National Institute of Technology Uttarakhand

Department of Computer Science & Engineering

Project File - Student Record Management System

Subject: DatabaseManagement Systems (DBMS)

SUBMITTED TO: Mr. Sanjay Kumar Tamta

SUBMITTED BY: Ayush Payal

DATE: 3 Nov 2025

🏫 Student Record Management System

⚙️ Tools and Technologies Used

Component	**Description**
Programming Language	Python
GUI Library	Tkinter
Database	MySQL
Database Connector	`mysql.connector`
Development Environment	VS Code
Database Tool	MySQL Workbench
Operating System	Windows 11

🎯 Objective

The main objective of this project is to develop a **Student Record Management System** that provides an integrated platform for **Admins**, **Faculty**, and **Students** to manage academic data such as student details, attendance, marks, and grades.

The system supports full **CRUD (Create, Read, Update, Delete)** operations, **automatic grade calculation**, and **secure login authentication**, ensuring both accuracy and efficiency in data handling.

🧩 System Overview

This project provides a centralized, role-based management system with the following access levels:

- 👨‍💼 **Admin**: Manages student data and user accounts (add, view, delete).
- 👩‍🏫 **Faculty**: Adds marks and attendance records for students.
- 🧑‍🎓 **Student**: Logs in to view their own marks, grades, and attendance.

The backend is built using **MySQL**, while the frontend is a **Tkinter-based GUI** connected through **Python's mysql.connector** library.

🗄️ Database Design

The database is named **`student_records`** and was created in **MySQL Workbench**.
It includes several interrelated tables with **foreign keys** ensuring data consistency and relational integrity.

Main Tables

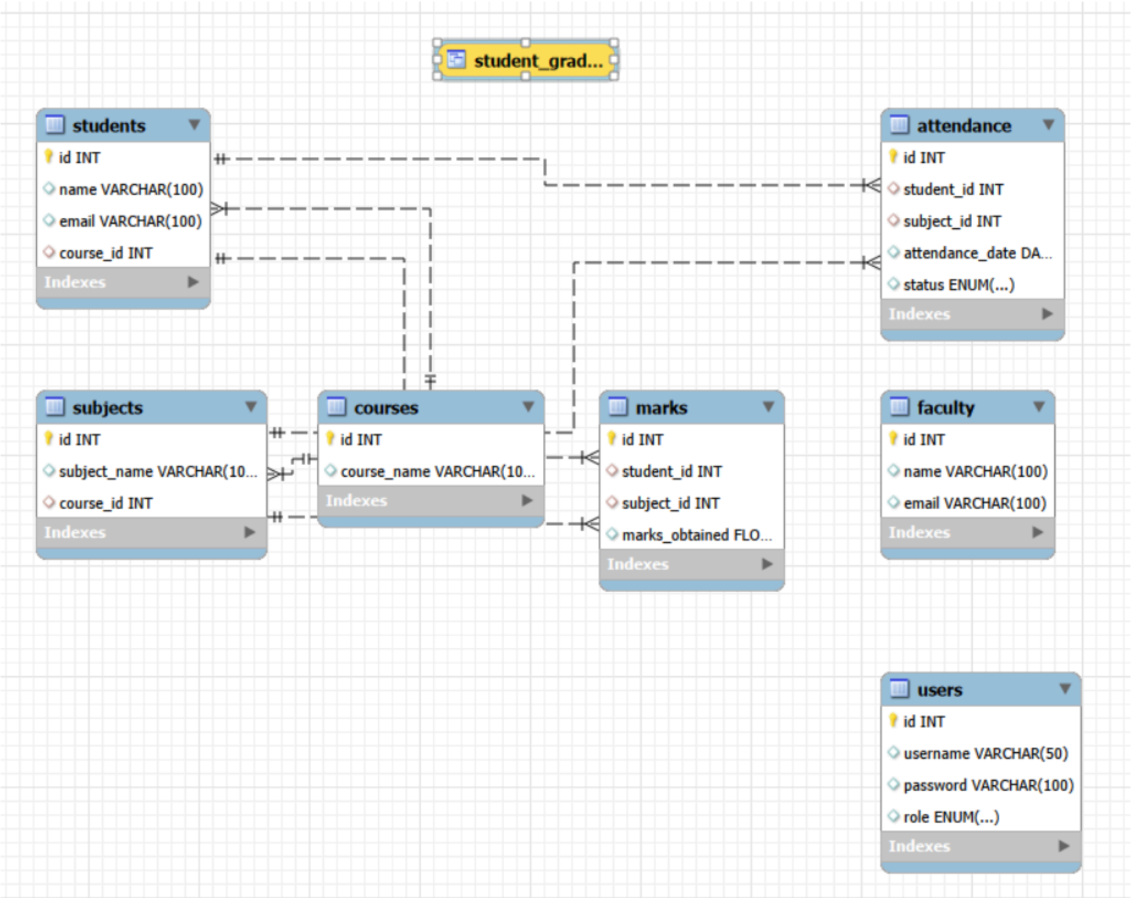
Table	**Description**
courses	Stores course information
subjects	Contains subjects related to each course
faculty	Holds faculty details
students	Stores student details (ID, Name, Email, Course)
marks	Records marks obtained by students
attendance	Stores attendance records
users	Stores login credentials and roles
student_grades (View)	Displays grades auto-calculated from marks

---

Key Database Features

- **Foreign Key Constraints** → Maintain data integrity
- **ON DELETE CASCADE** → Deletes dependent records automatically
- **Auto-Incremented IDs** → Ensures unique entries
- **Auto Grade Calculation** → Derived dynamically using a SQL View
- **Role-Based Authentication** → Managed through `users` table

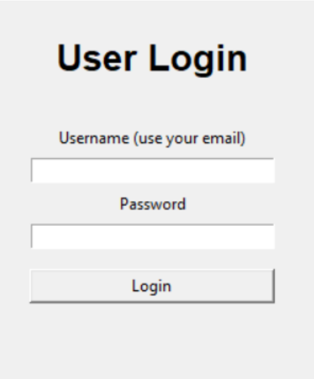
📊 EER Diagram



💻 Application Modules

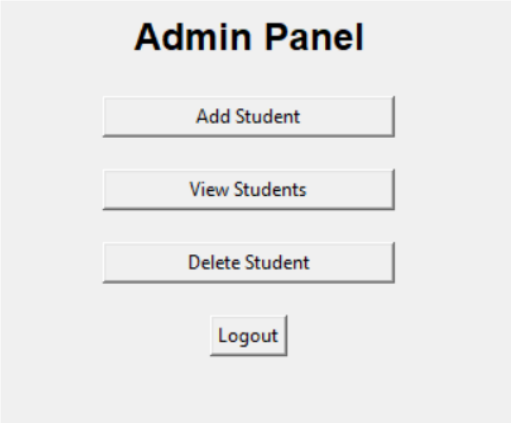
1️⃣ Login System

- Secure login for **Admin**, **Faculty**, and **Student** using credentials from the `users` table.
-
- Role-based redirection to different panels after successful login.
- Password-based authentication using `validate_user()` from the backend.



2️⃣ Admin Panel

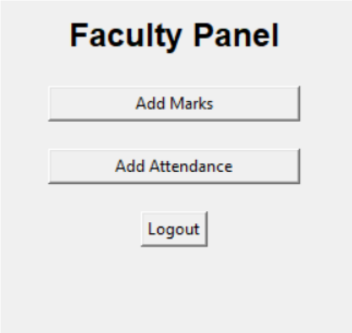
- Admin can:
  - ○ Add new students (auto-creates login in `users` table).
  - ○ View all students in a table view.
  - ○ Delete student records (removes both `students` and corresponding `users` entries).

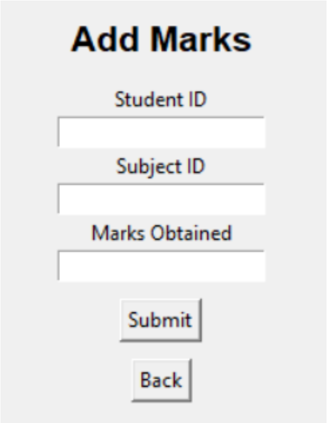


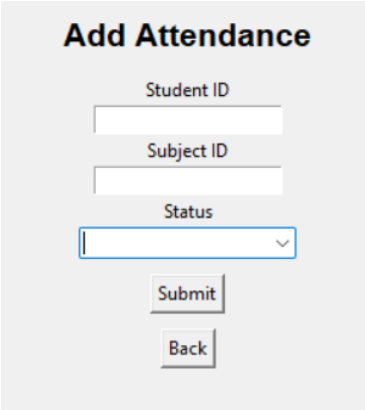
- IDs are auto-generated using a **“Next Available ID”** function to fill gaps from deleted entries.

3️⃣ Faculty Panel

- Faculty can:
  - ○ Add student marks per subject.
  - ○ Record attendance status (Present/Absent).
  - ○ View or modify attendance records.

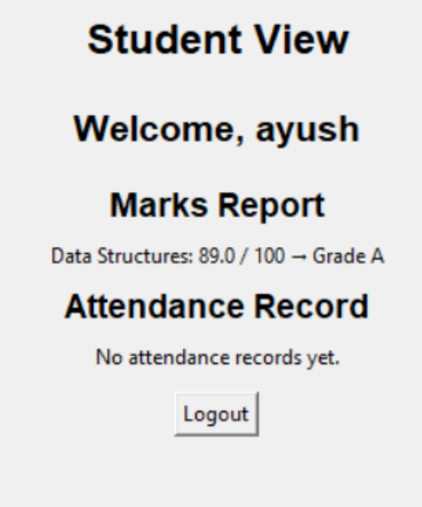






4️⃣ Student Panel

- Students log in using their email and password.
- Can view:
  - Marks and auto-calculated grades per subject.
  - Attendance records by subject and date.
- Automatically creates missing student entries in case of login mismatch.



Student View

Welcome, ayush

Marks Report

Data Structures: 89.0 / 100 → Grade A

Attendance Record

No attendance records yet.

🔄 CRUD Operations Summary

Operation	**Description**	**Example**
Create	Add new student/faculty/subject entries	Add Student form
Read	View student and marks data	Student records table
Update	Modify attendance or marks	Faculty marks entry
Delete	Remove records (auto-delete from linked tables)	Delete Student

📊 Automatic Grade Calculation

Grades are generated automatically from marks, ensuring transparency and accuracy.This logic is implemented both in Python and via a **MySQL View (`student_grades`)**.

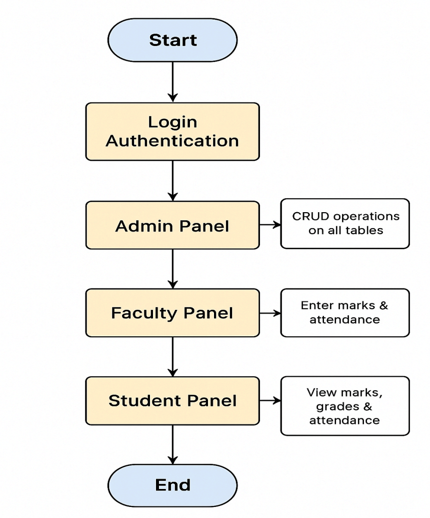
Marks Range	**Grade**
90 and above	A+
80–89	A
70–79	B
60–69	C
50–59	D
Below 50	F

🔐 Data Security and Consistency

- **Foreign Keys & Constraints** → Prevents invalid entries.
- **Role-Based Access** → Limits actions based on login type.
- **Cascade Deletion** → Maintains data cleanliness.
- **Transaction Control** → Ensures rollback on failed inserts.
- **Login Authentication** → Uses secure validation via `users` table.

🧠 Working Flow

1. Database and tables created in **MySQL Workbench**.
2. Relationships defined using **foreign keys**.
3. GUI created in **Tkinter** (Python).
4. Each CRUD operation is linked to SQL queries using **mysql.connector**.
5. Login validated via `validate_user()` function.
6. Grade calculation and attendance auto-updated in the database.



🧾 Conclusion

The **Student Record Management System** integrates a **Tkinter GUI frontend** with a **MySQL backend** to provide a complete, role-based academic management solution.
It allows Admins, Faculty, and Students to seamlessly manage and access educational records with reliability and ease.

✅ Key Achievements

- Auto ID generation for missing entries
- Auto grade calculation
- Role-based login and data access
- Real-time CRUD operation linking GUI ↔ Database
- Full consistency and rollback protection

🚀 Future Enhancements

- Add performance analytics and graphs
- Email/SMS notifications for attendance or marks
- Cloud-hosted MySQL for multi-user access
- Exportable student performance reports (PDF/Excel)