K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Recognised under the section 2 (f) of the UGC Act 1956

Empowering the Youth; Empowering the Nation

K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

STUDENT RESULT MANAGEMENT SYSTEM

First Year Mini Project
Submitted by

ROLL NO.	NAME
2401010193	Anurag Upadhyay
2401010225	Yuvraj Singh Rathore
2401010209	Akshita Singh
2401010179	Diya Goel

Internal Mentor: Dr. Mansi Kajal

Project Overview

The **Student Result Management System (SRMS)** is a Python-based desktop application developed to help educational institutions efficiently manage their academic records. The system centralizes data related to courses, students, and exam results, eliminating the challenges of maintaining records manually.

It provides a graphical user interface (GUI) built with **Tkinter**, making it easy for users to interact with the system. All data is stored securely using **SQLite**, a lightweight and fast relational database.

The project includes modules for:

•**Course Management** – to manage course names, duration, charges, and descriptions.

•**Student Management** – to register, update, and delete student information.

•**Result Management** – to enter and calculate marks, ensuring accurate percentage computation.

•**Result Report** – to search and view results based on student roll numbers.

•**Dashboard** – to navigate between modules and display summary data.

Problem Statement

Managing student records, academic results, and course details is a vital task in any educational institution. However, many institutions still rely on outdated, manual processes, which come with several problems:

1. 🧾 **Manual Record Keeping**: Writing student details and results in registers or Excel files is tedious and error-prone.
2. ❌ **Lack of Centralization**: Information is scattered—courses in one file, student info in another, and results in a third—making it hard to find or update records quickly.
3. 🕒 **Time-Consuming**: Searching for a single student's record or result can take a long time when done manually.
4. 📉 **Human Errors**: Mistakes in entering marks, calculating percentages, or copying data between files can lead to inaccurate results.

Objectives

This project aims to:

- To **simplify** the process of handling student and result information.
- To provide an easy-to-use **graphical interface** for managing courses, students, and results.
- To ensure **accuracy** and **consistency** when storing or updating records.
- To allow **quick search** and display of results for any student.

Methodology

🧱 Architecture Overview:

1. Frontend (User Interface):

- Built using **Tkinter**, Python’s standard GUI library.
- Designed to be simple, clean, and intuitive for non-technical users.
- Provides separate windows for different operations like adding courses, registering students, entering results, and viewing reports.

Methodology

2. Backend (Database and Logic):

- Uses **SQLite**, a lightweight and embedded relational database.
- All data (courses, student details, and results) is stored locally in the **rms.db** file.
- SQL queries are used for inserting, updating, deleting, and retrieving records.

3. Modules:

- **Course Module**: Manages course names, duration, fees, and descriptions.
- **Student Module**: Handles registration with complete details including contact, address, and enrolled course.
- **Result Module**: Allows entering marks, calculates percentage, and prevents duplicate result entries.
- **Report Module**: Enables viewing and deleting student results by roll number.
- **Dashboard**: A main control panel that connects all modules and shows total counts.

Methodology

4. Data Handling:

- Inputs are collected through form fields.
- Basic validation is performed to ensure required fields are filled.
- Data is stored and retrieved using SQLite commands via Python’s **sqlite3** module.

Methodology

⚙️ Workflow Example (How it works):

- The user starts the application and reaches the **Dashboard**.
- They choose an option like **"Student"** to open the student management window.
- Data is entered in the form and saved to the database.
- If needed, the user can search, update, or delete records.
- The **Result** module allows assigning marks to a student, which automatically calculates the percentage.
- The **Report** module displays all result details for a searched student.

Technologies Used

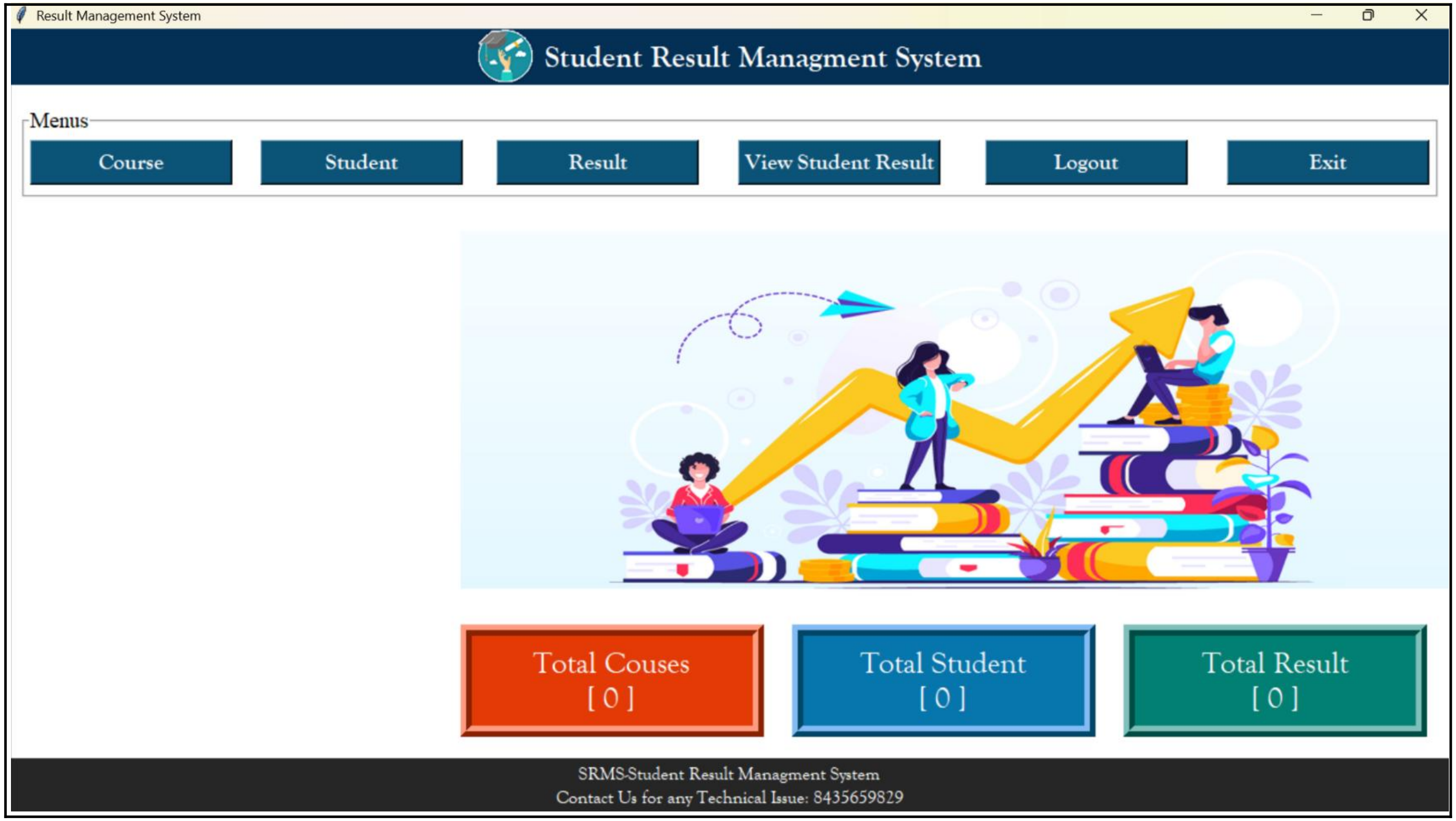
Components	Tool/Technology	Purpose
Programming	Python 3	Core development language
GUI Library	Tkinter	Creating desktop user interfaces
Database	SQLite	Storing and Managing data
Image Handling	Pillow (PIL)	Display images in the GUI
Optional Backend	Flask + SocketIO (Prototype)	For possible web-based extension

KEY FEATURES

- ❑ 🧑‍🏫 **Course Management**: Add, update, search, and delete course details.
- ❑ 🧑‍🎓 **Student Management**: Register students with full details including address and course.
- ❑ 📝 **Result Management**: Enter marks and automatically calculate percentage.
- ❑ 📋 **Result Viewer**: View results by searching with roll number.
- ❑ 🧭 **Dashboard Interface**: Easily navigate to any module.
- ❑ ❌ **Error Handling**: Prevents duplicate entries and shows helpful error messages.
- ❑ 📈 **Expandable**: Has the potential to become a web-based system in the future.

SCREENSHOTS

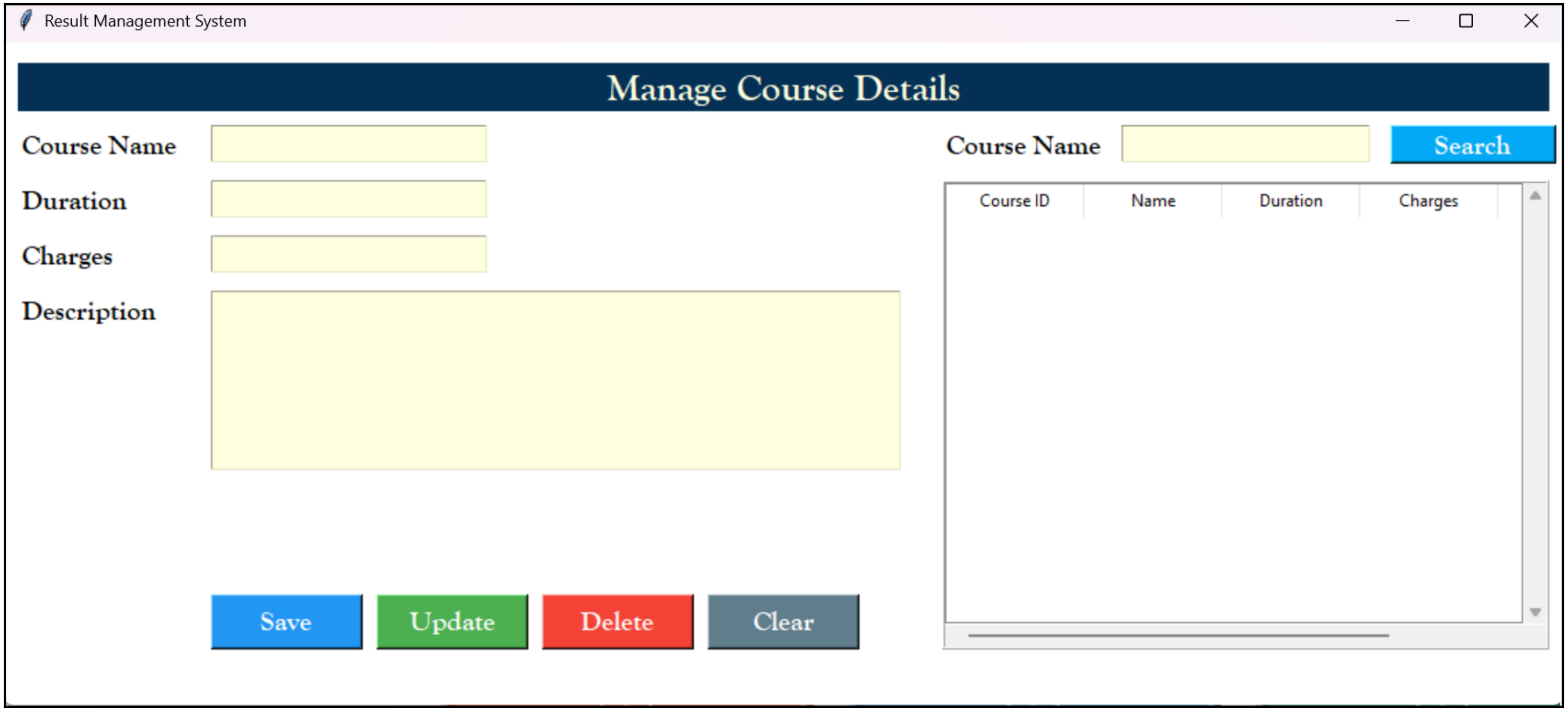
DASHBOARD:



SRMS-Student Result Managment System
Contact Us for any Technical Issue: 8435659829

SCREENSHOTS

COURSE MANAGEMENT:



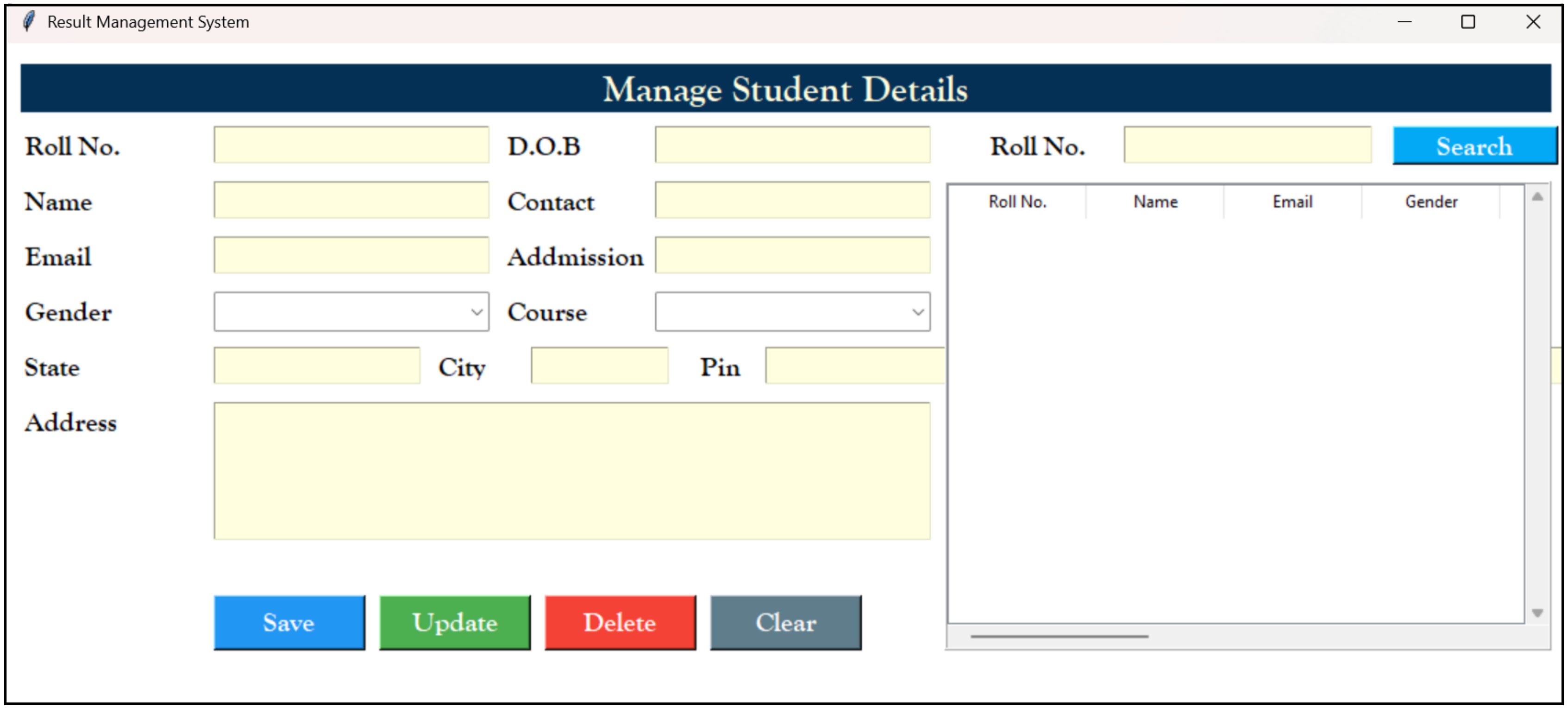
Result Management System

Manage Course Details

Course Name	Duration	Charges	Description

SCREENSHOTS

STUDENT MANAGEMENT:



Result Management System

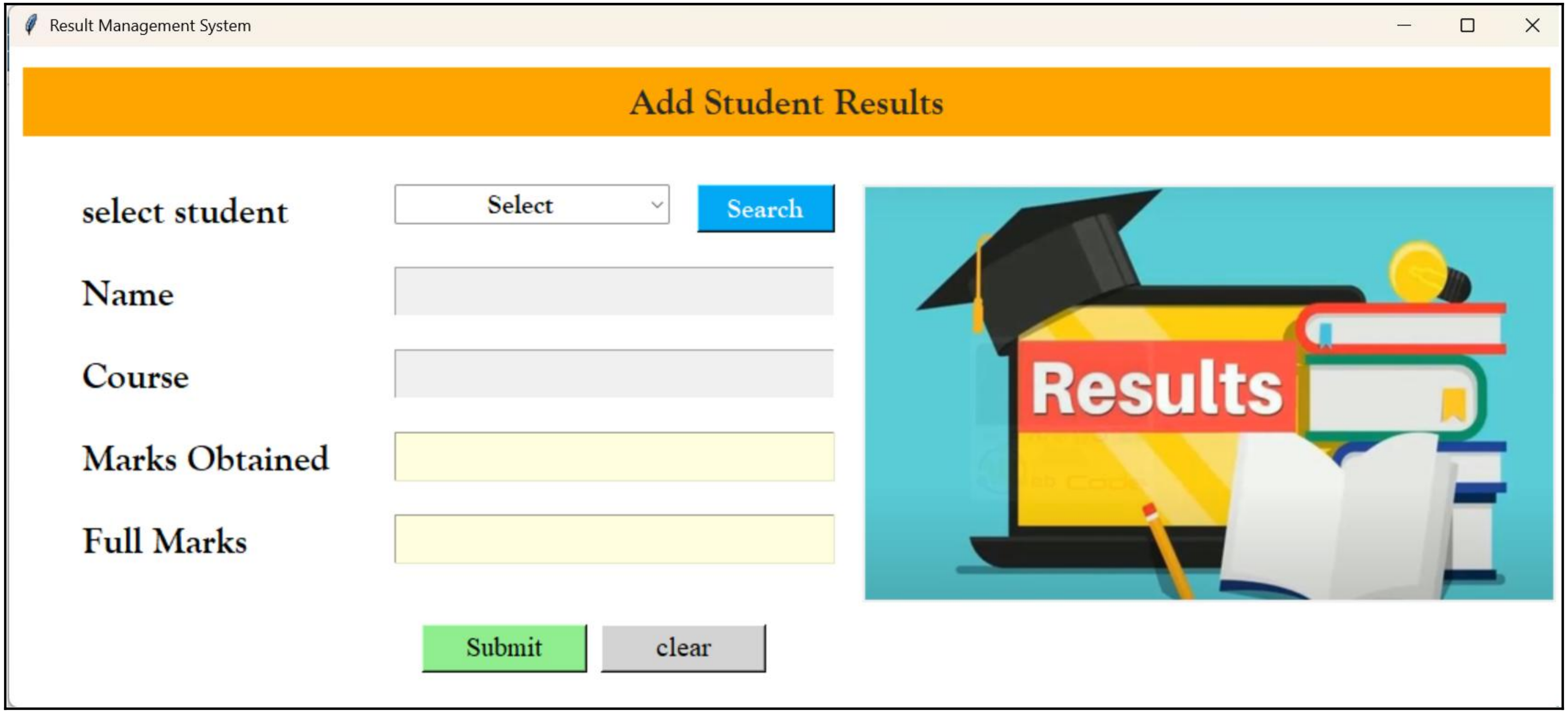
Manage Student Details

Roll No.		D.O.B	
Name		Contact	
Email		Addmission	
Gender		Course	
State		City	
Pin			
Address			

Roll No.	Name	Email	Gender

SCREENSHOTS

RESULT MANAGEMENT:



Result Management System

Add Student Results

select student

Name

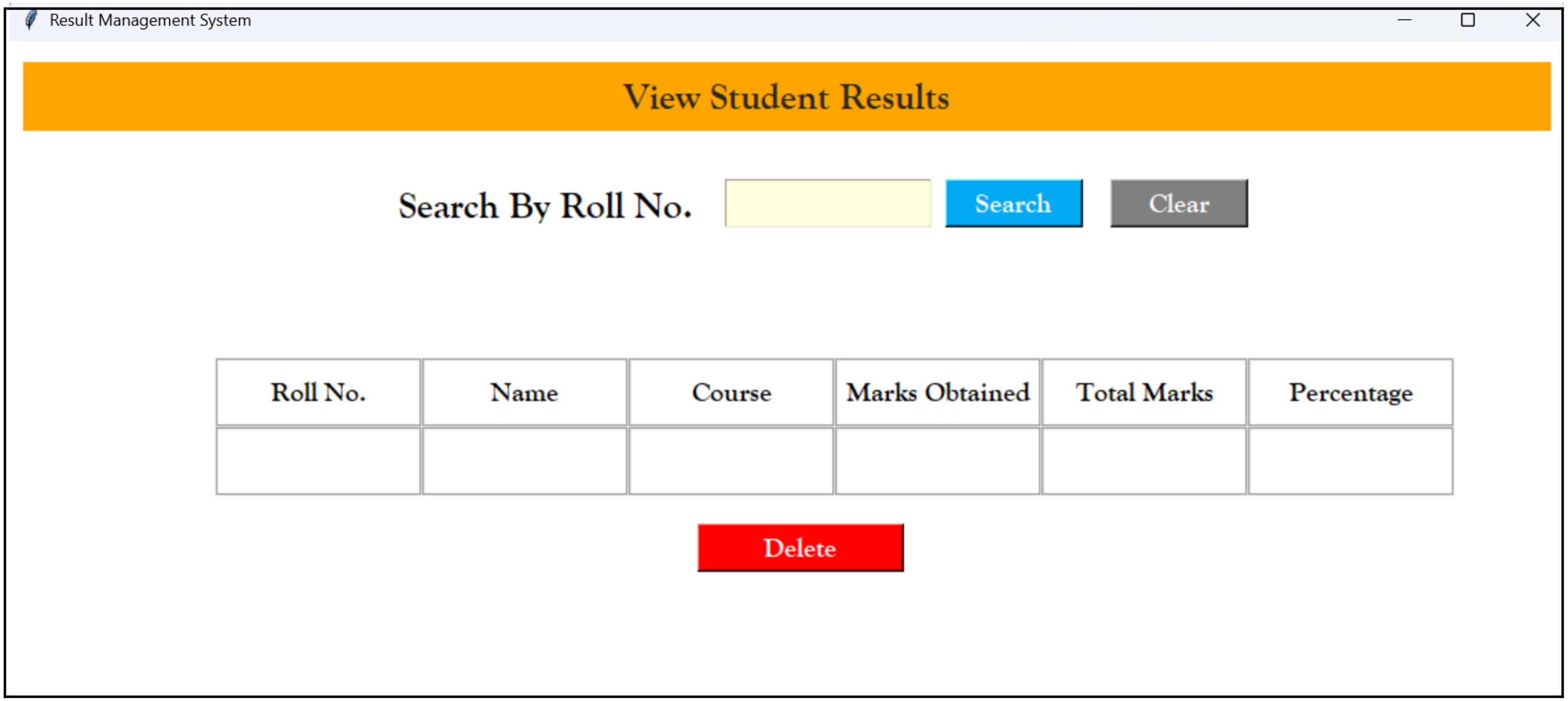
Course

Marks Obtained

Full Marks

SCREENSHOTS

RESULT VIEWER:



Result Management System

View Student Results

Search By Roll No.

Roll No.	Name	Course	Marks Obtained	Total Marks	Percentage

FUTURE WORK

- ✓ Add **login authentication** to protect access to data.
- ✓ Generate **PDF reports** of student results.
- ✓ Display **charts and statistics** of student performance.
- ✓ Convert the system into a **web-based platform** using Flask or Django.
- ✓ Add features for **bulk data upload** from Excel or CSV files.

CONCLUSION

This project—**Student Result Management System (SRMS)**—was built to make academic record-keeping easier and more efficient for schools and colleges. Instead of relying on registers or spreadsheets, which can be messy and error-prone, this system gives users a clean and simple way to manage student details, courses, and results all in one place.

Using Python and a lightweight database (SQLite), we've created a system where admins or staff can add students, assign courses, enter marks, and instantly view results. The interface is user-friendly, even for people with limited technical knowledge, and the system helps avoid common mistakes like duplicate entries or missing data.

Overall, this project solves a real problem and makes day-to-day work much smoother for educational institutions. It also has room to grow—features like login security, result printing, and even turning it into a website could easily be added in the future.

THANK YOU