# Result Management System

## C Programming Mini Project

Presented by: Arjun Kumar Mahato

Course: C Programming

College : Rungta International Skills University

Guide By: Naina devi

# Presentation Agenda

This presentation outlines the development and functionality of our C Programming Mini Project.

01

Introduction

02

Tools & Technology

03

Project Objectives

04

System Working

05

Applications

06

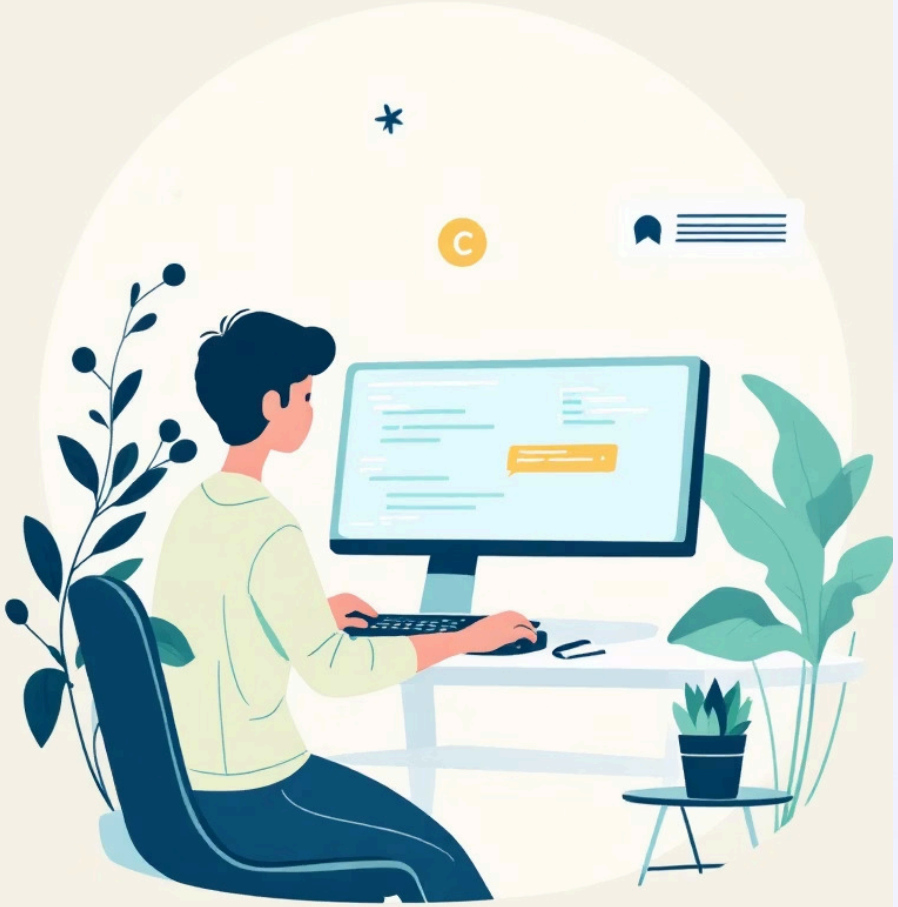
Advantages

07

Future Scope

08

Conclusion



# Introduction to the System

The Result Management System is a console-based application meticulously developed using C Programming. Its primary purpose is to automate the processing of student results, significantly reducing manual effort and minimising potential errors in academic record-keeping.

- A robust console-based application
- Developed entirely in C Programming for efficiency
- Automates the complex process of student result management
- Drastically reduces the scope for human error

# Core Project Objectives

Our system is designed with clear objectives to enhance academic administration and student performance analysis.

1

### Manage Academic Records

Efficiently organise and maintain comprehensive student academic records.

2

### Calculate Marks & Percentage

Accurately compute total marks and overall percentage for each student.

3

### Generate Rank List

Automatically create a rank-ordered list of students based on their performance.

4

### Identify Toppers

Pinpoint subject-wise toppers, recognising academic excellence.

5

### Display Marksheets

Generate and display detailed student marksheets for easy access.

# Tools & Technology Utilised

The project's development relied on standard and effective tools within the C programming ecosystem.

### Programming Language: C

The foundational language for system development, chosen for its performance and control.

### Compilers

Utilised GCC, Turbo C, or Code::Blocks for efficient code compilation and execution.

### Header Files

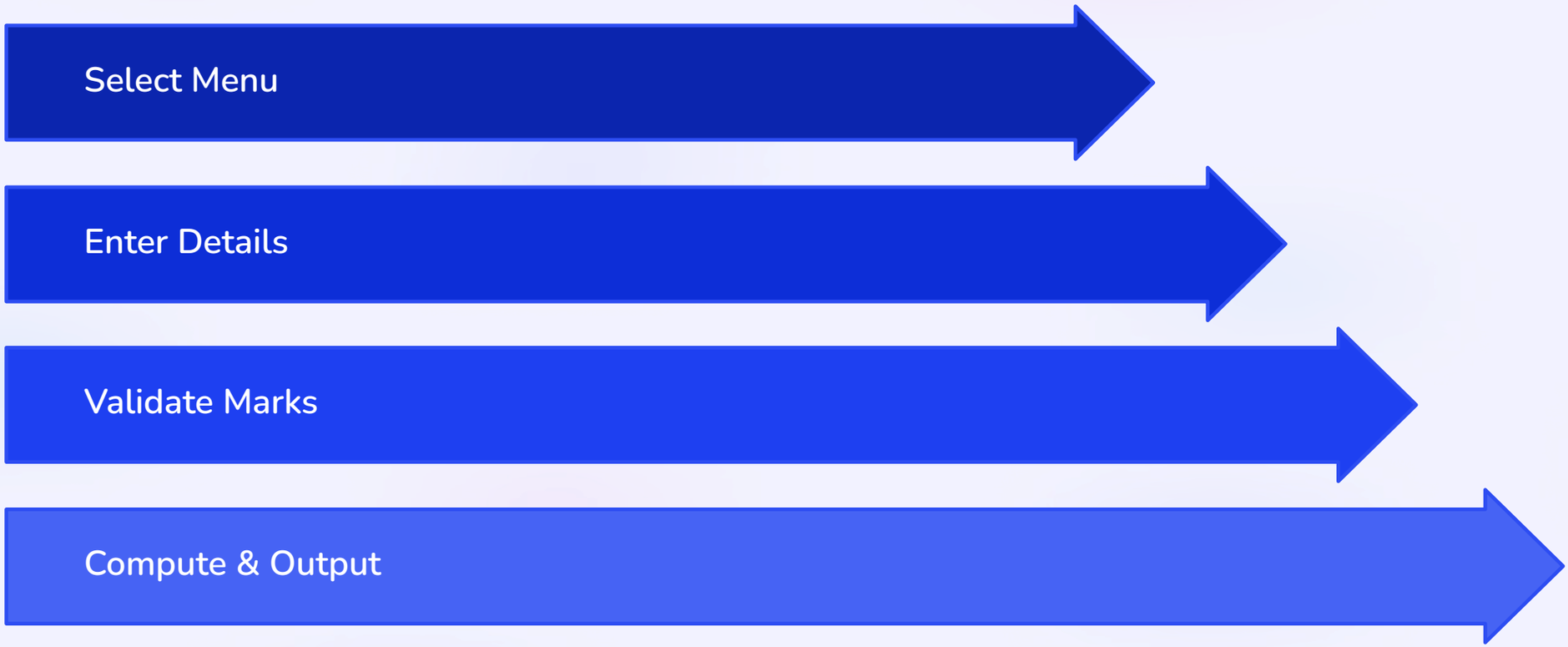
Essential libraries such as `stdio.h` for input/output and `string.h` for string manipulation.

### Platform

Developed and tested across both Windows and Linux operating systems.

# System Flow & Working Mechanism

The system operates through a streamlined process, guiding the user from data input to result generation.




- User selects desired option from a menu
- Input of student and subject-specific details
- Validation of marks (ensuring values between 0 and 100)
- Automated calculation of total marks and overall percentage
- Generation of comprehensive result outputs

# Key Features & Functionalities

The Result Management System offers a suite of features designed for ease of use and comprehensive result processing.

### Menu-Driven Interface

Intuitive navigation for all system operations.

### Multiple Student Handling

Capability to process and manage data for numerous students.

### Rank List Generation

Automated creation of performance-based student rankings.

### Subject Topper Identification

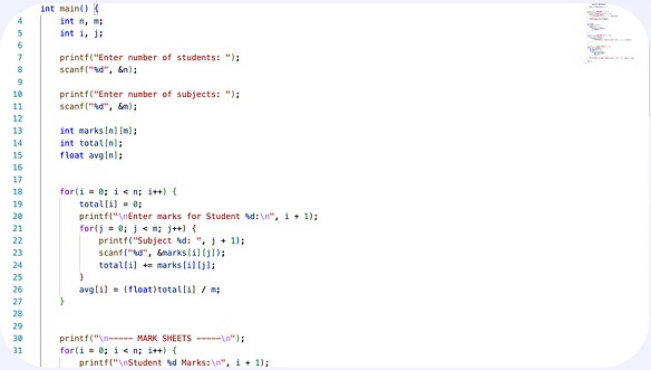
Highlights top performers in individual subjects.

### Pass/Fail Evaluation

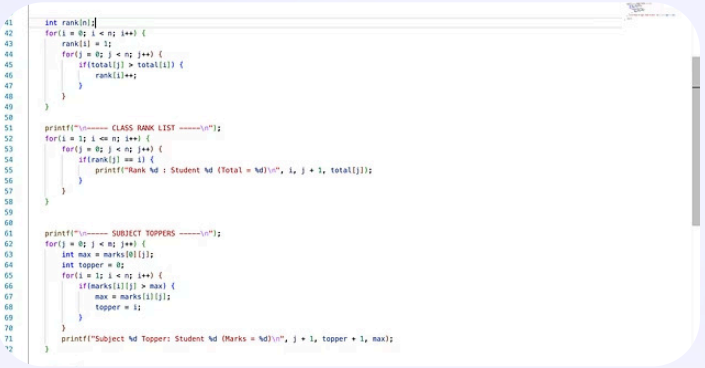
Automatic determination of student pass/fail status.

# Project Working: Interface Snapshots

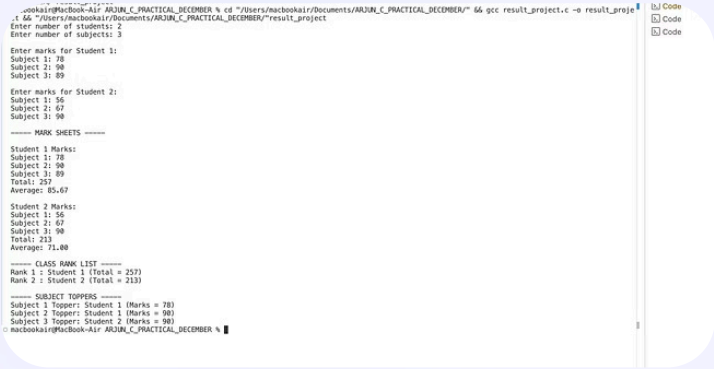
Visual examples showcasing the system's interactive elements and output displays.



```c
int main() {
    int n, m;
    int i, j;

    printf("Enter number of students: ");
    scanf("%d", &n);

    printf("Enter number of subjects: ");
    scanf("%d", &m);

    int marks[n][m];
    int total[n];
    float avg[n];


    for(i = 0; i < n; i++) {
        total[i] = 0;
        printf("\nEnter marks for Student %d:\n", i + 1);
        for(j = 0; j < m; j++) {
            printf("Subject %d: ", j + 1);
            scanf("%d", &marks[i][j]);
            total[i] += marks[i][j];
        }
        avg[i] = (float)total[i] / m;
    }


    printf("\n===== MARK SHEETS =====\n");
    for(i = 0; i < n; i++) {
        printf("\nStudent %d Marks:\n", i + 1);
```



```c
int rank[n];
for(i = 0; i < n; i++) {
    rank[i] = 1;
    for(j = 0; j < n; j++) {
        if(total[j] > total[i]) {
            rank[i]++;
        }
    }
}

printf("\n===== CLASS RANK LIST =====\n");
for(i = 1; i <= n; i++) {
    for(j = 0; j < n; j++) {
        if(rank[j] == i) {
            printf("Rank %d : Student %d (Total = %d)\n", i, j + 1, total[j]);
        }
    }
}


printf("\n===== SUBJECT TOPPERS =====\n");
for(j = 0; j < m; j++) {
    int max = marks[0][j];
    int topper = 0;
    for(i = 1; i < n; i++) {
        if(marks[i][j] > max) {
            max = marks[i][j];
            topper = i;
        }
    }
    printf("Subject %d Topper: Student %d (Marks = %d)\n", j + 1, topper + 1, max);
}
```



```
...ookair@MacBook-Air ARJUN_C_PRACTICAL_DECEMBER % cd "/Users/macbookair/Documents/ARJUN_C_PRACTICAL_DECEMBER/" && gcc result_project.c -o result_proje
ct && "/Users/macbookair/Documents/ARJUN_C_PRACTICAL_DECEMBER/"result_project
Enter number of students: 2
Enter number of subjects: 3

Enter marks for Student 1:
Subject 1: 78
Subject 2: 90
Subject 3: 89

Enter marks for Student 2:
Subject 1: 56
Subject 2: 67
Subject 3: 90

===== MARK SHEETS =====

Student 1 Marks:
Subject 1: 78
Subject 2: 90
Subject 3: 89
Total: 257
Average: 85.67

Student 2 Marks:
Subject 1: 56
Subject 2: 67
Subject 3: 90
Total: 213
Average: 71.00

===== CLASS RANK LIST =====
Rank 1 : Student 1 (Total = 257)
Rank 2 : Student 2 (Total = 213)

===== SUBJECT TOPPERS =====
Subject 1 Topper: Student 1 (Marks = 78)
Subject 2 Topper: Student 1 (Marks = 90)
Subject 3 Topper: Student 2 (Marks = 90)
macbookair@MacBook-Air ARJUN_C_PRACTICAL_DECEMBER %
```

These images provide a glimpse into the system's functional user interface, from initial menu selection to the final presentation of results.

# Diverse Applications of the System

The Result Management System provides versatile utility across various educational and developmental contexts.

### Schools and Colleges

Streamlining result processing and academic record keeping in educational institutions.

### Academic Projects

Serving as a foundational example for further development in computer science curricula.

### Student Performance Analysis

Tools for educators to quickly assess and understand student academic trends.

### Learning Tool for C Beginners

An excellent practical project for students learning the C programming language.

# Future Scope & Enhancements

The current system lays a strong foundation, with numerous possibilities for future expansion and advanced features.

1

### File Handling

Implementing permanent data storage to retain results across sessions.

2

### GUI-Based Interface

Transitioning to a graphical user interface for enhanced user experience.

3

### Database Integration

Connecting with robust databases for scalable and secure data management.

4

### Online System

Developing a web-based platform for remote result access and management.

5

### Grade System

Incorporating a comprehensive grading system alongside numerical scores.

# THANK YOU!

Arjun Kumar Mahato | Guide By Naina Devi