
STUDENT RESULT MANAGEMENT SYSTEM

Madhu Singh^{*1}, Prakhar Dev^{*2}

^{*1,2}Students, Department Of Computer Science And Engineering, School Of Management Sciences
(Aff. APJ Abdul Kalam Technical University), Lucknow, Uttar Pradesh, India.

ABSTRACT

The Student Result Management System is a Python-based desktop application that efficiently manages and organizes student data in educational institutions. The system utilizes an SQLite database and the tkinter library for GUI development. Its user-friendly interface enables easy navigation and management of student information, including personal details, course details, and exam results. The system offers various functions, such as adding and deleting student and course records, updating exam results, and generating student results. To ensure student data privacy, the system incorporates security features like password protection. The Student Result Management System can significantly improve educational institutions' performance and student data management.

Keywords: Desktop Application, Student Data, Personal Details, Course Details, Exam Results, Security Features.

I. INTRODUCTION

The Student Result Management System is user-friendly interface enables users to store, manage, and analyze student data with ease. Using paper-based systems to manage student data can often be inefficient and prone to errors, making computer-based systems a better alternative. The Student Result Management System aims to provide an efficient and reliable tool for managing student data, offering a wide range of operations such as adding, deleting, and updating student records, updating exam results, generating student results based on marks, and analyzing student performance. The system is also secure, with password protection to ensure the privacy of student data. To build the system, Python programming language, known for its power and versatility, is used. Python makes it easy to develop and maintain applications. The system uses SQLite database for storing and managing student data, which provides reliable and efficient data storage. Additionally, the tkinter library is used for GUI development, providing an intuitive and user-friendly interface. It simplifies navigation within the system, making student data management much easier. The Student Result Management System is a reliable and efficient tool for managing student data, enhancing the performance of educational institutions.

II. LITERATURE SURVEY

After conducting a literature survey on student result management systems, it has become evident that there is a growing need for efficient and dependable systems to manage student data in educational institutions. Traditional paper-based systems are deemed inadequate as they are inefficient and prone to errors, which has led to the development of computer-based systems. Several studies have explored the use of various programming languages and database systems for developing student result management systems. Python programming language has proven to be a popular and effective language for developing desktop applications, while the SQLite database is a reliable and efficient means of data storage. User interface design has also been a significant focus of many studies in the development of student result management systems. A user-friendly interface is essential for managing student data effectively. The tkinter library, which is a standard GUI toolkit for Python, is a proven and effective tool for developing user-friendly interfaces in Python-based applications. Furthermore, some studies have focused on the security aspects of student result management systems. They emphasize the importance of data privacy and security and propose solutions such as password protection to ensure the privacy of student data. To summarize, the literature survey highlights the importance of developing efficient and reliable computer-based systems for managing student data, with a focus on user interface design and security features. The Student Result Management System project, using Python GUI and SQLite database, addresses these issues and provides an effective solution for managing student data in educational institutions.

III. METHODOLOGY

To develop a student result management system using Python GUI-based and SQLite3, the methodology includes several steps such as requirements gathering, user interface design, database schema design, implementation, testing, deployment, and maintenance. Python libraries like Tkinter, PyQt, or wxPython can be utilized to create an intuitive user interface, and SQLite3 can be utilized to store the student data efficiently. The implementation process involves coding the system using Python programming language and extensively testing it for bugs and errors. Once the testing is complete, the system can be deployed and maintained according to the user's feedback and requirements. Overall, this system offers an efficient and scalable solution for managing student data in educational institutions.

IV. SYSTEM REQUIREMENTS

Hardware Requirements:

The hardware requirements for the system are as follows:

Intel Core i3 processor or above: A quad-core processor is a chip with four independent units called cores that read and execute central processing unit (CPU) instructions such as add, move data, and branch. Each core operates in conjunction with other circuits such as cache, memory management, and input/output (I/O) ports. The individual cores in a quad-core processor can run multiple instructions at the same time, increasing the overall speed for programs compatible with parallel processing. Manufacturers typically integrate the cores onto a single semiconductor wafer, or onto multiple semiconductor wafers within a single IC (integrated circuit) package.

RAM: A minimum of 500MB RAM is required. RAM, or random-access memory, is a type of computer main memory in which certain contents may be retrieved directly by the central processing unit in a very short amount of time, independent of the order (and hence location) in which they were recorded. Random-access circuits can support two forms of memory: static RAM (SRAM) and dynamic RAM (DRAM).

Hard disk: A minimum of 2GB hard disk space is required. A hard disk, also called a hard disk drive or hard drive, is a magnetic storage medium for a computer. Hard disks are flat circular plates made of aluminum or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information. Data are stored on their surfaces in concentric tracks. A small electromagnet, called a magnetic head, writes a binary digit (1 or 0) by magnetizing tiny spots on the spinning disk in different directions and reads digits by detecting the magnetization direction of the spots. A computer's hard drive is a device consisting of several hard disks, read/write heads, a drive motor to spin the disks, and a small amount of circuitry, all sealed in a metal case to protect the disks from dust. In addition to referring to the disks themselves, the term hard disk is also used to refer to the whole of a computer's internal data storage. Beginning in the early 21st century, some personal computers and laptops were produced that used solid-state drives (SSDs) that relied on flash memory chips instead of hard disks to store information.

V. SOFTWARE REQUIREMENTS

The software requirements for the system are as follows:

Python: Python is a popular programming language that was created by Guido van Rossum and released in 1991. It is used for web development (server-side), software development, mathematics, and system scripting. Python works on different platforms such as Windows, Mac, Linux, Raspberry Pi, etc. It has a simple syntax similar to the English language, and its syntax allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way, or a functional way.

SQLite3: SQLite in general is a server-less database that can be used within almost all programming languages, including Python. Server-less means there is no need to install a separate server to work with SQLite, so you can connect directly with the database. SQLite is a lightweight database that can provide a relational database management system with zero-configuration because there is no need to configure or set up anything to use it.

Operating System: The system requires Windows 10 or above as an operating system (OS). An operating system

is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer.

VI. MODULES

The Registration Module is used to register new users with their details. The Login Module is used for managing login credentials. The Course Module is used for managing details of courses offered. The Student Module is used for managing information and details of students enrolled in courses. The Result Module is used for managing the results of students. The View Result Module allows users to view their results by entering their credentials. These modules provide an efficient and organized way to manage student data in an educational institution.

VII. IMPLEMENTS

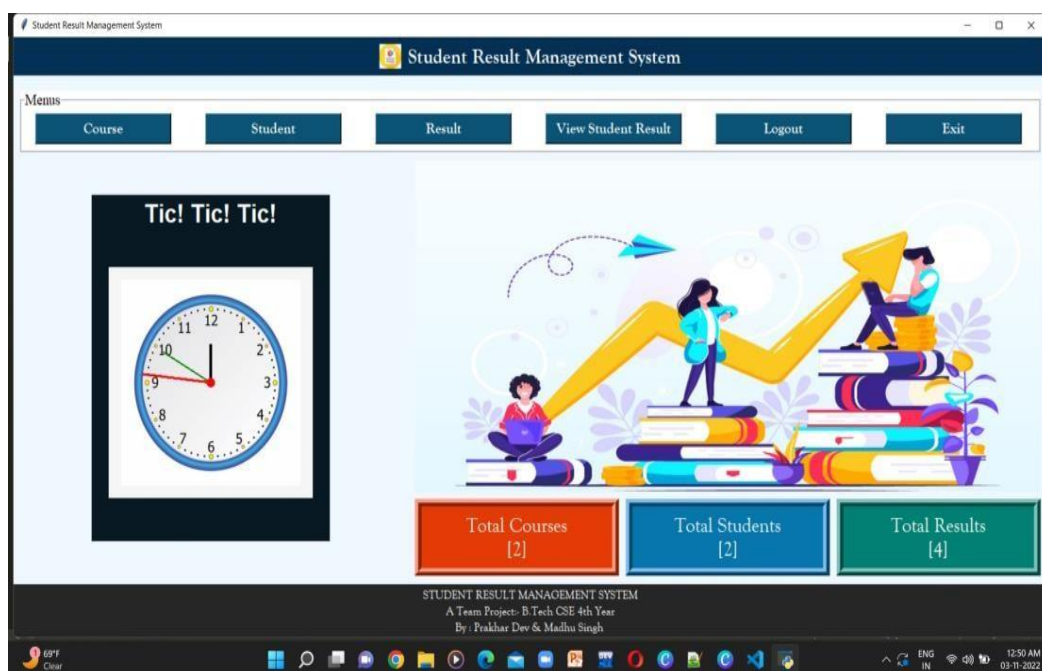


Figure 1: Dashboard

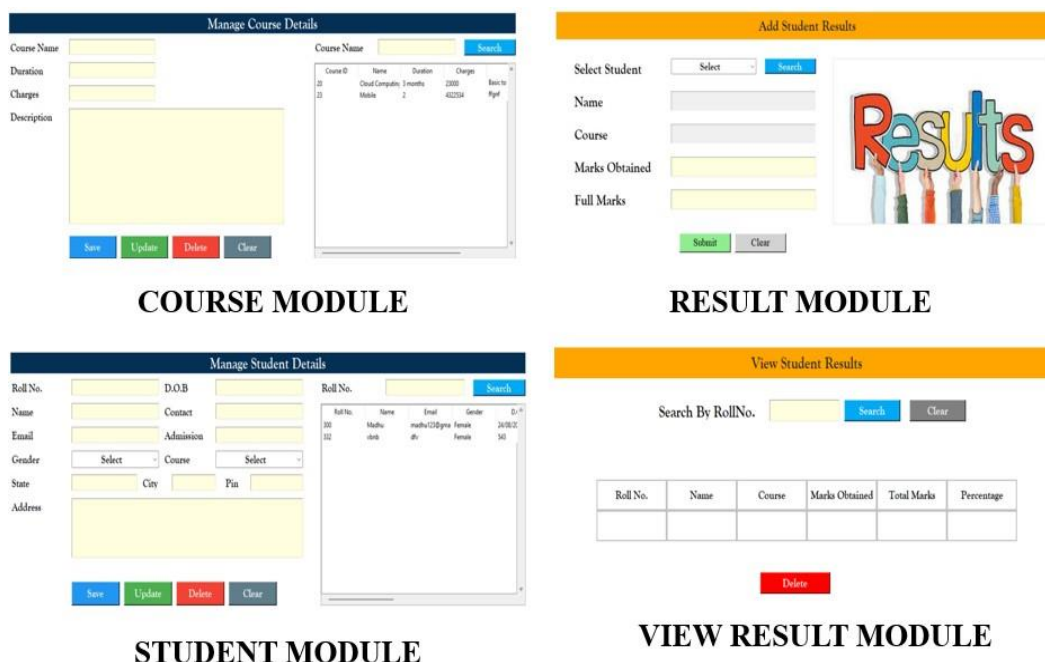


Figure 2: Modules

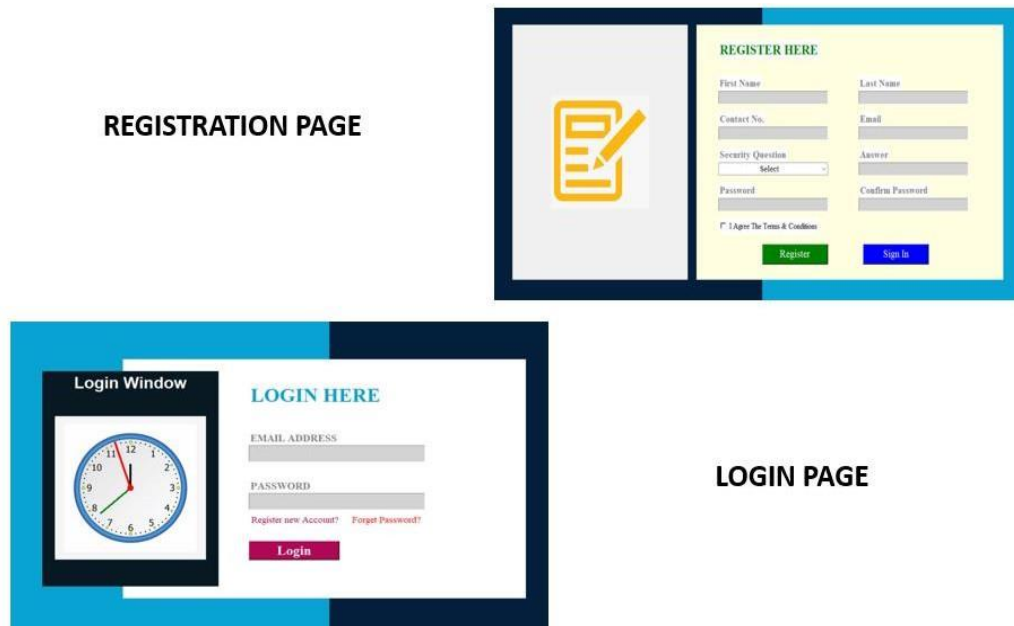


Figure 3: Registration and Login Page

VIII. FUTURE SCOPE

In summary, the future scope of the project includes:

- Generating certificates for students
- Adding more advanced features to the Student Result Management System
- Hosting the platform on online servers for worldwide accessibility In the past, data used to be manually inserted for result analysis, but now the project uses SQL files for data extraction. In the future, data can be fetched and parsed in various formats such as doc, csv, and odt. Additionally, there is potential for data visualization using pie charts, graphs, and other methods. The project can be made more interactive and user-friendly by adding features such as a group chat for students to discuss various engineering issues, thus fulfilling each user's needs in the best way possible.

IX. CONCLUSION

The Student Result Management System is a web-based application that can be accessed by anyone, anywhere, and at any time. This system simplifies the process of calculating and visualizing results for both students and faculty. Our project aims to fulfill the requirements of managing project work with user-friendly coding. This package is expected to satisfy all the needs of the organization and prove to be a powerful tool. The objective of the software planning is to provide a framework that enables managers to make reasonable estimates at the beginning of the software project and update them regularly as the project progresses.

X. REFERENCES

- [1] Available online at https://en.wikipedia.org/wiki/Student_information_system.
- [2] Available online at https://www.youtube.com/Web_tech.
- [3] Available online at <https://www.python.org/>.
- [4] Available online at <https://www.w3schools.com/>.
- [5] Kamthane, A.N., & Kamthane, A. A. (2017). Programming and Problem Solving with Python. Pearson Education India.
- [6] Walia, E. S., & Gill, E. S. K. (2014). A framework for web-based student result management system using Python. International Journal of Computer Science and Mobile Computing, 3(8), 24-33.
- [7] Liu, Z., Wang, H., & Zan, H. (2010, October). Design and implementation of student result management system. In 2010 International symposium on intelligence information processing and trusted computing (pp. 607-610). IEEE.