



UNIVERSITY INSTITUTE OF COMPUTING

PROJECT REPORT ON “STUDENT MANAGEMENT SYSTEM”

Program Name: BCA
ADVANCED WEB DEVELOPMENT

23 CAH-302

SUBMITTED BY:

Name : Kapil / Navdeep / Hariom

Uid : 23BCA10584 / 23BCA10425

/23BCA10389

Branch : BCA

Section/ Group: 2-B

Semester : 5th

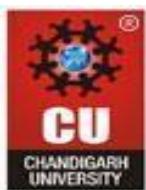
D.O.Submission : 6 Nov 2025

SUBMITTED TO:

Name : Abhineet Thakur
Designation: Assistant Professor

D.O.Submission : 6 Nov 2025

sign: _____



UNIVERSITY INSTITUTE *of*
COMPUTING
Asia's Fastest Growing University

BONAFIDE CERTIFICATE

Certified that this project report "**STUDENT MANAGEMENT SYSTEM**" is the Bonafide work of "**Kapil / Navdeep / Hariom**" under UID "**23BCA10584 / 23BCA10425 /23BCA10389**" who carried out the project work under my supervision.

.....
SIGNATURE

UIC DEPARTMENT

ASSISTANT PROFESSOR

VINOD BAGGA

ADVANCED WEB DEVELOPMENT

PROJECT BASED LEARNING REPORT ON STUDENT MANAGEMENT SYSTEM

2. Abstract

The *Student Management System* is a web-based application developed using Laravel PHP framework to simplify the management of student records. The system provides administrators with functionalities to manage student data, including adding, updating, viewing, and deleting records. It also includes user management features, authentication, and password recovery mechanisms.

The main goal of this project is to create a reliable and user-friendly interface for maintaining academic details such as marks, attendance, and fee status. The application enhances the efficiency of record-keeping and reduces manual work through digital automation.

3. Introduction

Educational institutions often face challenges in managing large volumes of student data. Manual record-keeping is prone to errors, redundancy, and inefficiency. To address these problems, this project aims to build a centralized Student Management System using Laravel PHP.

The system ensures data accuracy, quick retrieval of information, and secure management of users. Administrators can log in, add new students, update their academic details, and manage user roles effectively.

Key Features:

□

Secure login and authentication system

- CRUD operations for student records (Add, Edit, Delete, View)
 - Manage users (Admin/User roles)
 - Password recovery and update module
-

4. Problem Statement / Objectives

Problem Statement:

Manually maintaining student records can be time-consuming and error-prone. There is a need for an automated solution to store, retrieve, and manage data efficiently while ensuring security and user accessibility.

Objectives:

- To develop a web-based application for managing student information.
 - To enable CRUD functionalities for student records.
 - To design a simple, attractive, and responsive user interface.
 - To provide secure authentication and authorization mechanisms.
 - To facilitate user role management (Admin/User).
-

5. System Design / Approach

Architecture:

□

The system follows an **MVC (Model-View-Controller)** architecture provided by Laravel.

Model: Handles database operations related to students and users.

- **View:** Displays the data to users via web pages.
- **Controller:** Acts as an intermediary between the Model and View.

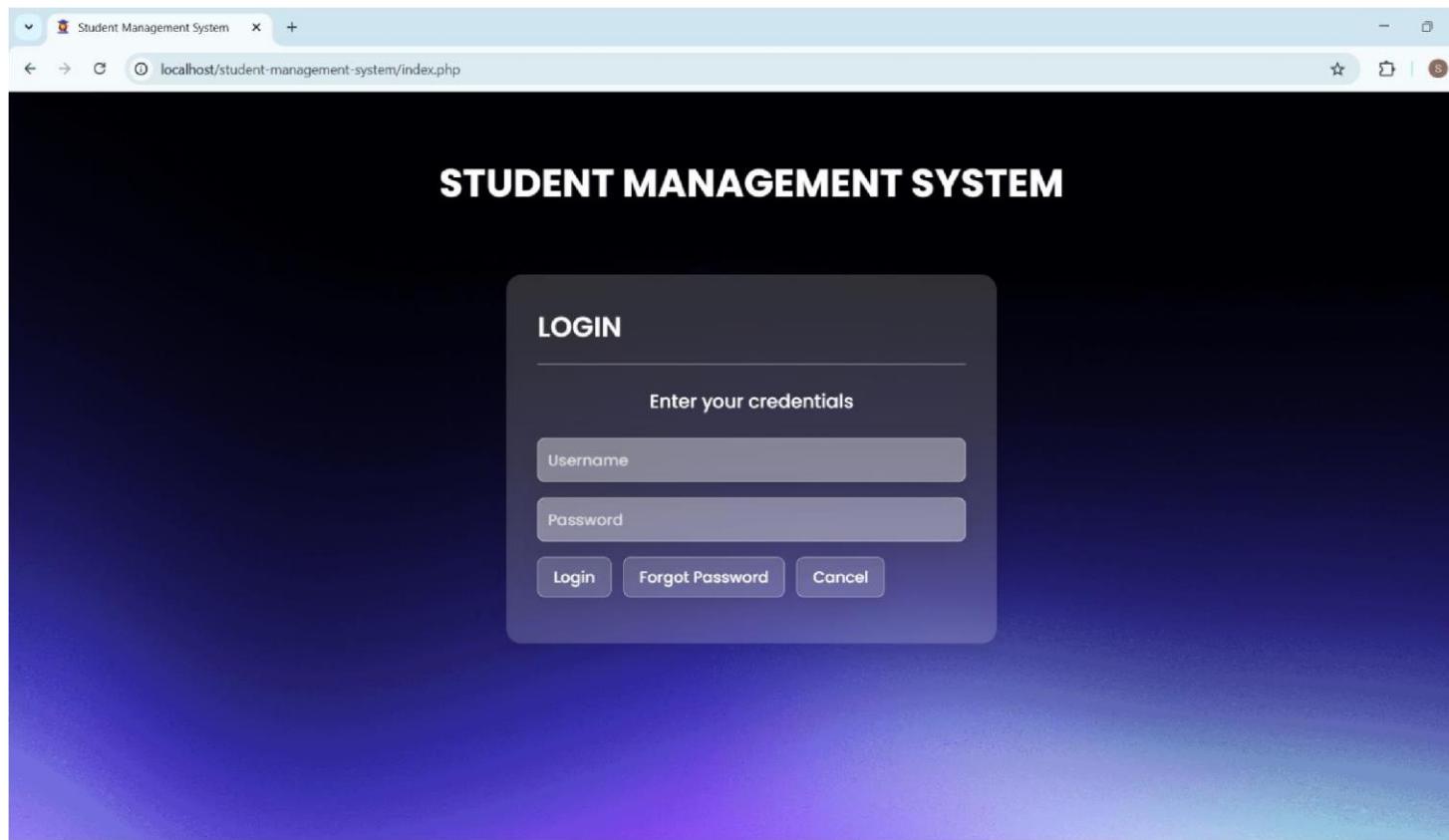
Modules:

1. **Login Module** – Authenticates users (Admin/User).
2. **Dashboard Module** – Displays all student records in tabular form.
3. **Student Management Module** – Add, edit, or delete student details.
4. **User Management Module** – Add or remove users with role-based access.
5. **Password Module** – Allows users to reset or change their passwords.

Tools Used:

- **Backend:** PHP (Laravel Framework)
- **Frontend:** HTML, CSS, JavaScript
- **Database:** MySQL
- **Server:** XAMPP / Apache
- **IDE:** Visual Studio Code

(Insert a system architecture diagram or flowchart here if available.)



6. Implementation

Main Modules & Features:

1. Login Page

- Users enter credentials to access the dashboard.
Option to reset forgotten password.

2. Dashboard

- Displays student records in a table with columns for Enrollment Number, Name, Marks, Attendance, and Fee Status.
- Admin can perform Edit/Delete operations.



Student Dashboard localhost/student-management-system/dashboard.php

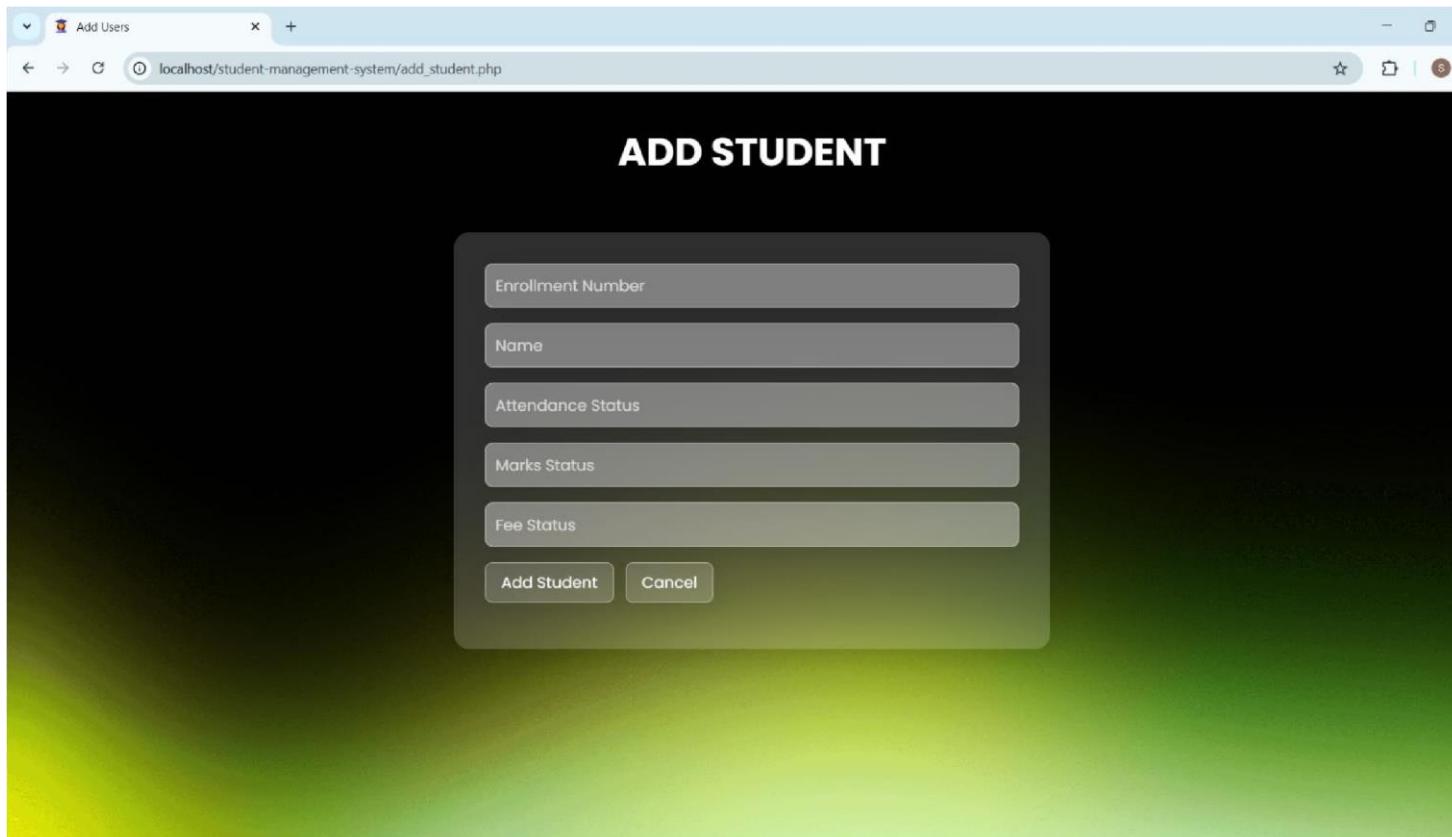
STUDENT DASHBOARD

WELCOME, DEV ADD RECORD MANAGE USERS LOG OUT

ENROLLMENT NUMBER	NAME	MARKS	ATTENDANCE	FEES	EDIT	DELETE
22000214	Abhiraj Chaudhuri	99%	92%	Paid	<button>Edit</button>	<button>Delete</button>
22000215	Abhishek Yadav	92%	69%	Paid	<button>Edit</button>	<button>Delete</button>
22000216	Ajin K James	95%	87%	Paid	<button>Edit</button>	<button>Delete</button>
22000269	Rohan Agarkar	89%	75%	Paid	<button>Edit</button>	<button>Delete</button>
22000577	Sandra Maria Wilson	94%	87%	Paid	<button>Edit</button>	<button>Delete</button>
22000781	Parita Tarvani	92%	91%	Paid	<button>Edit</button>	<button>Delete</button>
22000782	Ruby Singh	85%	69%	Paid	<button>Edit</button>	<button>Delete</button>
22000783	Riya Shah	88%	78%	Not Paid	<button>Edit</button>	<button>Delete</button>

3. Add Student

- Admin can add a new student by entering details like enrollment



The screenshot shows a web browser window titled "Add Users" with the URL "localhost/student-management-system/add_student.php". The main content is a form titled "ADD STUDENT" with the following fields:

- Enrollment Number
- Name
- Attendance Status
- Marks Status
- Fee Status

At the bottom of the form are two buttons: "Add Student" and "Cancel".

number, name, marks, attendance, and fee status.



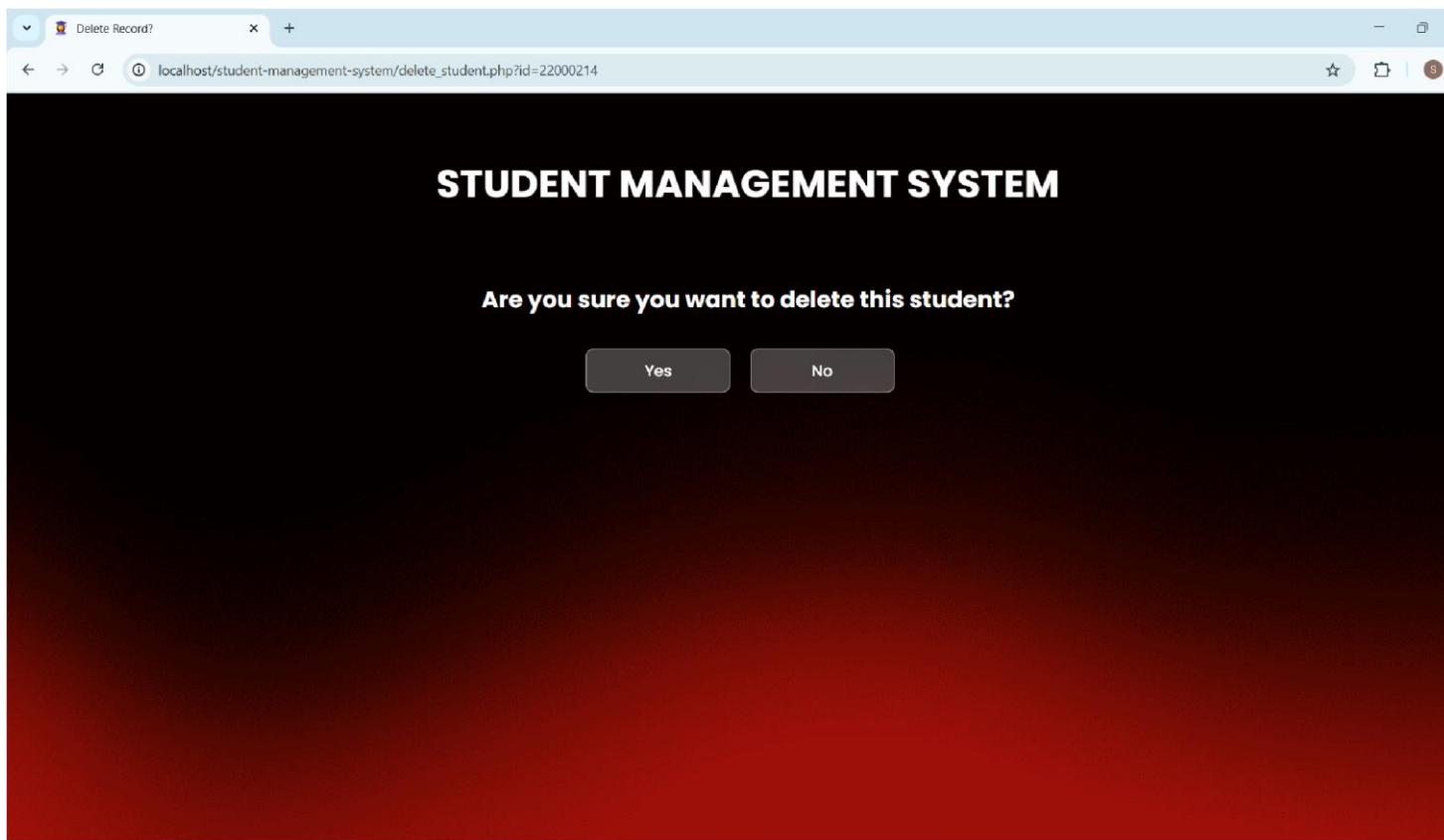
The screenshot shows a web browser window titled "Edit Student". The URL in the address bar is "localhost/student-management-system/edit_student.php?id=22000214". The main title of the page is "STUDENT MANAGEMENT SYSTEM". A modal dialog box is centered on the screen with the title "EDIT STUDENT". The modal contains the following data:

Enrollment No.	22000214
Name	Abhiraj Chaudhuri
Marks	99%
Attendance	92%
Fee Status	Paid

At the bottom of the modal are two buttons: "Update Student" and "Cancel".

4. Edit Student

- Enables editing of existing student records.



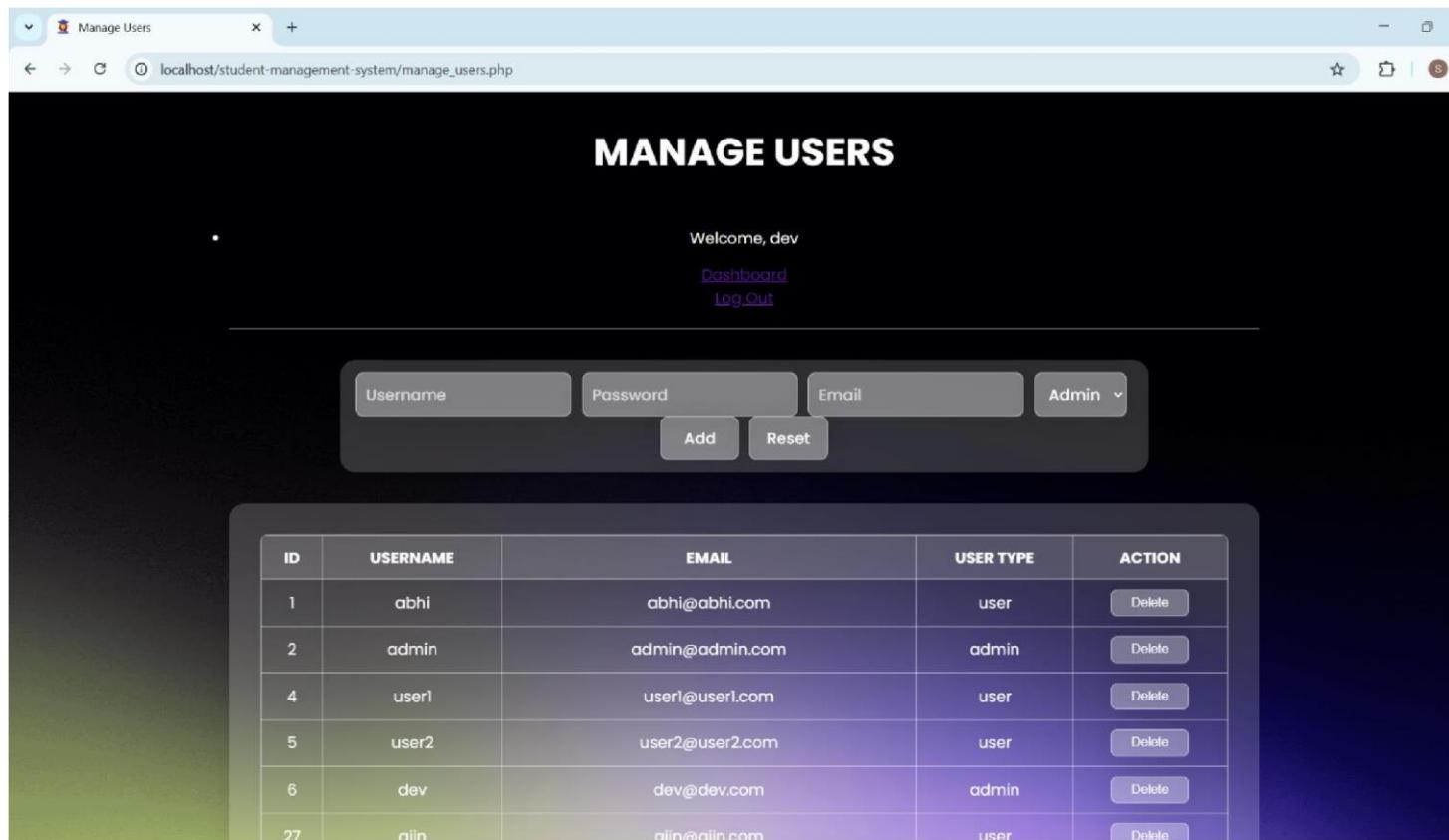
5. Delete Student

- Confirms before deleting a student record permanently.



6. Manage Users

- Allows admin to add, remove, or modify user accounts.

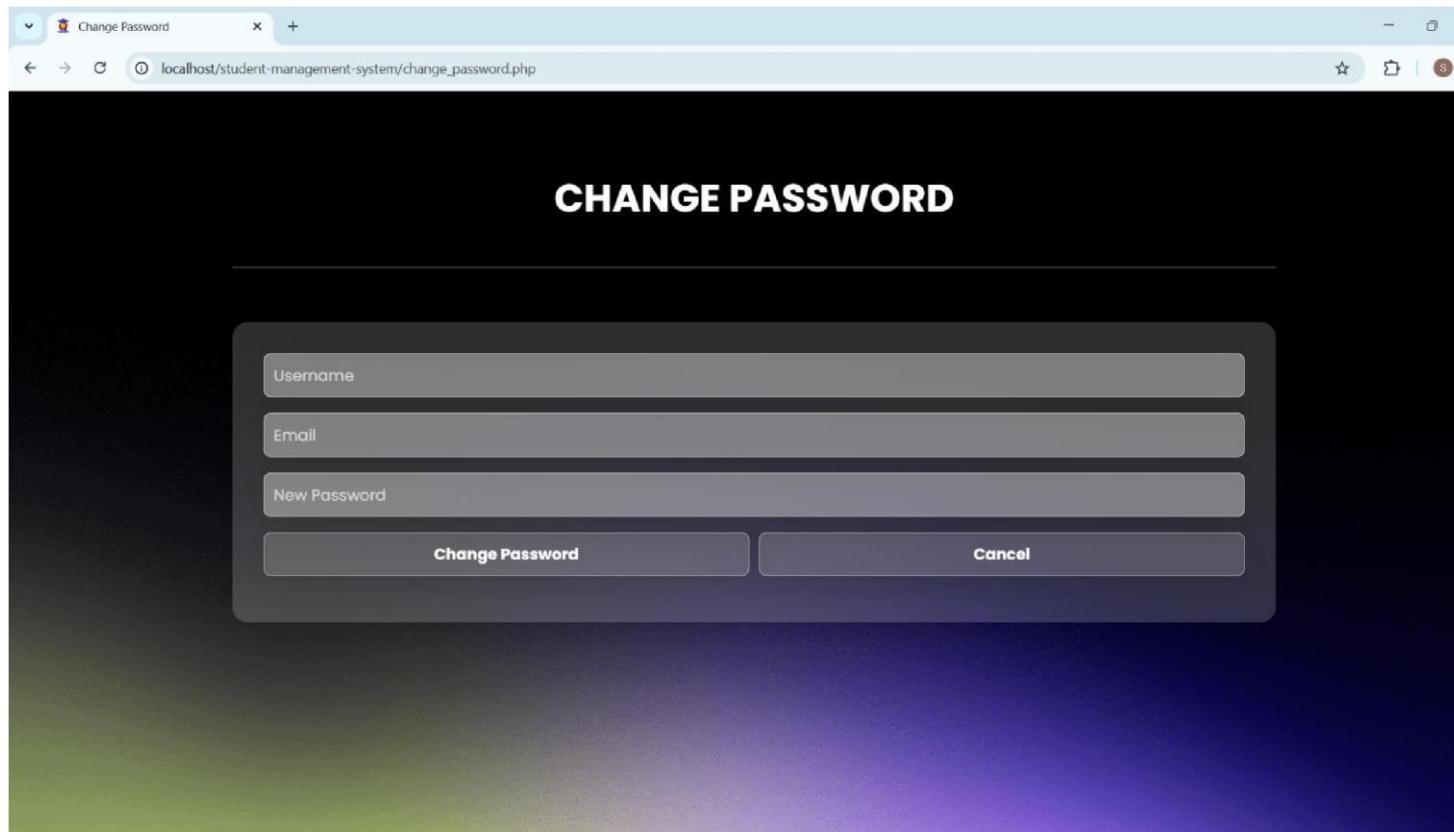


The screenshot shows a web application titled "Manage Users". At the top, there is a header with the text "Welcome, dev", a "Dashboard" link, and a "Log Out" link. Below the header is a search/filter bar with fields for "Username", "Password", "Email", and a dropdown menu for "Admin". There are also "Add" and "Reset" buttons. The main content area displays a table of user data:

ID	USERNAME	EMAIL	USER TYPE	ACTION
1	abhi	abhi@abhi.com	user	<button>Delete</button>
2	admin	admin@admin.com	admin	<button>Delete</button>
4	user1	user1@user1.com	user	<button>Delete</button>
5	user2	user2@user2.com	user	<button>Delete</button>
6	dev	dev@dev.com	admin	<button>Delete</button>
27	qilin	qilin@qilin.com	user	<button>Delete</button>

- User roles defined as “Admin” and “User”.

7. Change Password



The screenshot shows a web application titled "Change Password". The page features a large title "CHANGE PASSWORD" at the top. Below the title is a form with three input fields: "Username", "Email", and "New Password". At the bottom of the form are two buttons: "Change Password" and "Cancel".

- Lets users reset their password using username and email verification.
-

Code Snippets: students_DB.doc

Database students DB

Table structure for table admin

Column	Type	Null	Default
<i>id</i>	int(11)	No	username varchar(20) No email varchar(80)
		No	password varchar(60) No user_type enum('admin', 'user') No

Dumping data for table admin

1	abhi	abhi@abhi.com	abhi	user
2	admin	admin@admin.com	admin	admin
4	user1	user1@user1.com	user1	user 5 user2
		user2@user2.com	user2	user
6	dev	dev@dev.com		admin admin
27	ajin	ajin@ajin.com		ajin user

Table structure for table students

Column	Type	Null	Default
<i>stu_enrollment_number</i>	int(8)	No	stu_name varchar(50) No
stu_marks_status	varchar(20)	No	stu_attendance varchar(20) No
			enum('Paid', 'Not Paid') No

stu_fee_status Yes NULL
 Paid')

Dumping data for table students

22000214	Abhiraj Chaudhuri	99%	92%	Paid
22000215	Abhishek Yadav	92%	69%	Paid
22000216	Ajin K James	95%	87%	Paid
22000269	Rohan Agarkar	89%	75%	Paid
22000577	Sandra Maria Wilson	94%	87%	Paid
22000781	Parita Tarvani	92%	91%	Paid
22000782	Ruby Singh	85%	69%	Paid
22000783	Riya Shah	88%	78%	Not Paid
22000784	Rishi Agarwal	92%	61%	Paid
22000785	Maya Jain	87%	59%	Paid
22000786	Dev Patel	91%	60%	Not Paid
22000787	Priya Sharma	86%	58%	Paid
22000788	Rahul Singh	89%	59%	Paid
		22000789		Mon2 Sharma
		93%	61%	Paid
22000790	Anish Agarwal	88%	59%	Paid
		22000791		Nikita Jain
		92%	60%	Not Paid
22000792	Karan Patel	87%	58%	Paid
22000793	Rishi Sharma	91%	59%	Paid

22000794	Sonu Singh	86%	58%	Paid
22000795	Anjali Patel	89%	59%	Paid
22000796	Rajesh Agarwal	93%	61%	Paid
22000797	Monisha Jain	88%	59%	Not Paid
22000798	Vikram Patel	92%	60%	Paid
22000799	Rakesh Sharma	87%	58%	Paid
22000800	Sunil Singh	91%	59%	Paid
22000801	Ajay Patel	86%	58%	Paid
22000802	Ramesh Agarwal	89%	59%	Paid
22000803	Suresh Jain	93%	61%	Paid
22000804	Kamlesh Patel	88%	59%	Paid
		22000805		Suresh
	Sharma	92%	60%	Paid
22000806	Rakesh Singh	87%	58%	Paid
22000807	Sunil Patel	91%	59%	Not Paid
22000808	Ajay Agarwal	86%	58%	Paid
22000809	Ramesh Jain	89%	59%	Paid
22000810	Suresh Patel	93%	61%	Paid
22000811	Kamlesh Sharma	88%	59%	Paid
	22000812		Suresh Singh	
	92%	60%	Paid	22000813
			Rakesh Patel	87%
				58%
	Paid			

22000814	Sunil Agarwal	91% 59% Paid
22000815	Ajay Jain	86% 58% Not Paid
22000981	Yuvaraj Singh	65% 34% Not Paid
22000982	Riya Shah	56% 82% Not Paid
23000142	Keya Patel	86% 56% Paid
23000143	Anya Sharma	90% 60% Paid
23000144	Ishaan Singh	85% 58% Paid
23000145	Riya Patel	88% 59% Paid
23000146	Aryan Agarwal	92% 61% Paid
23000147	Maya Jain	87% 59% Paid
		23000148 Dev Patel
		91% 60% Paid
23000149	Priya Sharma	86% 58% Paid
23000150	Rahul Singh	89% 59% Paid
23000151	Neha Patel	93% 61% Paid
23000152	Anish Agarwal	88% 59% Paid
23000153	Nikita Jain	92% 60% Paid
		23000154 Karan Patel
		87% 58% Paid
23000155	Rishi Sharma	91% 59% Paid
23000156	Sonu Singh	86% 58% Paid
		23000157 Anjali Patel
		89% 59% Paid
23000158	Rajesh Agarwal	93% 61% Paid

23000159	Monisha Jain	88% 59% Paid	23000160	Vikram Patel	
		92% 60% Paid			
23000161	Rakesh Sharma	87% 58% Paid			
23000162	Sunil Singh	91% 59% Paid			
23000163	Ajay Patel	86% 58% Paid			
23000164	Ramesh Agarwal	89% 59% Paid			
23000165	Suresh Jain	93% 61% Paid			
23000166	Kamlesh Patel	88% 59% Paid	23000167	Suresh	
		Sharma 92% 60% Paid			
23000168	Rakesh Singh	87% 58% Paid			
23000169	Sunil Patel	91% 59% Paid			
23000170	Ajay Agarwal	86% 58% Paid			
23000171	Ramesh Jain	89% 59% Paid			
23000172	Suresh Patel	93% 61% Paid	23000174	Suresh Singh	
		92% 60% Paid			
23000175	Rakesh Patel		87%	58%	Paid
23000216	Keyan Engineer		75%	45%	Paid
23000693	Kento Bento	49% 54% Not Paid	24000214	Vidhi Patel	93% 82% Not Paid

Paid **students_DB.sql**

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: localhost
-- Generation Time: Oct 20, 2023 at 08:39 AM
-- Server version: 10.4.28-MariaDB
-- PHP Version: 8.2.4

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

--

-- Database: `students_DB`

--

--

-- Table structure for table `admin`

--

```
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `username` varchar(20) NOT NULL,
  `email` varchar(80) NOT NULL,
  `password` varchar(60) NOT NULL,
  `user_type` enum('admin','user') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `admin`

--

```
INSERT INTO `admin` (`id`, `username`, `email`, `password`, `user_type`)
VALUES
(1, 'abhi', 'abhi@abhi.com', 'abhi', 'user'),
(2, 'admin', 'admin@admin.com', 'admin', 'admin'),
(4, 'user1', 'user1@user1.com', 'user1', 'user'),
(5, 'user2', 'user2@user2.com', 'user2', 'user'),
(6, 'dev', 'dev@dev.com', 'admin', 'admin'),
(27, 'ajin', 'ajin@ajin.com', 'ajin', 'user');
```

```
--
```

```
-- Table structure for table `students`
```

```
--
```

```
CREATE TABLE `students` (
`stu_enrollment_number` int(8) NOT NULL,
`stu_name` varchar(50) NOT NULL,
`stu_marks_status` varchar(20) NOT NULL,
`stu_attendance` varchar(20) NOT NULL,
`stu_fee_status` enum('Paid','Not Paid') DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `students`

--

```
INSERT INTO `students`(`stu_enrollment_number`, `stu_name`,  
`stu_marks_status`, `stu_attendance`, `stu_fee_status`) VALUES  
(22000214, 'Abhiraj Chaudhuri', '99%', '92%', 'Paid'),  
(22000215, 'Abhishek Yadav', '92%', '69%', 'Paid'),  
(22000216, 'Ajin K James', '95%', '87%', 'Paid'),  
(22000269, 'Rohan Agarkar', '89%', '75%', 'Paid'),  
(22000577, 'Sandra Maria Wilson', '94%', '87%', 'Paid'),  
(22000781, 'Parita Tarvani', '92%', '91%', 'Paid'),  
(22000782, 'Ruby Singh', '85%', '69%', 'Paid'),  
(22000783, 'Riya Shah', '88%', '78%', 'Not Paid'),  
(22000784, 'Rishi Agarwal', '92%', '61%', 'Paid'),  
(22000785, 'Maya Jain', '87%', '59%', 'Paid'),  
(22000786, 'Dev Patel', '91%', '60%', 'Not Paid'),  
(22000787, 'Priya Sharma', '86%', '58%', 'Paid'),  
(22000788, 'Rahul Singh', '89%', '59%', 'Paid'),  
(22000789, 'Mon2 Sharma', '93%', '61%', 'Paid'),  
(22000790, 'Anish Agarwal', '88%', '59%', 'Paid'),  
(22000791, 'Nikita Jain', '92%', '60%', 'Not Paid'),
```

(22000792, 'Karan Patel', '87%', '58%', 'Paid'),
(22000793, 'Rishi Sharma', '91%', '59%', 'Paid'),
(22000794, 'Sonu Singh', '86%', '58%', 'Paid'),
(22000795, 'Anjali Patel', '89%', '59%', 'Paid'),
(22000796, 'Rajesh Agarwal', '93%', '61%', 'Paid'),
(22000797, 'Monisha Jain', '88%', '59%', 'Not Paid'),
(22000798, 'Vikram Patel', '92%', '60%', 'Paid'),
(22000799, 'Rakesh Sharma', '87%', '58%', 'Paid'),
(22000800, 'Sunil Singh', '91%', '59%', 'Paid'),
(22000801, 'Ajay Patel', '86%', '58%', 'Paid'),
(22000802, 'Ramesh Agarwal', '89%', '59%', 'Paid'),
(22000803, 'Suresh Jain', '93%', '61%', 'Paid'),
(22000804, 'Kamlesh Patel', '88%', '59%', 'Paid'),
(22000805, 'Suresh Sharma', '92%', '60%', 'Paid'),
(22000806, 'Rakesh Singh', '87%', '58%', 'Paid'),
(22000807, 'Sunil Patel', '91%', '59%', 'Not Paid'),
(22000808, 'Ajay Agarwal', '86%', '58%', 'Paid'),
(22000809, 'Ramesh Jain', '89%', '59%', 'Paid'),
(22000810, 'Suresh Patel', '93%', '61%', 'Paid'),
(22000811, 'Kamlesh Sharma', '88%', '59%', 'Paid'),
(22000812, 'Suresh Singh', '92%', '60%', 'Paid'), (22000813, 'Rakesh Patel', '87%', '58%', 'Paid'),
(22000814, 'Sunil Agarwal', '91%', '59%', 'Paid'),

- (22000815, 'Ajay Jain', '86%', '58%', 'Not Paid'),
(22000981, 'Yuvaraj Singh', '65%', '34%', 'Not Paid'),
(22000982, 'Riya Shah', '56%', '82%', 'Not Paid'),
(23000142, 'Keya Patel', '86%', '56%', 'Paid'),
(23000143, 'Anya Sharma', '90%', '60%', 'Paid'),
(23000144, 'Ishaan Singh', '85%', '58%', 'Paid'),
(23000145, 'Riya Patel', '88%', '59%', 'Paid'),
(23000146, 'Aryan Agarwal', '92%', '61%', 'Paid'),
(23000147, 'Maya Jain', '87%', '59%', 'Paid'),
(23000148, 'Dev Patel', '91%', '60%', 'Paid'),
(23000149, 'Priya Sharma', '86%', '58%', 'Paid'),
(23000150, 'Rahul Singh', '89%', '59%', 'Paid'),
(23000151, 'Neha Patel', '93%', '61%', 'Paid'),
(23000152, 'Anish Agarwal', '88%', '59%', 'Paid'),
(23000153, 'Nikita Jain', '92%', '60%', 'Paid'),
(23000154, 'Karan Patel', '87%', '58%', 'Paid'),
(23000155, 'Rishi Sharma', '91%', '59%', 'Paid'),
(23000156, 'Sonu Singh', '86%', '58%', 'Paid'),
(23000157, 'Anjali Patel', '89%', '59%', 'Paid'),
(23000158, 'Rajesh Agarwal', '93%', '61%', 'Paid'),
(23000159, 'Monisha Jain', '88%', '59%', 'Paid'), (23000160, 'Vikram Patel', '92%', '60%', 'Paid'),
(23000161, 'Rakesh Sharma', '87%', '58%', 'Paid'),

(23000162, 'Sunil Singh', '91%', '59%', 'Paid'),
(23000163, 'Ajay Patel', '86%', '58%', 'Paid'),
(23000164, 'Ramesh Agarwal', '89%', '59%', 'Paid'),
(23000165, 'Suresh Jain', '93%', '61%', 'Paid'),
(23000166, 'Kamlesh Patel', '88%', '59%', 'Paid'),
(23000167, 'Suresh Sharma', '92%', '60%', 'Paid'),
(23000168, 'Rakesh Singh', '87%', '58%', 'Paid'),
(23000169, 'Sunil Patel', '91%', '59%', 'Paid'),
(23000170, 'Ajay Agarwal', '86%', '58%', 'Paid'),
(23000171, 'Ramesh Jain', '89%', '59%', 'Paid'),
(23000172, 'Suresh Patel', '93%', '61%', 'Paid'),
(23000174, 'Suresh Singh', '92%', '60%', 'Paid'),
(23000175, 'Rakesh Patel', '87%', '58%', 'Paid'),
(23000216, 'Keyan Engineer', '75%', '45%', 'Paid'),
(23000693, 'Kento Bento', '49%', '54%', 'Not Paid'),
(24000214, 'Vidhi Patel', '93%', '82%', 'Not Paid');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admin`

--
ALTER TABLE `admin`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `students`

--

ALTER TABLE `students`

ADD PRIMARY KEY (`stu_enrollment_number`);

--

-- AUTO_INCREMENT for dumped tables

--

--

-- AUTO_INCREMENT for table `admin`

--

ALTER TABLE `admin`

MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=28;

COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS

```
*;/
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;
```

7. Results / Testing

Testing Performed:

Test Case	Input	Expected Output	Result
Login Valid Credentials	admin/admin123	Redirect to dashboard	Pass
Add Student Record	Valid student details	Record added successfully	Pass
Edit Student Record	Modify marks	Updated successfully	Pass
	Click Delete	Record deleted after confirmation	Pass
Delete Student	Valid username &	confirmation	
Change Password	email	Password updated	Pass

Output Screenshots:

- Login Page
- Dashboard
- Add Student

- Edit Student
 - Delete Confirmation
 - Manage Users
 - Change Password
-

8. Conclusion

The *Student Management System* provides a digital solution for managing student information efficiently. It reduces manual effort, improves data accuracy, and allows secure multi-user access. The system can be extended in the future to include advanced features like data analytics, attendance reports, and student performance charts.

Future Scope:

- Integration with cloud storage for data backup.
 - Email/SMS notifications for students.
 - Role-based dashboards and analytics visualization.
-

9. References

- [Laravel Documentation](#)
- PHP.net – Official PHP Documentation
- W3Schools – HTML, CSS, JavaScript Tutorials
- Stack Overflow (Community support for troubleshooting)