**Student Grade Tracker – Project Report**

**Project Details**

**Project Title**: Student Grade Tracker  
**Internship Program**: Code Alpha  
**Student Name**: Pranitha  
**College**: SR University

**Objective**

To develop a **Student Grade Tracker** application using **Java** that allows:

* Adding student records
* Viewing all students
* Updating grades
* Deleting records
* Searching students
* Generating summary reports (average, highest, lowest grades)

**Technologies Used**

* Java Programming Language
* Object-Oriented Programming (OOP)
* ArrayList for dynamic data management
* Console-based User Interface

**Sample Students Added**

| **Name** | **Grade** |
| --- | --- |
| Pranitha | 85 |
| Ramesh | 74 |
| Ayesha | 92 |
| Kiran | 67 |
| Meena | 78 |
| Farhan | 90 |
| Sneha | 88 |
| Arjun | 69 |
| Lakshmi | 81 |
| Rajesh | 76 |

**Sample Output Demonstration**

* **View All Students** → Lists all records.
* **Search “Ayesha”** → Output: Found, Grade = 92.
* **Update “Kiran”** → Change grade from 67 to new grade.
* **Delete “Meena”** → Record removed.
* **Summary Report** →
  + Total Students: 9
  + Average: ≈ 80
  + Highest: 92
  + Lowest: 67

**✅ Conclusion**

This project demonstrates practical implementation of **Java OOP**, user input handling, and real-world data management using console-based applications. It is ideal for college-level or internship projects.

**Full Java Source Code**

java

CopyEdit

import java.util.\*;

public class StudentGradeTracker {

static class Student {

private String name;

private int grade;

public Student(String name, int grade) {

this.name = name;

this.grade = grade;

}

public String getName() { return name; }

public int getGrade() { return grade; }

public void setGrade(int grade) { this.grade = grade; }

}

private static ArrayList<Student> studentList = new ArrayList<>();

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

public static void main(String[] args) {

preloadStudents();

System.out.println("Welcome to the Student Grade Tracker!");

boolean running = true;

while (running) {

printMenu();

int choice = getUserChoice();

switch (choice) {

case 1 -> addStudent();

case 2 -> viewAllStudents();

case 3 -> updateStudentGrade();

case 4 -> deleteStudent();

case 5 -> searchStudent();

case 6 -> printSummary();

case 7 -> {

System.out.println("Exiting... Thank you!");

running = false;

}

default -> System.out.println("Invalid choice.");

}

}

}

private static void preloadStudents() {

studentList.add(new Student("Pranitha", 85));

studentList.add(new Student("Ramesh", 74));

studentList.add(new Student("Ayesha", 92));

studentList.add(new Student("Kiran", 67));

studentList.add(new Student("Meena", 78));

studentList.add(new Student("Farhan", 90));

studentList.add(new Student("Sneha", 88));

studentList.add(new Student("Arjun", 69));

studentList.add(new Student("Lakshmi", 81));

studentList.add(new Student("Rajesh", 76));

}

private static void printMenu() {

System.out.println("\n=== Menu ===");

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

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

System.out.println("3. Update Grade");

System.out.println("4. Delete Student");

System.out.println("5. Search Student");

System.out.println("6. Summary Report");

System.out.println("7. Exit");

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

}

private static int getUserChoice() {

while (!scanner.hasNextInt()) {

System.out.print("Enter a number: ");

scanner.next();

}

int choice = scanner.nextInt();

scanner.nextLine();

return choice;

}

private static void addStudent() {

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

String name = scanner.nextLine();

int grade = getValidGrade();

studentList.add(new Student(name, grade));

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

}

private static int getValidGrade() {

int grade;

while (true) {

System.out.print("Enter grade (0-100): ");

while (!scanner.hasNextInt()) {

System.out.print("Enter a number: ");

scanner.next();

}

grade = scanner.nextInt();

if (grade >= 0 && grade <= 100) break;

System.out.println("Grade must be 0-100.");

}

scanner.nextLine();

return grade;

}

private static void viewAllStudents() {

if (studentList.isEmpty()) {

System.out.println("No records found.");

return;

}

System.out.printf("\n%-20s %-5s\n", "Name", "Grade");

for (Student s : studentList) {

System.out.printf("%-20s %-5d\n", s.getName(), s.getGrade());

}

}

private static void updateStudentGrade() {

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

String name = scanner.nextLine();

for (Student s : studentList) {

if (s.getName().equalsIgnoreCase(name)) {

int newGrade = getValidGrade();

s.setGrade(newGrade);

System.out.println("Grade updated.");

return;

}

}

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

}

private static void deleteStudent() {

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

String name = scanner.nextLine();

Iterator<Student> it = studentList.iterator();

while (it.hasNext()) {

if (it.next().getName().equalsIgnoreCase(name)) {

it.remove();

System.out.println("Student deleted.");

return;

}

}

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

}

private static void searchStudent() {

System.out.print("Enter name to search: ");

String name = scanner.nextLine();

for (Student s : studentList) {

if (s.getName().equalsIgnoreCase(name)) {

System.out.printf("Found: %s - Grade: %d\n", s.getName(), s.getGrade());

return;

}

}

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

}

private static void printSummary() {

if (studentList.isEmpty()) {

System.out.println("No data.");

return;

}

int total = 0, max = Integer.MIN\_VALUE, min = Integer.MAX\_VALUE;

for (Student s : studentList) {

int g = s.getGrade();

total += g;

if (g > max) max = g;

if (g < min) min = g;

}

double avg = (double) total / studentList.size();

System.out.printf("\nTotal: %d\nAverage: %.2f\nHighest: %d\nLowest: %d\n",

studentList.size(), avg, max, min);

}

}