

**Dr. D. Y. Patil Institute of Technology,
Pimpri, Pune-411018**
Department of Computer Engineering

A MINI PROJECT REPORT ON
Student Management System

SUBMITTED BY

Mr. Gawade Disha Dadaso Exam Seat No - T1902404271



DEPARTMENT OF COMPUTER ENGINEERING

Dr. D. Y. PATIL INSTITUTE OF TECHNOLOGY,

PIMPRI, PUNE

SAVITRIBAI PHULE PUNE UNIVERSITY

2024 -2025



This is to certify that the project report entitled
“Student Management System”

Submitted by

Mr. Gawade Disha Dadaso Exam No : **T1902404271**

is a bonafide student of this institute and the work has been carried out by him under the supervision of **Prof. Pranal Kakde** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Computer Engineering.

(Prof. Pranal Kakde)
Guide
Department of Computer Engineering

(Dr. Vinod Kimbahune)
Head,
Department of Computer Engineering

(Dr. Lalit Kumar Wadhwa)
Principal,
Dr. D. Y. PATIL INSTITUTE OF TECHNOLOGY,
PIMPRI, PUNE

Place : Pune

Date :22/10/2024

DECLARATION OF THE STUDENT

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Mr. Gawade Disha Dadaso

T1902404271

Date:- 22/10/2024

ABSTRACT

The Student Management System (SMS) is an advanced software solution designed to efficiently manage and maintain student data within schools, colleges, and universities. It automates the process of handling various student-related information, including personal details, academic records, course enrolment, and college details. By digitizing these processes, the system reduces the time, effort, and cost traditionally involved in manual data handling.

The SMS serves as a central hub for the administration to track and update student information in real-time. This system also streamlines communication by providing easy access to updated records and information for both students and faculty. The adoption of Student Information Systems (SIS) has significantly improved the performance and efficiency of educational institutions by reducing administrative workload, enabling better data organization, and fostering transparency.

In recent years, the widespread use of SMS has revolutionized the education industry, transforming how student records are managed and how information is shared. This tool ensures the seamless flow of academic processes, making it an essential part of modern educational institutions.

TABLE OF CONTENTS

Sr. No.	Title of Chapter	Page No.
01	Introduction	6
02	Software Requirements Specification	6 - 9
03	ER Diagram	10
04	Results	11
	4.1 Outcomes	11 - 15
	4.2 Screen Shots	11 - 15
05	Conclusions	16
	5.1 References	17

1. Introduction

A Student Management System can handle all the details about a student. The details include College details, Course details, Students personal details, Academic details etc. If we do these activities manually it is very time consuming and costly so this system can help to manage all data of student.

Schools and Universities are the foundation of knowledge and an educational body on which students rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

Most schools and Universities count on an advanced software tool known as 'Student Information System (SIS)' to keep all their student records.

Over the recent years, the performance and efficiency of the education industry have been enhanced by using the Student Management System. This tool has productively taken over the workload of the admin department with its well-organized, easy, and reliable online school management software.

•Objective

The primary objective of the Student Management System (SMS) is to provide an efficient, user-friendly platform for managing and maintaining student-related data within educational institutions. Key objectives include:

1. **Automate Administrative Tasks:** Streamline and automate routine tasks such as student registration, course enrolment, academic record management, and fee processing to reduce manual effort and errors.
2. **Centralized Data Management:** Create a centralized database that holds comprehensive information about students, including personal details, academic performance, attendance, and course details, making it easily accessible to authorized users.
3. **Improve Communication:** Facilitate better communication between students, faculty, and administrative staff by providing a platform to share information and updates quickly and efficiently.
4. **Enhance Data Security:** Ensure the security and privacy of student data by providing controlled access to sensitive information, reducing the risk of unauthorized access and data breaches.
5. **Increase Efficiency:** Optimize the overall functioning of academic and administrative processes, resulting in increased productivity, faster decision-making, and a reduction in administrative costs.

2. Software Requirements Specification /Methodology

Software Requirements:

- Operating System: Windows 11
- Software: jdk 1.6, notepad, MS office, MS access.

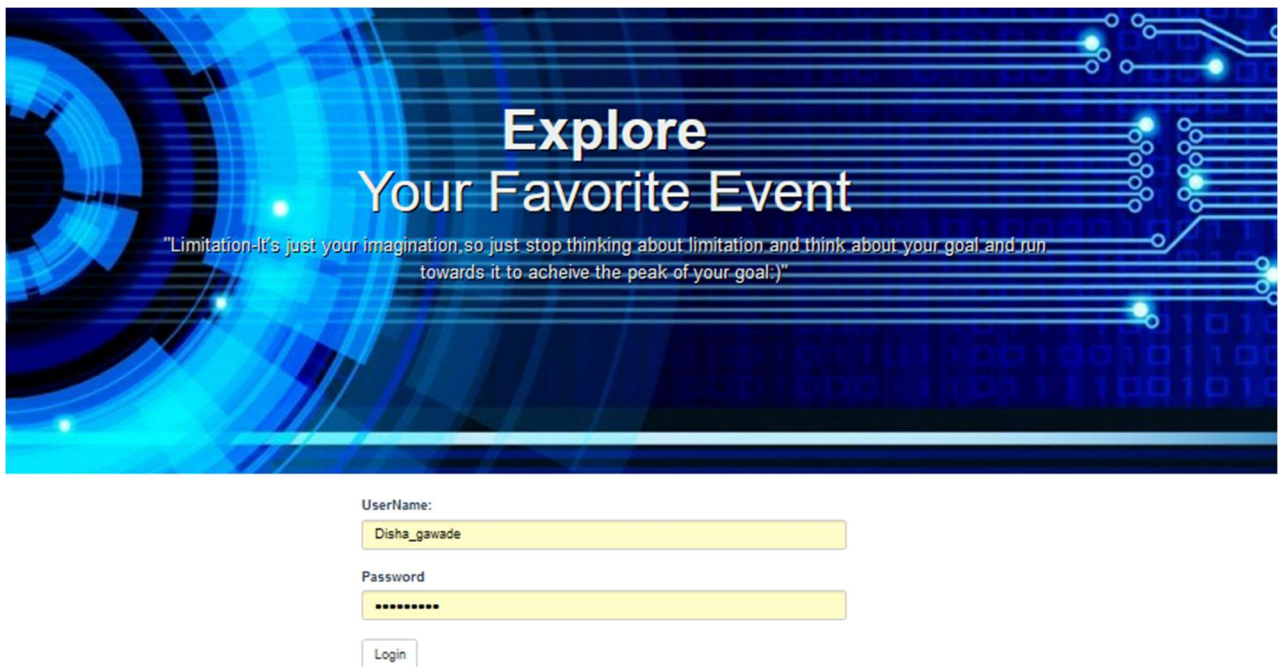
Hardware Requirements:

- Processor: AMD A9-9420 RADEON R5, 5 COMPUTER CORES 2C+3G 3.00GHz
- RAM : 4 GB

IMPLEMENTATION AND RESULT

Following are the screens of the Student Management System where you can see all the features of this system in use and you can also see the GUI of the system:

1. **Login frame** – This is the login frame of this system where Student have to enter the required credentials to have access for the main dashboard.



Explore
Your Favorite Event

"Limitation-It's just your imagination, so just stop thinking about limitation and think about your goal and run towards it to achieve the peak of your goal:)"

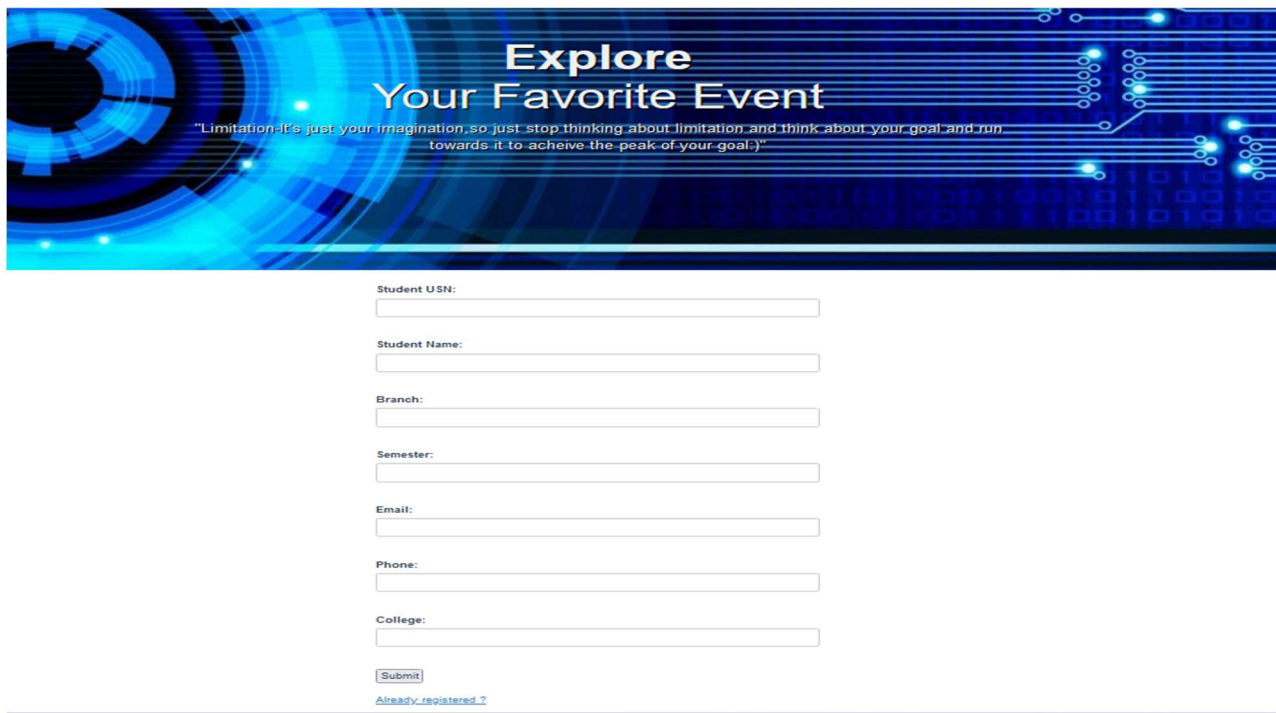
UserName:
Disha_gawade

Password

Login

Fig. 1

2. **Main Dashboard** – After login in, user is directed to the main dashboard of this system where user can perform various operations like adding an Students, deleting an Students



The image shows a registration form titled "Explore Your Favorite Event" with a motivational quote: "Limitation-It's just your imagination,so just stop thinking about limitation and think about your goal and run towards it to achieve the peak of your goal:)". The form includes input fields for Student USN, Student Name, Branch, Semester, Email, Phone, and College. A "Submit" button and a link "Already registered ?" are at the bottom.

Student USN:

Student Name:

Branch:

Semester:

Email:

Phone:

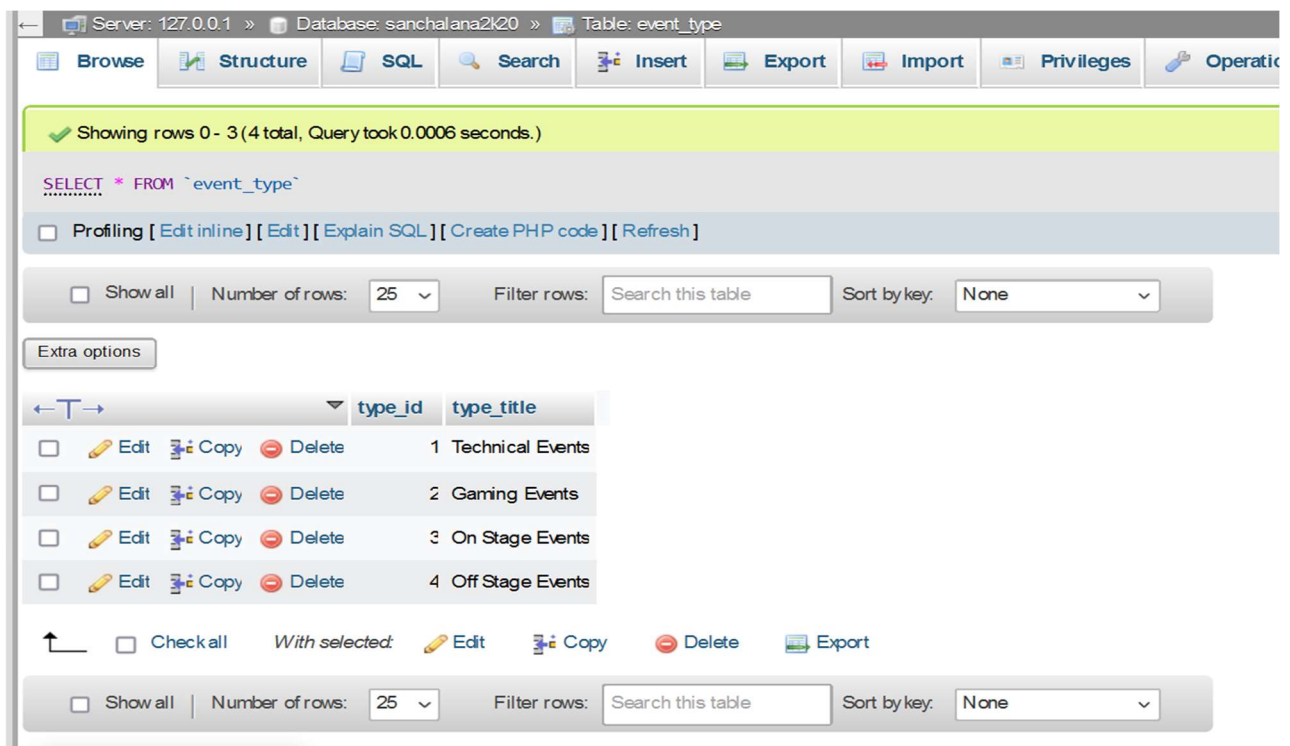
College:

[Already registered ?](#)

Fig. 2

3. **Add student** – Here user have to enter all the required credentials to add a new employee to the system.

Fig. 3

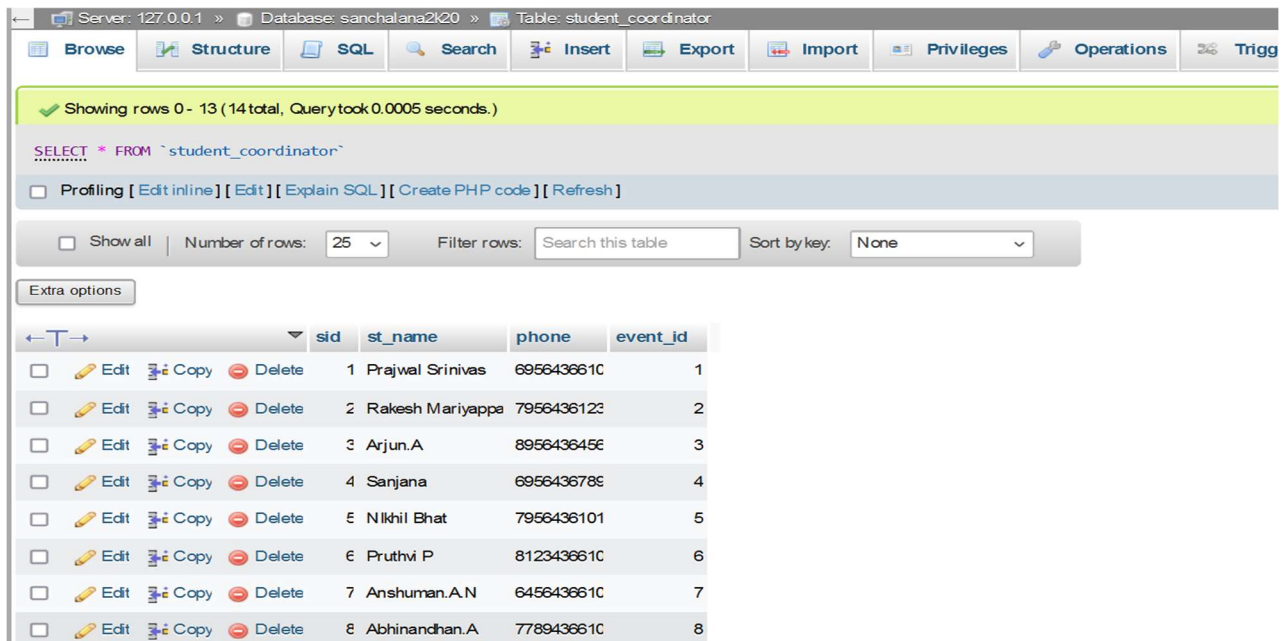


The image shows a database management interface for the 'event_type' table. The top bar indicates the server is 127.0.0.1, the database is sanchalana2k20, and the table is event_type. The interface includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operations. A status bar shows "Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)". The SQL query is "SELECT * FROM `event_type`". Below the query are options for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. A table view shows 4 rows with columns type_id and type_title. The table contains the following data:

type_id	type_title
1	Technical Events
2	Gaming Events
3	On Stage Events
4	Off Stage Events

Below the table are options for Show all, Number of rows (25), Filter rows (Search this table), and Sort by key (None). There are also buttons for Edit, Copy, Delete, and Export.

4. **Remove student** – User has to enter the student id in order to delete his information from the system.

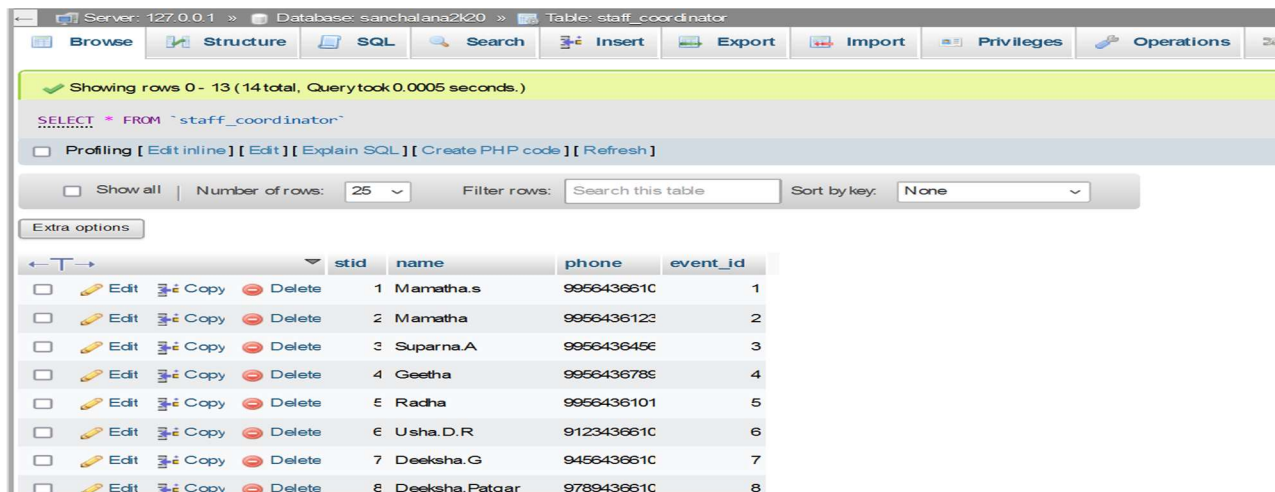


The screenshot shows the phpMyAdmin interface for the 'student_coordinator' table. The table has 14 rows and 4 columns: sid, st_name, phone, and event_id. The interface includes a top navigation bar with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the navigation bar, there is a status bar indicating 'Showing rows 0 - 13 (14 total, Query took 0.0005 seconds.)'. The SQL query editor shows 'SELECT * FROM `student_coordinator`'. Below the query editor, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (Search this table), and 'Sort by key' (None). The table data is displayed in a grid with columns for 'sid', 'st_name', 'phone', and 'event_id'. Each row has a checkbox for selection and buttons for 'Edit', 'Copy', and 'Delete'.

	sid	st_name	phone	event_id
<input type="checkbox"/>	1	Prajwal Srinivas	6956436610	1
<input type="checkbox"/>	2	Rakesh Mariyappa	7956436123	2
<input type="checkbox"/>	3	Arjun.A	8956436456	3
<input type="checkbox"/>	4	Sanjana	6956436789	4
<input type="checkbox"/>	5	Nikhil Bhat	7956436101	5
<input type="checkbox"/>	6	Pruthvi P	8123436610	6
<input type="checkbox"/>	7	Anshuman.A.N	6456436610	7
<input type="checkbox"/>	8	Abhinandhan.A	7789436610	8

Fig. 4

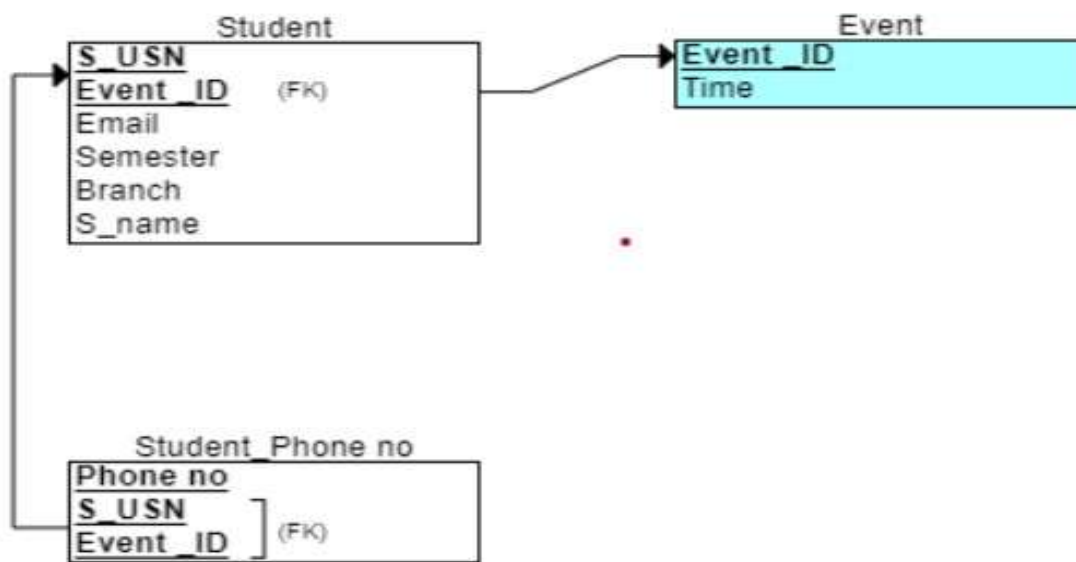
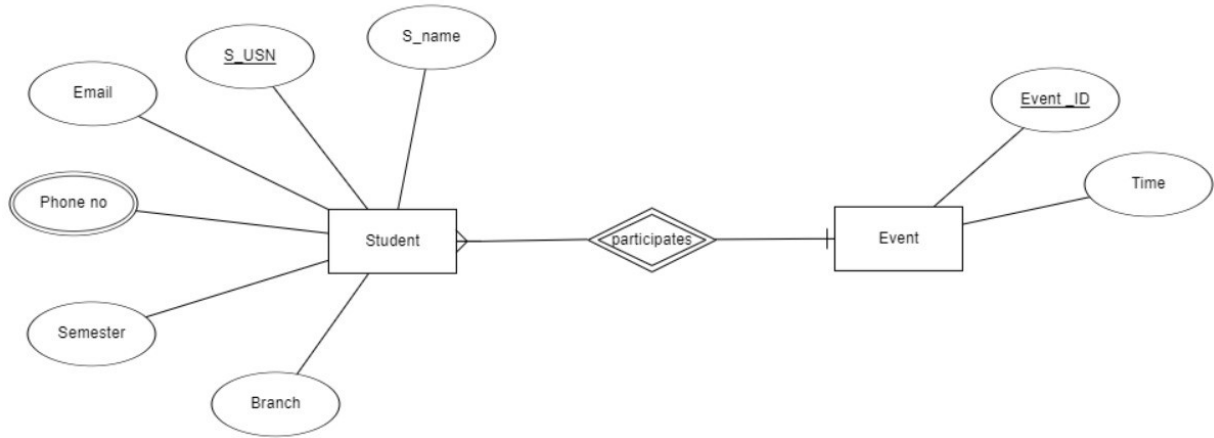
5. **View and update student** – In order to view and update employee information, the user have to enter student ID.



The screenshot shows the phpMyAdmin interface for the 'staff_coordinator' table. The table has 8 rows and 4 columns: stid, name, phone, and event_id. The interface includes a top navigation bar with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the navigation bar, there is a status bar indicating 'Showing rows 0 - 13 (14 total, Query took 0.0005 seconds.)'. The SQL query editor shows 'SELECT * FROM `staff_coordinator`'. Below the query editor, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (Search this table), and 'Sort by key' (None). The table data is displayed in a grid with columns for 'stid', 'name', 'phone', and 'event_id'. Each row has a checkbox for selection and buttons for 'Edit', 'Copy', and 'Delete'.

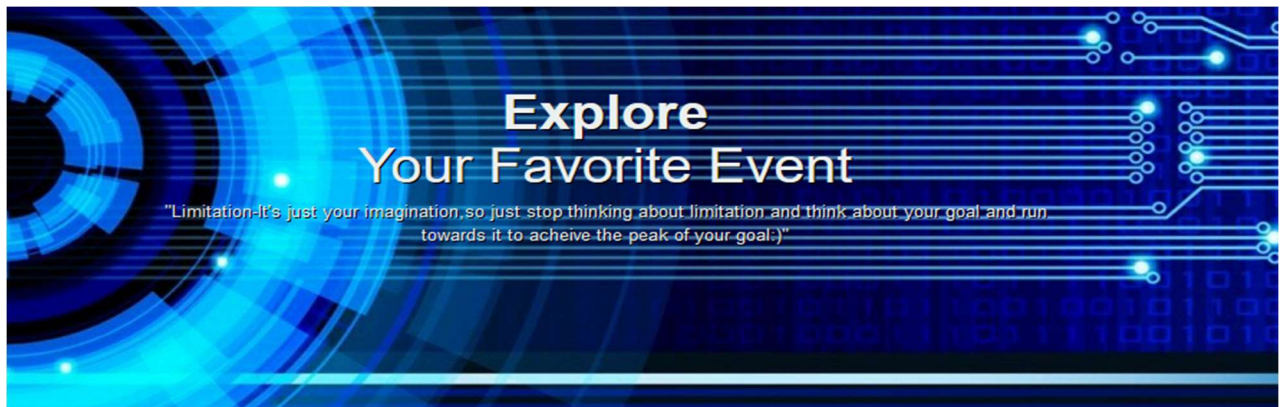
	stid	name	phone	event_id
<input type="checkbox"/>	1	Mamatha.s	9956436610	1
<input type="checkbox"/>	2	Mamatha	9956436123	2
<input type="checkbox"/>	3	Suparna.A	9956436456	3
<input type="checkbox"/>	4	Geetha	9956436789	4
<input type="checkbox"/>	5	Radha	9956436101	5
<input type="checkbox"/>	6	Usha.D.R	9123436610	6
<input type="checkbox"/>	7	Deeksha.G	9456436610	7
<input type="checkbox"/>	8	Deeksha.Patgar	9789436610	8

2. ER DIAGRAM:-



1. RESULTS:-

4.1 OUTCOMES:



UserName:

Password



Student USN:

Student Name:

Branch:

Semester:

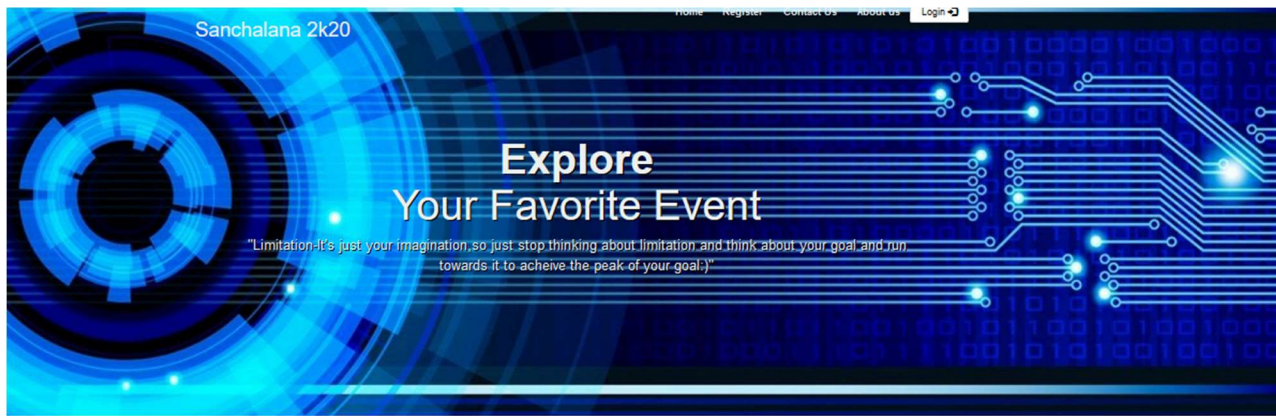
Email:

Phone:

College:

[Already registered ?](#)





Register your Favourite events:



Technical Events

EMBRACE YOUR TECHNICAL SKILLS BY PARTICIPATING IN OUR DIFFERENT TECHNICAL EVENTS!

[View Technical Events](#)

Server: 127.0.0.1 » Database: sanchalana2k20 » Table: event_type

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)

✓ Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT * FROM `event_type`
```

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Sort by key: None

Extra options

	type_id	type_title
<input type="checkbox"/> Edit Copy Delete	1	Technical Events
<input type="checkbox"/> Edit Copy Delete	2	Gaming Events
<input type="checkbox"/> Edit Copy Delete	3	On Stage Events
<input type="checkbox"/> Edit Copy Delete	4	Off Stage Events

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Sort by key: None

Browse
 Structure
 SQL
 Search
 Insert

Showing rows 0 - 13 (14 total, Query took 0.0018 seconds.)

SELECT * FROM `event_info`

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all
 |
 Number of rows:
 25
 Filter rows:
 Search this table

Extra options

Server: 127.0.0.1 » Database: sanchalana2k20 » Table: staff_coordinator
 Browse
 Structure
 SQL
 Search
 Insert
 Export
 Import
 Privileges
 Operations

Showing rows 0 - 13 (14 total, Query took 0.0005 seconds.)

 SELECT * FROM `staff_coordinator`

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all
 |
 Number of rows:
 25
 Filter rows:
 Search this table
 Sort by key:
 None

Extra options

			stdid	name	phone	event_id
<input type="checkbox"/>				1 Mamatha.s	9956436610	1
<input type="checkbox"/>				2 Mamatha	9956436123	2
<input type="checkbox"/>				3 Suparna.A	9956436456	3
<input type="checkbox"/>				4 Geetha	9956436789	4
<input type="checkbox"/>				5 Radha	9956436101	5
<input type="checkbox"/>				6 Usha.D.R	9123436610	6
<input type="checkbox"/>				7 Deeksha.G	9456436610	7
<input type="checkbox"/>				8 Deeksha.Patgar	9789436610	8

Server: 127.0.0.1 » Database: sanchalana2k20 » Table: student_coordinator

Browser

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Trigg

Showing rows 0 - 13 (14 total, Query took 0.0005 seconds.)

SELECT * FROM `student_coordinator`

☐ Profiling

[\[Edit inline \]](#)

[\[Edit \]](#)

[\[Explain SQL \]](#)

[\[Create PHP code \]](#)

[\[Refresh \]](#)

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

←

→

sid

st_name

phone

event_id

☐

[Edit](#)

[Copy](#)

[Delete](#)

1

Prajwal Srinivas

6956436610

1

☐

[Edit](#)

[Copy](#)

[Delete](#)

2

Rakesh Mariyappa

7956436123

2

☐

[Edit](#)

[Copy](#)

[Delete](#)

3

Arjun.A

8956436456

3

☐

[Edit](#)

[Copy](#)

[Delete](#)

4

Sanjana

6956436789

4

☐

[Edit](#)

[Copy](#)

[Delete](#)

5

Nikhil Bhat

7956436101

5

☐

[Edit](#)

[Copy](#)

[Delete](#)

6

Pruthvi P

8123436610

6

☐

[Edit](#)

[Copy](#)

[Delete](#)

7

Anshuman.A.N

6456436610

7

☐

[Edit](#)

[Copy](#)

[Delete](#)

8

Abhinandhan.A

7789436610

8

2. Conclusion

- Student management System helpful to perform paperless work and manage all data of students.
- It provides easy, accurate, unambiguous and faster data access.
- It provides a user friendly, error free environment to manage large amount of data.
- The application provides appropriate information to users according to the chosen service.
- Student Management System can be used by educational institutions to maintain their student records easily.

Future Scope

The Student Management System (SMS) has significant potential for growth and improvement to further enhance the educational experience for institutions, administrators, teachers, and students. Some areas of future scope include:

1. **AI and Machine Learning Integration:** The integration of AI and machine learning could enable predictive analytics to monitor student performance, identify at-risk students, and

recommend personalized learning paths and interventions based on academic history and behavior patterns.

2. **Mobile Application Development:** Expanding the system into mobile applications for students, teachers, and administrators would improve accessibility and convenience, allowing stakeholders to access and update information from anywhere at any time.
3. **Cloud-based Solutions:** Implementing cloud-based SMS will make it easier for institutions to scale their operations, handle larger data volumes, and ensure data availability and backup without the need for complex in-house IT infrastructure.
4. **Integration with Learning Management Systems (LMS):** Seamless integration with Learning Management Systems can allow real-time synchronization of student academic records, making it easier to track coursework, assessments, and grades across multiple platforms.

References:

Reference books and websites used during the entire project.....

Reference book

Author

1) Advance Java Programming.

-Ravi Majithia

2) Advance Java Programming.

-Dr.Meenakshi A. Thalor

Web Reference -

I. <https://www.javatpoint.com/java-jdbc>

II. <https://www.cwipedia.in/2020/12/22517-notes-pdf-ajp-msbte.html>