

A Web Based Application for Student Management System

A Project Report submitted in partial fulfillment of the requirement for the award of
degree of
Bachelor of Technology in Computer Science and Engineering

A. BHANU TEJA (22B81A05CG)

B. SRIDHAR (22B81A05EG)

S. VARSHITH (22B81A05EK)



Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,

Accredited by NBA, and NAAC)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.) - 501510, Telangana State

2023-24



CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,
Accredited by NBA, and NAAC) Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Miniproject work entitled “**A Web Based Application for Student Management System**” is being submitted by **A. BHANU TEJA (22B81A05CG), B. SRIDHAR (22B81A05EG) and S. VARSHITH (22B81A05EK)** in partial fulfillment of the requirement for the award of the Minor degree of **Bachelor of Technology in Computer Science and Engineering**, during the academic year 2023-2024.

Professor Incharge

Professor and Head, CSE

Dr. A. Vani Vathsala

External Examiner

DECLARATION

I hereby declare that this Mini project report titled “**A Web Based Application for Student Management System**” submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by us. The information and data given in the report is authentic to the best of our knowledge. This Mini project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

A. Bhanu Teja(22B81A05CG)

B. Sridhar(22B81A05EG)

S. Varshith(22B81A05EK)

Date: 27/05/2024

Place: Hyderabad

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our principal **Dr. K. Ramamohan Reddy, Dr.A.Vani Vathsala**., Head of CSE Department, CVR College of Engineering, for having been kind enough to arrange necessary facilities for executing the project in the college.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

ABSTRACT

A web Based Application for Student Management System deals with the maintenance of university, college, faculty, student information within the university. SGS is an automation system, which is used to store the college, faculty, student, courses and information of a college. Starting from registration of a new student in the college, it maintains all the details regarding the attendance and marks of the students. The project deals with retrieval of information through an INTRANET based campus wide portal. It collects related information from all the departments of an organization and maintains files, which are used to generate reports in various forms to measure individual and overall performance of the students.

Development process of the system starts with System analysis.

System analysis involves creating a formal model of the problem to be solved by understanding requirements.

TABLE OF CONTENTS

Chapter			Page No.
		List of Figures	
		Abbreviations	I
1		Introduction	1
	1.1	Motivation	2
	1.2	Problem statement	2
	1.3	Project Objectives	3
2		Literature Survey	4
	2.1	Existing work	4
	2.2	Limitations of Existing work	5
3		Software & Hardware specifications	7
	3.1	Software requirements	7
	3.2	System specifications	7
	3.3	User requirements	7
4		Proposed System Design	9
	4.1	Proposed methods	9
	4.2	Architecture Diagram	10
	4.3	Use case Diagram	10
	4.4	Technology Description	11
5		Implementation & Testing	13
	5.1	Implementation	13
	5.2	Testing	19
6		Conclusion & Future scope	
	6.1	Conclusion	22
	6.2	Future Scope	22
		References	23

Abbreviations

- SMS – Student Management System
- OpenEMIS - open-source educational management information system
- GDPR - General Data Protection Regulation
- FERPA. - Family Educational Rights and Privacy Act
- RFID – Radio Frequency Identification

CHAPTER 1

INTRODUCTION

A Student Management System is an environment that manages all the data of the students who are studying in an educational institution. This data is computerized through an automated system. Here, computerization is more advantageous than the usual method. Thus, a student management system offers many benefits to an educational institution. It allows teachers to easily change and access student data, and parents can easily focus on children with a clear environment to meet state level compliance and other regulatory requirements.

This system we have created also facilitates the entry, maintenance and viewing and of all authorized student details. Here we mainly focus on the examination of students, their subjects, the registration department which conducts the registration process, the examination department which conducts the examinations, the IT division which is important component of the result and the smart ID components. The special thing is that we have given all the students a unique ID and the degree program they are involved in has also focused on age, gender and contact number. Each department has a unique ID. For convenience, a designated location and e-mail address are also used.

In contemporary educational settings, the effective management of student-related data and administrative processes stands as a cornerstone for the successful operation of educational institutions. Traditional methods of paper-based record-keeping and disjointed software solutions often fall short in meeting the dynamic demands of modern education. Consequently, there arises a pressing need for a comprehensive solution that can streamline administrative tasks, foster seamless communication among stakeholders, and elevate overall operational efficiency. This project endeavors to address this need by developing a robust web-based Student Management System (SMS), tailored to cater to the specific requirements of educational institutions. By providing a centralized platform for managing student data, academic records, and administrative processes, the proposed SMS aims to revolutionize the management of educational institutions, facilitating informed decision-making, enhancing collaboration, and promoting student success.

1.1. MOTIVATION

The impetus behind the development of a web-based Student Management System stems from a recognition of the myriad challenges faced by educational institutions in managing student-related information using outdated or fragmented systems. These challenges manifest in various forms:

- Inefficiencies in administrative workflows: Manual data entry, reliance on paper-based records, and the use of disparate systems contribute to inefficiencies and errors in managing student information.
- Communication barriers: Limited or outdated communication channels impede effective collaboration and engagement among administrators, teachers, students, and parents.
- Lack of real-time access to data: Difficulty in accessing up-to-date student records, academic progress, and attendance information hampers timely interventions and informed decision-making.

The development of a web-based SMS is motivated by a desire to alleviate these challenges and empower educational institutions with a modern, integrated solution for managing student information, fostering communication, and enhancing operational efficiency.

1.2. PROBLEM STATEMENT

The problem at hand revolves around the absence of an efficient and comprehensive system for managing student-related information and administrative processes within educational institutions. Current systems often exhibit deficiencies such as:

- Fragmentation in data management: Student data is often dispersed across multiple systems or maintained in disparate spreadsheets, leading to duplication of efforts and inconsistencies in information.
- Communication bottlenecks: Insufficient communication tools hinder effective collaboration between stakeholders, resulting in delays and misunderstandings.
- Manual administrative tasks: Labor-intensive processes for tasks like enrollment, attendance tracking, and grade management are prone to errors and consume valuable time and resources.

1.3. PROJECT OBJECTIVES

The primary objectives of the Student Management System are as follows:

- **Efficiency:** Streamline administrative processes, reduce manual tasks, and improve overall operational efficiency within the educational institution.
- **Transparency:** Provide stakeholders with transparent access to student data, academic records, and communication channels, fostering trust and accountability.
- **Engagement:** Enhance student engagement and academic success by providing access to resources, grades, attendance records, and communication tools.
- **Collaboration:** Facilitate collaboration and communication among administrators, teachers, students, and parents, promoting a cohesive learning environment.
- **Data-driven Decision Making:** Enable data-driven decision-making by providing comprehensive reporting and analytics capabilities to track student performance, enrollment trends, and institutional metrics.

CHAPTER 2

LITERATURE REVIEW

A literature review on (SMS) explores existing research, technologies, challenges, and trends in the development and implementation of such systems. The review covers various aspects, including historical context, technological advancements, functional components, and the impact of these systems on educational institutions. Here is an organized overview for a literature review on student management systems:

2.1 EXISTING WORK

In the domain of student management systems, a multitude of solutions exist, each with its unique features, strengths, and weaknesses. These existing systems serve as foundational pillars for educational institutions worldwide, facilitating efficient management of student data, academic progress tracking, and administrative tasks. Below are some of the prominent existing systems:

1. **OpenEMIS:** OpenEMIS stands out as an open-source educational management information system. It offers a comprehensive suite of tools for managing student records, teacher information, and administrative tasks. OpenEMIS emphasizes accessibility and affordability, making it particularly attractive to resource-constrained educational institutions, especially in developing regions.
2. **PowerSchool:** Widely recognized as a leading student information system, PowerSchool boasts a user-friendly interface and a robust set of features. From attendance tracking to grade management and parent communication, PowerSchool caters to the needs of K-12 schools and districts, streamlining administrative processes and enhancing collaboration between educators, students, and parents.

3. Schoology: Schoology transcends the boundaries of a traditional student management system by integrating learning management functionalities. This platform empowers educators to create engaging online courses, manage curriculum content, and assess student learning outcomes effectively. Schoology's seamless integration of course management with student information makes it a popular choice among educators seeking to leverage technology for educational excellence.
4. Fedena: Fedena emerges as a comprehensive school management software solution tailored to the diverse needs of educational institutions. With modules covering admissions, attendance, examinations, finance, and more, Fedena offers a holistic approach to student management. Its flexibility and scalability make it suitable for schools of all sizes, from small independent schools to large university campuses.
5. CampusVue: Catering specifically to higher education institutions, CampusVue specializes in providing advanced student information system functionalities. From admissions management to academic advising and student financials, CampusVue addresses the unique challenges faced by colleges and universities in managing student data and supporting student success initiatives.

2.2 LIMITATIONS OF EXISTING WORK

Despite their significant contributions to education management, existing student management systems are not without their limitations. Understanding these limitations is essential for informing the development of more effective and inclusive solutions:

1. Cost: Many commercial student management systems come with hefty licensing fees and ongoing maintenance costs, posing a significant financial barrier to adoption, particularly for cash-strapped educational institutions with limited budgets.

2. **Scalability:** Some student management systems struggle to accommodate the scalability requirements of growing educational institutions or large-scale deployments. Issues such as performance degradation and system instability may arise when handling a high volume of data or concurrent user interactions.
3. **Customization:** While certain systems offer customization options, achieving a tailored solution that aligns precisely with an institution's unique workflows and requirements may necessitate extensive development efforts or third-party integrations, thereby increasing complexity and cost.
4. **Integration Challenges:** Integrating student management systems with other educational software applications, such as learning management systems (LMS), financial management systems, or library management systems, can be complex and time-consuming. Incompatibilities between systems, data synchronization issues, and API limitations may hinder seamless interoperability and data exchange.
5. **Usability Concerns:** Despite efforts to improve user interfaces and user experiences, some student management systems remain complex and unintuitive, requiring extensive training and technical support for administrators, teachers, and staff to effectively navigate and utilize the system's features.
6. **Data Security Risks:** Safeguarding sensitive student data from unauthorized access, data breaches, or cyberattacks is paramount. However, not all student management systems incorporate robust security measures, leaving educational institutions vulnerable to potential data security threats and compliance risks, such as GDPR or FERPA violations.
7. **Support and Maintenance:** The level of technical support and ongoing maintenance provided by vendors varies widely. Educational institutions may encounter challenges in obtaining timely assistance, troubleshooting issues, or receiving software updates and bug fixes, potentially leading to disruptions in operations and user frustration.

CHAPTER 3

REQUIREMENT ANALYSIS

Creating a student management system involves identifying various user requirements to ensure that the system is efficient, user-friendly, and meets the needs of all stakeholders. Here are the typical user requirements for a student management system, categorized by user roles:

3.1 SOFTWARE REQUIREMENTS

1. Operating system: Windows/Linux
2. Execution Environment: VS Code
3. Programming Language: Python 3.10.9
4. Libraries: Tkinter, PIL, typing

3.2 HARDWARE REQUIREMENTS

1. Processor-Intel i5 or higher
2. RAM – 8GB or higher
3. Storage-256GB SSD or above
4. Internet connection-40Mbps or more
5. Display-High-resolution monitor

3.3 USER REQUIREMENTS

Creating a student management system involves identifying various user requirements to ensure that the system is efficient, user-friendly, and meets the needs of all stakeholders. Here are the typical user requirements for a student management system, categorized by user roles:

- **User Authentication and Authorization:** The system should allow different users such as administrators, teachers, students, and parents to log in with secure authentication methods and access only the features and information appropriate to their roles.

- **Student Information Management:** It should be able to store and manage student data including personal information, academic records, attendance, disciplinary records, and health records.
- **Course and Curriculum Management:** The system should allow administrators and teachers to create, manage, and update courses, syllabi, and curriculum materials. This includes scheduling classes, assigning teachers to courses, and managing class resources.
- **Reporting and Analytics:** The system should provide reporting and analytics capabilities for administrators to track key metrics such as student performance, attendance rates, and demographic trends. This can help in making data-driven decisions and improving overall educational outcomes.
- **Security and Data Privacy:** Data security and privacy should be a top priority, with features like role-based access control, encryption, regular backups, and compliance with relevant data protection regulations such as GDPR or FERPA.
- **User Support and Training:** The system should provide adequate user support resources such as documentation, tutorials, and helpdesk assistance, as well as training for administrators, teachers, students, and parents to use the system effectively.

CHAPTER 4

SYSTEM DESIGN

Designing a Student Management System (SMS) involves creating a detailed architecture and defining the components that will interact to fulfill the requirements gathered during the analysis phase. The system design includes both high-level architecture and detailed component design. Below is a comprehensive guide to the system design of a Student Management System:

4.1 PROPOSED METHODS

A student management system can employ various methods to effectively manage student information and academic processes. One approach is through a centralized database that stores comprehensive student records, including personal details, academic achievements, attendance, and disciplinary history. This database can be accessed by authorized users through a secure login system, ensuring data privacy and security. Additionally, the system can incorporate features for course and curriculum management, allowing administrators and teachers to create, schedule, and update courses and syllabi. Automated attendance tracking methods, such as biometric or RFID systems, can streamline the process of recording student attendance, reducing administrative workload. Grading and assessment functionalities can facilitate the timely input of grades, calculation of GPA, and generation of report cards, providing valuable feedback to students and parents. Communication and collaboration tools, such as messaging platforms and discussion forums, foster effective communication between teachers, students, and parents, promoting engagement and involvement in the educational process. Finally, integration with other systems, such as learning management systems and school management software, ensures seamless data exchange and workflow automation, enhancing overall efficiency and productivity.

4.2 ARCHITECTUR DIAGRAM

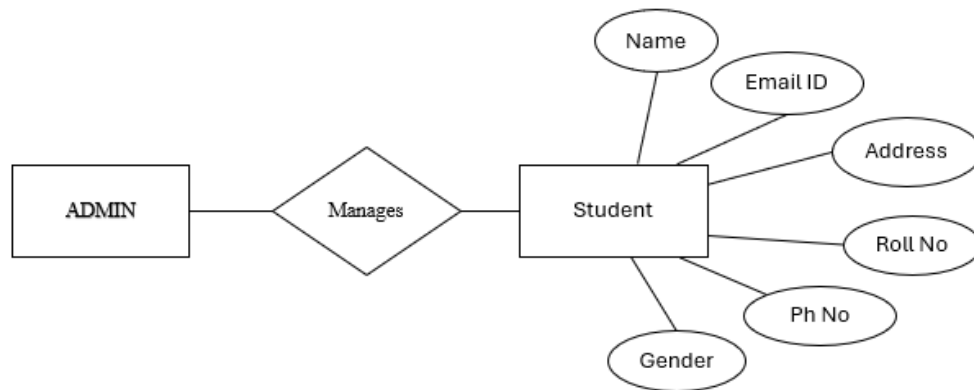


Fig 4.2.1 Architecture diagram

4.3 USE CASE DIAGRAM

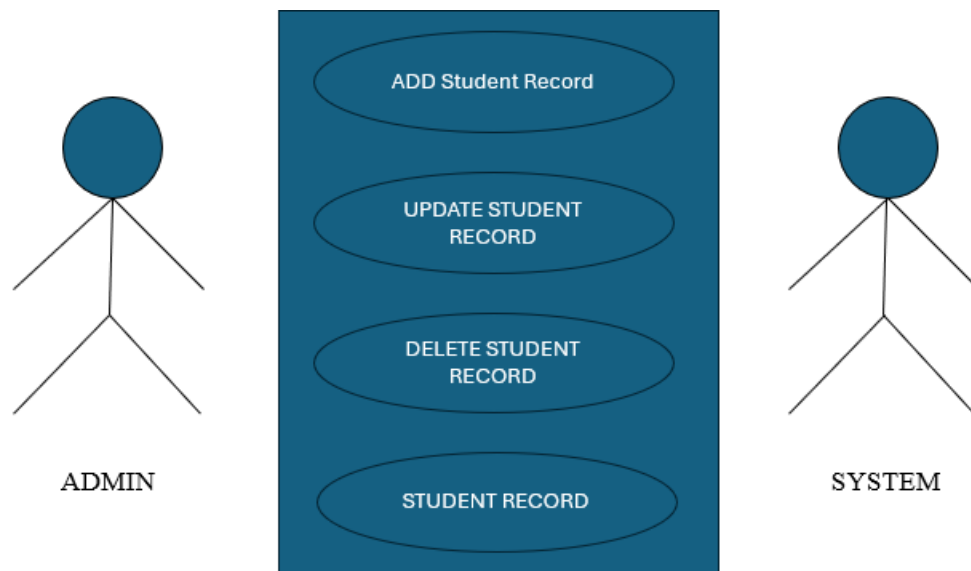


Fig 4.3.1 Use case diagram

4.4 TECHNOLOGY DESCRIPTION

A student management system typically leverages a variety of technologies to fulfill its functionalities efficiently and securely. Here's a breakdown of some key technologies commonly used in such systems:

1. Python: Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

2. Tkinter Tk: Tkinter is the standard Python interface to the Tk GUI toolkit. It is a lightweight, easy-to-learn, and cross-platform GUI library for creating desktop applications. Tkinter provides a wide range of widgets, such as buttons, labels, text boxes, menus, canvases, and more. It is included in the standard Python distribution, making it a convenient choice for rapid application development.
3. Widgets: TKinter provides a variety of widgets (UI components) such as buttons, labels, text boxes, check buttons, radio buttons, list boxes, menus, and more. These widgets can be arranged and configured to create interactive GUIs.
4. Geometry Management: TKinter offers several geometry managers (pack, grid, and place) to organize and layout widgets within a window or frame.
5. Event Handling: TKinter allows developers to bind functions or methods to events triggered by user interactions with the GUI components. This enables event-driven programming, where actions are performed in response to user inputs.

6. Customization: Developers can customize the appearance and behaviour of widgets by specifying various options such as colors, fonts, sizes, and event bindings.
7. Integration with Python: Tkinter is included with Python standard library, so there's no need to install any additional packages. It provides a simple and easy-to-use interface for creating GUI applications directly within Python scripts.
8. Tkinter ttk: ttk (themed Tkinter) is a themed widget set for Tkinter. It provides a more modern and visually appealing appearance for Tkinter applications. ttk includes a variety of themes and styles, which can be customized to create a consistent and visually appealing user interface.
9. PIL (Pillow): PIL (Python Imaging Library) is a powerful Python library for working with images. It provides features for loading, manipulating, and saving various image formats, such as JPEG, PNG, GIF, and BMP. Pillow is a friendly fork of PIL, which is more actively maintained and includes additional features.

CHAPTER 5

IMPLEMENTATION & TESTING

Implementing and testing a Student Management System (SMS) involves a series of structured steps to ensure the system meets all requirements and functions as intended. Below is a detailed guide for the implementation and testing process.

5.1 IMPLEMENTATION

Importing the necessary libraries: By importing the tkinker library we used widgets like button, label, label frame to set and get the data from the interface and from typing library we use text widget to display text. Also from PIL library we use Image and ImageTk widgets to display the image properly.

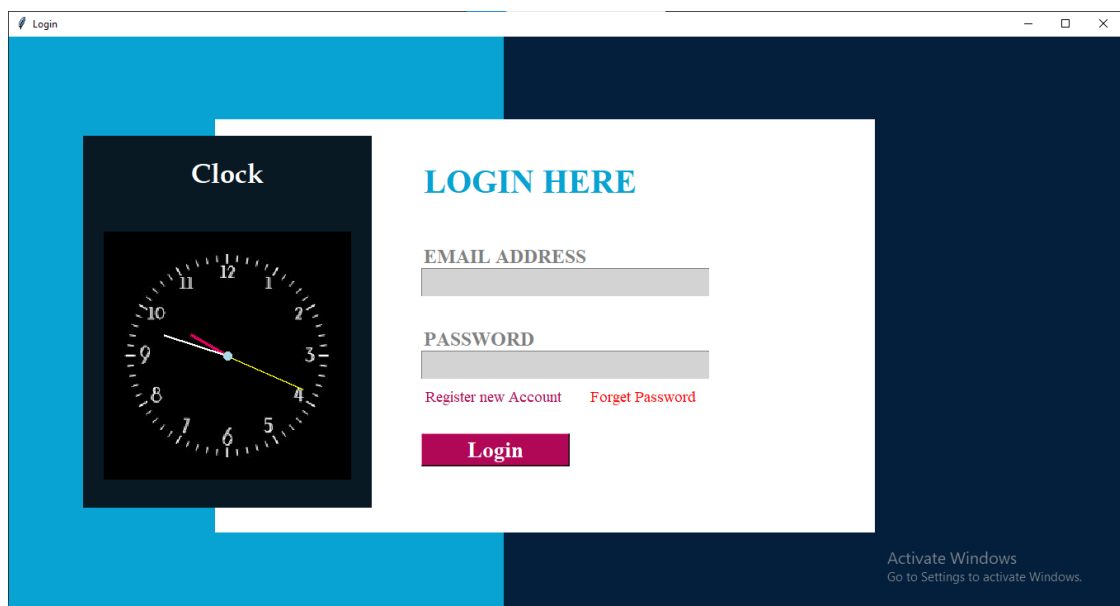


Fig 5.1.1 Login Page

Code:

```
def login(self):
    if self.txt_email.get()==" or self.txt_pass_.get()=="":
        messagebox.showerror("Error","All fields are required",parent=self.root)
    else:
        try:
            con=sqlite3.connect(database="sms.db")

            cur=con.cursor()
            cur.execute("select * from employee where email=? and
password=?",(self.txt_email.get(),self.txt_pass_.get()))
            row=cur.fetchone()
            if row==None:
                messagebox.showerror("Error","Invalid USERNAME &
PASSWORD",parent=self.root)

            else:
                messagebox.showinfo("Success",f"Welcome:
{self.txt_email.get()} ",parent=self.root)
                self.root.destroy()
                os.system("python dashboard.py")
                con.close()
        except Exception as es:
            messagebox.showerror("Error",f"Error Due to: {str(es)}",parent=self.root)
```

Description: Here after running the code First comes the login page where we have to enter the required Email and password if they previously exist. (Fig 5.1.1) Then we get logged in or else it pops an error saying email or password incorrect.

Fig 5.1.2 Forget password page

Description: This page gets popped when user needs to reset the password. For that user need to give correct answer for Security question then the password gets reset. (Fig 5.1.2)

Fig 5.1.3 Register page

Description: This page gets open when there in some new user and they wants to register in this application. For that they have to press Register here in the login page so that the get redirected into this page. By giving the required detains they can get registered. (Fig 5.1.3)

In the login page if they get logged in the immediately get redirected into dashboard.

Description: Then we go to the main dashboard and here we get all the options about the student. (Fig 5.1.4)

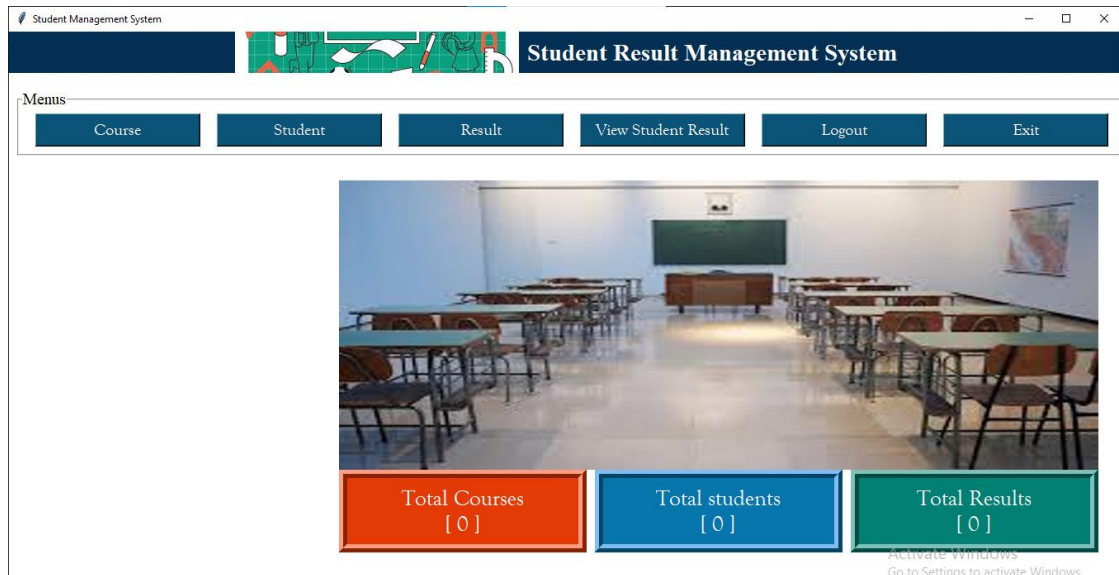


Fig 5.1.4 Dashboard view

Description: when we go to course then we get options regarding course like course name duration etc and we have to enter the required details and these details will get stored in a database created. (Fig 5.1.5)

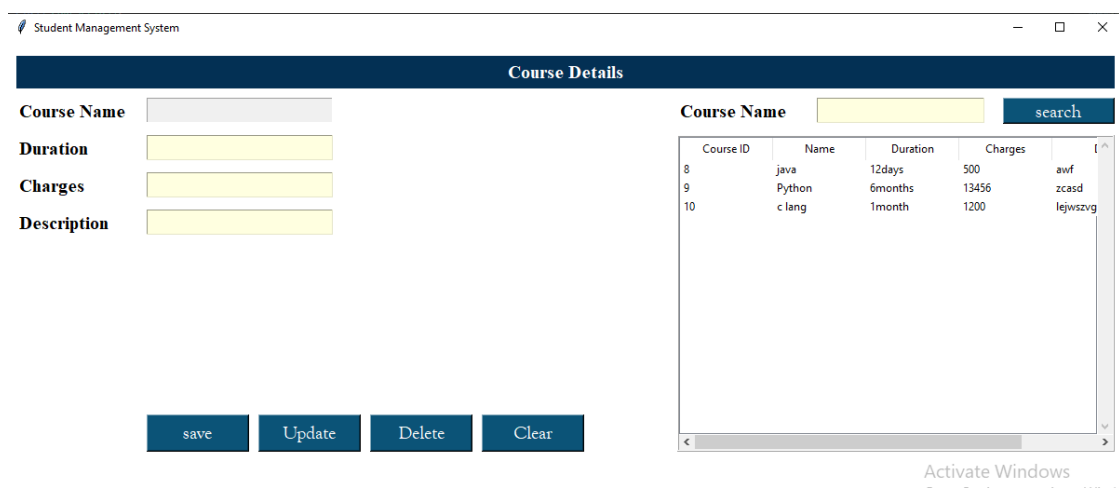


Fig 5.1.5 Course view

Description:

Here by the below code we create database named sms.db and init we create different tables like course, student, result. these tables are used to store the data entered by the user from the interfaces.

Database code:

```
import sqlite3
def create_db():
    con=sqlite3.connect(database="sms.db")
    cur=con.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS course(cid INTEGER PRIMARY
KEY AUTOINCREMENT,name text,duration text,charges text,description text)")
    con.commit()

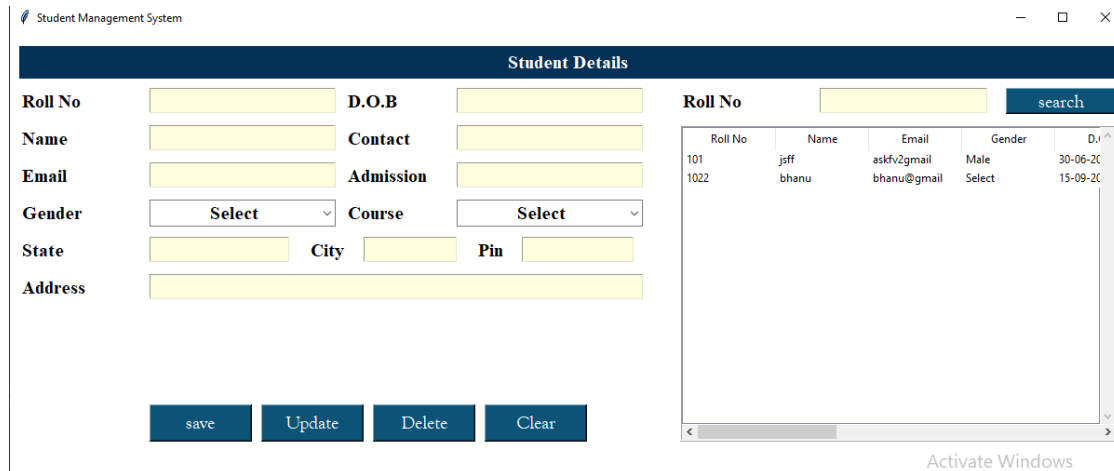
    cur.execute("CREATE TABLE IF NOT EXISTS student(roll INTEGER PRIMARY
KEY AUTOINCREMENT,name text,email text,gender text,dob text,contact
text,admission text,course text,state text,city text,pin text,address text)")
    con.commit()

    cur.execute("CREATE TABLE IF NOT EXISTS result(rid INTEGER PRIMARY
KEY AUTOINCREMENT,roll text,name text,course text,marks_ob text,full_marks
text,per text)")
    con.commit()

    con.close()

create_db()
```


Description: Here Student interface is same like course interface ,here also we enter the required details and store the details in the data base. (Fig 5.1.6)



The screenshot shows a web application titled "Student Management System". It features a "Student Details" form on the left and a table of student records on the right.

Student Details Form:

- Roll No:
- D.O.B:
- Name:
- Contact:
- Email:
- Admission:
- Gender:
- Course:
- State:
- City:
- Pin:
- Address:

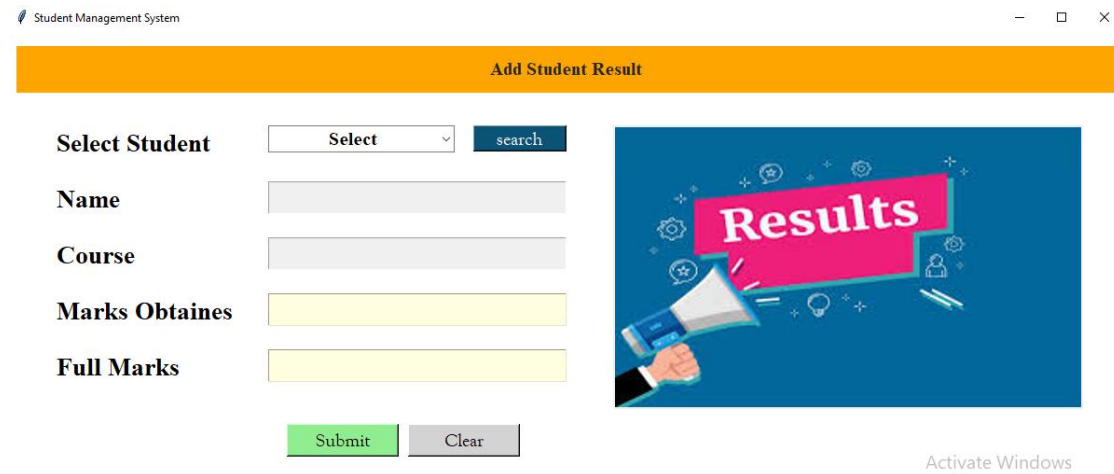
Buttons: save, Update, Delete, Clear

Student Records Table:

Roll No	Name	Email	Gender	D.O.B
101	jsff	askfv2gmail	Male	30-06-20
1022	bhanu	bhanu@gmail	Select	15-09-20

Fig 5.1.6 Student view

Description: from student interface we go to result and here we select the roll number and it displays the name and course of the selected roll no and we enter the marks here and submit. (Fig 5.1.7)



The screenshot shows a web application titled "Student Management System". It features an "Add Student Result" form on the left and a "Results" banner on the right.

Add Student Result Form:

- Select Student:
- Name:
- Course:
- Marks Obtained:
- Full Marks:

Buttons: Submit, Clear

Results Banner:

A banner with the word "Results" in a pink box, surrounded by icons of a megaphone, a lightbulb, and a pencil.

Fig 5.1.7 Student view

Description: Then from view student we can display the student report in the assigned columns. These student report details are fetched from the tables created in the sms.db database and also we can delete the report of the respective roll number. (Fig 5.1.8)

Student Management System

View Student Result

Search By Roll No

Roll No	Name	Course	Marks Obtained	Total Marks	Percentage

Activate Windows

Fig 5.1.8 Result view

Description: And also we can update, delete, clear the fields of the student.

5.2 TESTING

Testing a Student Management System (SMS) is a critical phase to ensure that the system is robust, reliable, and meets the specified requirements. This involves several types of testing, each aimed at validating different aspects of the system.

Description: In the dashboard in course we entered the details and it successfully got saved and displayed on the side screen. We also searched the details by referring the course name.

And the duration, charges, description are updated from the previous stored data. The data got deleted by pressing delete. Clear cleared all the fields on the panel.

Course Details

Course Name: Duration: Charges: Description:

Buttons:

Course ID	Name	Duration	Charges	
8	java	12days	500	awf
9	Python	6months	13456	zcasd
10	c lang	1month	1200	lejwszvg

Fig 5.2.1 Course view

Description: In the student details after entering the required details when we press save it get saved into the table and displayed on the screen. Search fetched the details of student. Update, Delete, Clear worked same as the previous interface.

Student Details

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

Buttons:

Roll No	Name	Email	Gender	D.O.B
101	jsff	askfv2gmail	Male	30-06-20
1022	bhanu	bhanu@gmail	Select	15-09-20

Fig 5.2.1 Student view

Description: In this result interface we select the roll number from the list and the name and course got displayed. Also after entering the marks percentage got calculated.

Student Management System

— □ ×

Add Student Result

Select Student

Name

Course

Marks Obtained

Full Marks



Activate Windows

Fig 5.2.3 Course view

Description: In this interface by searching the roll number we got displayed the report of the selected roll number.

Student Management System

— □ ×

View Student Result

Search By Roll No

Roll No	Name	Course	Marks Obtained	Total Marks	Percentage
101	jsff	c lang	800	1000	80.0

Activate Windows

Fig 5.2.4 Course view

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION

The conclusion of a real-time project student management system using Python would highlight the efficiency and effectiveness of the system in managing student data. The project would have demonstrated features such as comprehensive student profiles, efficient attendance tracking, academic progress monitoring, communication portals, and integrated financial management. These features would have contributed to streamlined administrative processes, reduced manual errors, and promoted informed decision-making.

6.2 FUTURE SCOPE

The future scope of such a system is vast. There are several areas for potential development and enhancement:

Artificial Intelligence (AI) Integration: The system could incorporate AI to enable predictive analytics, personalized learning paths, and automated decision-making processes. This would make the system more intelligent and proactive. Mobile

Accessibility: With the increasing reliance on mobile devices, the system could be made more mobile-friendly. This would provide on-the-go access to information for both administrators and stakeholders.

Blockchain Security: As data security becomes a top priority, the system could incorporate blockchain technology. This would ensure a secure and transparent system, safeguarding sensitive student information.

Machine Learning: The system could use machine learning algorithms to analyze student data and provide insights. This could help in identifying trends, predicting student performance, and providing personalized feedback.

REFERENCES

1. <https://pywebio.readthedocs.io/en/latest/>
2. <https://geeksforgeeks.org/python-gui-tkinter/>
3. <https://www.javatpoint.com/python-tkinter>
4. <https://www.geeksforgeeks.org/introduction-to-sqlite/>
5. <https://chatgpt.com/>