**Student Management System**

## Goal:

To build a role-based system where **Admins**, **Teachers**, and **Students** can perform specific operations like registering, verifying users, marking/viewing attendance, requesting changes, and managing classes.

1. **User Roles & Functionalities:**

2.1. **Admin**

1. Can verify and update any user
2. Can assign teachers to classrooms
3. Can view system-wide attendance reports

2.2. **Teacher**

1. Can view students in their assigned classroom
2. Can mark attendance
3. Can view attendance history
4. Can handover class to another teacher
5. Can promote students to the next academic year

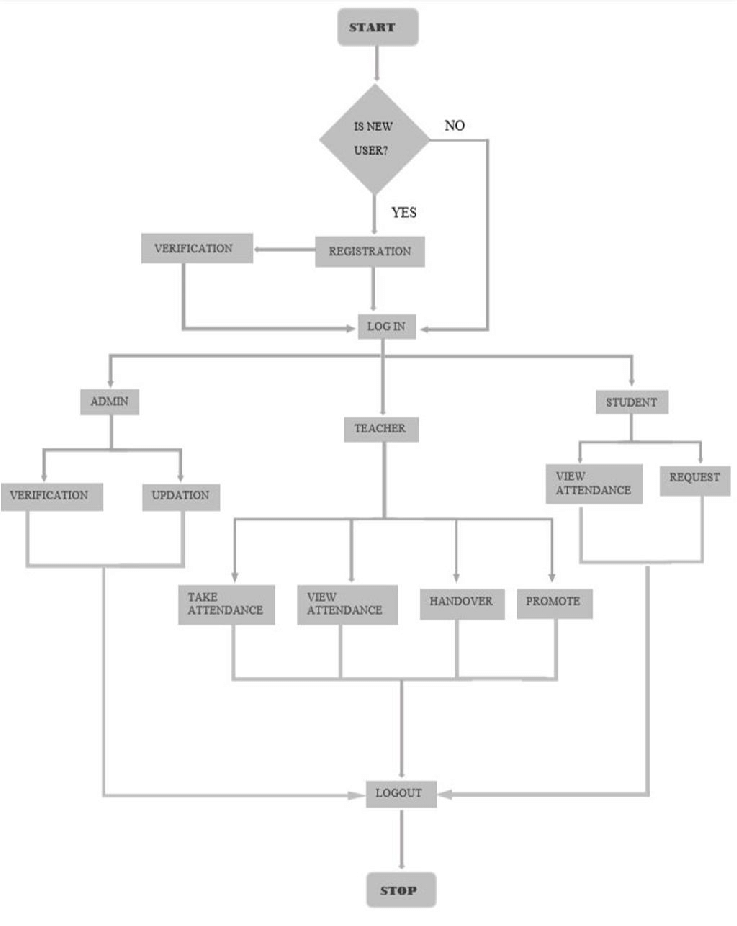
2.3 . **Student**

1. Can view own attendance
2. Can request corrections if attendance is incorrect

## 3.Tech Stack Overview

| **Layer** | **Tool** |
| --- | --- |
| Frontend | React.js, Axios, React Router |
| Backend | Django, Django REST Framework |
| Auth | JWT (SimpleJWT) |
| Database | MySQL |

**4.FLOWCHART**



## 5.Database Models (Conceptual)

| **Entity** | **Attributes** |
| --- | --- |
| **User** | id, name, email, password, role (admin/teacher/student), is\_verified |
| **Classroom** | id, name, assigned\_teacher |
| **StudentProfile** | user (FK), classroom (FK) |
| **Attendance** | student (FK), date, status, recorded\_by (FK) |
| **AttendanceRequest** | student (FK), date, reason, status (pending/approved/rejected) |
| **Handover** | classroom, from\_teacher, to\_teacher, date |
| **Promotion** | student, from\_class, to\_class, date |

1. **Users api endpoints ( http://127.0.0.1:8000 )**

| **Method** | **Endpoint** | **Description** | **Required Payload (JSON)** | **Response Example** |
| --- | --- | --- | --- | --- |
| POST | /api/register/ | Register a new user (student/teacher) | { "id": "T001", "name": "Alice", "email": "alice@example.com", "password": "pass123", "role": "teacher" } | { "message": "User registered successfully." } |
| POST | /api/login/ | Login and receive JWT token | { "id": "T001", "password": "pass123" } | { "access": "...", "refresh": "..." } |
| GET | /api/students/ | Get all students (Auth required) | **Header:** Authorization: Bearer <access\_token> | [{"id": "S001", "name": "John", "email": "john@example.com"}] |
| GET | /api/teachers/ | Get all teachers (Auth required) | **Header:** Authorization: Bearer <access\_token> | [{"id": "T001", "name": "Alice", "email": "alice@example.com"}] |
| DEL | /api/delete-user/<str:user\_id>/ | Delete Specified User  (Auth required) | **Header :**  Authorization: Bearer <access\_token> | {"detail": "User deleted successfully."  } |