

A Project Report On

## **School Administrative System (SAS)**

### **Prepared By:**

Parmar Raxesh  
(CE-122)(23CEUBS099)

B. Tech CE, Semester V  
Subject: Advanced Technologies (23CE520)

### **Guided By:**

Prof. Siddharth P. Shah  
Assistant Professor



Department of Computer  
Engineering Faculty of  
Technology  
Dharmsinh Desai University  
College Road, Nadiad-387001  
Gujarat, INDIA



Department of Computer Engineering  
Faculty of Technology  
Dharmsinh Desai University

## **CERTIFICATE**

This is to certify that the practical/term work carried out in the subject of Advanced Technologies (23CE520) and recorded in this report is the bonafide work of

**Parmar Raxesh , ID No: 23CEUBS099**

of B.Tech semester V in the branch of Computer Engineering during the academic year 2025-2026.

Prof. Siddharth P. Shah

Dr. C.K. Bhensdadia

Assistant Professor

Head of the Department

# **Table of Contents (Final, Updated Based on Your Project)**

1. [\*\*Introduction\*\*](#)
  - 1.1 [Problem Definition](#)
  - 1.2 [Objectives](#)
  - 1.3 [Scope of the Project](#)
2. [\*\*System Requirements\*\*](#)
  - 2.1 [Hardware](#)
  - 2.2 [Software](#)
3. [\*\*System Design\*\*](#)
  - 3.1 [Architecture](#)
  - 3.2 [Database Collections](#) (MongoDB)
  - 3.3 [Typical Schema Mapping](#)
4. [\*\*Implementation\*\*](#)
  - 4.1 [Frontend](#) (React – Vite)
  - 4.2 [Backend](#) (Node.js & Express)
5. [\*\*Results & Screenshots\*\*](#)
  - 5.1 [User Interface Screens](#)
  - 5.2 [Sample API Responses](#)
  - 5.3 [MongoDB Data](#)
6. [\*\*Testing\*\*](#)
  - 6.1 [Frontend Testing](#)
  - 6.2 [API Testing](#) (Postman)
7. [\*\*Limitations\*\*](#)
8. [\*\*Future Enhancements\*\*](#)
9. [\*\*Conclusion\*\*](#)
10. [\*\*References\*\*](#)

# 1. Introduction

## Problem Definition

School operations like student records, notices, timetable, billing, and staff details are often managed manually. This creates delays and increases the risk of errors. The goal is to provide a web application that reduces paperwork and keeps daily operations organized.

## Objectives

- Provide role-based dashboards for Teacher, Staff, Principal, and Trust.
- Manage students, previous students, notices, bills, and user details.
- Allow users to report system issues through a bug report module.
- Use MongoDB to store all school-related data.
- Handle login sessions using a simple HTTP session on the frontend, while the backend is capable of JWT authentication.

## Scope of the Project

The system includes:

- User authentication
  - Student management
  - Staff and teacher management
  - Notices, timetable, and bills
  - Bug reporting
  - Access control based on user roles
- 

# 2. System Requirements

## Hardware

**Minimum:** i3, 4GB RAM

**Recommended:** i5+, 8GB RAM

## Software

- Node.js 18+
  - MongoDB 6+
  - React (Vite)
  - Express.js
  - Axios
  - Mongoose
  - Postman
  - VS Code
- 

## 3. System Design

### Architecture

**React Frontend → Express Backend → Local MongoDB**

Frontend uses HTTP session to maintain logged-in state.  
Backend supports JWT logic but can also work without it.

### Database Collections

1. **users**
2. **user\_details**
3. **students**
4. **previousStudents**
5. **bills**
6. **notices**

## Typical Schema Mapping

- **User**  
Handles login credentials and user roles ( Staff, Teacher, Principal, Trustee).
  - **User Details**  
Stores extended profile info such as name, password (hash), and role-specific data.
  - **Students**  
Active student records with class, div, timetable mapping, etc.
  - **Previous Students**  
Records of students who left or graduated.
  - **Bills**  
payment-related entries.
  - **Notices**  
Announcements posted by principal, or trustee.
- 

## 4. Implementation

### Frontend

Structure under /frontend/src:

#### Components

- **Navbar.jsx** – Navigation across pages for different roles
- **Footer.jsx** – Footer area
- **ProfileCard.jsx** – Shows user details
- **ReportBug.jsx** – Component to submit bug reports
- **UseBills.jsx** – Component of bills
- **UseNotice.jsx** – Component of notices
- **UseStudents.jsx** – Component of students
- **UserList.jsx** – Displays users

- **UseTimeTable.jsx** – Component of timetable
- **hooks/UseLogoutOnMount.jsx** – Clears session on mount

## Pages

- **Home.jsx**
- **Login.jsx**
- **Signup.jsx**
- **Profile.jsx** : for updated the email , name , and password
- **Details.jsx**
- **Teacher.jsx**
- **Staff.jsx**
- **Principal.jsx**
- **Trustee.jsx**
- **ReportBugForm.jsx**

## Services

- **api.jsx** – Axios instance connecting frontend → backend

## Backend

Your backend routes manage:

- User authentication
- User profiles
- Students
- Bills
- Notices
- Bug reporting
- Staff / Teacher / Principal / Trustee operations

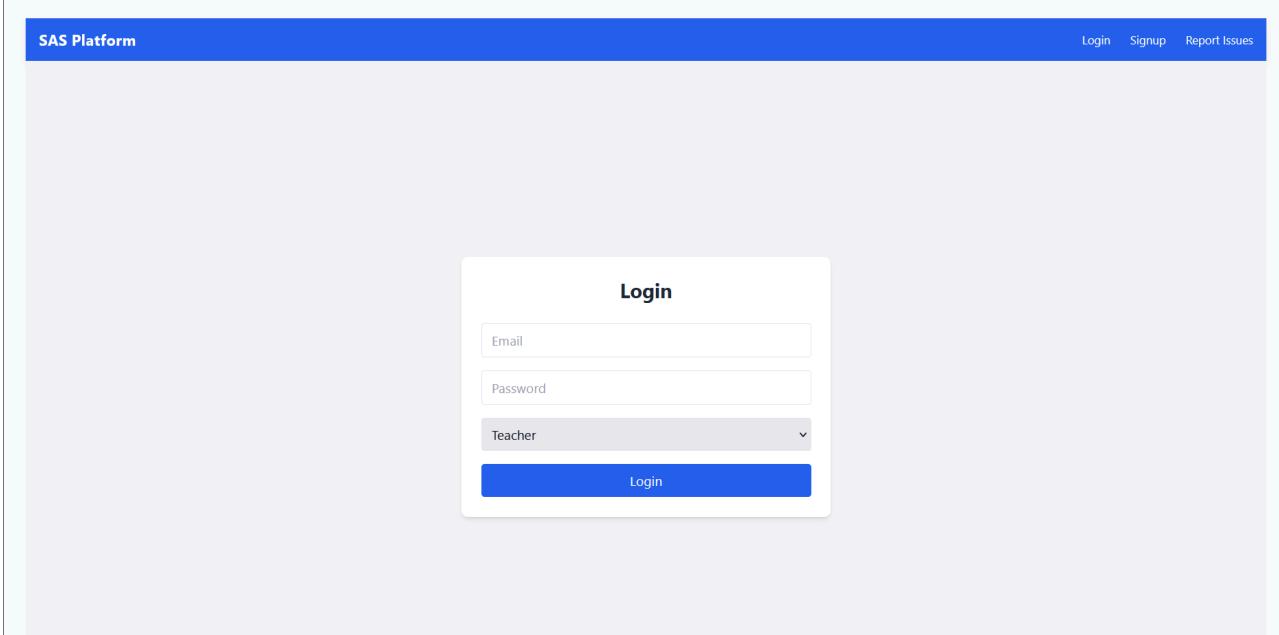
Backend supports:

- HTTP sessions
  - JWT tokens (optional as per your comment)
- 

## 5. Results & Screenshots

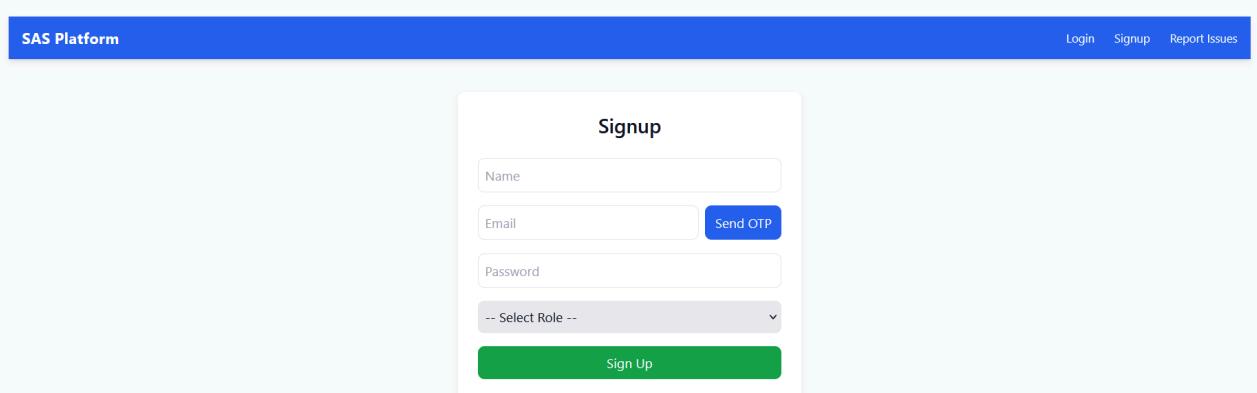
### User Interface Screens

- Login



The screenshot shows the login interface for the SAS Platform. At the top, there is a blue header bar with the text "SAS Platform" on the left and "Login" "Signup" "Report Issues" on the right. Below the header is a large white area containing a "Login" form. The form has the title "Login" at the top. It includes three input fields: "Email", "Password", and a dropdown menu set to "Teacher". At the bottom of the form is a blue "Login" button.

- Signup



The screenshot shows the signup interface for the SAS Platform. At the top, there is a blue header bar with the text "SAS Platform" on the left and "Login" "Signup" "Report Issues" on the right. Below the header is a large white area containing a "Signup" form. The form has the title "Signup" at the top. It includes four input fields: "Name", "Email" (with a "Send OTP" button next to it), "Password", and a dropdown menu labeled "-- Select Role --". At the bottom of the form is a green "Sign Up" button.

- Dashboards (Teacher, Staff, Principal, Trustee)

**SAS Platform**

Change Password & Email Personal Details Logout

### My Profile

Name: Rakesh  
Email: goturakesh@gmail.com

### Notices

<b>Test1</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>
<b>Test2</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>

[Students Tab](#) [Report a Bug](#) [Generate Timetable](#) [Manage Bills](#)

▶ [View All Teachers](#)

▶ [View Staff](#)

© 2025 SAS Platform

**SAS Platform**

Change Password & Email Personal Details Logout

### My Profile

Name: user1  
Email: 23ceuos102@ddu.ac.in

### Notices

<b>Test1</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>
<b>Test2</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>

[Students Tab](#) [Report a Bug](#)

© 2025 SAS Platform

**SAS Platform**

Change Password & Email Logout

### My Profile

Name: trustee  
Email: patel.aksh1810@gmail.com

### Notices

<b>Test1</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>
<b>Test2</b> Testing <a href="#">Documents:Doc_1</a> By: Rakesh (goturakesh@gmail.com)   Date: 2025-09-20	<a href="#">Edit</a> <a href="#">Delete</a>

[Report a Bug](#)

▶ [View All Teachers](#)

▶ [View Principal](#)

▶ [View Staff](#)

© 2025 SAS Platform

## My Profile

Name: Rakesh  
Email: 23ceubs099@ddu.ac.in

## Notices

## Test1

Testing

Documents:[Doc 1](#)

By: Rakesh (goturakesh@gmail.com) | Date: 2025-09-20

## Test2

Testing

Documents:[Doc 1](#)

By: Rakesh (goturakesh@gmail.com) | Date: 2025-09-20

[Report a Bug](#)[Manage Bills](#)

© 2025 SAS Platform

## Sample API Responses

- Student list

The screenshot shows a browser developer tools interface with the "Response" tab selected. The JSON response body is displayed, showing two student records (index 1 and 6). Each record contains fields like studentName, fatherName, motherName, email, std, div, rollNo, phoneNo, aadharNo, photoUrl, and \_id.

```
JSON
[
  {
    "studentName": "student2",
    "fatherName": "father2",
    "motherName": "mother2",
    "email": "student2@example.com",
    "std": 1,
    "div": "A",
    "rollNo": 1,
    "phoneNo": 1234567001,
    "aadharNo": 123456789012,
    "photoUrl": "https://res.cloudinary.com/sasapp/image/upload/v1758356156/photos/qzopc1ykvcxqg3pnbu8.jpg",
    "_id": "68ce62ba95d34a5edad8bf85"
  },
  {
    "studentName": "student4",
    "fatherName": "f4",
    "motherName": "m4",
    "email": "st@eaxmple.com",
    "std": 6,
    "div": "A",
    "rollNo": 4,
    "phoneNo": 1234500004,
    "aadharNo": 123456789004,
    "photoUrl": "https://res.cloudinary.com/sasapp/image/upload/v1758627077/photos/thqm2sowhvau9hhv3bh.jpg",
    "_id": "68d284d99268d3073c5deb4c"
  }
]
```

- Bills

```

[{"_id": "68ce5ad15a72c16cc29dd555", "amount": 10000, "billDate": "2023-01-01", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758354132/feesSem1.pdf/r4ihdcl9bwnlcokiazob.pdf"}, {"url": "https://res.cloudinary.com/sasapp/image/upload/v1758639573/doc_0/tvo7azbwi3p3uzvmdm5x.jpg"}], "type": "Electricity", "vendorEmail": "goturaxesh@gmail.com", "vendorName": "Raxesh"}, {"_id": "68d2b5d3364669feec64638b", "amount": 5000, "billDate": "2025-09-23", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758647662/notice_documents/oxxchv10sweqltcjy70h.jpg"}], "type": "Water", "vendorEmail": "goturaxesh@gmail.com", "vendorName": "Raxesh"}, {"_id": "68ce58e55465c36c9318b225", "amount": 15000, "billDate": "2025-09-23", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758639573/doc_0/tvo7azbwi3p3uzvmdm5x.jpg"}], "type": "Examination", "vendorEmail": "goturaxesh@gmail.com", "vendorName": "Raxesh"}, {"_id": "68ce5a187c20b4b7e2f3d7e1", "amount": 10000, "billDate": "2025-10-03", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758353947/notice_documents/nfcqaxpxnsmrddpxpfkr3.pdf"}], "type": "Internet", "vendorEmail": "goturaxesh@gmail.com", "vendorName": "Raxesh"}]

```

- Notices

```

[{"_id": "68ce58e55465c36c9318b225", "message": "Testing", "subject": "Test1", "user": {"email": "goturaxesh@gmail.com", "name": "Raxesh", "role": "principal"}, "date": "2025-09-20", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758647662/notice_documents/oxxchv10sweqltcjy70h.jpg"}]}, {"_id": "68ce5a187c20b4b7e2f3d7e1", "message": "Testing", "subject": "Test2", "user": {"email": "goturaxesh@gmail.com", "name": "Raxesh", "role": "principal"}, "date": "2025-09-20", "documents": [{"url": "https://res.cloudinary.com/sasapp/image/upload/v1758353947/notice_documents/nfcqaxpxnsmrddpxpfkr3.pdf"}]}]

```

## MongoDB Data

- MongoDB Compass (collections)

The screenshot shows the MongoDB Compass application interface. On the left, there's a sidebar titled 'Compass' with a 'Connections' section containing 'SAS'. The main area is titled 'SAS > test > users' and shows a 'Documents' tab with 4 results. Each document contains the following fields:

- `_id: ObjectId('68ce3e5ee481e237f073c038')`
- `name : "Rakesh"`
- `email : "goturakesh@gmail.com"`
- `role : "principal"`
- `password : "$2b$10$M1bNIGfs/g78T984U5qFk.WHnoDm5pPM8wD/zeCUjvfjKhOZVpDay"`

Below this, another set of documents is shown with similar structure and data.

## 6. Testing

### Frontend Testing

- Verified routing and protected pages using session-based checks.

### API Testing

Performed using Postman:

- User login
- Fetch students
- Add notices or bills
- Create/Delete user
- Submit bug reports

## 7. Limitations

- No attendance tracking
  - No real JWT usage in frontend
  - Not optimized for mobile view
  - Not deployed on cloud
- 

## 8. Future Enhancements

- Convert session-based login to full JWT auth
  - Deploy to AWS/Vercel
  - Add attendance module
  - Add SMS/email notifications
- 

## 9. Conclusion

The project successfully implements a real school administration workflow using the MERN stack. It provides structured dashboards, student management, notices, bills, and role-specific features. It reduces paperwork and makes data management more reliable.

---

# 10. References

- Node.js Official Documentation: <https://nodejs.org>
- Express.js Documentation: <https://expressjs.com>
- React.js Documentation: <https://react.dev>
- Vite Documentation: <https://vitejs.dev>
- MongoDB Documentation: <https://www.mongodb.com/docs>
- Mongoose Documentation: <https://mongoosejs.com>
- Axios GitHub: <https://github.com/axios/axios>
- Postman Documentation: <https://learning.postman.com>
- MDN Web Docs: <https://developer.mozilla.org>
- StackOverflow Community Discussions: <https://stackoverflow.com>
- Tutorials and guides from YouTube channels covering MERN stack development