NAME- PARTH GOYAL
ROLL NO- 2401420028
COURSE- BTECH-CSE-DS
SEMESTER- 3rd

JAVA ASSIGNMENT

STUDENT RESULT MANAGEMENT SYSTEM

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class StudentResultManagementSystem{

    private static int MAX_STUDENTS = 100;
    private Student[] students = new student[MAX_STUDENTS];
    private int count=0;
    Scanner Input = new Scanner(System.in);
    private static final int PASS_MARK = 33;

    public void AddStudent() throws InvalidMarksException{
        System.out.print("Enter Roll Number: ");
        int Roll = Input.nextInt();
        Input.nextLine();

        if (FindIndexByRoll(Roll) != -1){
            System.out.println("Error! Already Exists!");
            return;
        }

        System.out.print("Enter student Name: ");
        String Name = Input.nextLine();
```

```
        int[] Marks = new int[3];
        for (int i=0; i<3; i++) {
            System.out.print("Enter marks" + (i+1) + ": ");
            Marks[i] = Input.nextInt();
        }
        Input.nextLine();

        Student S = new Student(Roll, Name, Marks);
        S.ValidateMarks();
        if (Count < MAX_STUDENTS) {
            Students[Count++] = S;
            System.out.println("Successfully added!");
        }
        else {
            System.out.println("Full! Cannot add more!");
        }
    }

    public void ShowStudentDetails() {
        System.out.print("Enter Roll Number to Search: ");
        int Roll = Input.nextInt();
        Input.nextLine();

        int Idx = FindIndexByRoll(Roll);
        if (Idx == -1) {
            System.out.println("Not found!");
        }
        else {
            Students[Idx].DisplayResult();
            System.out.println("Search completed!");
        }
    }
}
```

DATE

```
private int FindIndexByRoll(int Roll){
    for(int i=0; i<Count; i++){
        if(students[i] != null && students[i].GetRoll()==Roll){
            return i;
        }
    }
    return -1;
}

public void Menu(){
    boolean Running = true;
    try{
        while(Running){
            System.out.println();
            System.out.println("SRMS");
            System.out.println("1. Add Student");
            System.out.println("2. Show Student Details");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");

            int Choice;
            try{
                Choice = Input.nextInt();
                Input.nextLine();
            }
            catch(InputMismatchExcebtion IME){
                System.out.println("Invalid Input!");
                Input.nextLine();
                continue;
            }
```

DATE

```
switch (choice) {
    case 1:
        try {
            AddStudent();
        }
        catch (InvalidMarksException IME) {
            System.out.println(" Error!");
        }
        catch(InputMismatchException IME2) {
            System.out.println(" Input Error!");
            Input.nextLine();
        }
        break;
    case 2:
        try {
            ShowStudentDetails();
        }
        catch(InputMismatchException IME) {
            system.out.println(" Input error!");
            Input.nextLine();
        }
        break;
    case 3:
        System.out.println(" Exiting---- Thank You!");
        Running=false;
        break;
    default:
        System.out.println(" Invalid choice! ");
        }
    }
}
finally {
```

DATE

```
            if (Input != null) {
                Input.close();
            }
        }
    }

    public static void main(String[] args) {
        StudentResultManagementSystem System = new SRMS();
        System.Menu();
    }

    static class Student {
        private int Roll;
        private String Name;
        private int[] Marks;

        public Student(int Roll, String Name, int[] Marks) {
            this.Roll = Roll;
            this.Name = Name;
            this.Marks = Marks;
        }

        public int GetRoll() {
            return Roll;
        }

        public void ValidateMarks() throws InvalidMarksExceptian {
            if (Marks == null || Marks.length != 3) {
                throw new InvalidMarksException("Incomplete!");
            }
            for (int i = 0; i < Marks.length; i++) {
                if (Marks[i] < 0 || Marks[i] > 100) {
                    throw new InvalidMarksException("Invalid!");
                }
            }
        }
    }
```

DATE

```
public double CalculateAverage() {
    double sum = 0;
    for (int M : Marks) sum += M;
    return sum / Marks.length;
}

public void DisplayResult() {
    System.out.println("Roll Number: " + Roll);
    System.out.println("Student Name: " + Name);
    System.out.print("Marks: ");
    for (int M : Marks) {
        System.out.print(M + " ");
    }
    System.out.println();
    double AVG = CalculateAverage();
    System.out.println("Average: " + AVG);

    boolean Pass = true;
    for (int M : Marks) {
        if (M < PASS_MARK) {
            Pass = false;
            break;
        }
    }
    System.out.println("Result: " + (Pass ? "Pass" : "Fail"));
}
}

static class InvalidMarksException extends Exception {
    public InvalidMarksException(String message) {
        super(message);
    }
}
}
```

pragya

PAGE