

C-TAG

Project Name: Fee Report Management System

Project Description:

The Fee Report Management System is a Python application designed to manage student and accountant information within an educational institution. It incorporates a MySQL database to handle data persistence and offers role-based access to ensure secure and efficient operation. The system supports two types of users: Admin and Accountant, each with specific permissions.

Objective

The main objective of this project is to create a robust system where:

- **Admin:** Can manage accountant records and control access to the system.
- **Accountant:** Can manage student records, check due fees, and handle basic administrative tasks.

Users of the System

1. Admin:

- Role: System Administrator
- Responsibilities: Manage accountant records, oversee system access.

2. Accountant:

- Role: Financial Administrator
- Responsibilities: Manage student records, track due fees, and perform daily operations.

Functional Requirements

1. Admin Functionality:

- **Accountant Management:**
 - **Add Accountant:** Admin can add new accountant records to the system.
 - **View Accountants:** Admin can view details of all accountants.
 - **Edit Accountant:** Admin can update information of existing accountants.
 - **Delete Accountant:** Admin can remove accountant records from the system.
- **Logout:** Admin can log out of the system.
- **Authentication:** Admin must authenticate using Admin ID and Password.

2. Accountant Functionality:

- **Student Management:**
 - **Add Student:** Accountant can add new student records to the system.
 - **View Students:** Accountant can view details of all students.
 - **Edit Student:** Accountant can update student information.
 - **Delete Student:** Accountant can remove student records from the system.
- **Fee Management:**
 - **Check Due Fee:** Accountant can view the due fees for each student.
- **Logout:** Accountant can log out of the system.

- **Authentication:** Accountant must authenticate using their credentials.

1. Admins Table

Purpose: Stores information about the system administrators who manage accountants. Admins have high-level access to the system and can perform operations such as adding, viewing, editing, and deleting accountant records.

Schema:

Column	Type	Constraints	Description
`id`	INT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each admin
`admin_id`	VARCHAR(50)	NOT NULL, UNIQUE	Unique ID used for admin login
`password`	VARCHAR(255)	NOT NULL	Hashed password for admin authentication

Details:

- `id`: A unique identifier for each admin. It's an auto-incremented integer.
- `admin_id`: A unique login ID for the admin. This field is required and must be unique.
- `password`: store password to check authentic user.

2. Accountants Table

Purpose: Holds information about the accountants who manage student records. Accountants perform operations such as adding, viewing, editing, and deleting student records, as well as checking due fees.

Schema:

Column	Type	Constraints	Description
`id`	INT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each accountant
`username`	VARCHAR(50)	NOT NULL, UNIQUE	Unique username used for accountant login
`password`	VARCHAR(255)	NOT NULL	Hashed password for accountant authentication

Details:

- **id:** A unique identifier for each accountant. It's an auto-incremented integer.
- **username:** A unique username for the accountant's login. This field is required and must be unique.
- **password:** store password to check authentic user.

3. Students Table

Purpose: Contains records of students, including their names and fee-related information. Accountants can manage these records to track student information and manage fee dues.

Schema:

Column	Type	Constraints	Description
<code>id</code>	INT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for each student
<code>name</code>	VARCHAR(100)	NOT NULL	Name of the student
<code>fee_due</code>	DECIMAL(10,2)	NOT NULL	Amount of fee due from the student

Details:

- **id:** A unique identifier for each student. It's an auto-incremented integer.
- **name:** The full name of the student. This field is required.
- **fee_due:** Represents the amount of fee due from the student. This field uses the DECIMAL type to handle currency values accurately.

Summary

1. Admins Table:

- Manages system administrators with unique IDs and hashed passwords for secure access.

2. Accountants Table:

- Manages accountants who handle student records and perform fee management, with unique usernames and hashed passwords.

3. **Students Table:**

- Stores student information and fee due amounts, enabling accountants to track and manage student records effectively.

CTAG