

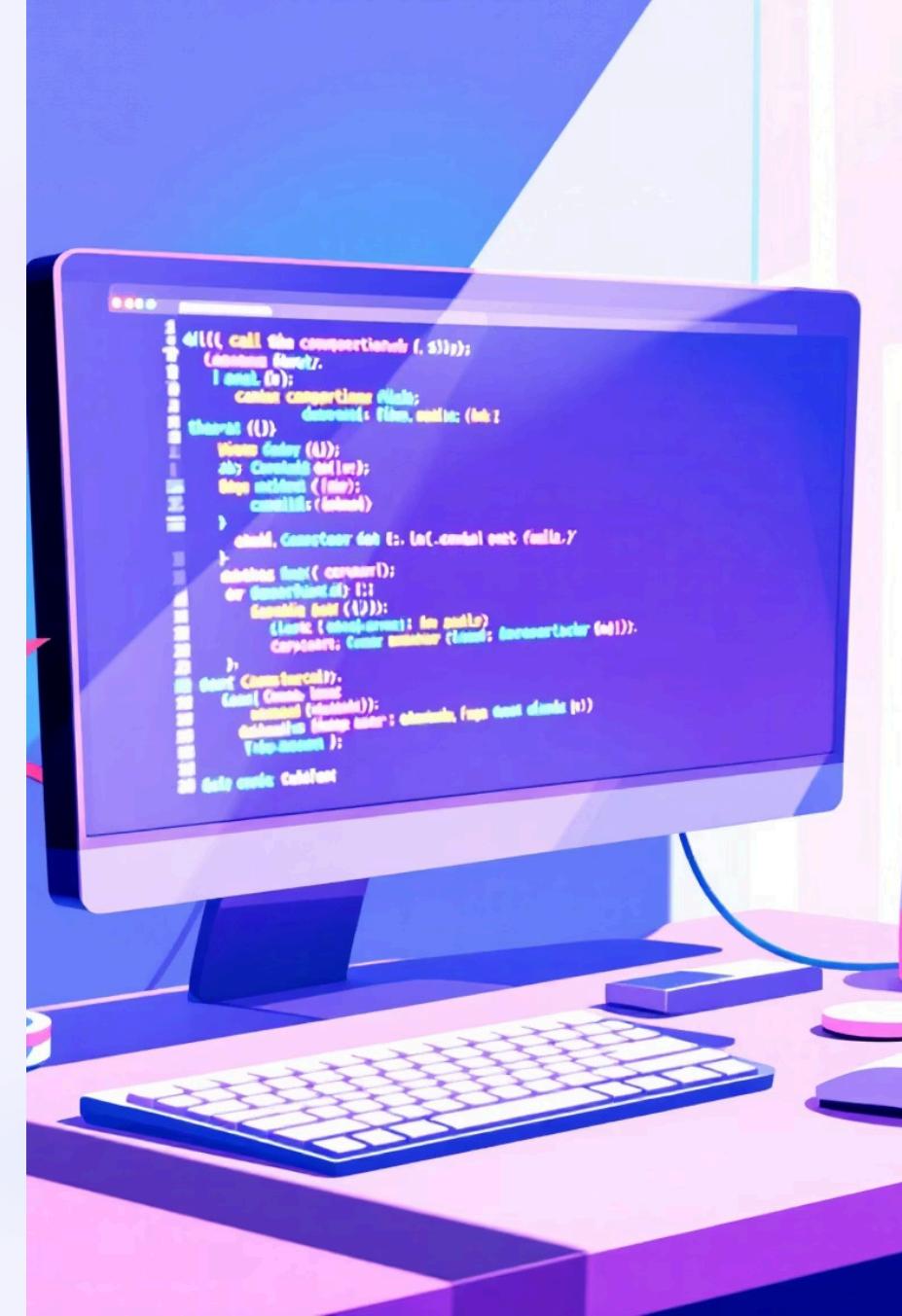
Result Management System

C PROGRAMMING MINI PROJECT

A console-based application automating student result processing,
reducing manual effort and minimizing errors in academic record keeping.

Presented by: Aditya Kumar | **Guide:** Naina Devi

Rungta International Skills University



Presentation Agenda

01

Introduction

System overview and purpose

03

Tools & Technology

Development environment

05

Applications

Real-world use cases

07

Future Scope

Enhancement opportunities

02

Project Objectives

Core goals and targets

04

System Working

Flow and mechanisms

06

Advantages

Key benefits

08

Conclusion

Summary and takeaways



Introduction to the System

The Result Management System is a console-based application developed using C Programming, designed to automate student result processing and enhance academic administration.

Console-Based

Robust application developed entirely in C Programming for maximum performance and control.

Automated Processing

Automates complex student result management, streamlining academic workflows.

Error Reduction

Drastically reduces the scope for human error in recordkeeping and calculations.

Core Project Objectives

Our system enhances academic administration through five key objectives that streamline result management and student performance analysis.



Manage Academic Records

Efficiently organize and maintain comprehensive student academic records.



Calculate Marks & Percentage

Accurately compute total marks and overall percentage for each student.



Generate Rank List

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



Identify Toppers

Pinpoint subject-wise toppers, recognizing academic excellence across disciplines.



Display Marksheets

Generate and display detailed student mark sheets with comprehensive information.

Tools & Technology Utilized

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

Programming Language: C

Foundational language chosen for its performance, control, and efficiency in system development.

Compilers

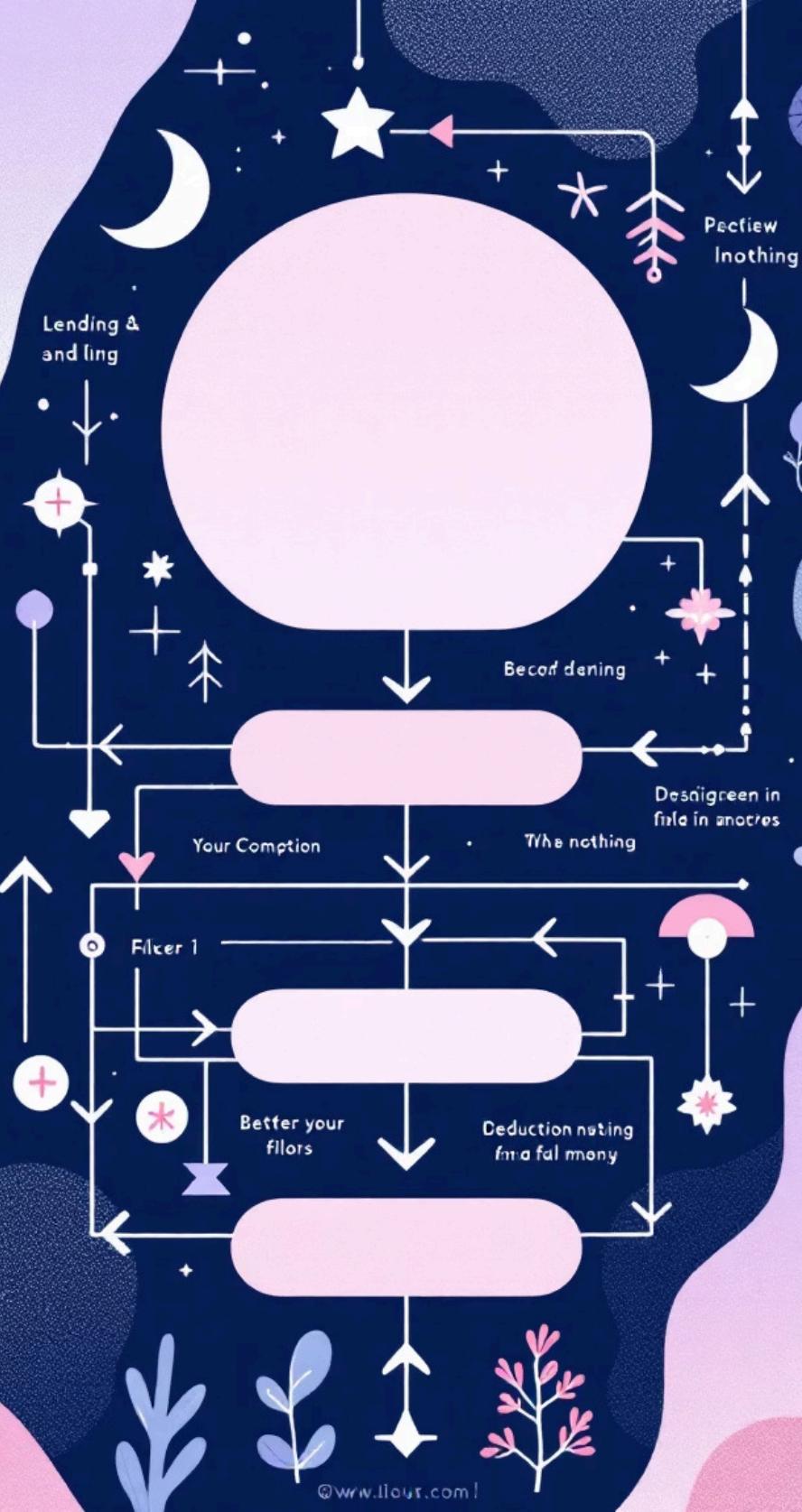
Utilized GCC, Turbo C, or Code::Blocks for efficient code compilation and execution.

Header Files

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

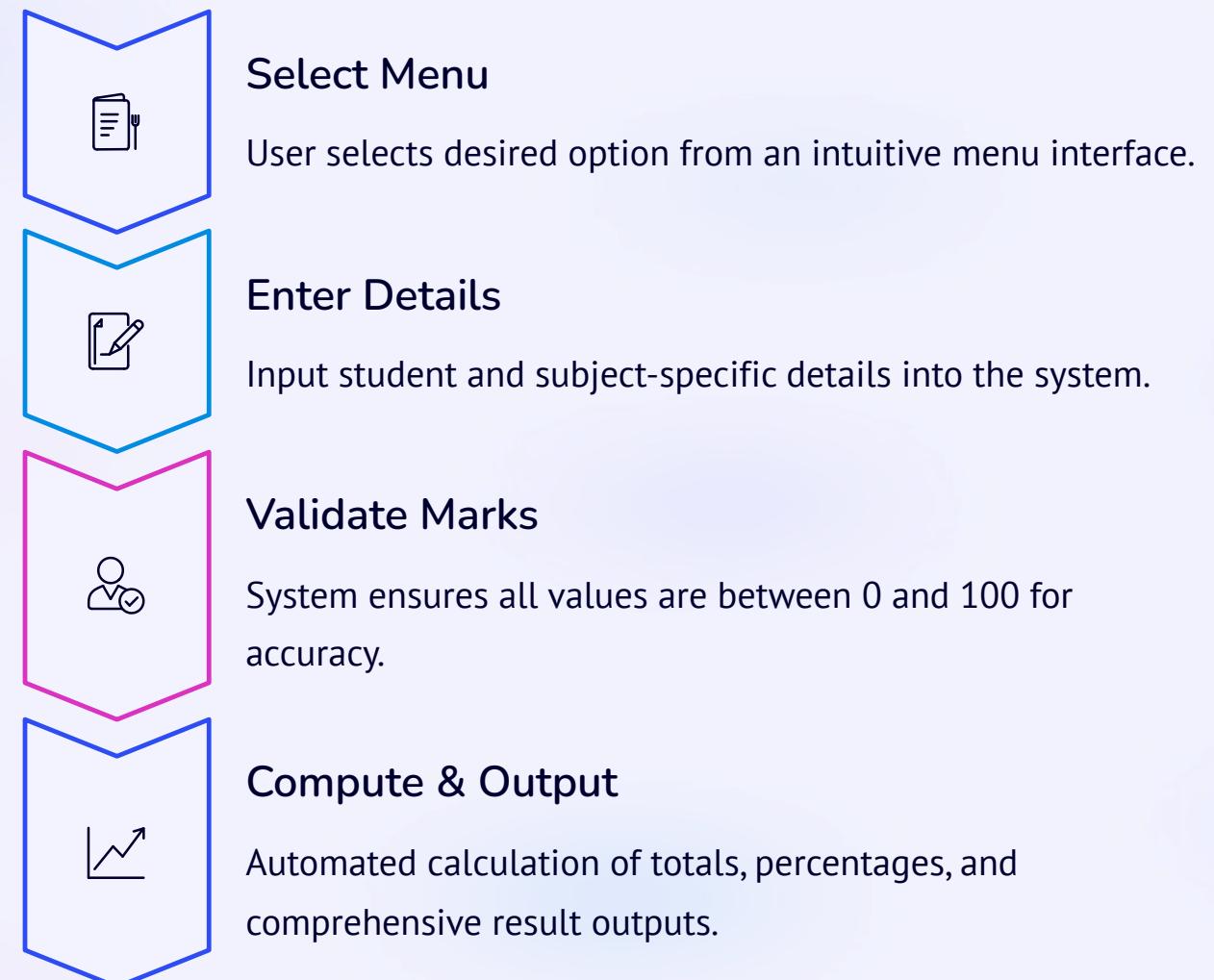
Platform

Developed and tested across both Windows and Linux operating systems for maximum compatibility.



System Flow & Working Mechanism

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



Key Features & Functionalities

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

—0— Menu-Driven Interface

Intuitive navigation system for all operations, making the system accessible to users of all skill levels.



Multiple Student Handling

Capability to process and manage data for numerous students simultaneously with efficiency.



Rank List Generation

Automated creation of performance-based student rankings for quick assessment.



Subject Topper Identification

Highlights top performers in individual subjects, recognizing academic excellence.



Pass/Fail Evaluation

Automatic determination of student pass/fail status based on predefined criteria.

Project Working: Code Implementation

```
#include <stdio.h>

#define MAX_STUDENTS 50 #define MAX_SUBJECTS 5

int main() { int n, i, j; int marks[MAX_STUDENTS][MAX_SUBJECTS]; int total[MAX_STUDENTS]; float percentage[MAX_STUDENTS];
int roll[MAX_STUDENTS]; int topper[MAX_SUBJECTS]; int temp;

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

for (i = 0; i < n; i++) {
    roll[i] = i + 1;
    total[i] = 0;

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

for (j = 0; j < MAX_SUBJECTS; j++) {
    topper[j] = 0;
    for (i = 1; i < n; i++) {
        if (marks[i][j] > marks[topper[j]][j]) {
            topper[j] = i;
        }
    }
}
for (i = 0; i < n - 1; i++) {
    for (j = i + 1; j < n; j++) {
        if (total[i] < total[j]) {
            temp = total[i];
            total[i] = total[j];
            total[j] = temp;

            temp = roll[i];
            roll[i] = roll[j];
            roll[j] = temp;
        }
    }
}
printf("\n==MARKSHEETS ==\n");
for (i = 0; i < n; i++) {
    printf("\nStudent %d\n", i + 1);
    for (j = 0; j < MAX_SUBJECTS; j++) {
        printf("Subject %d: %d\n", j + 1, marks[i][j]);
    }
    printf("Total: %d\n", total[i]);
    printf("Percentage: %.2f\n", percentage[i]);
}

printf("\n==CLASS RANK LIST ==\n");
printf("Rank\tRoll No\tTotal Marks\n");
for (i = 0; i < n; i++) {
    printf("%d\t%d\t%d\n", i + 1, roll[i], total[i]);
}

printf("\n== SUBJECT TOPPERS ==\n");
for (j = 0; j < MAX_SUBJECTS; j++) {
    printf("Subject %d Topper: Student %d (Marks: %d)\n",
           j + 1, topper[j] + 1, marks[topper[j]][j]);
}

return 0;
}
```

Applications & Future Enhancements

Diverse Applications

Schools and Colleges

Streamlining result processing and academic recordkeeping for educational institutions.

Academic Projects

Foundational example for computer science curricula and student learning.

Performance Analysis

Tools for educators to assess student academic trends and patterns.

Learning Tool

Excellent practical project for students learning C programming fundamentals.

Future Scope

File Handling

Implementing permanent data storage across sessions.

GUI Interface

Transitioning to graphical user interface for enhanced experience.

Database Integration

Connecting with robust databases for scalable data management.

Online System

Developing web-based platform for remote access.

Grade System

Incorporating comprehensive grading system with letter grades.



Thank You!

Presented By

Aditya Kumar

Guided By

Naina Devi

Institution

Rungta International Skills
University

This Result Management System demonstrates the power of C programming in solving real-world academic challenges through automation, accuracy, and efficiency.