

**STUDENT RESULT MANAGEMENT SYSTEM (MINI PROJECT)
REPORT**

SUBMITTED TO:

K.R. MANGALAM UNIVERSITY



K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(CORE)**

SUBMITTED BY: Y1-2024-25-G-53

Anurag Upadhyay (2401010193)

Yuvraj Singh Rathore (2401010225)

Diya Goel (2401010179)

Akshita Singh (2401010208)

SCHOOL OF ENGINEERING TECHNOLOGY

K.R. MANGALAM UNIVERSITY

SOHNA, HARYANA 122103, INDIA

INTRODUCTION:-

In today's digital world, everything is becoming smarter and more efficient—from smartphones to smart classrooms. But if you take a closer look at how many schools and colleges still manage their academic records, you'll find that a lot of institutions are still using outdated, manual methods. These traditional systems—registers, spreadsheets, handwritten marksheets—may have worked in the past, but they're simply not built to handle the fast-paced, high-volume demands of modern education. That's where a project like the Student Result Management System (SRMS) comes in.

This project was developed to address a real-world need: helping schools and colleges organize and manage student information, course details, and exam results in a more efficient way. Instead of relying on paperwork or scattered Excel files, SRMS puts everything into one place—a simple desktop application built with Python's Tkinter library, using SQLite for database management. The idea is to make academic administration smoother, faster, and less stressful for everyone involved.

Let's face it: managing hundreds of student records isn't easy. Every semester, institutions have to deal with course registrations, new admissions, exam result calculations, and progress reports. If this is done manually, even a small mistake—like writing the wrong percentage or duplicating a record—can cause confusion, complaints, or worse. SRMS is designed to minimize these errors by offering a clean, well-organized system where data is automatically validated and stored securely.

What makes SRMS truly helpful is how user-friendly it is. You don't need to be a programmer or IT expert to use it. The system is broken into clear, easy-to-understand modules. For example, there's one section for managing courses, another for student admissions, one for entering and storing results, and another for viewing detailed reports. Each of these modules is accessed through a central dashboard, making navigation simple and logical.

OBJECTIVES:-

Every project begins with a goal, a reason why it's worth building in the first place. The Student Result Management System (SRMS) is no different. At its core, this project was created to solve very real and practical problems faced by educational institutions—especially when it comes to handling student data, managing courses, and processing exam results.

Let's think for a moment about what it's like to manage these tasks manually. Imagine a school administrator or teacher trying to maintain paper files for every student—keeping track of their names, contact details, the courses they've enrolled in, and the marks they've scored in each exam. As the number of students grows, these tasks become harder to manage, and even small mistakes can lead to big consequences. A missing mark, a wrongly calculated percentage, or a duplicate record could create confusion for students, stress for parents, and unnecessary work for teachers and staff.

With this challenge in mind, one of the main objectives of SRMS is to simplify and organize the management of student academic records. Instead of dealing with piles of papers or scattered spreadsheets, administrators can use a clean, easy-to-use application to do everything in one place. Adding new students, enrolling them in courses, entering their results, and viewing reports can all be done quickly, safely, and without hassle.

Another big goal of this project is to make the system user-friendly, especially for people who aren't tech-savvy. Not every school or college has a trained IT department, so SRMS is built in such a way that anyone—whether they're a teacher, a clerk, or a school principal—can learn to use it within minutes. The buttons are clearly labeled, the windows are cleanly laid out, and the actions (like save, update, delete, search) are exactly what you'd expect. There's no complicated terminology or confusing menus—just straightforward features that work.

The project also aims to reduce human errors, which are extremely common when dealing with academic data manually. SRMS includes built-in validation features, meaning it will alert the user if they forget to enter a student's name or try to add a result for someone who hasn't been registered yet. These little safeguards make a huge difference, especially when you're working with dozens or hundreds of records.

TECHNOLOGY USED:-

<u>LAYER</u>	<u>TECHNOLOGY</u>
FRONTEND	Python's Tkinter library for GUI
Image Handling	Pillow (PIL) for image rendering
Database	SQLite (Lightweight, serverless DB)
Backend Logic	Python (OOP structure across modules)
Web Expansion	Flask with Flask-SocketIO for polling/voting
Deployment Tools	Python IDLE, CLI; PyInstaller

SYSTEM REQUIREMENTS:-

Hardware Requirements:

- Processor: Dual Core or better
- RAM: Minimum 2 GB
- Hard Disk: At least 500 MB free space
- Display: 1024x768 resolution recommended

Software Requirements:

- OS: Windows/Linux/MacOS
- Python 3.6 or higher
- Required Libraries: pillow, flask, flask-socketio

SYSTEM COMPONENTS:-

To understand how the Student Result Management System works, it helps to look at it in two main parts: the frontend and the backend. Think of it like a restaurant—what customers see and interact with is the frontend (the menu, tables, and waiters), while the backend is the kitchen, where all the real work happens behind the scenes.

Frontend (What Users See & Use)

The frontend is the part of the system that users directly interact with. In SRMS, the frontend is built using Tkinter, which is a tool in Python for creating windows, buttons, forms, and tables.

Here's what the frontend includes:

Dashboard Window:

- The central control panel where users can go to different parts of the app—like courses, students, or results.

Course Module:

- A form to add or update course names, durations, and fees.

Student Module:

- Lets users enter student details like name, contact info, and enrolled course.

Result Module:

- A form to input marks and automatically calculate percentages.

Report Viewer:

- A screen to search for a student by roll number and instantly view their result.

The goal of the frontend is to make the system easy to use—even for someone with no technical background.

Backend (How It All Works Internally)

The backend is where all the data is stored and managed. In SRMS, this is handled using SQLite, a simple but powerful local database. It's like the brain of the system—it remembers everything, from student records to result data.

The Backend Includes:

Database Tables:

- There are separate tables for students, courses, and results.

SQL Queries:

- These are used to add, update, delete, and search records.

Python Logic:

- Code that connects the frontend forms to the database. For example, when a user clicks "Save", the backend makes sure the data goes into the right place.

Error Handling:

- If something goes wrong—like entering invalid data—the backend catches it and shows a helpful message.

The backend makes sure the system runs smoothly, data stays safe, and every action taken in the frontend is properly processed.

FEATURES IMPLEMENTED:-

- **Modular Design with Separate Functional Files**

The system is organized into multiple Python files, each handling a specific task—like managing students, courses, or results. This makes the code easier to understand, maintain, and scale in the future.

- **Scrollable Tables with Treeview**

Data is displayed in clean, scrollable tables using the Treeview widget from Tkinter, allowing users to view and interact with many records without clutter or confusion.

- **Data Validation with MessageBox Alerts**

The system checks for missing or incorrect inputs and shows helpful error messages using popup alerts. This ensures data is accurate and reduces mistakes.

- **Dynamic Updates and Real-Time Data Display**

Any time data is added, updated, or deleted, the interface instantly reflects the changes without needing to restart the application.

- **Flask WebSocket Polling Prototype**

An additional web feature was added using Flask and WebSocket technology, allowing real-time voting or feedback functionality—demonstrating how the system can expand online.

FLOW OF THE APPLICATION:-

1. Launch Application via dashboard.py.

2. Navigate to:

- > Course Management: Create/modify/delete/search course records.
- > Student Management: Register or update student profiles.
- > Result Entry: Search a student, enter obtained marks and full marks

3. All modules interact with rms.db for persistent storage.

DEPOLYMENTS:-

Desktop Version:

- ❖ **Run via Python:** `python dashboard.py`

Web Version:

- ❖ **Run Flask server:** `python from flask import Flask, render_template.py`

Executable Packaging:

- ❖ **Use PyInstaller:** `pyinstaller -- onefile dashboard.py`

WORKING:-

Add Course: Admin inputs name, duration, and charges.

Add Student: Admin fills student details and selects course.

Add Result: Admin selects student, enters marks, and the system calculates the percentage.

View/Delete Results: Admin can search results by roll number and remove them if needed.

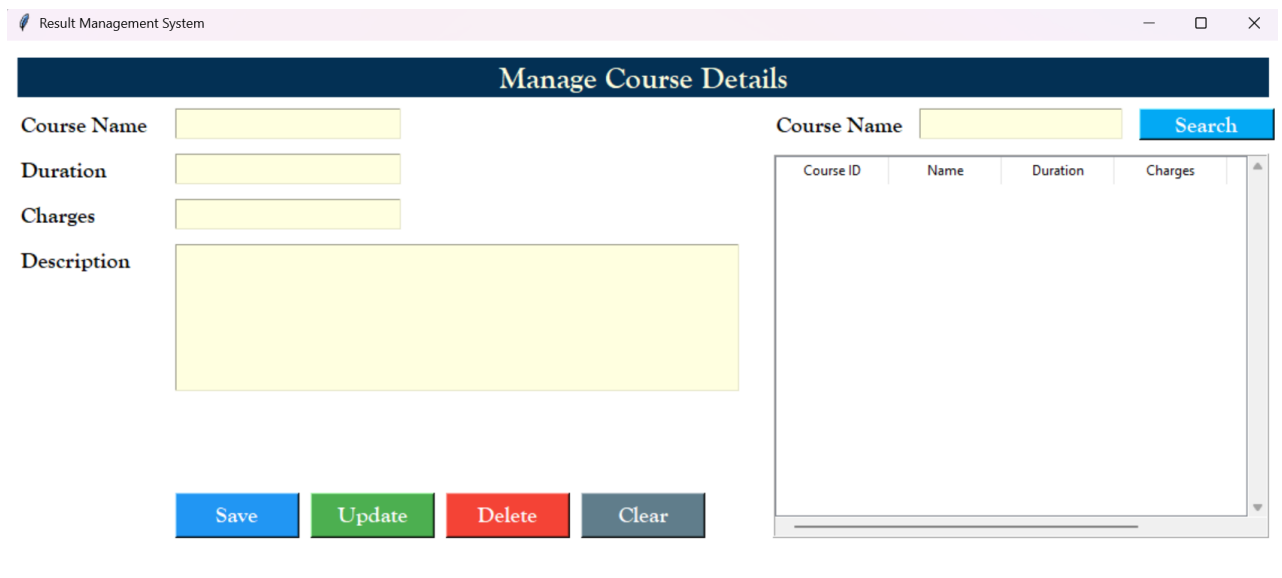
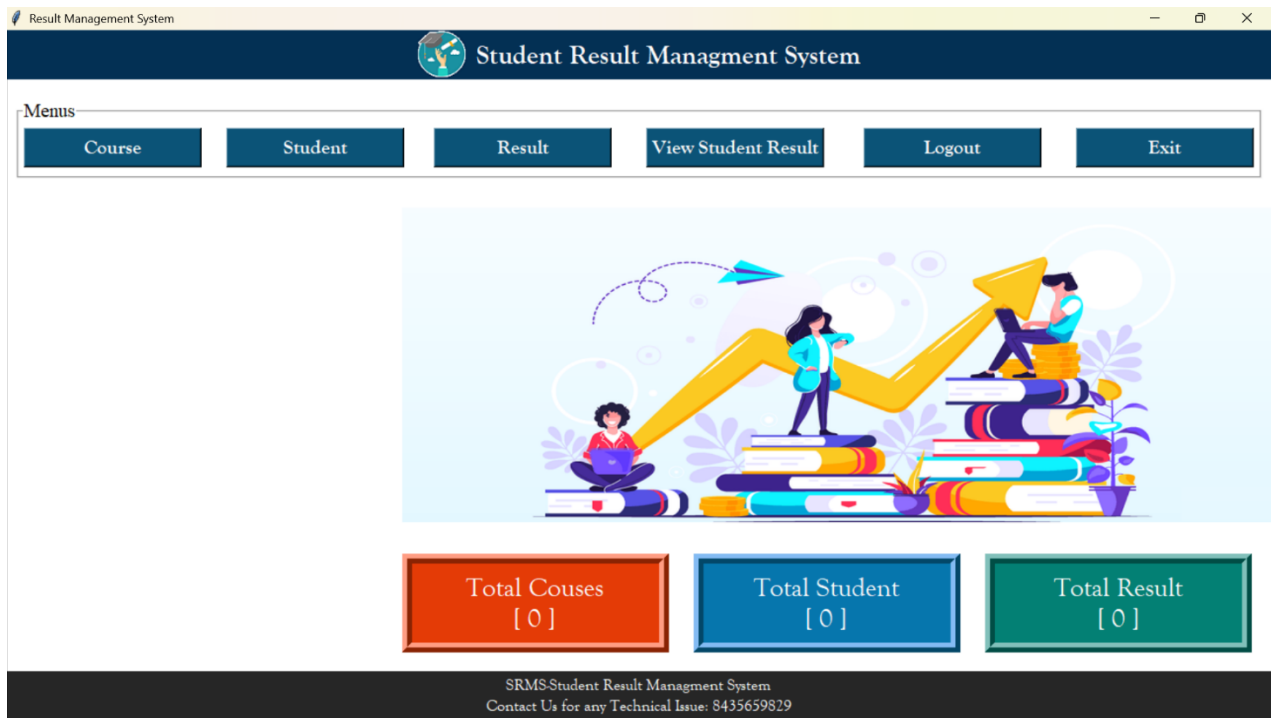
LIMITATIONS:-

- Single-user access only
- No authentication/authorization mechanism
- Local database without cloud sync
- No export functionality
- Web functionality is minimal

FUTURE IMPROVEMENTS:-

- ✓ Add user login and roles
- ✓ Move to cloud-based backend
- ✓ Full web version using Flask/Django
- ✓ Export to PDF/Excel
- ✓ Add data visualization and student portal

SCREENSHOTS:-



Result Management System

Manage Student Details

Roll No.

D.O.B

Name

Contact

Email

Admission

Gender

Course

State

City

Pin

Address

Roll No.

Search

Roll No.	Name	Email	Gender

Save

Update

Delete

Clear

Result Management System

Add Student Results

select student

Select

Search

Name


Course

Marks Obtained

Full Marks

Submit

clear



Result Management System

View Student Results

Search By Roll No.

Search

Clear

Roll No.	Name	Course	Marks Obtained	Total Marks	Percentage

Delete

CONCLUSION:-

The **Student Result Management System (SRMS)** was created to solve a very real problem that many schools and colleges face—managing student data, courses, and exam results efficiently. In many institutions, these tasks are still done manually or using basic tools like spreadsheets, which can lead to errors, lost records, and a lot of wasted time. SRMS steps in to make that process smoother, faster, and far more reliable.

One of the best parts about SRMS is its **user-friendly design**. Built using Python's Tkinter library, the system offers a clean and simple interface that anyone can use—no special training or technical knowledge is required.

Whether you're adding a new student, updating course information, or entering exam marks, everything is straightforward and just a few clicks away. The system is divided into different modules—courses, students, results, and reports—each with its own window and functionality. This **modular structure** keeps the app organized and makes it easy to update or expand in the future. The use of **SQLite** as the backend ensures that all data is stored safely and can be retrieved quickly when needed.

We've also taken care to **reduce human error** by adding input validation and helpful pop-up alerts. Users get clear messages if they forget to fill in something important or try to perform an invalid action. This helps maintain the accuracy of records and saves time in the long run.

While SRMS currently functions as a desktop application, it's also built with **future growth in mind**. A basic web-based polling feature using Flask and WebSocket has already been added to demonstrate how the system can be expanded to support online access, voting, or student feedback systems in the future.

In the end, SRMS isn't just a coding project—it's a practical tool that can make a real difference in how educational institutions handle their academic data. It's reliable, efficient, and ready to use, with plenty of room to grow. Most importantly, it shows how technology can make everyday tasks simpler and more accurate—giving teachers and administrators more time to focus on what really matters: educating students.