

# Student Record Management System



# Team Members

1. Shaik Jasmine- AP24110010003
- 2 . Sidagam Lakshmi Tanmayee - AP24110010005
3. Edara Jahnavi - AP24110010015
4. Lahari Nanduru - AP24110010044
5. Manthena Ashmitha - AP24110010053
6. Medasani Pujitha - AP24110010061



# Introduction

1. This project manages student records digitally using C++.
2. Supports storing CGPA, marks, remarks, and personal data.
3. Removes the need for manual record management.
4. Uses file handling to store data securely.
5. Provides different access levels based on user roles.



# Problem Statement

## Time-Consuming Records

Manual record-keeping slows down work and wastes faculty time.

## No Error Reporting System

Students can't report wrong details or request corrections.

## Frequent Data Errors

Mistakes commonly occur in marks, attendance, and remarks.

## No Centralized Access

Different roles (Admin/HOD/Professor) lack a unified platform.

## Slow Data Updates

Updating student information takes too long in manual systems.

# Project Objectives

To store and manage student details efficiently.

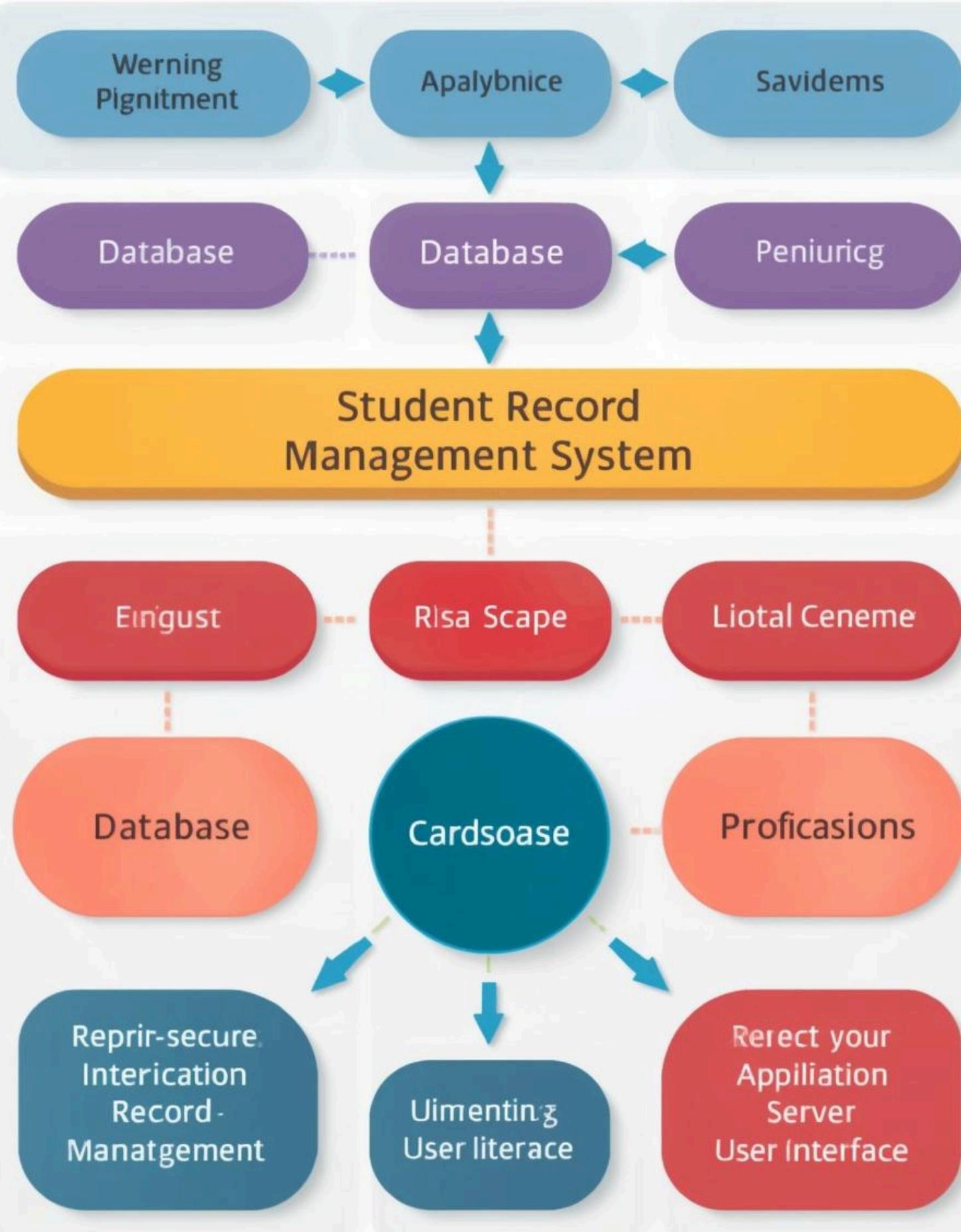
To allow easy updating of marks, CGPA, and remarks.

To provide secure role-based access (Admin, HOD, Professor)

To allow students to raise tickets for incorrect information.

To automate data retrieval and reduce manual errors.

## Architecture Diagram



# System Architecture Overview

## Structured Modular Design

A clean module-based structure (Student, Ticket, Admin, HOD, Professor) makes the system easy to manage, update, and scale.

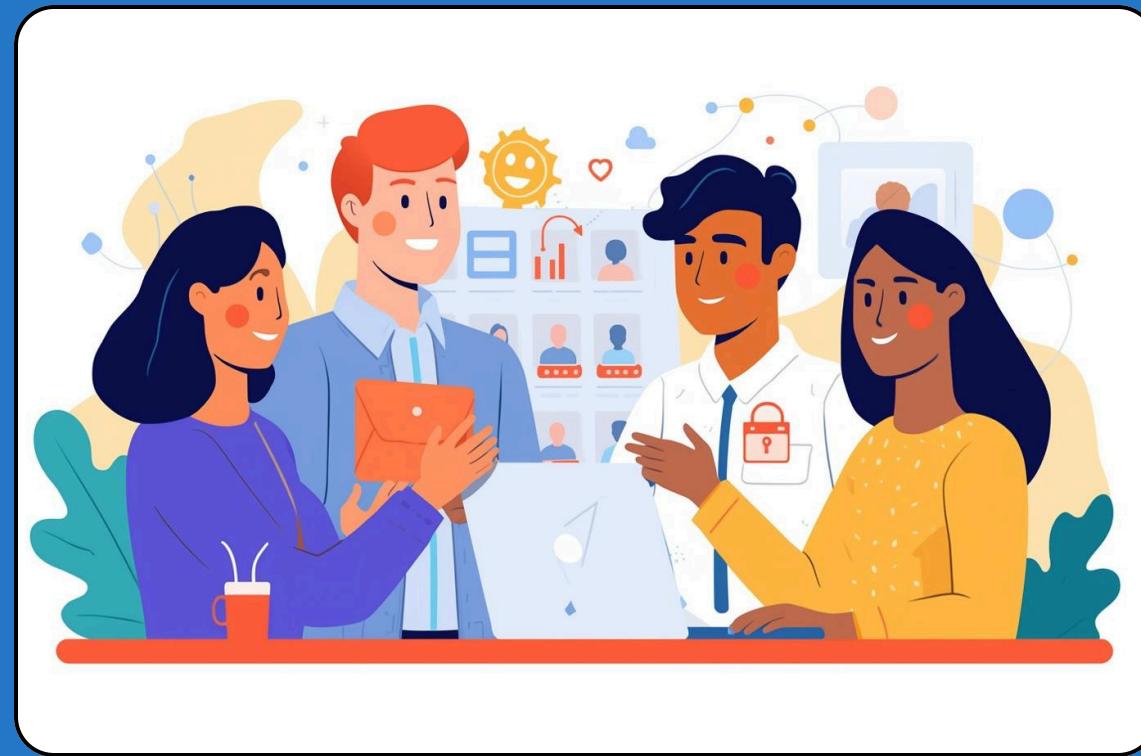
## File-Based Data Storage

Student records and tickets are securely handled using binary file storage, ensuring safe retrieval and efficient data management.

## Role-Based Processing Flow

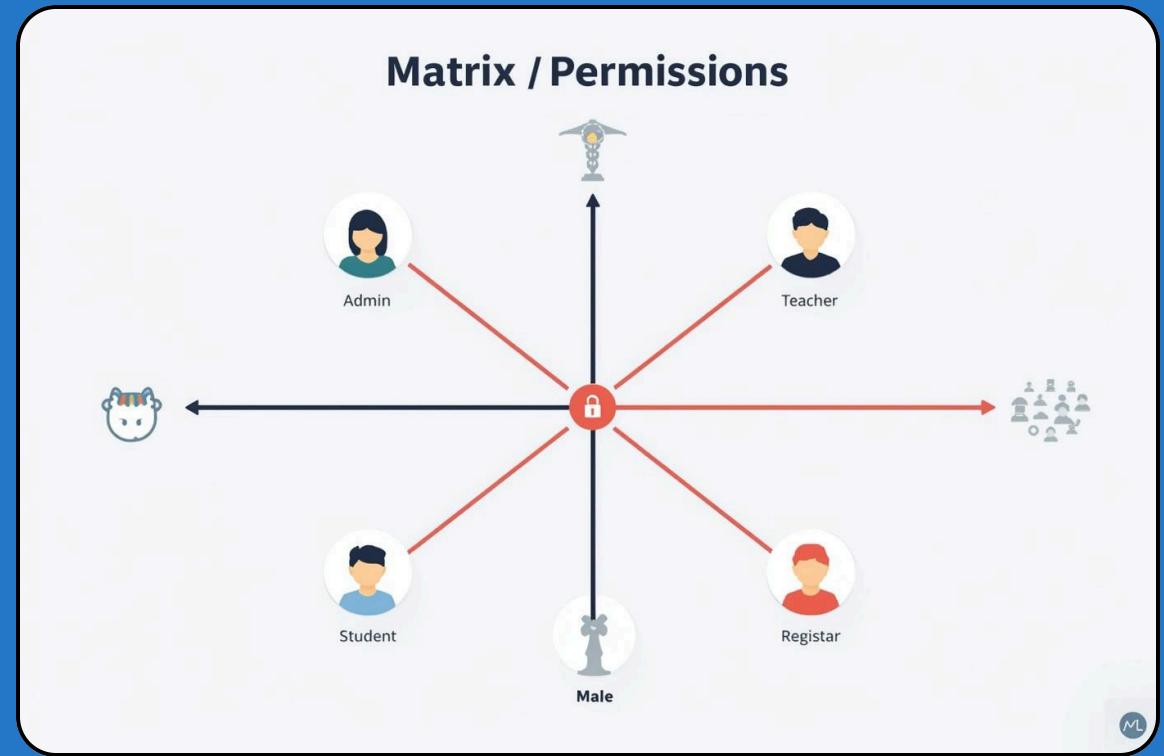
Each user role follows a controlled flow from input to processing and storage, ensuring proper access, accuracy, and smooth system operation.

# User Roles and Permissions



## Admin

Add, update, view students, and view tickets.



## HOD and Professor

HOD can View all students and check raised tickets and professor will update marks and remarks for students.



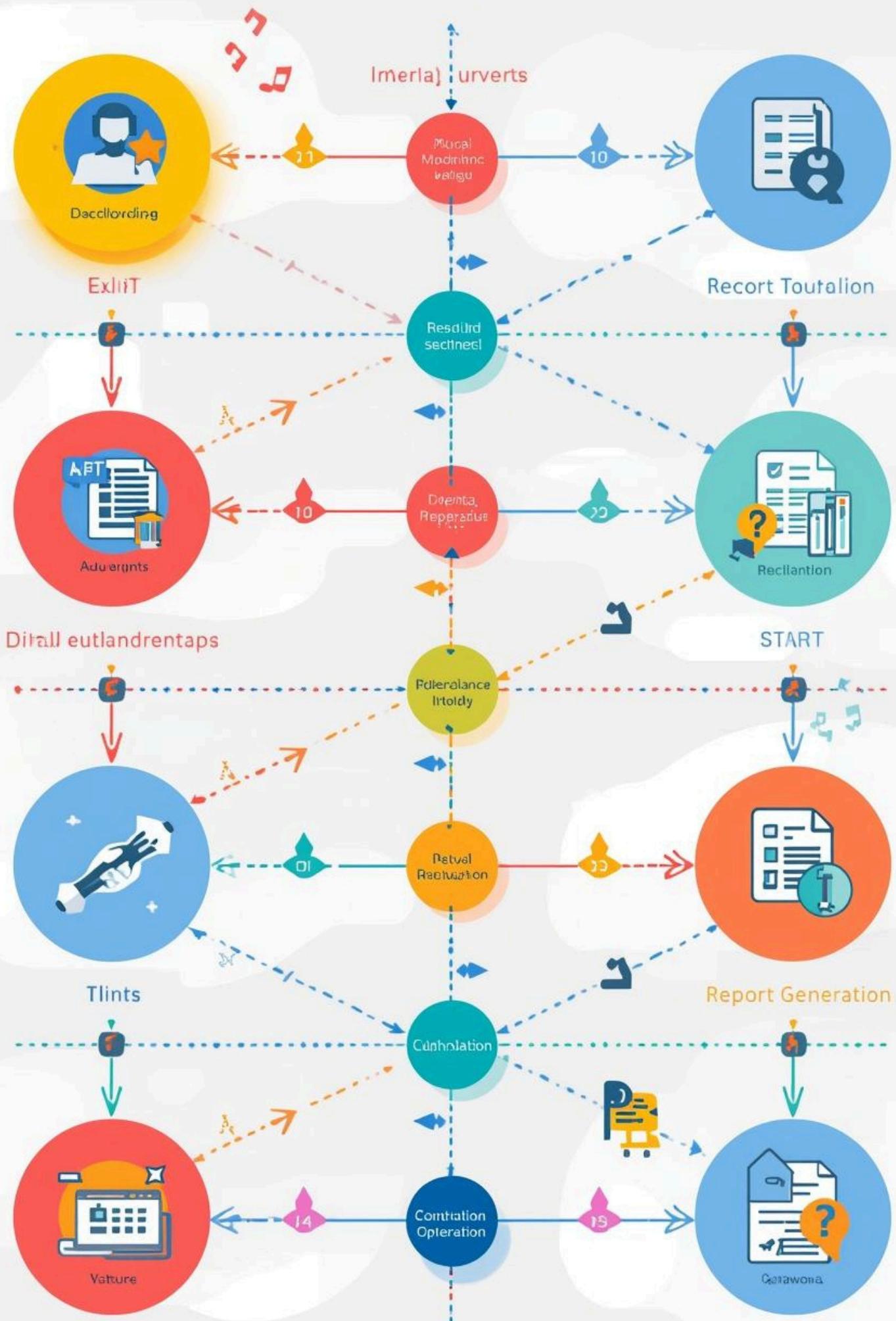
## Student

Raise tickets for incorrect or missing information.

# Effective Data Management

1. Add new student details (name, roll, CGPA, marks, remarks).
2. Update existing student records through admin/professor/HOD.
3. Store data in binary files for long-term usage.
4. Retrieve and display all student records instantly.
5. Maintain remarks (e.g., “Active in sports”) for better profiling.





# System Flowchart

STA

# User Selects Role (Admin/HOD/ Professor/Student)

## System Verifies Role & Displays Corresponding Menus

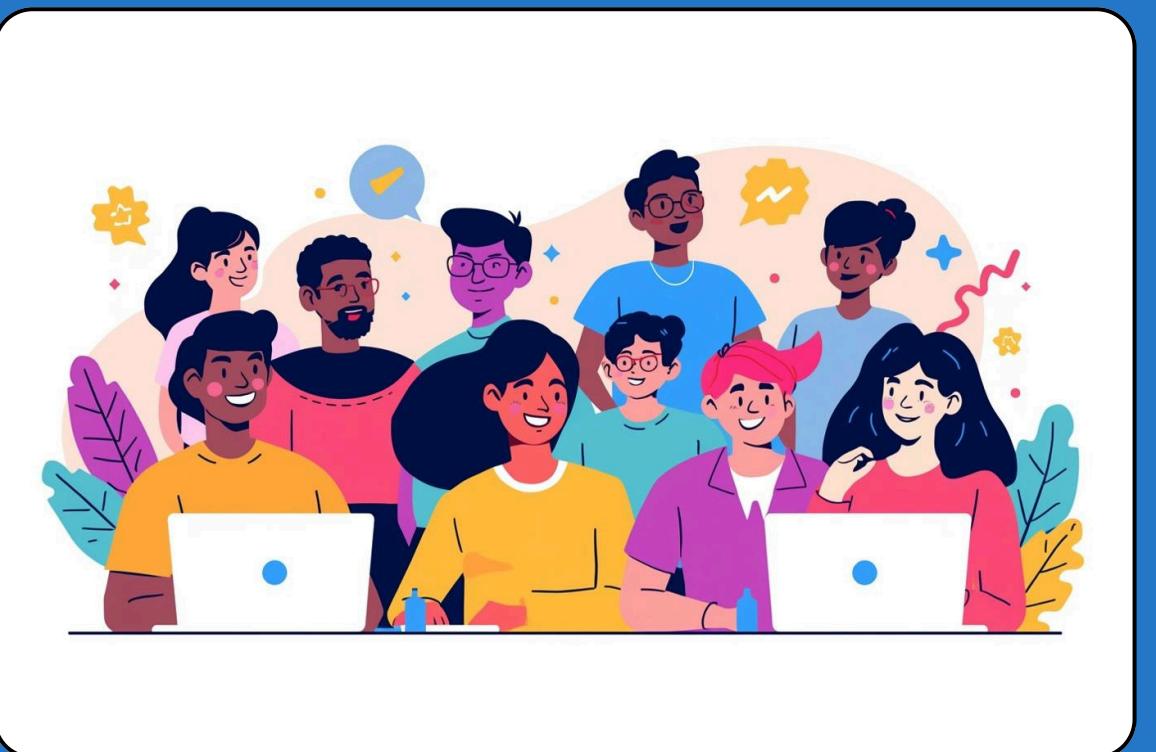
## User Performs Action: Add / Update / View / Tick

## Data Saved or Retrieved from Binary Files

[Return to Login Menu OR Ex](#)

# Features Implemented

- 1. Student data entry with CGPA, marks, and remarks.**
- 2. Ticket raising system for students with incorrect data.**
- 3. Role-based login: Admin, HOD, Professor, Student.**
- 4. Update and view functions for all roles.**
- 5. File handling for secure and permanent data storage.**



# Future Enhancements



## 1. GUI-Based Modern Application

The system can be upgraded from a console program to a clean graphical interface using Qt, Java, or Python, making it easier for users to navigate and manage records.

## Database-Driven Storage

Instead of storing data in files, integrating MySQL or MongoDB will make the system faster, more secure, and capable of handling large student datasets efficiently.

## Smart Automation & Mobile Access

Features like mobile apps, automatic report card generation, and AI-based performance prediction will improve accessibility and provide smarter insights for students and faculty.

# Conclusion

- 1. The system simplifies student data management.**
- 2. Reduces manual errors and saves time.**
- 3. Ensures secure access through role-based permissions.**
- 4. Allows students to report incorrect data easily.**
- 5. Useful and expandable system for educational institutions.**



thank  
you