Project Title: Student Result System (Java)

Student Name: Anirudh Sharma

Reg No. : 24MIM10039

Course: Vityarthi – Programming in Java

Submitted To: (Dr. Anand Motwani)

Date: (24.11.25)

What to do ?

- The project is a Java console application
- It records student details and marks
- Calculates total, percentage, and grade
- Uses interfaces, abstract classes, custom exceptions
- Saves report files into storage
- Demonstrates modular Java coding

Problem Statement

The aim is to build a simple Java-based result processing system that takes a student's details and their marks (3 subjects), calculates total, percentage, grade, and generates a report card stored as a file.
The project must demonstrate core Java OOP concepts and basic file handling.

Objectives

- Accept and validate student data
- Calculate total marks and percentage
- Compute grade using an interface
- Generate formatted report
- Save output as a text file
- Demonstrate:
  - Interfaces
  - Abstract classes
  - Packages
  - Exception handling
  - File I/O
  - Anonymous class

Scope of the Project

- Single-user console program
- Handles any number of students
- Saves each report file automatically
- No database required
- Ideal for small-scale academic environments

System Design

Architecture Diagram (Text Outline)

src/

├── main/ (MainApp.java)

├── model/ (Student.java, Marks.java)

├── services/ (GradeCalculator, CalculatorImpl, ReportBase, ReportGenerator)

└── utils/ (InputValidator, FileManager, InvalidMarksException)

storage/

out/

Implementation

Explain each module briefly:

model package

- **Student.java:** Holds name, roll, 3 marks; calculates total and percentage.
- **Marks.java:** Optional structure for marks.

services package

- **GradeCalculator.java:** Interface for grade logic
- **CalculatorImpl.java:** Implements grade rules
- **ReportBase.java:** Abstract class
- **ReportGenerator.java:** Saves report files via FileManager

utils package

- **InputValidator.java**: Ensures marks are valid

- **InvalidMarksException.java**: Custom exception
- **FileManager.java**: Writes report text into storage/

main package

- **MainApp.java:** CLI menu
    - Add student
    - Generate report
    - List students
    - Exit

OUTPUT

----- Student Result System -----

1. Add Student

2. Generate Report

3. List Students

4. Exit

Enter Name: Ravi

Enter Roll: 101

Enter marks for Subject 1: 78

Enter marks for Subject 2: 65

Enter marks for Subject 3: 80

[INFO] Student Input Completed.

Student Added Successfully!

Enter Roll for Report: 101

----- REPORT CARD -----

Name: Ravi

Roll: 101

Marks: [78, 65, 80]

Total: 223

Percentage: 74.33

Grade: A

Conclusion

- Successfully built modular Java project
- Demonstrated all required OOP concepts
- Achieved complete input → process → output flow
- File generation works as expected