* *Team 4 (Roll:07,09,21,26,28,37): School Management System*
* *Description:*
* *Create a system to manage students, teachers, and courses.*
* *Features:*
* *Student Management:*
* *○ Register and manage student information.*
* *○ Assign students to courses.*
* *Teacher Management:*
* *○ Add teacher profiles.*
* *○ Assign teachers to specific courses.*
* *Course Management:*
* *○ Create and manage courses (online, offline).*
* *○ Generate a class schedule.*
* *Grade Tracking:*
* *○ Record and display student grades.*
* *○ Generate grade reports.*

**Creating a School Management System in Java involves designing classes that represent students, teachers, courses, and grade tracking. Below is a simple implementation using Object-Oriented Programming (OOP) principles**.

**#**

Features of the System

Student Management: Add, remove, and list students.

Teacher Management: Add, remove, and list teachers.

Course Management: Assign courses to students and teachers.

Grade Tracking: Store and retrieve student grades.

**#**

Implementation in Java

Step 1: Define Models (Student, Teacher, Course, Grade)

***1. Student Class***

import java.util.ArrayList;

import java.util.List;

class Student {

private int id;

private String name;

private List<Course> courses;

private List<Grade> grades;

public Student(int id, String name) {

this.id = id;

this.name = name;

this.courses = new ArrayList<>();

this.grades = new ArrayList<>();

}

public void enrollInCourse(Course course) {

courses.add(course);

}

public void addGrade(Grade grade) {

grades.add(grade);

}

public void displayInfo() {

System.out.println("Student ID: " + id + ", Name: " + name);

}

}

***2. Teacher Class***

class Teacher {

private int id;

private String name;

public Teacher(int id, String name) {

this.id = id;

this.name = name;

}

public void displayInfo() {

System.out.println("Teacher ID: " + id + ", Name: " + name);

}

}

***3. Course Class***

class Course {

private int courseId;

private String courseName;

private Teacher teacher;

public Course(int courseId, String courseName, Teacher teacher) {

this.courseId = courseId;

this.courseName = courseName;

this.teacher = teacher;

}

public void displayInfo() {

System.out.println("Course ID: " + courseId + ", Name: " + courseName + ", Taught by: " + teacher);

}

}

***4. Grade Class***

class Grade {

private Student student;

private Course course;

private double score;

public Grade(Student student, Course course, double score) {

this.student = student;

this.course = course;

this.score = score;

}

public void displayGrade() {

System.out.println("Student: " + student + ", Course: " + course + ", Score: " + score);

}

}

***Step 2: Main Class (School Management System)***

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class SchoolManagementSystem {

private List<Student> students = new ArrayList<>();

private List<Teacher> teachers = new ArrayList<>();

private List<Course> courses = new ArrayList<>();

public void addStudent(int id, String name) {

students.add(new Student(id, name));

System.out.println("Student added successfully.");

}

public void addTeacher(int id, String name) {

teachers.add(new Teacher(id, name));

System.out.println("Teacher added successfully.");

}

public void addCourse(int courseId, String courseName, int teacherId) {

Teacher teacher = teachers.stream()

.filter(t -> t.getId() == teacherId)

.findFirst()

.orElse(null);

if (teacher != null) {

courses.add(new Course(courseId, courseName, teacher));

System.out.println("Course added successfully.");

} else {

System.out.println("Teacher not found.");

}

}

public void displayStudents() {

for (Student student : students) {

student.displayInfo();

}

}

public void displayTeachers() {

for (Teacher teacher : teachers) {

teacher.displayInfo();

}

}

public static void main(String[] args) {

SchoolManagementSystem sms = new SchoolManagementSystem();

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\nSchool Management System");

System.out.println("1. Add Student");

System.out.println("2. Add Teacher");

System.out.println("3. Display Students");

System.out.println("4. Display Teachers");

System.out.println("5. Exit");

System.out.print("Enter choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter Student ID: ");

int studentId = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter Student Name: ");

String studentName = scanner.nextLine();

sms.addStudent(studentId, studentName);

break;

case 2:

System.out.print("Enter Teacher ID: ");

int teacherId = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter Teacher Name: ");

String teacherName = scanner.nextLine();

sms.addTeacher(teacherId, teacherName);

break;

case 3:

sms.displayStudents();

break;

case 4:

sms.displayTeachers();

break;

case 5:

System.out.println("Exiting system.");

break;

default:

System.out.println("Invalid choice. Try again.");

}

} while (choice != 5);

scanner.close();

}

}

***How the System Works***

1. Run the program: A menu appears with options to manage students and teachers.

2. Add Students and Teachers: Users can enter details, and the system will store them in lists.

3. Display Students and Teachers: Shows all stored students and teachers.

4. Exit System: Terminates the program when the user selects the exit option.

***Future Enhancements***

Implement a database (MySQL, PostgreSQL) for persistent storage.

Improve course and grade management.

Create a GUI using JavaFX or Swing for better user interaction.

Implement user authentication for different roles (admin, teacher, student).