**Smart Academic Performance Evaluator**

A Project Report Submitted in the partial fulfillment of the requirements of the course titled

“Problem Solving Through Programming (JAVA)”

BACHELOR OF TECHNOLOGY

**In**

# DEPARTMENT OF FRESHMAN ENGINEERING

**By**

# POLISETTY SREEHARSHA 2520030123

KANITHI SHANKAR 2520030063

MOHIT VENKATA SAI DUGGANABOINA 2520030106

M SAI AJITH REDDY 2520090186

POTLABATHINI SHASHANTH 2520090211

M VIJAYA LAKSHMI 2520090201

Under the Esteemed Guidance of

**Mrs. NM. Deepika**

**Designation**

**Department of Freshman Engineering**

**K L (Deemed to be) University**

**DEPARTMENT OF FRESHMAN ENGINEERING**

****

**Declaration**

The Project Report entitled “Smart Academic Performance Evaluator” is a record of Bonafide work of Polisetty Sreeharsha – 2520030123, Kanithi Shankar – 2520030063, Mohit Venkata Sai Dugganaboina– 2520030106, M. Sai Ajith Reddy –2520090186,

M. Vijaya Lakshmi – 2520090201, Potlabathini Shashanth – 2520090211 submitted in partial fulfillment of the requirements of the course titled “Problem Solving Through Programming (JAVA)” under the B. Tech Ist Year Trimester - I program in Department of Freshman Engineering at K L University. The results presented in this report have not been copied from any other department, university, or institute.

# POLISETTY SREEHARSHA – 2520030123

# KANITHI SHANKAR - 2520030063

# MOHIT VENKATA SAI DUGGANABOINA - 2520030106

# M . AJITH SAI REDDY - 2520090186

# M . VIJAYA LAKSHMI – 2520090201

# POTLABATHINI SHASHANTH - 2520090211

**K L (Deemed to be) University**

# DEPARTMENT OF FRESHMAN ENGINEERING

****

**CERTIFICATE**

This is certify that the java project based report entitled **“Smart Academic Performance Evaluator”** is a bonafide work done and submitted by **Polisetty Sreeharsha – 2520030123 , Kanithi Shankar – 2520030063, Mohit Venkata Sai Dugganaboina– 2520030106, M. Sai Ajith Reddy –2520090186, M. Vijaya Lakshmi – 2520090201, Potlabathini Shashanth – 2520090211,** in partial fulfillment of the requirements of the course titled “Problem Solving Through Programming (JAVA)” under the B.Tech Ist Year Trimester - I program in Department of Freshman Engineering, K L (Deemed to be University), during the academic year **2025-2026.**

**Signature of the Guide**

**Signature of the Course Coordinator Signature of the HOD**

**ACKNOWLEDGEMENT**

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our wish to express my sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my Course Coordinator **Dr Y Ashok**, and my guide **Ms. N.M .Deepika,** Department of Freshman Engineering which cannot be expressed in words for his/her tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. N. Chaitanya Kumar**, Head of the ***Department for Freshman Engineering*** for providing us with adequate facilities, ways and means by which we are able to complete this project-based Lab.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project. Finally, I sincerely thank my parents, friends and classmates for their kind help and cooperation during my work.

# POLISETTY SREEHARSHA – 2520030123

# KANITHI SHANKAR - 2520030063

# MOHIT VENKATA SAI DUGGANABOINA - 2520030106

# M . AJITH SAI REDDY - 2520090186

# M . VIJAYA LAKSHMI – 2520090201

# POTLABATHINI SHASHANTH - 2520090211

**ABSTRACT**

**The Smart Academic Performance Evaluator is an efficient and user-friendly tool designed to simplify the process of calculating student results. Instead of manually adding marks and determining grades, the system automates the entire evaluation process. Users simply enter the student’s name, number of subjects, and marks obtained, and the program instantly computes the total, average percentage, and final grade. This helps teachers and institutions save time, reduce calculation errors, and ensure consistent evaluation—especially when handling multiple students.**

**The application is developed using the Java programming language, the project demonstrates core programming concepts such as loops, conditional statements, arrays, arithmetic operations, string handling, exception handling, and event-driven GUI design using Swing. It serves as an excellent beginner-to-intermediate project for understanding practical application development and applying problem-solving techniques in real scenarios.**

**The system currently supports dynamic input validation and real-time result generation, making it both robust and reliable. In the future, the application can be extended to include multi-student management, data storage using files or databases, automated report generation, graphical performance charts, and enhanced UI features. Overall, the Smart Academic Performance Evaluator provides a smart, accurate, and effective way to evaluate academic performance with ease.**

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Chapters** | **Topics** | **Page.no** |
|  |  | Acknowledgement | iv |
|  |  | Abstract | v |
| 1 | Introduction | * 1. Background of the project   2. Problem statement | 1  2 |
| 2 | System Architecture | * 1. High-level architecture diagram/Block diagram   2. Class Diagram | 3,4  5 |
| 3 | CO’s Attainments | * 1. CO1 Attainment   2. CO2 Attainment   3. CO3 Attainment   4. CO4 Attainment   5. CO5 Attainment   3.6 CO6 Attainment | 6,7  8,9  10,11  11,12  13,14  14,15 |
| 4 | Screen Shots | * 1. Screen Shots | 16-22 |
| 5 | Testing | * 1. Test cases and results | 23-26 |
| 6 | Future Enhancements | * 1. Planned features   2. Possible integrations or optimizations | 27 |
| 7 | Conclusion | * 1. Summary of the project   2. What was achieved   7.3 Skills learned during development | 28 |
| 8 | References | - Books, tutorials, documentation sites used | 29,30 |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | Appendices | * Installation/setup instructions * User manual or guide   -Geo Tag photos with guide  -Review forms with guide signatures | 31-35 |

# CHAPTER -1 INTRODUCTION

## Background of the Project

**In schools and colleges, teachers regularly calculate student’s results based on their performance in multiple subjects. Although the process seems straightforward, it becomes repetitive and time-consuming when done for many students. Manual calculation of totals, averages, and grades also increases the chances of errors, especially during examination periods when teachers are under pressure.**

**With technology now playing a major role in education, there is a growing need for simple tools that can automate such routine academic tasks. Accurate result computation is essential because even minor mistakes can affect a student’s academic standing. To make this process faster, easier, and error-free, an automated solution becomes very useful.**

**The Smart Academic Performance Evaluator was developed to meet this need. It offers a simple and reliable way to input marks, validate them, and instantly compute the total, average, and grade. By reducing manual effort and minimizing the risk of calculation mistakes, this system supports teachers, students, and educational institutions in managing assessments more efficiently.**

* 1. **Problem statement**

**Manual calculation of student marks is time-consuming, repetitive, and prone to human error. Teachers often need to compute totals, averages, and grades for many students, which becomes tiring and may lead to mistakes. During manual evaluation, there is also no reliable way to detect incorrect or mismatched inputs, increasing the chances of inaccurate results.**

**Therefore, there is a need for an automated system that can:**

* **Accept student details and subject-wise marks**
* **Validate the input to prevent errors**
* **Instantly calculate total marks, average percentage, and grade**
* **Display results in a clear and organized format**
* **Reduce manual effort and ensure accurate evaluation**

**The Smart Academic Performance Evaluator addresses these challenges by offering a fast, reliable, and user-friendly solution for calculating academic performance.**

# CHAPTER -2 SYSTEM ARCHITECTURE

## High-level architecture diagram

# 

# A diagram of a software project AI-generated content may be incorrect.

* 1. **Class Diagram**

# A screenshot of a computer program AI-generated content may be incorrect.

# CHAPTER -3 CO’s ATTAINMENT

**3.1 CO1 Attainment**

|  |  |
| --- | --- |
| **CO1 Syllabus** | **CO1 Concepts Included in Project** |
| Apply fundamental programming constructs such as data types, operators, conditional and iterative statements in Java to develop logic-based solutions for basic computational problems. Students will learn to design simple algorithms, trace execution, and validate logic through hands-on coding task | Variables and Data Types  Arithmetic and Logical Operators  Conditional Statements (if–else)  Loops (for loop)  Basic Input Validation |

**3.1.1 Scenario’s for CO1 implementation.**

**The system stores student name, subject count, marks, total, average -Variables + Data types**

**It checks conditions like invalid inputs, empty fields, negative marks - if–else**

**It repeats operations for each subject -for loop**

**It calculates total, average, grade - operators**

**It prevents wrong inputs- basic validation**

**3.1.2 CO1 code screen shot.**

**Data type declarations**

**A black background with white text

AI-generated content may be incorrect.**

**Conditional Statements**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**Loop + Operators**

**A computer screen shot of a code

AI-generated content may be incorrect.**

**3.2 CO2 Attainment**

|  |  |
| --- | --- |
| **CO2 Syllabus** | **CO2 Concepts Included in Project** |
| **Design, trace, and optimize algorithms using one-dimensional and two-dimensional arrays to solve mathematical, quantitative, and real-world problems efficiently through search, sort, and matrix manipulation techniques. Students will also analyze the efficiency and accuracy of algorithmic approaches.** | **Arrays for storing marks**  **Looping to compute totals**  **Summation algorithm**  **Average and grade computation** |

**3.2.1 Scenario’s for CO2 implementation.**

**Marks are stored inside an array using**

**A loop calculates total marks and checks validity**

**Total and average percentage are computed using mathematical algorithms**

**Grade classification algorithm based on percentage**

**3.2.2 CO2 code screen shot.**

**Splitting marks into array**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Looping marks array**

**A computer screen shot of code

AI-generated content may be incorrect.**

**Average calculation**

**A screen shot of a computer

AI-generated content may be incorrect.**

**3.3 CO3 Attainment**

|  |  |
| --- | --- |
| **CO3 Syllabus** | **CO3 Concepts Included in Project** |
| **Construct and evaluate advanced problem-solving logic using strings, recursion, and bitwise operations for solving complex mathematical, pattern-based, and combinatorial problems relevant to competitive coding platforms. Students will be able to integrate mathematical reasoning and pattern recognition into coding strategies.** | **String Parsing**  **Structured Input Handling**  **Conversion of string data into numeric values**  **Splitting comma-separated values** |

**3.3.1 Scenario’s for CO3 implementation.**

**The program splits the marks entered as a single string into multiple values using .split()**

**Each value is trimmed and converted from String → Double**

**Any invalid numeric format is caught using exception handling**

**3.3.2 CO3 code screen shot.**

**Splitting comma-separated marks**

****

**Trimming and parsing inside loop**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**Error message for invalid string number**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**3.4 CO Attainment**

|  |  |
| --- | --- |
| **CO4 Syllabus** | **CO4 Concepts Included in Project** |
| **Develop structured and modular programs by applying object-oriented programming principles such as encapsulation, abstraction, and modularization using Java classes, methods, and constructors. Students will transition from procedural to modular design thinking** | **Class (AcademicPerformanceGUGradient)**  **Constructor for GUI initialization**  **Methods (addStudentAction())**  **Encapsulation (private fields)** |

**3.4.1 Scenario’s for CO4 implementation.**

**The main class extends JFrame and organizes the structure of the GUI**

**The constructor sets up layout, panels, buttons, and table**

**addStudentAction() handles all computations and validations**

**Variables like nameField, tableModel are private - encapsulation**

**3.4.2 CO4 code screen shot.**

**Class header**

****

**Private fields**

**A computer screen shot of a table

AI-generated content may be incorrect.**

**Constructor where GUI is built**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Method for calculation**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**3.5 CO Attainment**

|  |  |
| --- | --- |
| **CO5 Syllabus** | **CO5 Concepts Included in Project** |
| **Design extensible and reusable Java programs employing inheritance, polymorphism, abstract classes, interfaces, and reflection API to solve domain-oriented problems with clarity and maintainability. Students will model real-world entities and relationships through effective OOP architecture.** | **Inheritance**  **Method Overriding for custom rendering** |

**3.5.1 Scenario’s for CO5 implementation.**

**The class extends JFrame → inheritance**

**Custom GradientPanel class overrides paintComponent() → runtime polymorphism**

**Uses Swing’s component hierarchy for UI extension**

**3.5.2 CO5 code screen shot.**

**Inheritance**

**A blue background with white text

AI-generated content may be incorrect.**

**Overriding paint Component ()**

**A screen shot of a computer code

AI-generated content may be incorrect.**

* 1. **CO Attainment**

|  |  |
| --- | --- |
| **CO6 Syllabus** | **CO6 Concepts Included in Project** |
| **Implement robust, scalable, and generic Java applications integrating exception handling, file I/O, generics, and collections framework, along with functional programming constructs to handle real-world data-driven tasks. Students will demonstrate ability to write production-level, fault-tolerant programs.** | **try–catch blocks**  **Detailed validation**  **Error messages (JOptionPane)** |

* + 1. **Scenario’s for CO6 implementation.**

**Non-numeric or invalid subject count triggers NumberFormatException**

**Invalid marks (negative, >100) show error dialogs**

**Mismatched number of marks is detected**

**Empty fields prevent processing**

**All exceptions handled gracefully without program termination**

* + 1. **CO6 code screen shot.**

**try–catch for subject count**

**A computer screen shot of text

AI-generated content may be incorrect.**

**try–catch for marks parsing**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

**CHAPTER -4 SCREEN SHOTS**

## 4.1 Screen Shots

## Full Code

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer code on a black background

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

**Result**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

# CHAPTER -5 TESTING

## Test Cases and Results

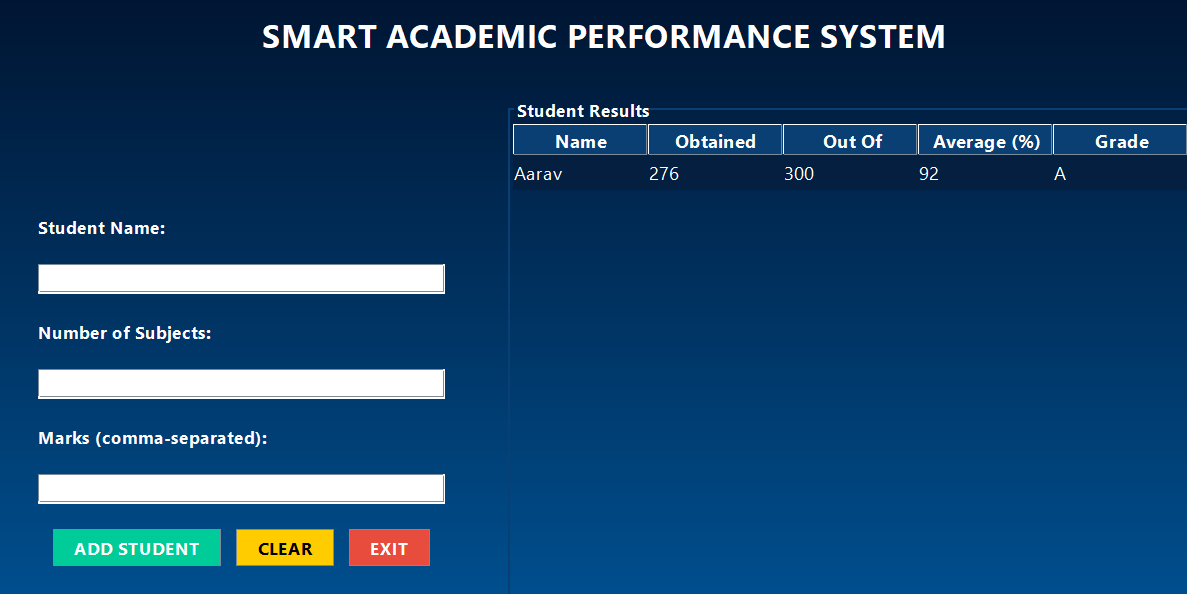
**Test case 1:**

**Input:**

**Enter Name: Aarav**

**Enter No of Subjects: 3**

**Enter marks: 99,87,90**

****

**Test case 2:**

**Input:**

**Enter Name: Ram**

**Enter No of Subjects: 4**

**Enter marks: 96,87,90,84**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Test case 3:**

**Input:**

**1)**

**Enter Name: Varun**

**Enter No of Subjects: 2**

**Enter marks: 90,82**

**2)**

**Enter Name: Girish**

**Enter No of Subjects: 3**

**Enter marks: 90,88,78**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Test case 4:**

**Input:**

**1)**

**Enter Name: Akhil**

**Enter No of Subjects: 2**

**Enter marks: 77,92**

**2)**

**Enter Name: Sharat**

**Enter No of Subjects: 3**

**Enter marks: 99,81,73**

**3)**

**Enter Name: Karun**

**Enter No of Subjects: 4**

**Enter marks: 80,98,91,92**

**A screenshot of a computer

AI-generated content may be incorrect.**

# CHAPTER -6 FUTURE ENHANCEMENTS

## Planned Features

## The current system works well for calculating student performance, but it can be improved with more useful features. In the future, we plan to add:

## Individual fields for each subject instead of comma-separated marks.

## Saving student data into a file or database so records don’t get lost.

## Automatic report card generation in PDF format.

## Graphs and charts for visual performance analysis.

## Login/teacher authentication for secure access.

## Possible Integrations / Optimizations

## Linking the system with databases like MySQL.

## Improving GUI speed to handle more data smoothly.

## Making the interface more responsive with modern UI libraries.

## CHAPTER -7 CONCLUSION

* 1. **Summary of the Project**

**This project was developed to make student result calculation simple and error-free. By entering basic details like name, number of subjects, and marks, the application automatically calculates total, average percentage, and grade using Java Swing.**

**7.2 What Was Achieved**

* **A clean and user-friendly GUI application.**
* **Accurate and fast result calculations.**
* **Helpful data validation to prevent wrong inputs.**
* **Practical use of Java concepts such as conditions, loops, arrays, OOP, and event handling.**

**7.3 Skills Learned**

* **Java programming and Swing GUI building.**
* **Debugging and problem-solving.**
* **Designing structured programs.**
* **Writing documentation and organizing project work.**

# CHAPTER -8 REFERENCES

## Books

Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner’s Approach* (8th ed.). McGraw- Hill Education.

Freeman, E., & Freeman, E. (2020). *Head First HTML and CSS* (2nd ed.). O’Reilly Media. Duckett, J. (2014). *HTML & CSS: Design and Build Websites*.

Duckett, J. (2015). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley. Sommerville, I. (2016). *Software Engineering* (10th ed.).

## Tutorials and Learning Resources

W3Schools. (2025). *HTML, CSS, and JavaScript Tutorials.* Retrieved from https://[www.w3schools.com](http://www.w3schools.com/)

MDN Web Docs. (2025). *Web Technologies Documentation.* Retrieved from https://developer.mozilla.org

FreeCodeCamp. (2025). *Responsive Web Design Certification.* Retrieved from https://[www.freecodecamp.org](http://www.freecodecamp.org/)

GeeksforGeeks. (2025). *Front-end Development and Web Technologies.* Retrieved from https://[www.geeksforgeeks.org](http://www.geeksforgeeks.org/)

Tutorialspoint. (2025). *HTML, CSS, JavaScript, and Web Development Tutorials.* Retrieved from https://[www.tutorialspoint.com](http://www.tutorialspoint.com/)

## APIs and Tools Used

Google Maps Platform. (2025). *Maps JavaScript API Documentation.* Retrieved from https://developers.google.com/maps

Unsplash API. (2025). *Free High-Resolution Image Access.* Retrieved from https://unsplash.com/developers

OpenWeatherMap. (2025). *Weather API for Travel Assistance.* Retrieved from https://openweathermap.org/api

RapidAPI Hub. (2025). *Travel and Tourism APIs.* Retrieved from https://rapidapi.com/hub

Firebase by Google. (2025). *Authentication and Database Services.* Retrieved from https://firebase.google.com

## Documentation and Research References

World Tourism Organization (UNWTO). (2024). *Tourism Data Dashboard.* Retrieved from https://[www.unwto.org](http://www.unwto.org/)

Booking.com. (2025). *Accommodation and Destination Insights.* Retrieved from https://[www.booking.com](http://www.booking.com/)

TripAdvisor. (2025). *Tourism and Destination Review Data.* Retrieved from https://[www.tripadvisor.com](http://www.tripadvisor.com/)

Travel + Leisure. (2025). *Top Global Destinations and Travel Trends.* Retrieved from https://[www.travelandleisure.com](http://www.travelandleisure.com/)

Tourism Review. (2025). *Digital Transformation in the Tourism Industry.* Retrieved from https://www.tourism- review.com

# CHAPTER -9 APPENDICES

**Installation / Setup**

1. **Install Java JDK.**
2. **Use an IDE like IntelliJ/Eclipse.**
3. **Create a project and paste the code into the src folder.**
4. **Run the program — the GUI will open.**

**User Guide**

1. **Enter Student Name.**
2. **Enter Number of Subjects.**
3. **Enter Marks.**
4. **Click Add Student to calculate results.**
5. **View total, average, and grade in the table.**
6. **Clear inputs or exit anytime.**

**Geo-Tag Photos**

A group of people standing in front of a screen

AI-generated content may be incorrect.

**A group of people standing in front of a screen

AI-generated content may be incorrect.**

**Review Forms**

**Geo Tag photos with guide**

****