**A**

**PROJECT REPORT ON**

**(STUDENT GRADE CALCULATOR)**

**REPORT ON STUDENT GRADE CALCULATOR**

# CHAPTER 1 - INTRODUCTION

* 1. **EXISTING SYSTEM**

The current manual grading system used in many educational institutions involves teachers calculating grades by hand or using spreadsheets. This process can be time-consuming, error-prone, and inefficient. Teachers have to repeatedly perform calculations for total marks, average marks, and grades, which increases the likelihood of mistakes, especially when managing a large number of students.

## Manual Calculations

Teachers often calculate total marks, averages, and grades manually or using basic tools like calculators. This repetitive process is prone to human errors, especially when dealing with multiple students or subjects.

## Time-Consuming

The manual system requires a significant amount of time to compute results, particularly for large classes with numerous subjects.

# PROBLEM DEFINITION- NEED OF COMPUTERIZATION

The manual grading system currently employed in many educational institutions faces significant challenges, which make the process inefficient, error-prone, and time-consuming. These challenges highlight the urgent need for computerization in student grade management. Below are the key aspects of the problem and why computerization is essential:

## Data Management Issues

Storing and retrieving student records is cumbersome in a paper-based system. Records can be misplaced, damaged, or lost over time, leading to inefficiencies and potential disputes.

## Scalability Challenges

As the number of students and subjects increases, managing the workload manually becomes impractical. A larger dataset also increases the likelihood of errors.

## Limited Accessibility and Analysis

Physical records are accessible only at specific locations and are challenging to analyze for trends or performance evaluations over time.

# CHAPTER 2 PROPOSED SYSTEM

* 1. **PROPOSED SYSTEM**

The proposed system aims to address the inefficiencies and limitations of the manual grading system by introducing an automated and computerized **Student Grade Management System**. This system streamlines the process of calculating student grades, ensures accuracy, and simplifies data management.

## Standardized Grading Criteria

Uses consistent and transparent grading algorithms, ensuring fair assessments for all students.

Customizable grading schemes can be applied if needed.

## Data Security and Backup

Protects sensitive student data through secure storage and user authentication.

Includes backup features to ensure data recovery in case of accidental loss.

## Improved Efficiency

Significantly reduces the time required for calculations and report generation. Frees up educators to focus more on teaching and less on administrative tasks.

# OBJECTIVES OF SYSTEM

## Automate Grade Calculation:

Simplify the process of determining total marks, average marks, and final grades based on subject scores.

## Enhance Accuracy:

Minimize manual errors by using automated calculations and a predefined grading scale.

## Provide Clear Reporting:

Generate a structured and easy-to-understand report that summarizes a student's performance across multiple subjects.

## Support Dynamic Inputs:

Handle varying numbers of subjects and adapt to diverse scoring requirements.

## Save Time and Effort:

Streamline the evaluation process for teachers, students, and administrators by providing instant results.

# USER REQUIREMENTS

## Functional Requirements

* + - **Grade Calculation**

Ability to input marks for multiple subjects per student.

Automatically calculate total marks, average, and assign grades based on predefined criteria.

## Student Record Management

Add, edit, delete, and retrieve student details (e.g., name, ID, subjects, and marks).

Maintain a centralized database of student records.

## Report Generation

Generate individual student reports showing marks, average, grade, and overall performance.

Provide class or batch-wise performance summaries for administrators.

## Standardized Grading

Use predefined grading criteria for assigning grades.

Allow customization of grading schemes to align with institutional policies.

## Non-Functional Requirements.

* + - **Reliability**

Deliver accurate results consistently with minimal downtime or errors. Ensure data integrity and prevent duplication or corruption.

## Compatibility

Ensure compatibility across different devices and operating systems. Integrate with existing school management software if required.

.

# OPERATING ENVIRONMENT – HARDWARE AND SOFTWARE

**Hardware Requirements**

**For Client Devices (User Systems)**

1. **Minimum Requirements:**
   * **Processor:** Dual-core processor (e.g., Intel Core i3 or equivalent)
   * **RAM:** 4 GB
   * **Storage:** 500 MB of free disk space
   * **Display:** 1024x768 resolution monitor
   * **Input Devices:** Keyboard and mouse or touchscreen (if applicable)

## Recommended Requirements:

* + **Processor:** Quad-core processor (e.g., Intel Core i5 or equivalent)
  + **RAM:** 8 GB
  + **Storage:** 1 GB of free disk space
  + **Display:** Full HD resolution (1920x1080)

**Software Requirements**

**For Client Devices (User Systems)**

1. **Operating System:**
   * Windows 10 or later / macOS 10.13 or later / Linux distributions (Ubuntu, Fedora, etc.)

## Software Dependencies:

* + Java Runtime Environment (JRE) 8 or later (if the system is built using Java).
  + Web browser (Google Chrome, Mozilla Firefox, or Microsoft Edge) for web- based access.
  + PDF viewer (e.g., Adobe Reader) for viewing reports.

**CHAPTER 3 - ANALYSIS AND DESIGN**

* 1. **FLOW CHART :**

****



Import

Required Class

Input student's name.



Input student's ID

Input the number of

subjects.

Initialize an array to store scores.



Loop to

input score of each

subject



Control statement

if else to determine the grade

Display the student

details total marks average marks and grade

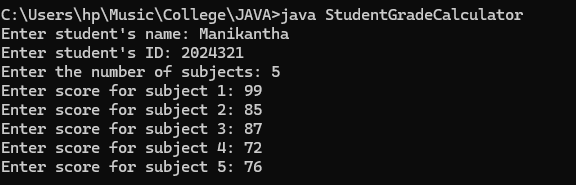
Close

scanner

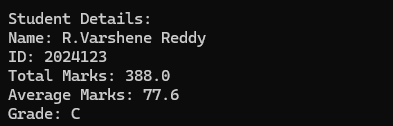
Calculate total of scores and average

* 1. **Screen Shots**

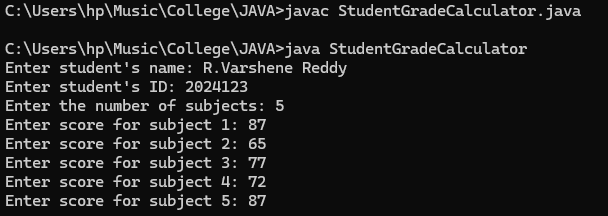
## EXAMPLE 1

****

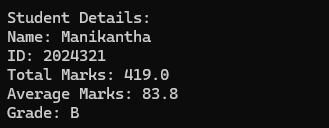
**OUTPUT FOR EXAMPLE 1**

****

## EXAMPLE 2

****

**OUTPUT FOR EXAMPLE 2**



# CHAPTER 4 CONCLUSION

* 1. **Limitations & Drawbacks**

## Lack of Support for Multiple Students

The program only calculates grades for one student at a time. It cannot handle multiple students or a class of students in one session.

## No Handling of Special Cases

The system does not take into account edge cases or special scenarios, such as a student who has zero scores for all subjects, or one with extra credit or bonus points.

## Fixed Grading Scale

The grading scale is hardcoded in the getGrade() method (e.g., A for 90+, B for 80-89, etc.), and it does not allow for flexibility or customization based on different institutions or grading policies.

## Limited Error Handling

The program does not include any exception handling to deal with potential runtime errors (such as InputMismatchException if the user enters a non-numeric value).

## No Report Generation or Export Options

The program displays the results on the console, but there is no option to save or export the results to a file (e.g., CSV or PDF format), making it difficult to share or store the

data.

# FUTURE ENHANCEMENT:

## Dynamic Grading Criteria

Allow users to define and modify grading scales dynamically. For instance,

teachers could adjust the grade boundaries (e.g., A: 85+, B: 75–84, etc.) according to their institution’s grading policy or customize the scoring weight for different assignments, exams, and projects

## Integration with Student Information Systems (SIS)

Integrate the grade calculator with existing Student Information Systems (SIS) or Learning Management Systems (LMS) used by educational institutions. This will allow automatic fetching of student data (name, ID, subjects, etc.) and grades

from centralized databases, minimizing data entry errors and saving time.

## Report Generation and Exporting

Introduce functionality to generate and export grade reports to different formats, such as PDF, Excel, or CSV. Teachers could easily generate detailed reports for students, parents, or administrators.