

Result Management System

C Programming Mini Project

Presented by: Pankaj Kumar

Course: C Programming

College: Rungta International Skills University

Year: 2025-26

Guide By: Naina Devi



Introduction



Project Objectives



Applications



Future Scope



Tools & Technology



System Working



Advantages



Conclusion

This presentation will cover the development and functionality of our C Programming mini-project.

Introduction to the System

- A robust console-based Result Management System
- Developed entirely using the C Programming language
- Designed to automate the processing of student results efficiently
- Significantly reduces manual effort and minimises human errors



Project Objectives

Academic Records

To effectively manage and organise student academic records.

Mark Calculation

To accurately calculate total marks and overall percentage for each student.

Rank Generation

To generate a comprehensive rank list based on student performance.

Identify Toppers

To identify and highlight subject-wise toppers within the system.

Display Marksheets

To facilitate the clear and concise display of individual student marksheets.



Tools & Technology Utilised



Programming Language

C Programming: The foundational language for this project, chosen for its efficiency and system-level capabilities.



Compiler Environment

GCC / Turbo C / Code::Blocks: Various compilers supported for development and execution, ensuring broad compatibility.



Essential Header Files

stdio.h, string.h: Standard libraries crucial for input/output operations and string manipulation.



Operating Platform

Windows / Linux: The system is designed to be functional across popular operating systems.

System Flow and Working Mechanism



User Menu Selection

1

The user initiates interaction by selecting an option from the main menu.



Data Input

2

Student and subject details, including marks, are entered into the system.



Marks Validation

3

Input marks are validated to ensure they fall within the 0-100 range.



Automatic Calculation

4

The system automatically calculates total marks and overall percentage.



Output Generation

5

Processed results, such as marksheets or rank lists, are generated and displayed.

Key Features of the System

- **Menu-Driven Interface**

An intuitive and easy-to-navigate interface for all functionalities.

- **Multiple Student Handling**

Capability to manage data for numerous students concurrently.

- **Rank List Generation**

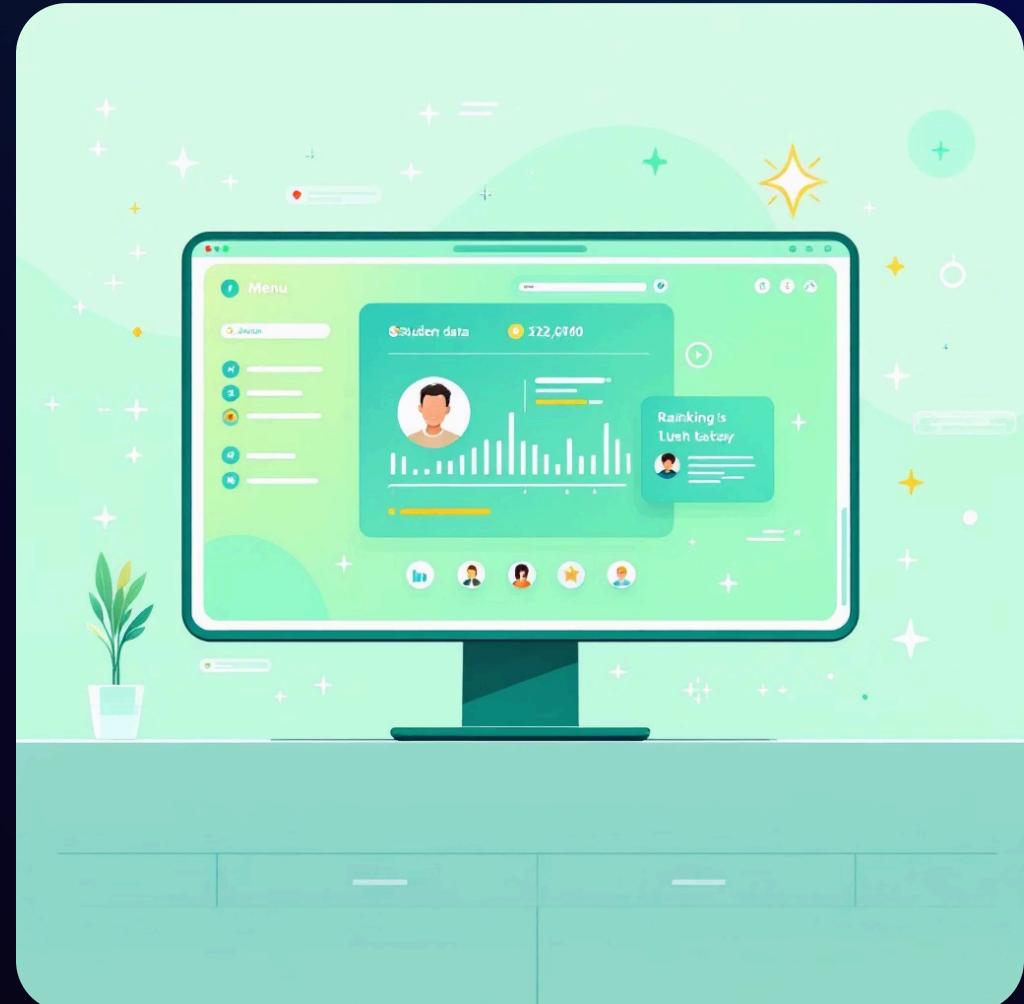
Automated generation of a sorted rank list based on performance.

- **Subject Topper Identification**

Pinpointing students who excel in specific subjects.

- **Pass/Fail Evaluation**

Automatic assessment and declaration of pass or fail status for students.



Project Working: Interface Snapshots

```
3
4 int n, s;
5 int roll[20];
6 char name[20][20];
7 int marks[20][10];
8 float total[20];
9 float percent[20];
10
11
12 void inputData();
13 void displayMarksheet();
14 void rankList();
15 void subjectTopper();
16
17 int main()
18 {
19     int choice;
20
21     do
22     {
23         printf("\n\n*** RESULT MANAGEMENT SYSTEM ***");
24         printf("\n1. Enter Student Data");
25         printf("\n2. Display Marksheets");
26         printf("\n3. Rank List");
27         printf("\n4. Subject Topper");
28         printf("\n5. Exit");
29         printf("\nEnter your choice: ");
30         scanf("%d", &choice);
31     }
```

```
32
33     switch (choice)
34     {
35         case 1:
36             inputData();
37             break;
38         case 2:
39             displayMarksheet();
40             break;
41         case 3:
42             rankList();
43             break;
44         case 4:
45             subjectTopper();
46             break;
47         case 5:
48             printf("\nThank you for using the Result Management System!");
49             break;
50         default:
51             printf("\nInvalid choice! Please select a valid option.");
52     }
53
54 } while (choice != 5);
55
56 return 0;
57
58
59 void inputData()
60 {
```

```
==== RESULT MANAGEMENT SYSTEM ====
1. Enter Student Data
2. Display Marksheets
3. Rank List
4. Subject Topper
5. Exit
Enter your choice: 1

Enter number of students: 2
Enter number of subjects: 3

Student 1 Roll No: 1
Student Name: pankaj
Marks of subject 1 (0 to 100): 76
Marks of subject 2 (0 to 100): 87
Marks of subject 3 (0 to 100): 89

Student 2 Roll No: 2
Student Name: arjun
Marks of subject 1 (0 to 100): 98
Marks of subject 2 (0 to 100): 86
Marks of subject 3 (0 to 100): 67

Data entered successfully!

==== RESULT MANAGEMENT SYSTEM ====
1. Enter Student Data
2. Display Marksheets
3. Rank List
4. Subject Topper
5. Exit
Enter your choice: 2

-----
Roll No: 1
Name: pankaj
Subject 1 Marks: 76
```

These screenshots illustrate the system's console interface during key operations.

Diverse Applications

Educational Institutions

Ideal for schools and colleges to manage examination results.

Academic Projects

Serves as a foundation for more complex academic reporting systems.

Performance Analysis

Supports basic analysis of student performance trends.

Learning Tool

An excellent practical learning tool for C Programming beginners.



Advantages & Future Enhancements

Key Advantages

→ User-Friendly

Simple and intuitive interface for seamless operation.

→ High Accuracy

Ensures precise and error-free result calculations.

→ Time-Saving

Automates processes, drastically reducing manual time spent.

→ Efficient Data Management

Organises and handles student data effectively.

→ Code Comprehension

Features easily understandable and maintainable code.

Future Scope

→ Persistent Storage

Implement file handling for permanent data storage.

→ Graphical Interface

Develop a user-friendly GUI-based interface.

→ Database Integration

Integrate with databases for scalable data management.

→ Online Accessibility

Expand into an online result management system.

→ Advanced Grading

Introduce a comprehensive grade system implementation.

THANK YOU!

Pankaj kumar | Guided By Naina Devi