**Project Report on**



**STUDENT MANAGEMENT SYSTEM**



**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Description** | **Page No.** |
| 1 | Problem Statement | **1** |
| 2 | Objectives | **2** |
| 3 | Approach | **3** |
| 4 | Code Implemention | **4-6** |
| 5 | Output | **7** |
|  |  |  |

**Problem Statement:**

The project aims to create a simple **Student Management System** that allows users to:

1. Add new students with ID, name, age, and marks.
2. Display all students.
3. Find the student with the highest marks.
4. Calculate the average marks of all students.
5. Delete a student by ID.

This system will serve as a demonstration of object-oriented programming and basic data management using Java.

**2. Objectives**

* To design a system for managing student records.
* To implement basic functionalities like adding, deleting, displaying, and computing statistics.
* To create a menu-driven interface for user interaction.
* To practice modular and structured programming using Java.

**3. Approach:**

* **Data Structure**: An ArrayList is used to store student objects.
* **Object-Oriented Principles**: A Student class encapsulates student attributes such as ID, name, age, and marks.
* **Modular Design**: Separate methods for each operation, such as adding, displaying, and deleting students.
* **Menu-driven Interface**: The system uses a loop-based menu to allow users to interactively manage student data.

**4. Solution / Code Implementation:**

class Student {

int id;

String name;

int age;

double marks;

public Student(int id, String name, int age, double marks) {

this.id = id;

this.name = name;

this.age = age;

this.marks = marks;

}

public void display() {

System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age + ", Marks: " + marks);

}

}

import java.util.ArrayList;

import java.util.Scanner;

public class StudentManagementSystem {

static ArrayList<Student> students = new ArrayList<>();

static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

boolean exit = false;

while (!exit) {

System.out.println("\n--- Student Management System ---");

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

System.out.println("2. Display All Students");

System.out.println("3. Find Student with Highest Marks");

System.out.println("4. Calculate Average Marks");

System.out.println("5. Delete Student by ID");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

switch (choice) {

case 1: addStudent(); break;

case 2: displayAllStudents(); break;

case 3: findHighestMarks(); break;

case 4: calculateAverageMarks(); break;

case 5: deleteStudentById(); break;

case 6: exit = true; break;

default: System.out.println("Invalid option! Try again.");

}

}

}

public static void addStudent() {

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

int id = scanner.nextInt();

scanner.nextLine();

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

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

double marks = scanner.nextDouble();

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

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

}

public static void displayAllStudents() {

if (students.isEmpty()) {

System.out.println("No students available.");

} else {

for (Student student : students) {

student.display();

}

}

}

public static void findHighestMarks() {

if (students.isEmpty()) {

System.out.println("No students available.");

return;

}

Student topStudent = students.get(0);

for (Student student : students) {

if (student.marks > topStudent.marks) {

topStudent = student;

}

}

System.out.println("Student with highest marks:");

topStudent.display();

}

public static void calculateAverageMarks() {

if (students.isEmpty()) {

System.out.println("No students available.");

return;

}

double totalMarks = 0;

for (Student student : students) {

totalMarks += student.marks;

}

double averageMarks = totalMarks / students.size();

System.out.println("Average marks of the class: " + averageMarks);

}

public static void deleteStudentById() {

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

int id = scanner.nextInt();

boolean found = false;

for (Student student : students) {

if (student.id == id) {

students.remove(student);

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

found = true;

break;

}

}

if (!found) {

System.out.println("Student with ID " + id + " not found.");

}

}

}

**5. Output:**

1. Adding Students:

**Enter Student ID: 101**

**Enter Student Name: Alice**

**Enter Student Age: 20**

**Enter Student Marks: 85.5**

**Student added successfully!**

1. Displaying All Students:

**id: 101, Name: Alice, Age: 20, Marks: 85.5**

1. Finding the Student with Highest Marks:

**Student with highest marks:**

**ID: 101, Name: Alice, Age: 20, Marks: 85.5**

1. Calculating Average Marks:

**Average marks of the class: 85.5**