## A PROJECT REPORT

## ON

## SCHOOL RECORD SYSTEM

## Submitted to

**University Of Mumbai**

## In the partial fulfillment of the degree of

**Bachelor of Computer Science**

**By**

**Vinayak Rajan Vishwakarma**

*Under the Guidance of*

**Prof. Janhavi Kshirsagar**

**DEPARTMENT OF COMPUTER SCIENCE**



**JNAN VIKAS MANDAL’S**

**Mohanlal Raichand Mehta College of Commerce, Diwali**

**Maa College of Science**

**Amritlal Raichand Mehta College of Arts, Dr. R.T. Doshi**

**College of Computer Science**

**(Sem 5)**

**(2024-2025)**



**JNAN VIKAS MANDAL’S**

**PADMASHREE DR. R.T.DOSHI DEGREE COLLEGE OF**

**INFORMATION TECHNOLOGY**

**MOHANLAL RAICHAND MEHTA COLLEGE OF COMMERCE DIWALIMAA DEGREE COLLEGE OF SCIENCE**

**AMRATLAL RAICHAND MEHTA COLLEGE OF ARTS JVM’S DEGREE COLLEGE OF INFORMATION TECHNOLOGY**

**AIROLI, NAVI MUMBAI – 400708**

## CERTIFICATE

This is to certify that **Vinayak Rajan Vishwakarma** Student of Third Year Bachelor of Computer Science of JVM’s Mehta Degree College bearing Roll no. **1847** has successfully completed the Project work titled “**School Record System**” in partial fulfillment in this educational year of University of Mumbai. This Project is the record of authentic work carried out during the academic year 2024- 2025. It is further certified that he has completed all required phases of the Project.

**Mrs.Janhavi Kshrisagar**

**Internal guide CS-IT Co-ordinator**

**Dr.B.R.Deshpande   
 External Examiner Principal**

**Date : College Seal**

**Abstract**

The School Record System (S.R.S.) is a digital solution that replaces traditional paper-based record-keeping in schools. It centralizes student, teacher, staff, and financial data in a secure, easy-to-access digital database, reducing errors and manual work. The system offers automated fee management, real-time reporting, and remote access, significantly improving administrative efficiency. With role-based access control, it ensures data security while enabling 24/7 availability. Scalable and user-friendly, the S.R.S. supports school growth, reduces costs by minimizing paper use, and includes robust backup and disaster recovery features to ensure long-term reliability.

**Acknowledgment**

It’s my great pleasure to take the opportunity and sincerely thank all those who have shown me the way to a successful project and helped me a lot during the completion of my project.

I greatly thank my Project Guide **Prof. Janhavi Kshirsagar** without whom the completion of this project couldn’t have been possible.

I take this opportunity to express my deep gratitude towards all the members of the Computer Science Department, for helping me in the completion of the project.

My sincere thanks to respected Principal **Dr.B.R.Deshpande** and the Head of the Computer Science Department **Prof.Janhavi Kshirsagar** for providing all the facilities including the availability of Computer Lab.

Further, I thank all my colleagues for their encouragement and support throughout the period of the completion.

**Index**

|  |  |  |
| --- | --- | --- |
|  | **Table of Contents** |  |
| **Sr. No** | **Phase Name** | **Signature** |
| **A)** | **Preliminary Investigation** |  |
| 1.1 | Current System |  |
| 1.2 | Limitations Of the system |  |
| 1.3 | Scope of proposed system |  |
| 1.4 | Advantages of proposed system |  |
| 1.5 | Feasibility Study |  |
| 1.6 | Gantt Chart |  |
| **B)** | **System Analysis** |  |
| 2.1 | Fact Finding Techniques |  |
| 2.2 | Event Table |  |
| 2.3 | Use Case Diagram |  |
| 2.4 | Activity Diagram |  |
| 2.5 | Class Diagram |  |
| 2.6 | Class Association Diagram |  |
| 2.7 | Object Diagram |  |
| 2.8 | Sequence Diagram |  |
| 2.9 | Collaboration Diagram |  |

|  |  |  |
| --- | --- | --- |
| **C)** | **System Design** |  |
| 3.1 | Database Tables with Attributes & Constraints |  |
| 3.2 | Component Diagram |  |
| 3..3 | Package Diagram |  |
| 3.4 | Deployment Diagram |  |
| 3.5 | System Flow Chart |  |
| 3.6 | Structure Chart |  |
| **D)** | **System Coding** |  |
| 4.1 | Menu Tree |  |
| 4.2 | Test Cases, Test Data, Test Result |  |
| 4.5 | Screen Layout |  |
| 4.6 | Future enhancement |  |
| 4.7 | Hardware/Software Requirement |  |
| 4.8 | Reference Bibliography |  |

**Phase - 1**

**Preliminary Investigation**

**Current System**

1. The school relies on a traditional, paper-based record-keeping system that has been in place for a long time.
2. All student, teacher, and staff records are physically stored in filing cabinets, binders, and paper documents.
3. Access to records is restricted to the physical location of the documents, making it inconvenient for staff members who need to access data remotely or after hours.
4. Records are scattered across various filing cabinets, folders, and documents, making it difficult to locate specific information when needed.
5. The current system maintains a historical record of the school's activities, including enrolment, staff changes, and financial transactions.
6. Fee collection and financial management are handled manually, with staff members performing calculations and managing transactions directly.

**Limitations of Current system**

1) Manual record-keeping is time-consuming and prone to errors.

2) Records are spread across multiple paper documents and files, making it hard to find specific information.

3) Limited access to records and risks of loss or damage.

4) Finding and retrieving specific records can be time-consuming.

5) Manually calculating and managing fees can lead to financial discrepancies.

6) Scaling up or accommodating more students and staff may be challenging.

7) Ongoing expenses for paper, storage, and printing costs.

8) Inefficient paper usage contributes to environmental waste.

9) Paper records require specialized knowledge for interpretation and management.

10) Vulnerable to data loss due to accidents or disasters

**Proposed system**

1. The proposed system is a comprehensive digital solution designed to streamline all aspects of school record-keeping, including student data, teacher information, staff records, and financial transactions.
2. All data is stored in a centralized and secure digital database, making it easy to access, manage, and retrieve information with efficiency and accuracy.
3. The SRS offers an intuitive and user-friendly interface, ensuring that staff members can quickly adapt to the system with minimal training.
4. Automation features reduce manual data entry and the risk of errors, significantly improving the efficiency of data management.
5. The system provides role-based access control, ensuring data security, and enables authorized personnel to access data remotely, promoting 24/7 availability.
6. The SRS offers automated fee management, including fee calculation, remaining fees, and the ability for students and parents to make online payments, reducing financial errors and simplifying the payment process.
7. The system generates real-time reports, analytics, and insights, enabling school administrators to make data-driven decisions and monitor performance effectively.
8. The SRS is designed to be easily adaptable and scalable, allowing the school to accommodate growth and expansion in student enrollment and staff.
9. By eliminating the need for paper, printing, and physical storage, the SRS provides significant cost savings over time.
10. The proposed system supports sustainability goals by reducing paper consumption and the school's carbon footprint.
11. The SRS includes features to ensure regulatory compliance and simplify record retention procedures, minimizing the risk of non-compliance.
12. Easy data backups and a disaster recovery plan protect against data loss due to accidents, fires, or other disasters.

**Advantages of proposed system**

***Efficiency and Time Savings:***

Automation of administrative tasks such as record-keeping, fee collection, and data entry significantly reduces the time required to manage these processes.

***Accuracy and Error Reduction:***

Automation and digital record-keeping minimize the risk of human errors, ensuring that records are accurate and consistent.

***Centralized Data:***

All student, teacher, and staff data is stored in one centralized database, making it easy to access and manage information from a single location.

***Accessibility*:**

Authorized personnel can access records and information remotely, improving accessibility and enabling 24/7 data retrieval.

***Financial Management:***

Automated fee management and online payment options reduce financial discrepancies, make fee collection more convenient, and provide accurate financial reporting.

***Real-time Reporting:***

The system generates real-time reports and analytics, helping school administrators make informed decisions and track performance effectively.

***Scalability:***

The system is easily adaptable and can accommodate growth in student enrolment and staff, making it a scalable solution for the school.

***Cost Savings:***

By eliminating the need for paper, printing, and physical storage, the system provides significant cost savings in the long run.

**Feasibility Study:**

***Technical Feasibility:***

The system has been developed in Visual Basic 2012 using ACCESS tables, hence system is technically feasible.

***Operational Feasibility****:*

As the user is familiar with the Windows environment and the system is developed in Visual Studio, there is no need for special training on the operating system. Hence system is operationally feasible.

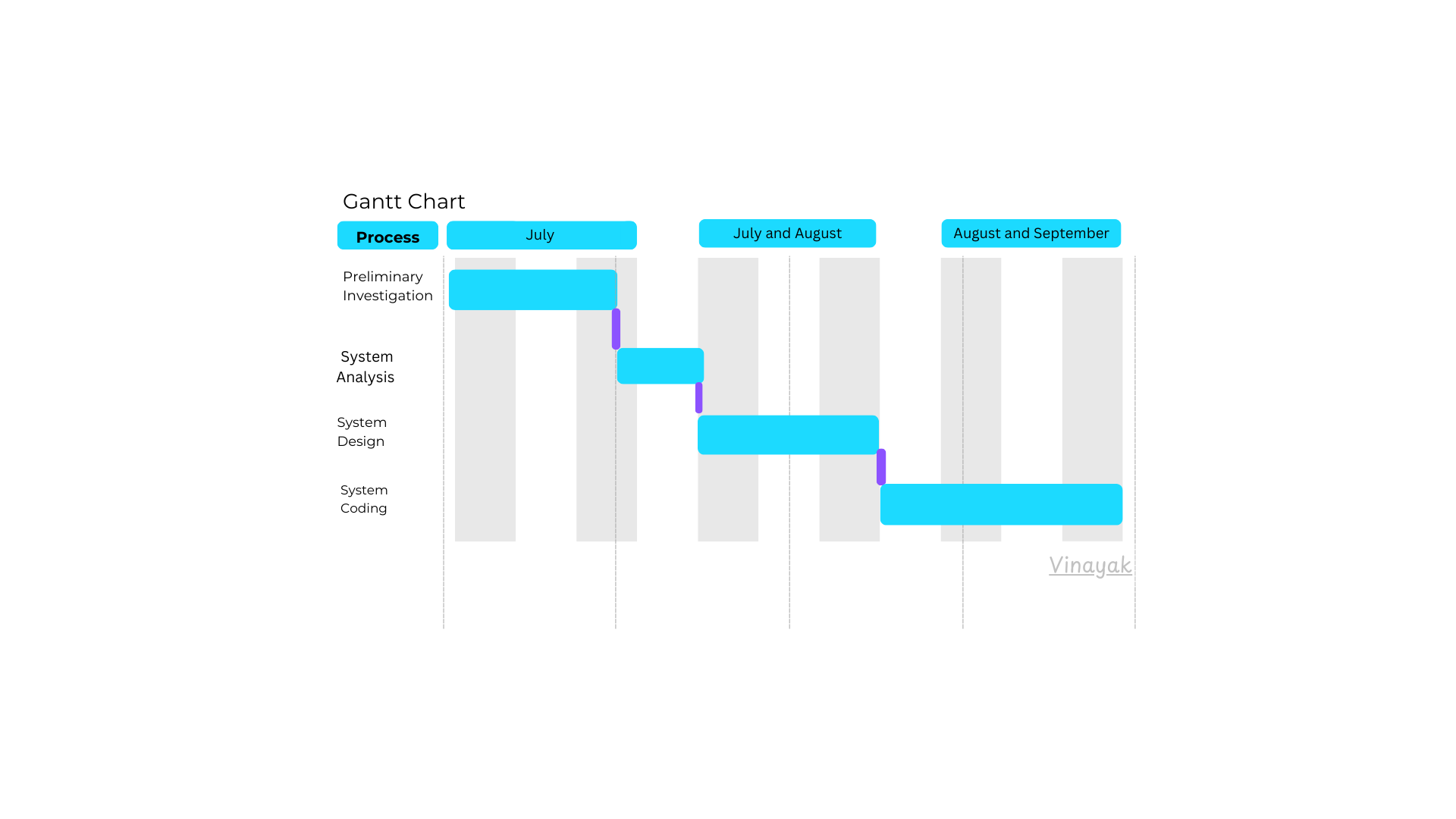
***Economical Feasibility:***

Since the hardware and software are already available there is no need for additional hardware and software. Hence the system is economically feasible.

***Schedule feasibility:***

Schedule feasibility means the project can be completed on time. Every schedule requires many assumptions and an estimate of the project

**Gantt chart**



**Phase - 2**

**System Analysis**

**Fact finding technique (Interviewing)**

* **Is Your System Manual or Computerized?**

1. No.

* **How do you Maintain your records ??**

1. We maintain our records through traditional paper-based methods.

* **How do you search previous record?**

1. To find a specific record, we typically rely on physical file organization. We'll go through filing cabinets and paper documents, utilizing indexing or labeling systems to identify and locate the desired record.

* **How do you maintain Information of student and staff?**

1. Each student has a physical file that contains their personal details, academic records, and any other relevant information. These files are organized alphabetically or by class/year in filing cabinets. Staff records are also organized in filing cabinets, and updates are made manually as needed.

* **What are the biggest challenges you face with your current manual system for managing school records?**

1. Our current manual record-keeping system is quite time-consuming due to the extensive data entry required. Managing data efficiently poses a challenge, and there is a limitation in accessing information. However, the most significant challenge we face is related to financial management.

* **How do you generate student fees receipt?**

1. By using a bill book. We keep one copy and give another to the customer.

**Event Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr no** | **Event** | **Trigger** | **Source** | **Activities** | **Response** | **Destination** |
| 1 | Add New students | New student added | Admin | Create new student record | Message:  Student added successfully | Admin |
| 2 | Update student details | 1.Search student record.  2.Update Student record | Admin | 1.Search student.  2.Update student | Message:  Student updated  successfully | Admin |
| 3 | Delete student record. | 1.Search student record.  2.Delete student record. | Admin | 1.Search student record.  2.Delete student record. | Message:  Student deleted successfully | Admin |
| 4 | Search student record. | Display student record. | Admin | Check student record. |  | Admin |
| 5 | Add new teacher | New teacher  added | Admin | Create new  teacher  record. | Message:  Teacher added successfully | Admin |
| 6 | Update teacher details | 1.Search teacher record.  2.Update teacher record | Admin | 1.Search teacher.  2.Update teacher | Message:  Teacher updated  successfully | Admin |
| 7 | Delete teacher record. | 1.Search teacher record.  2.Delete teacher record. | Admin | 1.Search teacher record.  2.Delete teacher record. | Message:  Teacher deleted successfully | Admin |
| 8 | Search teacher record. | Display teacher record. | Admin | Check teacher record. |  | Admin |
| 9 | Add new staff | New staff  added | Admin | Create new  staff  record. | Message:  Staff added successfully | Admin |
| 10 | Update staff details | 1.Search staff record.  2.Update staff record | Admin | 1.Search staff.  2.Update staff | Message:  Staff updated  successfully | Admin |
| 11 | Delete staff record. | 1.Search staff record.  2.Delete staff record. | Admin | 1.Search staff record.  2.Delete staff record. | Message:  Staff deleted successfully | Admin |
| 12 | Search staff record. | Display staff record. | Admin | Check staff record. |  | Admin |
| 13 | Search teacher record. | Display teacher record. | Admin | Check teacher record. |  | Admin |
| 9 | Add student fees | Fees added | Admin | Create new  Fees  record. | Message:  Fees added successfully | Admin |
| 10 | Update student fees | 1.Search student record.  2.Update student fees record | Admin | 1.Search student.  2.Update student fees. | Message:  Fees updated  successfully | Admin |
| 11 | Search student fees record. | Display student record. | Admin | Check manage fees record. |  | Admin |

**Use Case Diagram**

Admin

**Manage Student**

<<uses>>

<<extends>>

<<extends>>

Admin

**Manage Teacher**

<<uses>>

<<extends>>

<<extends>>

Admin

**Manage Staff**

<<uses>>

<<extends>>

<<extends>>

Admin

**Manage Fees**

<<uses>>

<<extends>>

Admin

**Activity Diagram**

1) To Add Student:

: Student

Add New Student

S1: Student

2) To Update Student:

: Student

Search Student

S1: Student

: Student

Edit Student

S1: Student

: Student

Update Student

S1: Student

3) To Delete Student:

: Student

Search Student

S1: Student

: Student

Delete Student

S1: Student

4) To Add Teacher:

: Teacher

Add New Teacher

T1: Teacher

5) To Update Teacher:

: Teacher

Search Teacher

T1: Teacher

: Teacher

Edit Teacher

T1: Teacher

: Teacher

Update Teacher

T1: Teacher

6) To Delete Teacher:

: Teacher

Search Teacher

T1: Teacher

: Teacher

Delete Teacher

T1: Teacher

7) To Add Staff:

: Staff

Add New Staff

T1: Staff

8) To Update Staff:

: Staff

Search Staff

S1: Staff

: Staff

Edit Staff

S1: Staff

: Staff

Update Staff

S1: Staff

9) To Delete Staff:

: Staff

Search Staff

S1: Staff

: Staff

Delete Staff

S1: Staff

10) To Add Student Fees:

: Fees

Add Student Fees

F1: Fees

11) To Update Student Fees:

: Fees

Search Student

F1: Fees

: Fees

Edit Student

F1: Fees

: Fees

Update Fees

F1: Fees

**Class Diagram**

|  |
| --- |
| Students |
| id - int (11)  full\_name - varchar (40)  parent\_name - varchar (40)  email - varchar (40)  phone\_no - varchar (10)  dob - date  gender - varchar (6)  address - varchar (80)  class - varchar (10)  roll\_no - int (5) |
| Add()  Update()  Delete()  Search |

|  |
| --- |
| Teachers |
| tr\_id - int (11)  full\_name - varchar (40)  email - varchar (40)  phone\_no - varchar (10)  dob - date  gender - varchar (6)  address - varchar (80)  qualification - varchar (20) |
| Add()  Update()  Delete()  Search |

|  |
| --- |
| Staff |
| stf\_id - int (11)  full\_name - varchar (40)  email - varchar (40)  phone\_no - varchar (10)  dob - date  gender - varchar (6)  address - varchar (80)  qualification - varchar (20) |
| Add()  Update()  Delete()  Search |

|  |
| --- |
| Fees |
| id - int (11)  student\_rollno - varchar (40)  student\_name - varchar (40)  student\_class - varchar (10)  paid - date  total - varchar (6)  remaining - varchar (80)  status - varchar (20) |
| Add()  Update()  Search |

**Class Association Diagram**

Staff

Fees

Teachers

Patient

\*

\*

1

1

1

\*

Students

\*

1

\*

**Sequence Diagram**

1. To Add Students:

Add New student

Ack for Adding student

: User Interface

S: Student

Student Added

Ack

Save student

Saved

Ack

Ack

1. To Update Students:

Search student info

Info searched Ack

: User Interface

S: Student

Student searched

Ack

Update student

Student updated

Ack

Ack

Save

Ack

Saved

Ack

1. To Delete Students:

Search student info

Ack

: User Interface

S: Student

Info searched

Ack

Delete student

Student deleted

Ack

Ack

1. To Add Teachers:

Add New teacher

Ack for Adding teacher

: User Interface

T: Teacher

Teacher Added

Ack

Save teacher

Saved

Ack

Ack

1. To Update Teacher:

Search teacher info

Info searched Ack

: User Interface

T: Teacher

teacher searched

Ack

Update teacher

Teacher updated

Ack

Ack

Save

Ack

Saved

Ack

1. To Delete Teacher:

Search teacher info

Ack

: User Interface

T: Teacher

Info searched

Ack

Delete teacher

Teacher deleted

Ack

Ack

1. To Add Staff:

Add New staff

Ack for Adding staff

: User Interface

S: Staff

Staff Added

Ack

Save staff

Saved

Ack

Ack

1. To Update Staff:

Search staff info

Info searched Ack

: User Interface

S: Staff

Staff searched

Ack

Update staff

staff updated

Ack

Ack

Save

Ack

Saved

Ack

1. To Delete Staff:

Search staff info

Ack

: User Interface

S: Staff

Info searched

Ack

Delete staff

Staff deleted

Ack

Ack

1. To Add Fees:

Add New student

Ack for Adding student & fees

: User Interface

F: Fees

Student & Fees Added

Ack

Save fees

Saved

Ack

Ack

1. To Update Fees:

Search student info

Info searched Ack

: User Interface

F: Fees

Student searched

Ack

Update fees

fees updated

Ack

Ack

Save

Ack

Saved

Ack

**Collaboration Diagram**

1. Add Student Record:
   1. Add Student
   2. Save Student
2. Add Student

: Student

: User Interface

Admin

1. Update Student Record:
   1. Add Student
   2. Edit Student
   3. Update Student

: Student

: User Interface

1. Search Student
2. Edit Student

Admin

1. Delete Student Record:
2. Search Student
3. Delete Student
   1. Search Student

1.2 Delete Student

: Student

: User Interface

Admin

1. Add Teacher Record:
   1. Add Teacher
   2. Save Teacher
2. Add Teacher

: Teacher

: User Interface

Admin

1. Update Teacher Record:
   1. Add Teacher
   2. Edit Teacher
   3. Update Teacher

: Teacher

: User Interface

1. Search Teacher
2. Edit Teacher

Admin

1. Delete Teacher Record:
2. Search Teacher
3. Delete Teacher
   1. Search Teacher

1.2 Delete Teacher

: Teacher

: User Interface

Admin

1. Add Staff Record:
   1. Add Staff
   2. Save Staff
2. Add Staff

: Staff

: User Interface

Admin

1. Update Staff Record:
   1. Add Staff
   2. Edit Staff
   3. Update Staff

: Staff

: User Interface

1. Search Staff
2. Edit Staff

Admin

1. Delete Staff Record:
   1. Search Staff

1.2 Delete Staff

: Staff

: User Interface

1. Search Staff
2. Delete Staff

Admin

1. Add Fees Record:
   1. Add Fees
   2. Save Fees
2. Add Student

: Fees

: User Interface

Admin

1. Update Fees Record:
   1. Add Fees
   2. Edit Fees
   3. Update Fees

: Fees

: User Interface

1. Search Student
2. Edit Fees

Admin

**Derive Table from Entities & Relationship**

**Student Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Attributes** |
| Id | Int | Hold student Id | Field size=11 |
| Full\_name | Varchar | Hold student full name | Field size=40 |
| Parent\_name | Varchar | Hold student parent name | Field size=40 |
| Email | Varchar | Hold student email | Field size=40 |
| Phone\_no | Varchar | Hold student phone number | Field size=10 |
| Dob | Date | Hold student birth date |  |
| Gender | Varchar | Hold student gender | Field size=6 |
| Address | Varchar | Hold student address | Field size=80 |
| Class | Varchar | Hold student class | Field size=10 |
| Roll\_no | Int | Hold student roll no | Field size=5 |

**Teacher Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Attributes** |
| Tr\_id | Int | Hold teacher Id | Field size=11 |
| Full\_name | Varchar | Hold teacher full name | Field size=40 |
| Email | Varchar | Hold teacher email | Field size=40 |
| Phone\_no | Varchar | Hold teacher phone number | Field size=10 |
| Dob | Date | Hold teacher birth date |  |
| Gender | Varchar | Hold teacher gender | Field size=6 |
| Address | Varchar | Hold teacher address | Field size=80 |
| Qualification | Varchar | Hold teacher class | Field size=20 |

**Staff Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Attributes** |
| Stf\_id | Int | Hold staff Id | Field size=11 |
| Full\_name | Varchar | Hold staff full name | Field size=40 |
| Email | Varchar | Hold staff email | Field size=40 |
| Phone\_no | Varchar | Hold staff phone number | Field size=10 |
| Dob | Date | Hold staff birth date |  |
| Gender | Varchar | Hold staff gender | Field size=6 |
| Address | Varchar | Hold staff address | Field size=80 |
| Qualification | Varchar | Hold staff class | Field size=20 |

**Fees Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Attributes** |
| Id | Int | Hold student Id | Field size=11 |
| Student\_rollno | Int | Hold student full name | Field size=5 |
| Student\_name | Varchar | Hold student email | Field size=40 |
| Student\_class | Varchar | Hold student phone number | Field size=5 |
| Paid | Int | Hold student birth date | Field size=11 |
| Total | Int | Hold student gender | Field size=11 |
| Remaining | Int | Hold student address | Field size=11 |
| Status | Varchar | Hold student class | Field size=15 |

**Component Diagram**

**Admin**

**Manipulate**

**Students**

**Database**

**Access**

**Manipulate**

**Teachers**

**Database**

**Access**

**Manipulate**

**Staff Database**

**Access**

**Report**

**Generate**

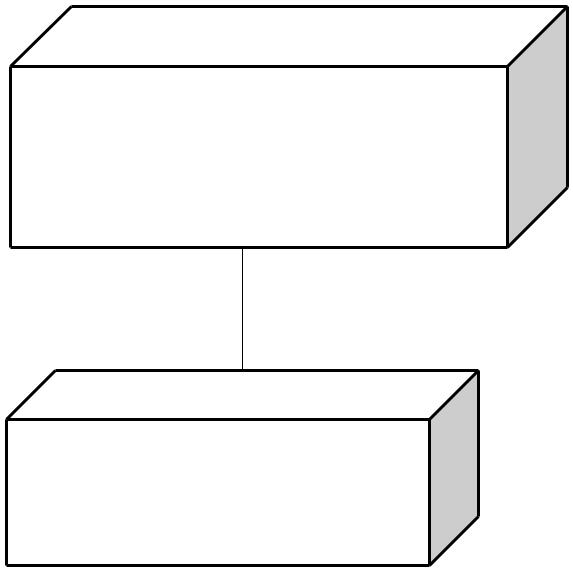
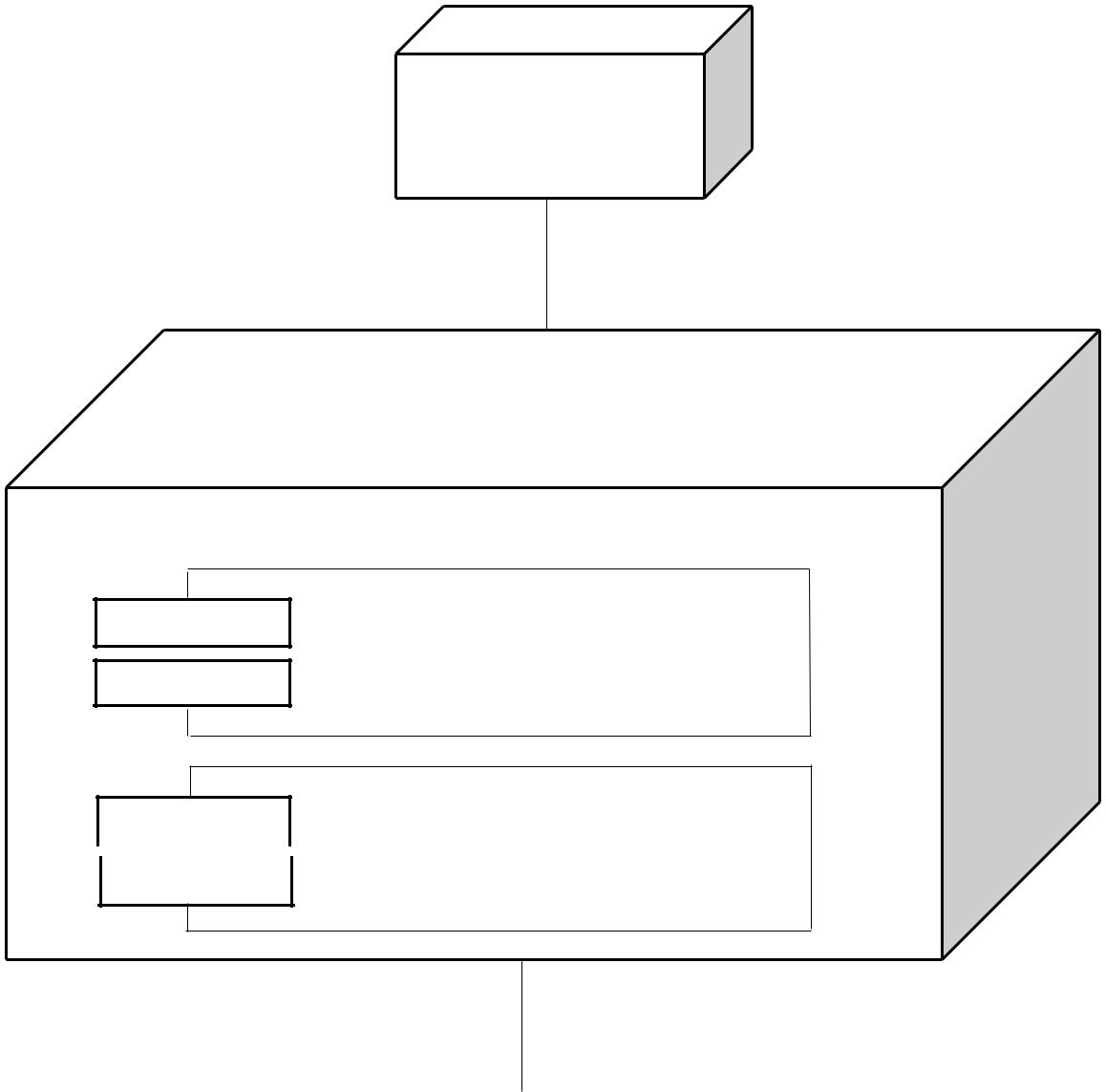
**Manipulate**

**Fees Database**

**Access**

**Print**

**Deployment diagrams**



Server

**System Administrator Executable**

**Common Administrator**

**Executable**

**User: Admin**

**Printer**

**Application Server**

**System Flow Chart**

Login

Maintain Login

Information

Login

Student

Maintain Student

Information

Student

Teacher

Maintain Teacher

Information

Teacher

Staff

Maintain Staff

Information

Staff

Fees

Maintain Fees

Information

Fees

Fees Reciept

**Structure Chart**

School Record System

Manage

Students

Manage

Teachers

Manage

Staff

Manage

Fees

C

B

A

D

A

B

Manage Student

id

fullname

parent name

email

phone no

dob

Student record

gender

address

class

roll no

Manage Teacher

tr\_id

fullname

email

phone no

dob

Teacher record

gender

address

class

roll no

Teacher details

Student details

Manage Staff

stf\_id

fullname

email

phone no

dob

Staff record

gender

address

class

roll no

D

C

Manage Fees

id

student rollno

student name

student class

paid

Fees record

total

remaining

status

roll no

Fees details

Staff details

**Phase - 4**

**System Coding**

**Menu Tree**

**School Record System**

**Login**

**Students**

**Staff**

**Teachers**

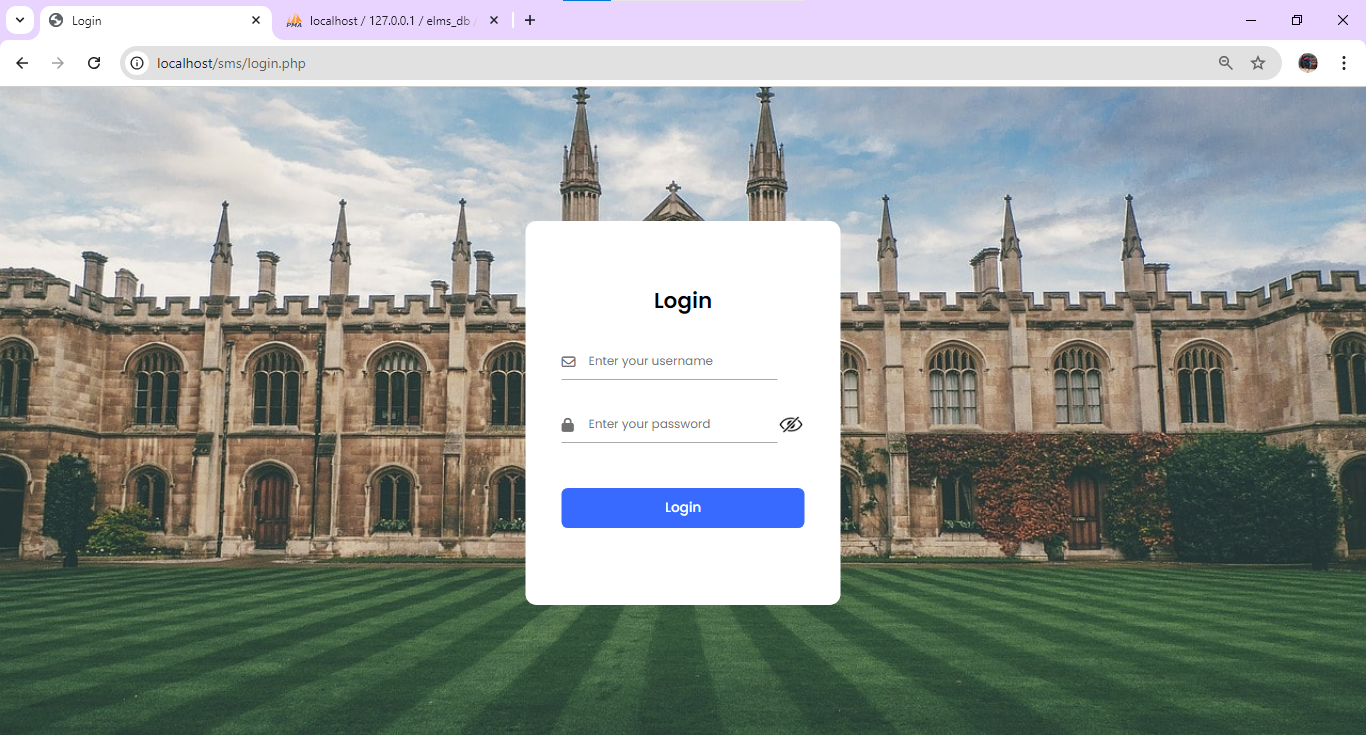
**Fees**

**Test Case and Validation**

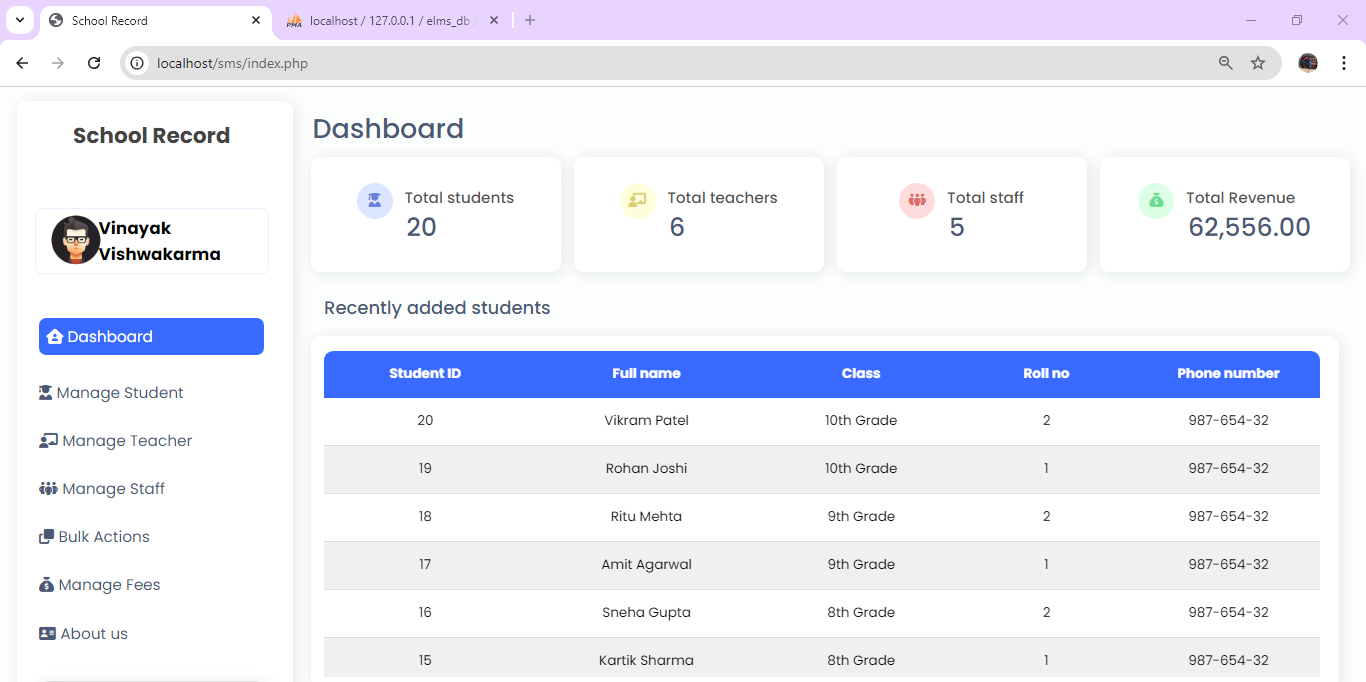
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case**  **id** | **Unit**  **to**  **test** | **Assumption** | **Test**  **Data** | **Steps to Execute** | **Accepted**  **Result** | **Actual**  **Result** | **Pass/**  **Fail** |
| 1 | Login | Username:  ”admin”  Password:  ”admin123” | Username:  ”admin”  Password:  ”admin123” | 1.Open Login Window  2.Enter Username  3.Enter Password  4.Click on Login  Button | Message:  Open’s Dashboad | Message:  Open’s Dashboard | pass |
|  |  |  | Username:  ”admin”  Password:  ”12345” | 1.Open Login Window  2.Enter Username  3.Enter Password  4.Click on Login  Button | Message:  Login failed. Please check your username & password | Message:  Login failed. Please check your username & password | pass |
| 2 | Student | Id: 1  Fullname: XYZ  Parentname: ABC  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Class: 10th  Rollno: 2 | Id: 1  Fullname: XYZ  Parentname: ABC  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Class: 10th  Rollno: 2 | 1.Visit Student  Form  2.Add Student Form info  3.Add Student Form | Message:  DATA INSERTED  SUCCESSFULLY | Message:  DATA INSERTED  SUCCESSFULLY | pass |
|  |  |  | Tr Id:  Fullname:  Parentname:  Email:  Phoneno:  Gender:  Address:  Class:  Rollno: | 1.Visit Student  Form  2.Add Student Form info  3.Add Student Form | Message:  PLEASE  ENTER THE  RECORD | Message:  PLEASE  ENTER THE  RECORD | pass |
| 3 | Teacher | Tr Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | Tr Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | 1.Visit Teacher  Form  2.Add Teacher Form info  3.Add Teacher Form | Message:  DATA INSERTED  SUCCESSFULLY | Message:  DATA INSERTED  SUCCESSFULLY | pass |
|  |  |  | Id:  Fullname:  Email:  Phoneno:  Gender:  Address: | 1.Visit Teacher  Form  2.Add Teacher Form info  3.Add Teacher Form | Message:  PLEASE  ENTER THE  RECORD | Message:  PLEASE  ENTER THE  RECORD | pass |
| 3 | Staff | Stf Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | Stf Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | 1.Visit Staff  Form  2.Add Staff Form info  3.Add Staff Form | Message:  DATA INSERTED  SUCCESSFULLY | Message:  DATA INSERTED  SUCCESSFULLY | pass |
|  |  |  | Stf Id:  Fullname:  Email:  Phoneno:  Gender:  Address: | 1.Visit Staff  Form  2.Add Staff Form info  3.Add Staff Form | Message:  PLEASE  ENTER THE  RECORD | Message:  PLEASE  ENTER THE  RECORD | pass |
| 3 | Staff | Stf Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | Stf Id: 1  Fullname: XYZ  Email: [abc@gmail.com](mailto:abc@gmail.com)  Phoneno: 256987156  DOB: 21-06-2023  Gender: Male  Address: Chiplun  Qualification: Msc | 1.Visit Staff  Form  2.Add Staff Form info  3.Add Staff Form | Message:  DATA INSERTED  SUCCESSFULLY | Message:  DATA INSERTED  SUCCESSFULLY | pass |
|  |  |  | Stf Id:  Fullname:  Email:  Phoneno:  Gender:  Address: | 1.Visit Staff  Form  2.Add Staff Form info  3.Add Staff Form | Message:  PLEASE  ENTER THE  RECORD | Message:  PLEASE  ENTER THE  RECORD | pass |
| 4 | Fees | Id: 1  Student rollno: 1  Student name: XYZ  Paid: 5000  Total: 10000  Remaining: 5000  Status: Remaining | Id: 1  Student rollno: 1  Student name: XYZ  Paid: 5000  Total: 10000  Remaining: 5000  Status: Remaining | 1.Visit Manage fees form  Form  2.Add Student Form button  3.Add Student Form | Message:  DATA INSERTED  SUCCESSFULLY | Message:  DATA INSERTED  SUCCESSFULLY | pass |
|  |  |  | Id:  Student rollno:  Student name:  Paid:  Total:  Remaining:  Status: | 1.Visit Manage fees form  Form  2.Add Student Form button  3.Add Student Form | Message:  PLEASE  ENTER THE  RECORD | Message:  PLEASE  ENTER THE  RECORD | pass |

**Screen Layout**

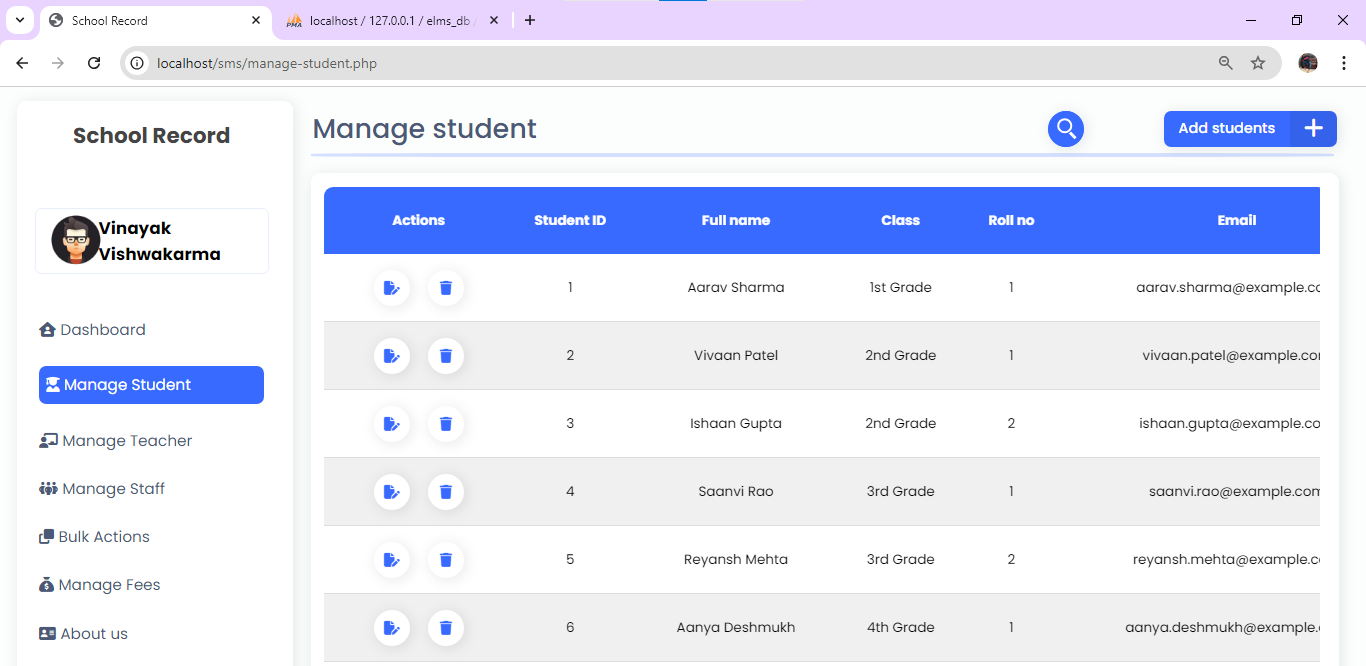
**Login:**

****

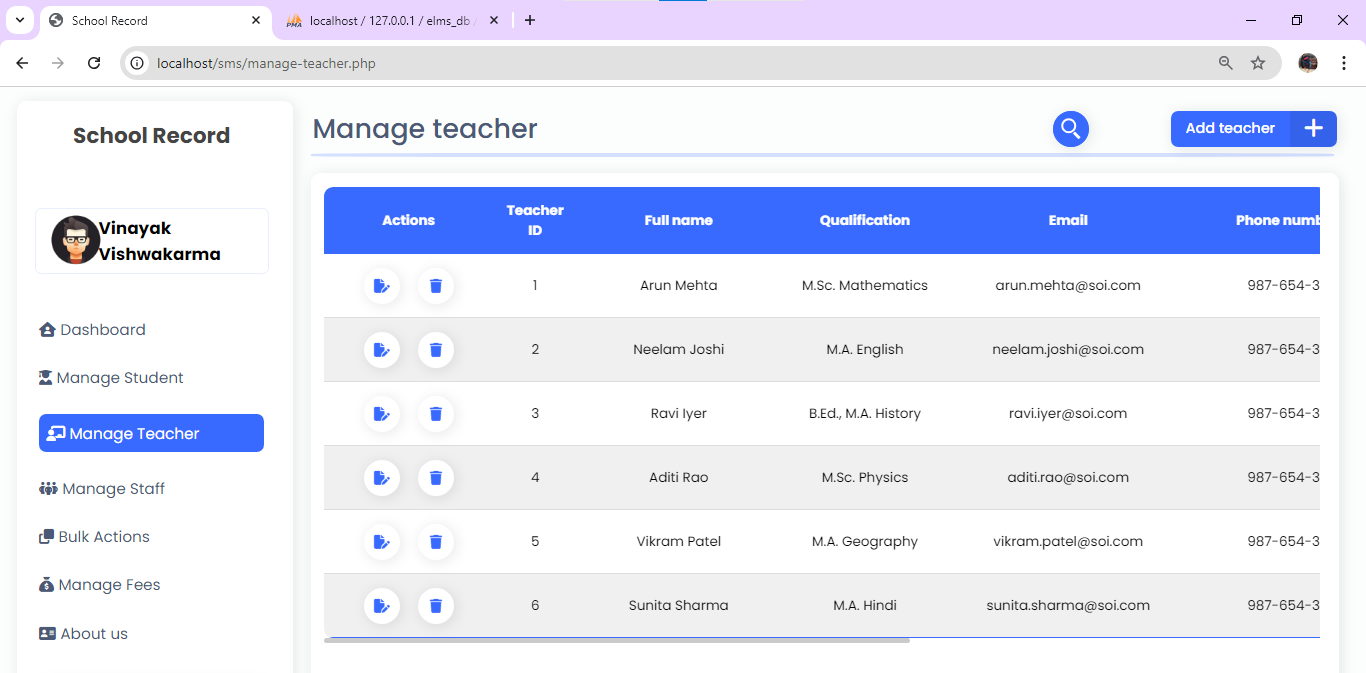
**Admin Dashboard:**

****

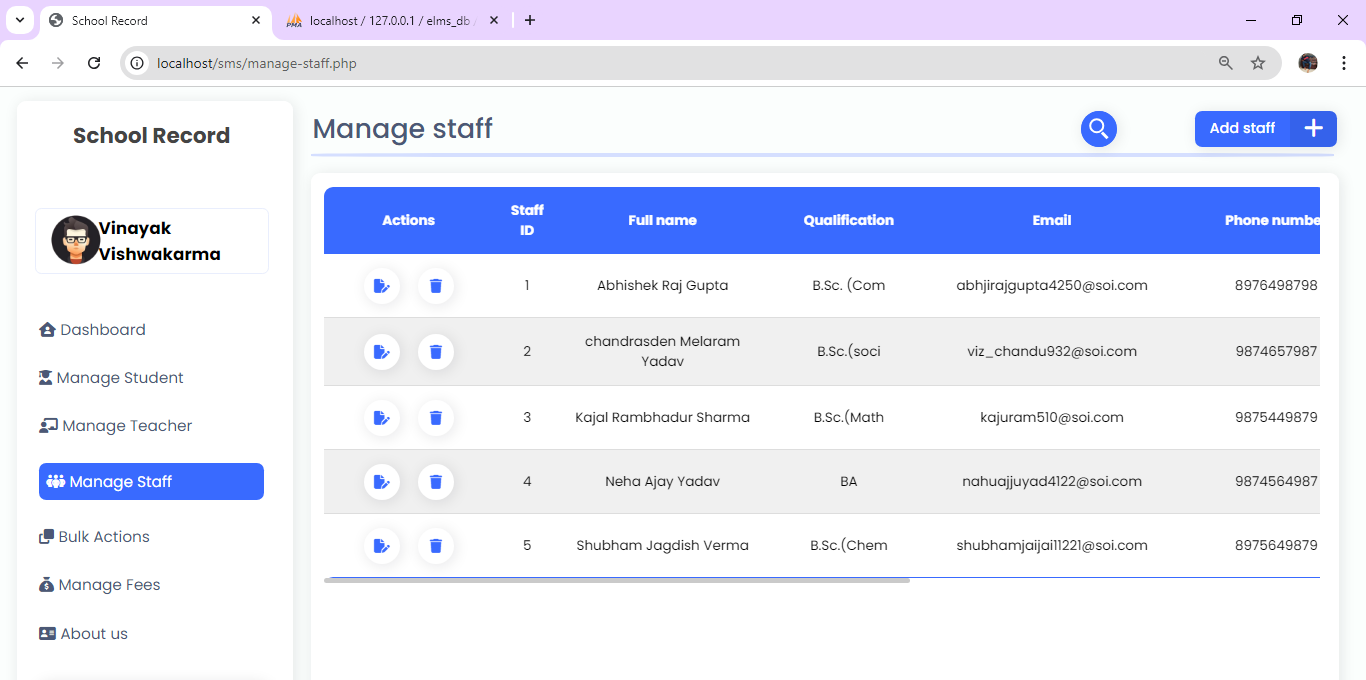
**Manage Students:**

****

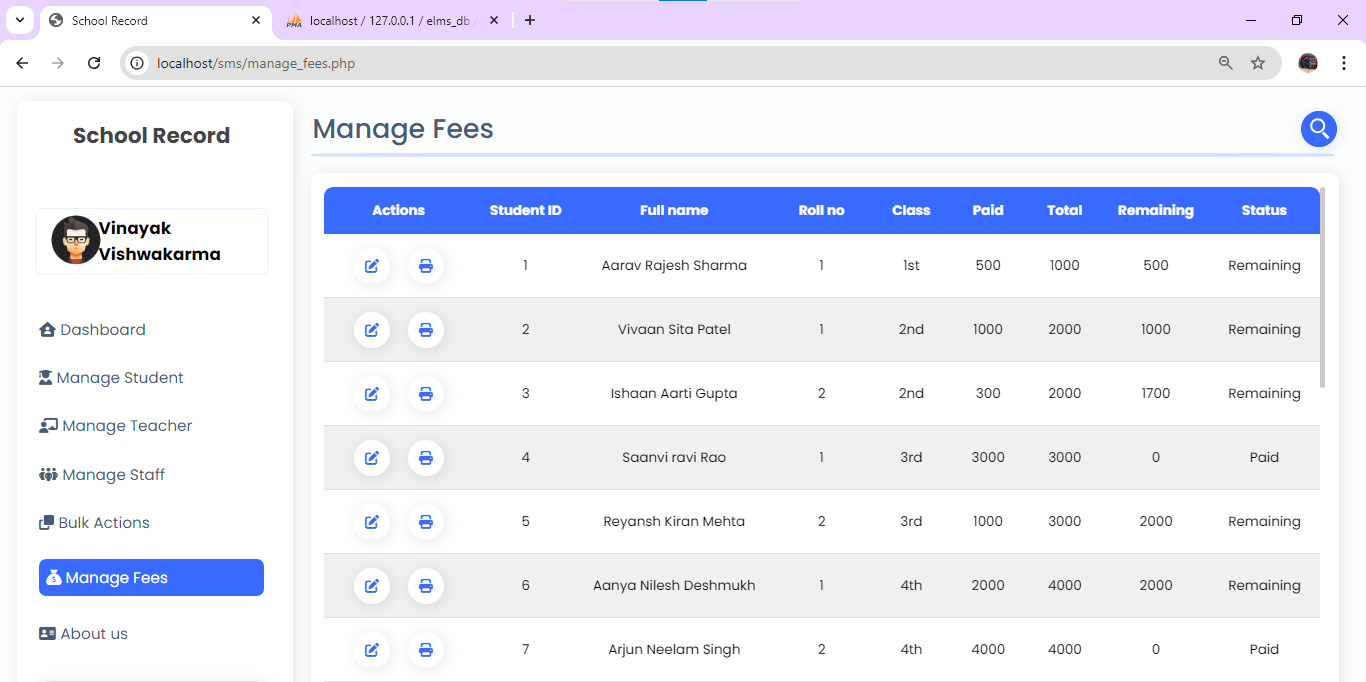
**Manage Teacher:**

****

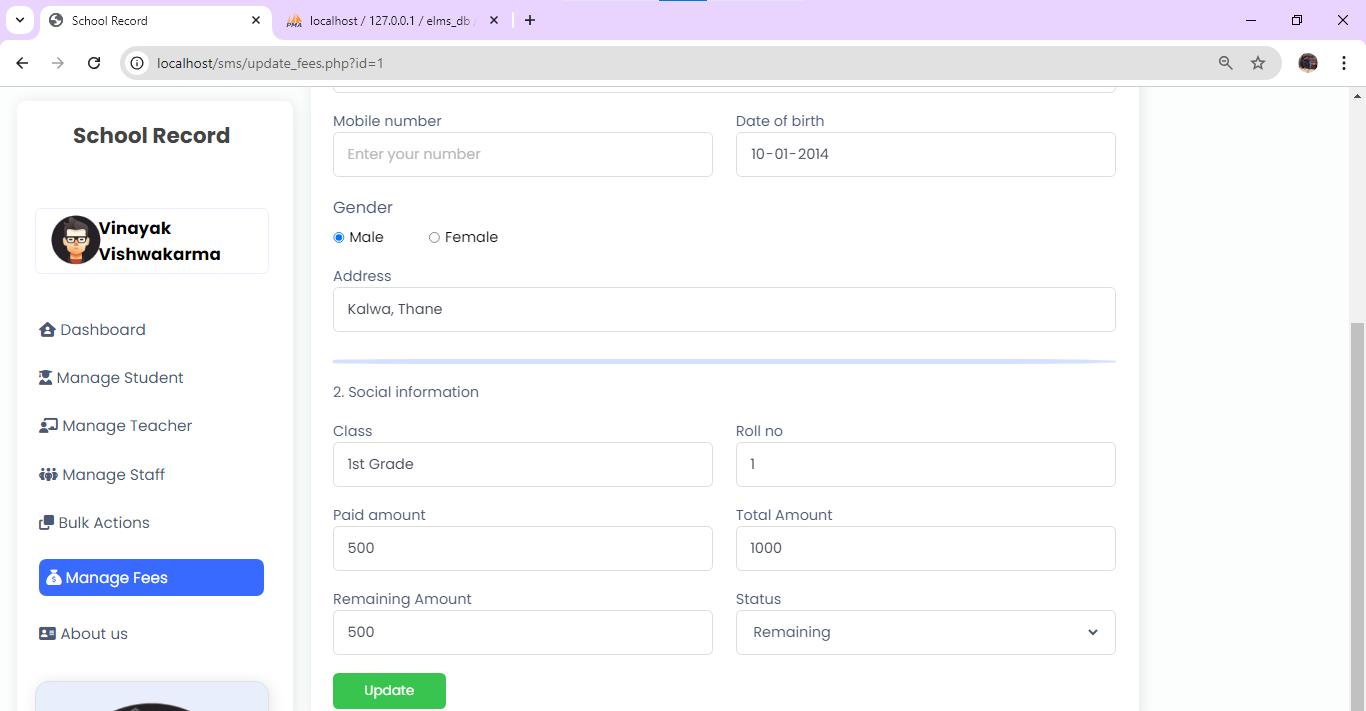
**Manage Staff:**

****

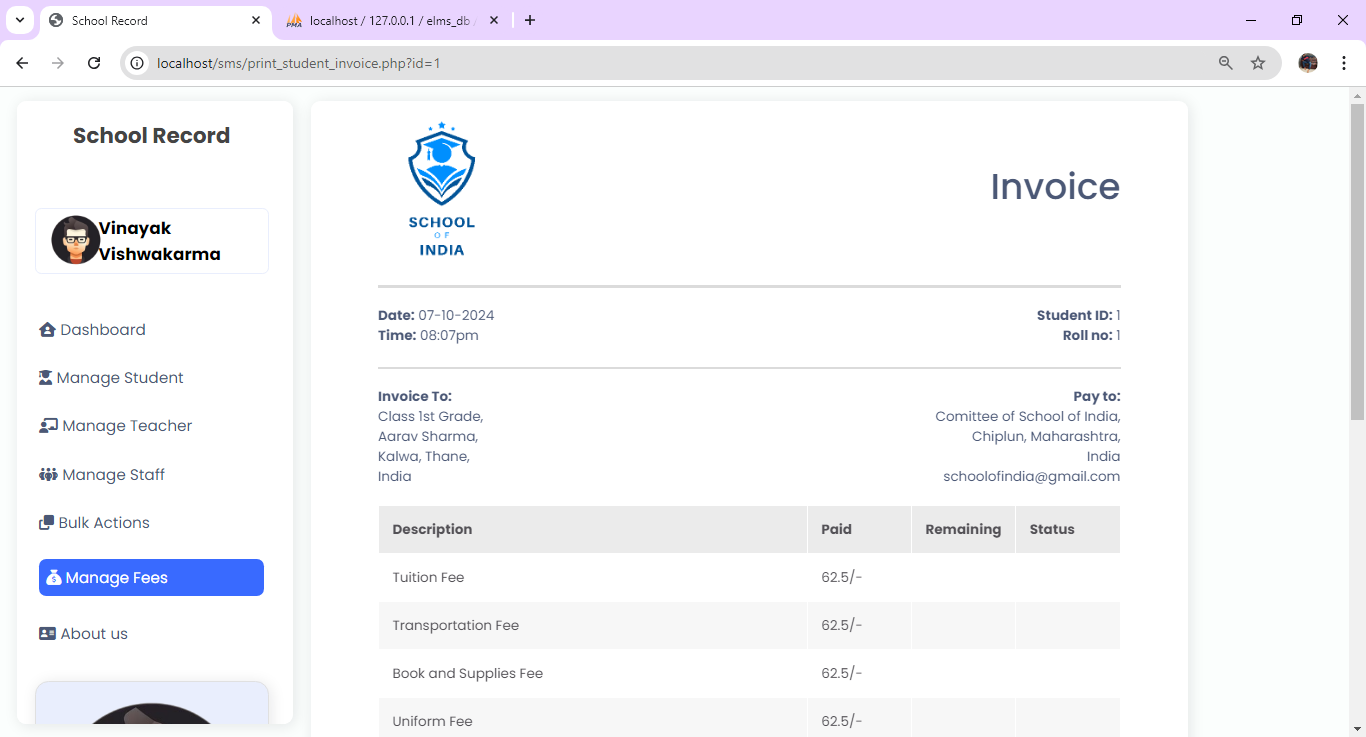
**Manage Fees:**

****

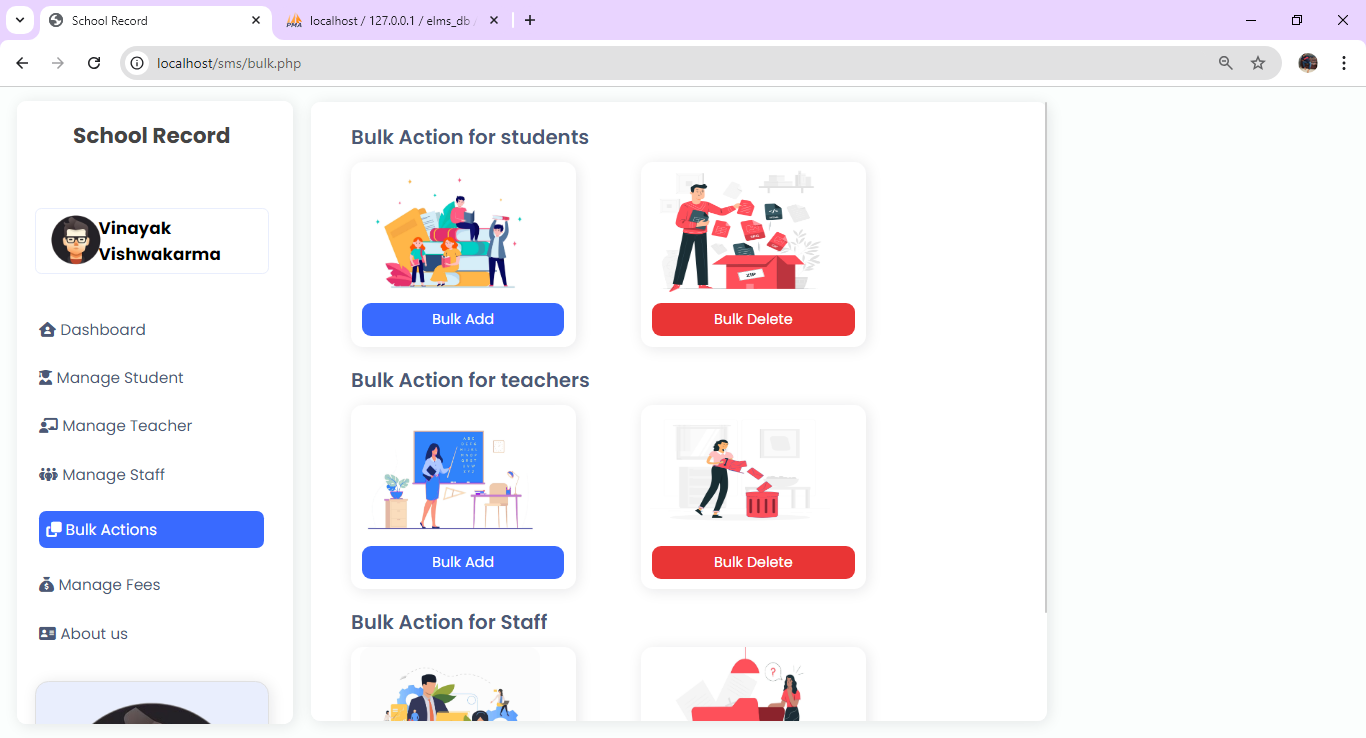
**Edit Fees:**

****

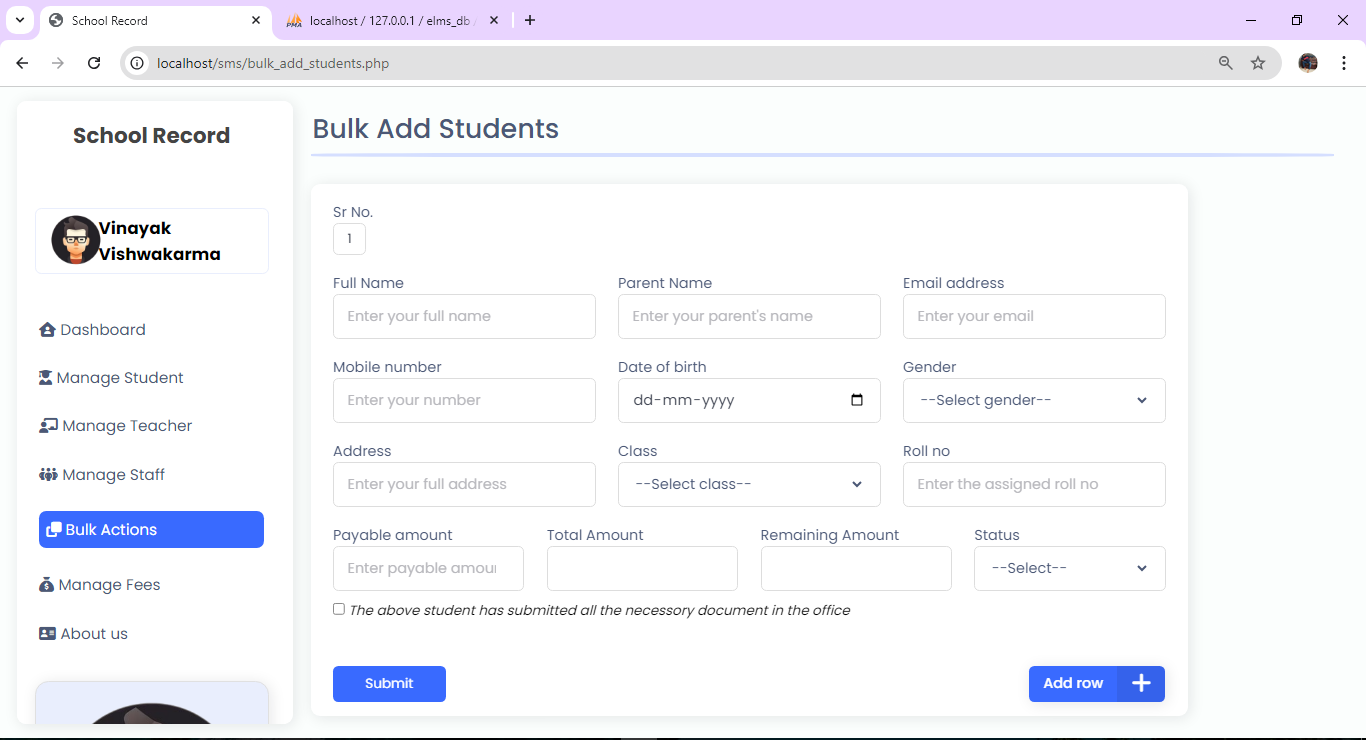
**Print Invoice:**

****

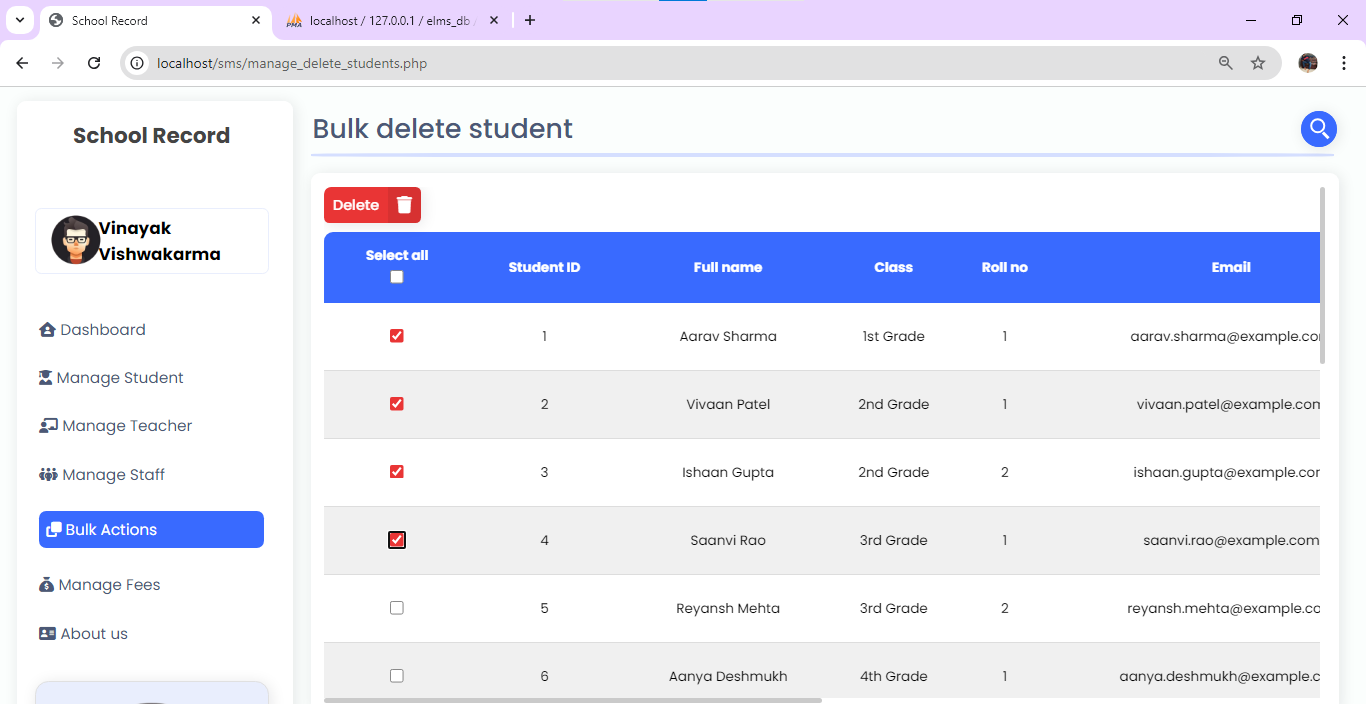
**Manage Bulk:**

****

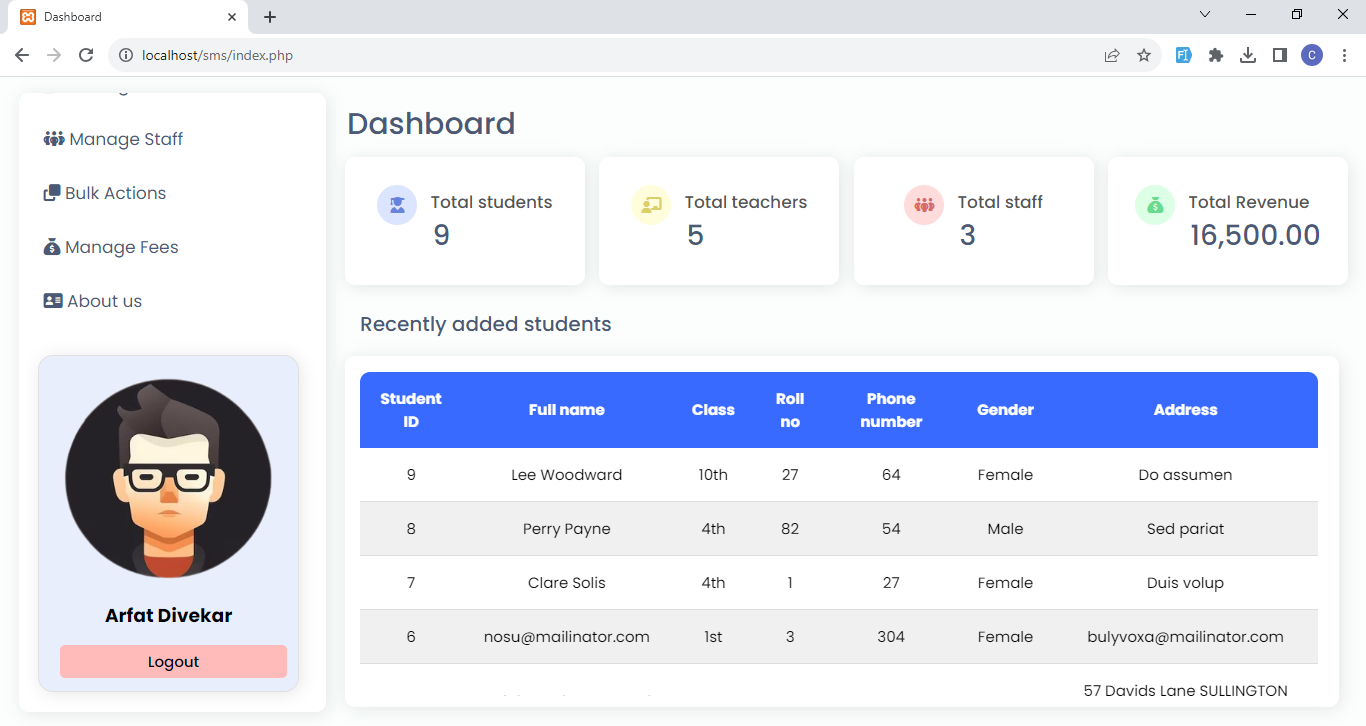
**Bulk add:**

****

**Bulk Delete:**

****

**Logout:**

****

**Future Enhancement**

**1. Mobile Application Integration:**

A dedicated mobile application can be developed to allow parents, teachers, and students to access the school management system on their smartphones. This would increase convenience and ensure better communication between all stakeholders.

**2. AI-Driven Analytics:**

Incorporate AI-driven analytics to predict student performance, identify students at risk of dropping out, and provide personalized learning suggestions based on academic history. AI could also help in predicting trends in enrollment and staffing needs.

**3. Biometric Attendance System:**

Integrating biometric systems (fingerprint or facial recognition) for both students and staff to enhance the accuracy of attendance management and reduce instances of fraud.

**4. Automated Alerts and Notifications:**

A system for sending automated notifications to parents and staff regarding important events, deadlines, or fee due dates via SMS, email, or in-app notifications. This would improve communication and ensure that no critical information is missed.

**5. Cloud Backup and Synchronization:**

Upgrade the system to use cloud technology for real-time backup and synchronization, ensuring that data is secure and accessible from any location without the risk of loss due to hardware failure or accidents.

**6. AI Chatbot for User Support:**

Implement a chatbot feature within the system that can provide quick answers to frequently asked questions by parents or students, assist with navigating the system, and offer basic troubleshooting help.

**7. Advanced Reporting and Visualization Tools:**

Enhance the reporting features by including customizable, graphical data visualizations. This would help administrators and teachers analyze trends, such as academic progress, attendance, or financial health, in a more intuitive manner.

**8. Integration with E-learning Platforms:**

Integrate the system with popular e-learning platforms to manage online classes, assignments, and student evaluations, enabling a hybrid learning model for schools.

**9. Parent-Teacher Communication Portal:**

A portal dedicated to parent-teacher interaction, where both parties can schedule meetings, discuss student performance, and share feedback, thus fostering better collaboration for student development.

**10. Multiple Language Support:**

Add support for multiple languages within the system to cater to a broader audience, ensuring that non-native English speakers can use the system without difficulty.

**11. Data Analytics for Fee Management:**

A feature that predicts patterns in fee payments, helps identify defaulters early, and suggests financial plans or reminders for parents based on their past payment behaviors.

**12. Remote Learning Management:**

Include features for managing remote classes, assignment submissions, and evaluations, making the system adaptable to both physical and virtual learning environments.

**Requirements**

**Hardware Requirements:**

Processor: Intel(R) Core(TM)2 Duo CPU

RAM: 4GB

Hard Disk: GB

**Software Requirements:**

Operating System: Windows 7/8/10

Frontend: HTML, CSS, JavaScript

Backend: PHP

**REFRENCE:**

* [**https://www.w3schools.com/**](https://www.w3schools.com/)
* [**https://www.codeofaninja.com/**](https://www.codeofaninja.com/)
* [**https://sider.ai/**](https://sider.ai/)

**BIBLOGRAPHY:**

* [The Joy of PHP Programming: A Beginner’s Guide – by Alan Forbes](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#joy-php-programming)
* [PHP & MySQL Novice to Ninja – by Kevin Yank](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#novice-to-ninja)
* [Head First PHP & MySQL – by Lynn Beighley & Michael Morrison](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#first-php-mysql)
* [PHP: A Beginner’s Guide – by Vikram Vaswani](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#vikram-vaswani)
* [Murach’s PHP & MySQL – by Joel Murach & Ray Harris](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#murach-harris)
* [Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites – by Robin Nixon](https://www.cloudways.com/blog/best-php-books-to-help-you-learn/#creating-dynamic-websites)