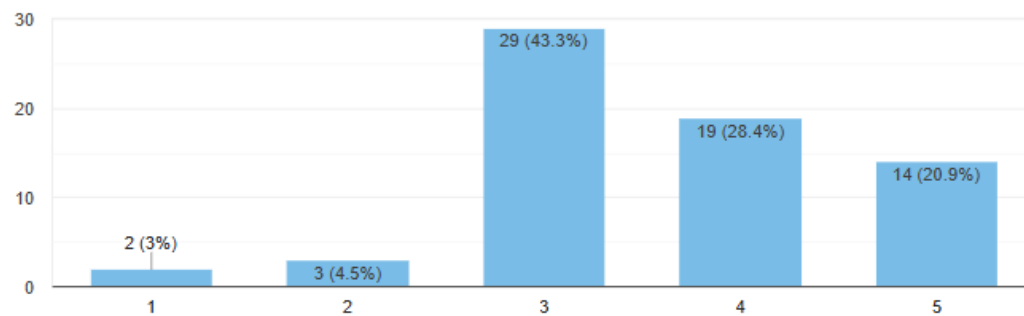
67 responses



### How easy is it to navigate the online school management system?

 [Copy chart](#)

67 responses

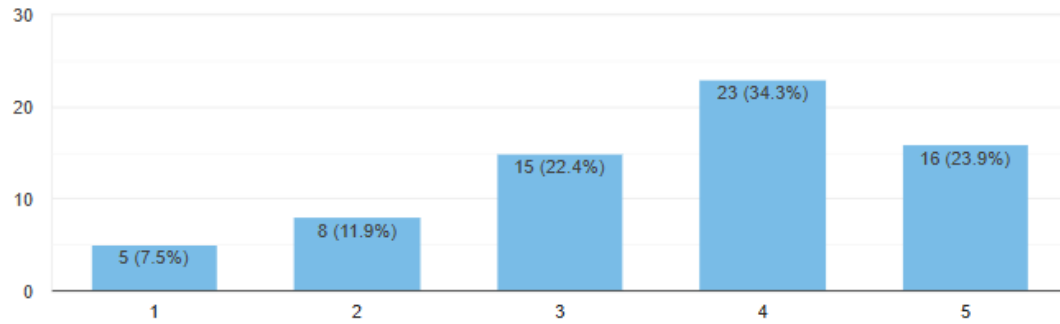




### How concerned are you about data security and privacy in an online school management system?

 [Copy chart](#)

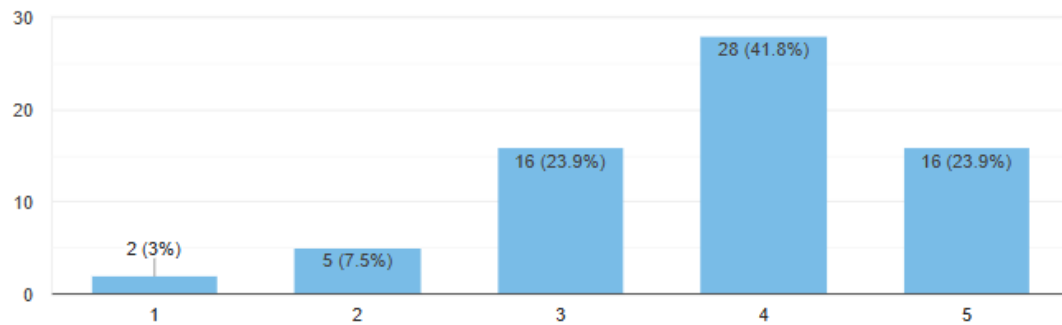
67 responses

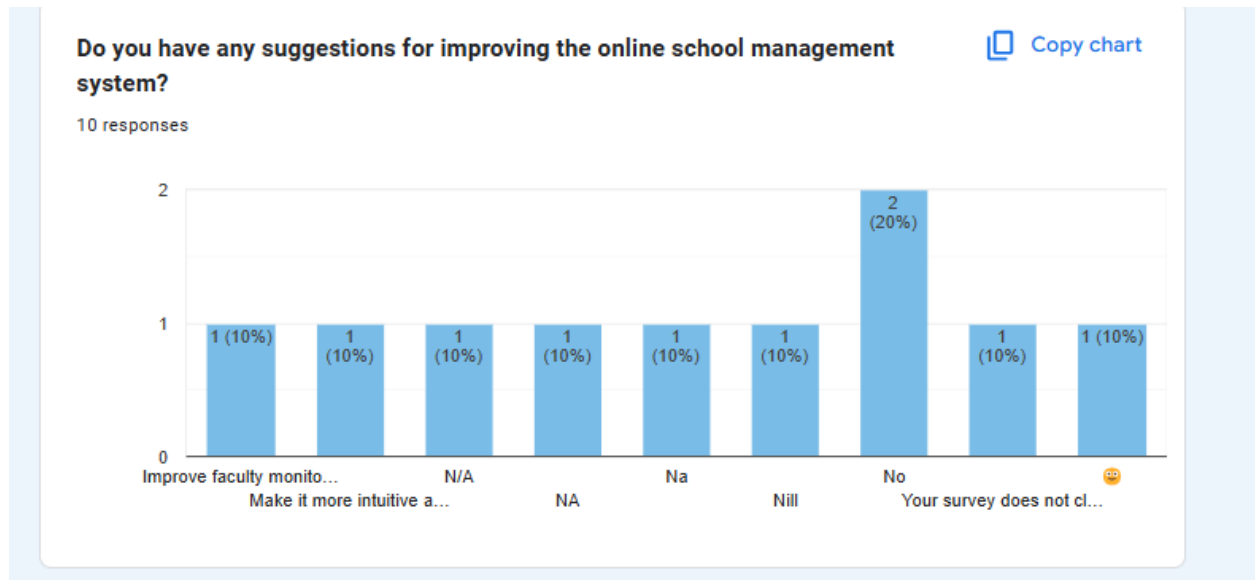


### How would you rate the overall experience?

 [Copy chart](#)

67 responses





## 2.4 Expert Advice

Input from educational technology consultants and IT specialists in school management software.

## 2.5 Current/Future requirements

- Current: Attendance management, notice management
- Future: Integration with AI for personalized learning and analytics, mobile apps for wider accessibility, fee management.

# 3 Overall Description

## 3.1 Product Functions

### 3.1.1 Hardware Requirement

- Server Requirements:
  - Processor: Minimum 2 GHz multi-core processor.
  - RAM: Minimum 8 GB.
  - Storage: Minimum 500 GB SSD.
- Client Requirements:
  - Device: Desktop, laptop, tablet, or smartphone.
  - Screen Resolution: 1280x720 or higher.

### 3.1.2 Software Requirement

- Server-side Software:
  - Operating System: Linux/Windows Server.
  - Database: MySQL or PostgreSQL.
  - Web Server: Apache or Nginx.
  - Programming Languages: PHP, Python, or Java.
- Client-side Software:

- Browser: Chrome, Firefox, or Edge (latest versions).
- Operating System: Windows, macOS, Android, or iOS.

## 3.2 Functional Requirements

### 3.2.1 Admin Registration/Login

- Description: Allows the administrator to log in or register a new account.
- Input: Admin username, password, and additional registration details (for new accounts).
- Output: Login or registration successful.
- Error:
  - Incorrect username or password.
  - Username already exists (during registration).

### 3.2.2 Create Class

- Description: Admin can create new classes for students and assign teachers.
- Input: Class name, teacher assignment, and optional schedule.
- Output: Class creation confirmation.
- Error:
  - Duplicate class name.
  - Missing required fields.

### 3.2.3 Add Student

- Description: Admin can register new students and assign them to classes.
- Input: Student details (name, age, class, contact info, etc.).
- Output: Student added successfully.
- Error:
  - Missing mandatory details.
  - Duplicate student record.

### 3.2.4 Add Subject

- Description: Admin can create new subjects and link them to classes.
- Input: Subject name, class, and teacher assignment.
- Output: Subject created successfully.
- Error:
  - Missing mandatory fields.
  - Subject name already exists.

### 3.2.5 Add Teacher

- Description: Admin can register new teachers and assign them to subjects and classes.
- Input: Teacher details (name, contact info, qualifications, etc.).
- Output: Teacher registration confirmation.
- Error:
  - Missing required details.
  - Duplicate teacher entry.

#### 3.2.6 Add Notice

- Description: Admin can create notices to be shared with students or teachers.
- Input: Notice title, content, and recipients.
- Output: Notice created and sent successfully.
- Error:
  - Empty notice fields.
  - System failure during notice saving.

#### 3.2.7 Teacher Login

- Description: Enables teachers to log into the system to manage attendance, grades, and notices.
- Input: Teacher username and password.
- Output: Login successful message or redirection to the teacher dashboard.
- Error:
  - Incorrect username or password.
  - Account deactivated.

#### 3.2.8 Create Notice

- Description: Teachers can create and post notices for students.
- Input: Notice title, description, and optional attachments.
- Output: Notice creation confirmation.
- Error:
  - Missing mandatory fields (e.g., title).
  - System error while saving the notice.

#### 3.2.9 Manage Attendance

- Description: Teachers can mark and manage attendance for their assigned classes.
- Input: Class list, date, and attendance status (e.g., present/absent).
- Output: Attendance saved successfully message.

□

Error:

- Invalid class selection.
- Attendance not saved due to server or database issue.

#### 3.2.10 Manage Grade

- Description: Teachers can assign or update grades for students in their classes.
- Input: Class, student ID, subject, and grade.
- Output: Grade successfully saved or updated.
- Error:
  - Invalid grade format.
  - Grade assignment failed due to system error.

#### 3.2.11 Student Login

- Description: Allows students to log into the system to access their details, attendance, notices, and grades.
- Input: Student username and password.
- Output: Login successful message or redirection to the student dashboard.
- Error:
  - Incorrect username or password.
  - Account locked or deactivated.

#### 3.2.12 Check Notice

- Description: Students can view notices posted by teachers or administrators.
- Input: Student's login session.
- Output: List of notices.
- Error:
  - No notices found.
  - Notice retrieval failure due to server error.

#### 3.2.13 Check Attendance

- Description: Allows students to view their attendance records.
- Input: Student ID or login session.
- Output: Attendance details for the selected period or subject.
- Error:
  - Attendance data not found.
  - Database connection error.

□

#### 3.2.14 Check Grade

- Description: Students can view their grades for assessments or subjects.  
Input: Student ID or login session.
- Output: Grades for subjects or exams.
- Error:
  - Grades not updated yet.
  - Server timeout or system error.

#### 3.2.15 Fee Management (Functional Requirement)

- Description: Enables students to view and pay fees, while admins manage fee structures and track payments.
- Input:
  - Student: Login credentials, payment details (e.g., method, amount).
  - Admin: Fee structure (class-wise), student fee records.
- Output:
  - Student: Payment receipt, updated fee status.
  - Admin: Fee reports (pending, paid, overdue).
- Errors:
  - Payment failure (e.g., insufficient funds, invalid details).
  - Missing fee details or server issues during updates.

### 3.3 Non-Functional Requirements

#### 3.3.1 Correctness Requirement

- Ensure accuracy and correctness of data processed by the system.
- Regular audits and validation checks.

#### 3.3.2 Portability requirement

- System should be easily transferable to different environments
- Support for various operating systems and devices

#### 3.3.3 Efficiency Requirement

- Optimal use of system resources.
- \Quick response time and minimal downtime.

□

#### 3.3.4 Usability Requirement

- Intuitive and easy-to-navigate interface.
- Minimal training required for users.

#### 3.3.5 Reusability Requirement

- Components should be reusable in different modules. Modular design to support reuse of code and functionalities

#### 3.3.6 Reliability Requirement

- System should be dependable and function consistently under defined conditions.
- High availability and error recovery mechanisms

#### 3.3.7 Maintainability Requirement

- Easy to maintain and update the system.
- Clear documentation and modular architecture to facilitate maintenance.

### 3.4 User Characteristics

- End Users: Students and parents with basic familiarity with online platforms for accessing schedules, grades, and communication.
- Admin Users: School staff and teachers with basic computer skills for managing records, attendance, and academic tasks. Design & Implementation Constraints

### 3.5 Design & Implementation Constrains

The OSMS is subject to several design and implementation constraints. Technically, the system must operate within the school's existing IT infrastructure, including server capacity and internet speed, and be compatible with commonly used devices and operating systems. Budget limitations mean the system must remain cost-effective, particularly for smaller schools. Rolebased access control is crucial to ensure the security of sensitive data such as financial records and academic grades. Additionally, the system must comply with data privacy regulations like GDPR or COPPA.

### 3.6 Assumptions & Dependencies

The development and operation of the OSMS are based on several assumptions and dependencies. It assumes that users will have access to a stable internet connection and adequate hardware for deployment and use. It also presumes that school staff and users will receive appropriate training for effective usage. The system depends on reliable integration with external tools such as payment gateways, messaging systems, and third-party APIs for notifications and

□

calendar synchronization. Continuous support and software updates are also critical for ensuring the system's functionality and longevity.



## **4 Interface Requirements**

### **4.1 User Interfaces**

The system features a web-based interface with a dashboard for administrators, teachers, students, and parents. It is user-friendly, responsive, and accessible via desktops, tablets, and smartphones.

### **4.2 Hardware Interfaces**

Supports devices like servers for backend operations and client devices such as desktops, laptops, tablets, and smartphones with internet connectivity.

### **4.3 Software Interfaces**

Integrates with databases (e.g., MySQL), web servers (e.g., Apache/Nginx), and third-party tools like payment gateways, messaging systems, and email services.

### **4.4 Communication Interfaces**

Uses standard internet protocols (HTTP/HTTPS) for web access and APIs for integrating email, SMS notifications, and cloud-based services.

## **5 Conclusion**

The Online School Management System (OSMS) is designed to streamline and automate school operations, improving efficiency, accuracy, and communication across all stakeholders, including administrators, teachers, students, and parents. By providing a centralized platform with user-friendly interfaces and seamless integration with necessary tools, OSMS ensures smooth management of academic and administrative tasks. Its scalability, adaptability, and compliance with modern standards make it a reliable solution for schools of all sizes, fostering a more organized and collaborative educational environment.