

DELHI TECHNOLOGICAL UNIVERSITY

DATA STRUCTURES LAB (CS-251)



Project Report

Topic:- Student Records Management System

Submitted by:

Nikhil Gupta (2K20/MC/086)

nikhilgupta_mc20a7_36@dtu.ac.in

Submitted to:

Dr. Goonjan Jain

Abstract

As we all know how important the Student Record management system is in school/college. The Student Record Management system offers users (Registrar) with a unified view of data from multiple sources. The main objective of this project is to build a student database system that will store records of students. It is purposed to reduce time spent on administrative tasks. The system is intended to provide better services to users, provide meaningful, consistent, and timely data and information and finally promotes efficiency by converting paper processes to electronic form. The system was developed using basic technologies namely Python and MySQL.

The system is free of errors and very efficient and less time consuming due to the care taken to develop it. All the phases of the software development cycle are employed and it is worthwhile to state that the system is user friendly and strong.

Index

S. No	Topic	Page No.
1.	Introduction	4
2.	Implementation	5
3.	Table Used	7
4.	Output And Interface's Screenshot	7
5.	Conclusion	13
6.	References	13

Introduction

The code for student management system is written in python. The main objective of using python is to provide the user a GUI based software as GUI is more user friendly. For this, I have used tkinter library of python, which is most widely and commonly used library for making such GUI based software. For maintaining database, I have used MySQL as backend technology. MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). For integrating both technology, I have used mysql.connector package of python.

While implementing the code for the Student Record Management System, a lot of concepts learned during the lectures such as functions, indentation, loops, inbuilt string functions, type conversions, etc. has been used. This helps me to get a deeper knowledge of all these concepts and also provides a hand on experience with these technologies.

Facilities Provided: -

The Student Management System provides all the basic CRUD (Create, Read, Update, Delete) operations needed for database management to the user. It ensures that there is no data loss with the help of MySQL.

Software Requirement: -

1. Python must be installed and added to path.
2. mysqlconnector and tkinter library must be installed.
3. MySQL server must be installed.

Implementation

Defining A Window: -

1. First of all, I defined a new window root using Tk() function.
2. Then I changed its properties geometry, title and configure.
3. After that I defined labels and buttons for each functionality using the inbuilt Label and Button functions.

Add Record Feature: -

1. For the add record feature, I first of all defined a new window and then take input using the Entry function.
2. After that I run the sql query and fetch all the records with matching student ids. If the program doesn't return anything, then again I run the Insert into command using the mysql.connector to add the data inside the table.
3. If the student id already exists, the program returns a messagebox showing the problem and the control moves back to main screen.

Search Record Feature: -

1. For Search record feature, I first of all defined a new window and then take student id as input from the user using the Entry function.
2. Then using the select query of MySQL, I fetch out all the records with matching id.
3. If the length of returned query is zero, it means no such record exists, the program returns a messagebox showing the problem and the control moves back to main screen.
4. Otherwise, the program displays the student's data using Entry and Label functions.

Modify Record Feature: -

1. For Modify record feature, I first of all defined a new window and then take student id as input from the user using the Entry function.
2. Then using the select query of MySQL, I fetch out all the records with matching id.
3. If the length of returned query is zero, it means no such record exists, the program returns a messagebox showing the problem and the control moves back to main screen.
4. Otherwise, the program displays the student's data using Entry and Label functions.

5. After that, you can modify the data other than id and click on modify to modify the data which will update the data using the update query of MySQL.

Delete Record Feature: -

1. For Delete record feature, I first of all defined a new window and then take student id as input from the user using the Entry function.
2. Then using the select query of MySQL, I fetch out all the records with matching id.
3. If the length of returned query is zero, it means no such record exists, the program returns a messagebox showing the problem and the control moves back to main screen.
4. Otherwise, the program displays the student's data using Entry and Label functions, then you can click on delete button to delete which uses the delete command to delete the data from the table.

View All Records Feature: -

1. For View All records feature, I first of all defined a new window.
2. After that using the treeview option, I will display the data.
3. I fetch all the data by using the select command and insert it one by one using the treeview.

Exit/ Back Feature: -

1. For Exit/ Back button, I have used the .destroy() function to destroy the window.

Table Used

Database Name:- student_data

Table Name: - Data

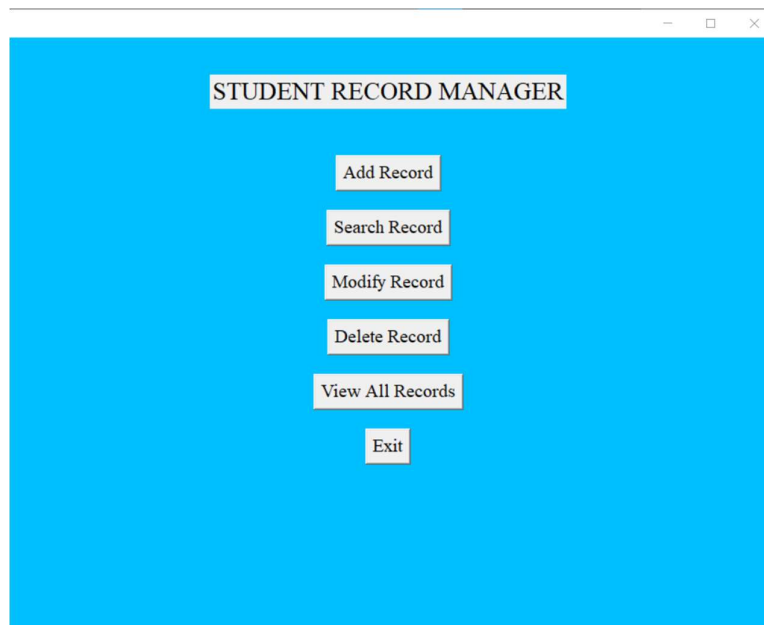
```
mysql> desc data;
```

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
name	varchar(50)	YES		NULL	
age	int	YES		NULL	
rollNo	int	YES		NULL	
branch	varchar(50)	YES		NULL	

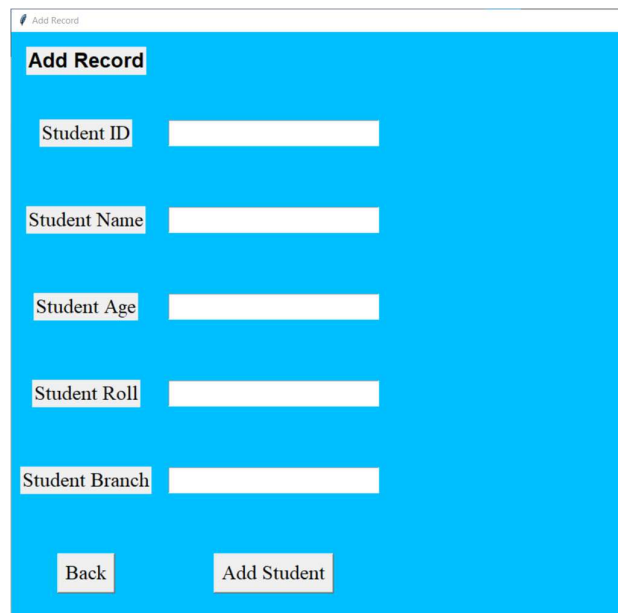
5 rows in set (0.01 sec)

Output and Interface's Screenshot

1. Main Window.

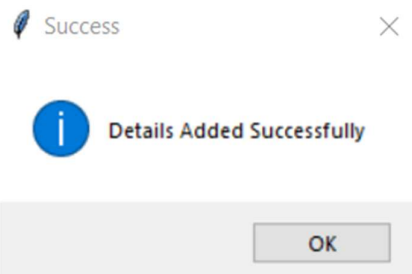


2. Add Record Option

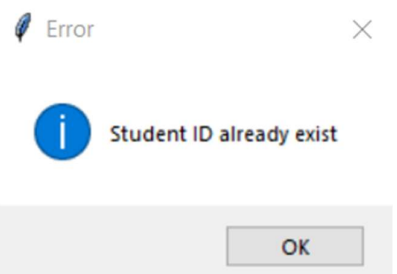


The screenshot shows a web application window titled 'Add Record'. The window has a blue header bar with the title 'Add Record'. Below the header, there are five input fields, each with a label to its left: 'Student ID', 'Student Name', 'Student Age', 'Student Roll', and 'Student Branch'. At the bottom of the form, there are two buttons: 'Back' and 'Add Student'.

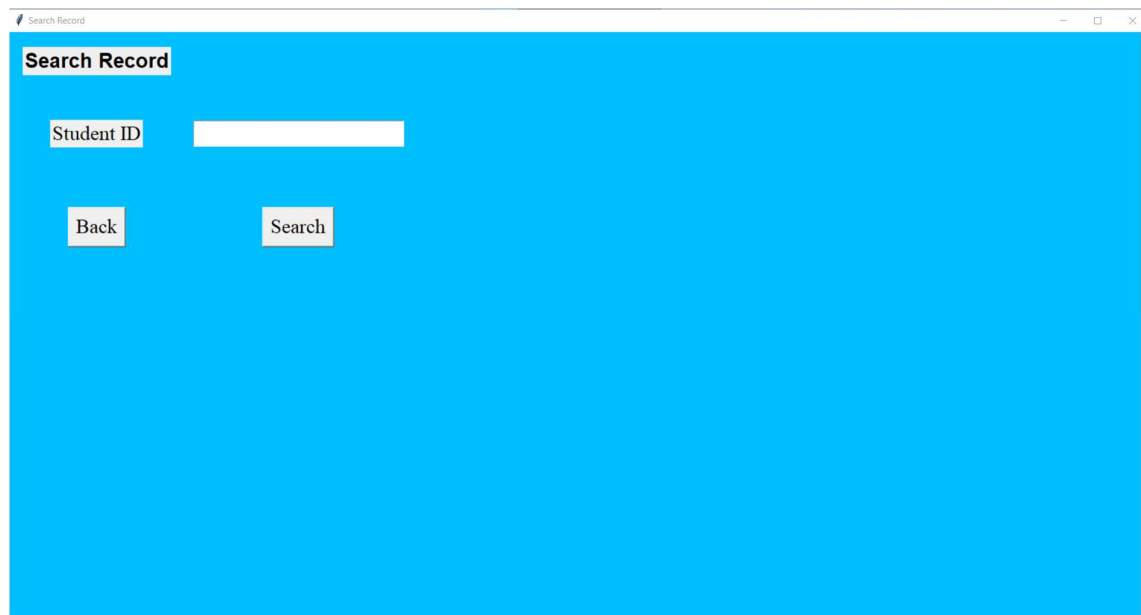
- If unique student id is entered, it shows success message



- If the entered student id is not unique, it shows error.

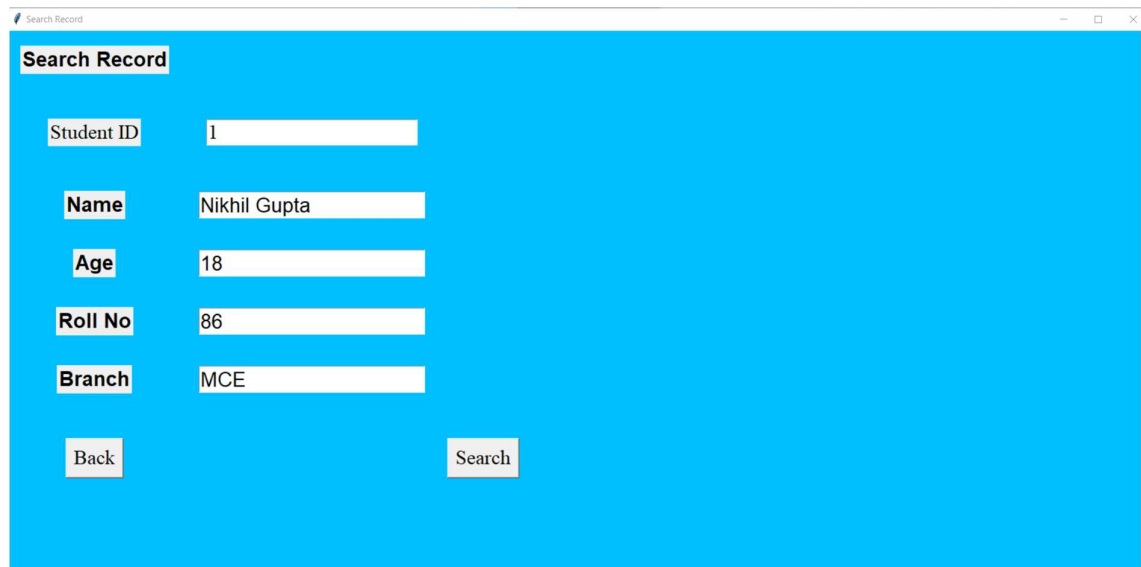


3. Search Record



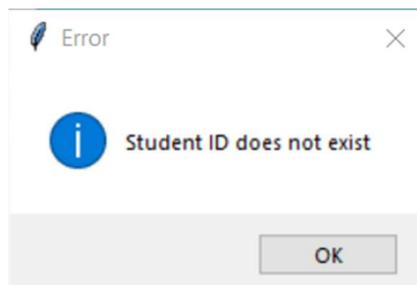
The screenshot shows a web application window titled "Search Record". The window has a blue background. At the top left, there is a title bar with a feather icon and the text "Search Record". Below the title bar, the text "Search Record" is displayed in a bold, black font. Underneath, there is a form with a label "Student ID" and an empty text input field. Below the input field, there are two buttons: "Back" and "Search".

- If entered ID is present in the data, it shows the data.

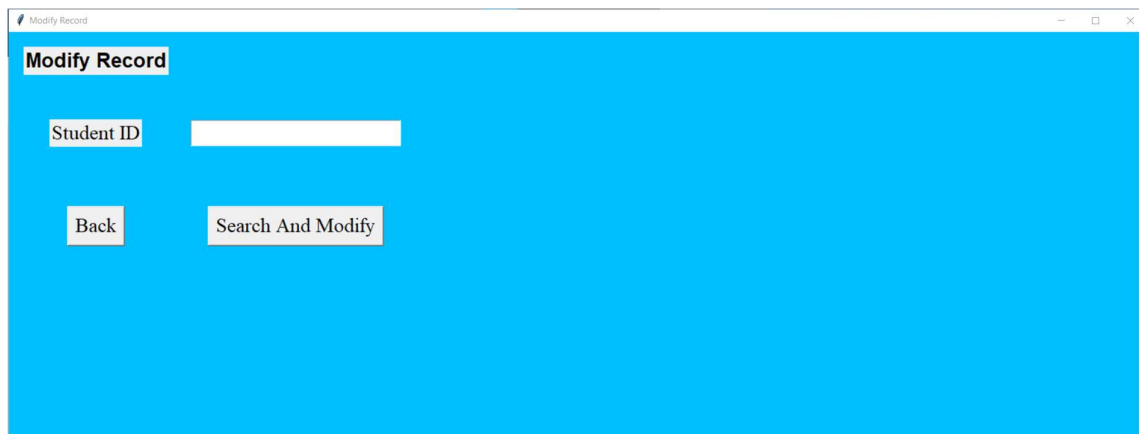


The screenshot shows the same "Search Record" application window, but now it displays search results. The "Student ID" input field contains the value "1". Below it, the "Name" field contains "Nikhil Gupta", the "Age" field contains "18", the "Roll No" field contains "86", and the "Branch" field contains "MCE". The "Back" and "Search" buttons are still present at the bottom.

- If the entered ID is not present in data, it shows error.



4. Modify Record



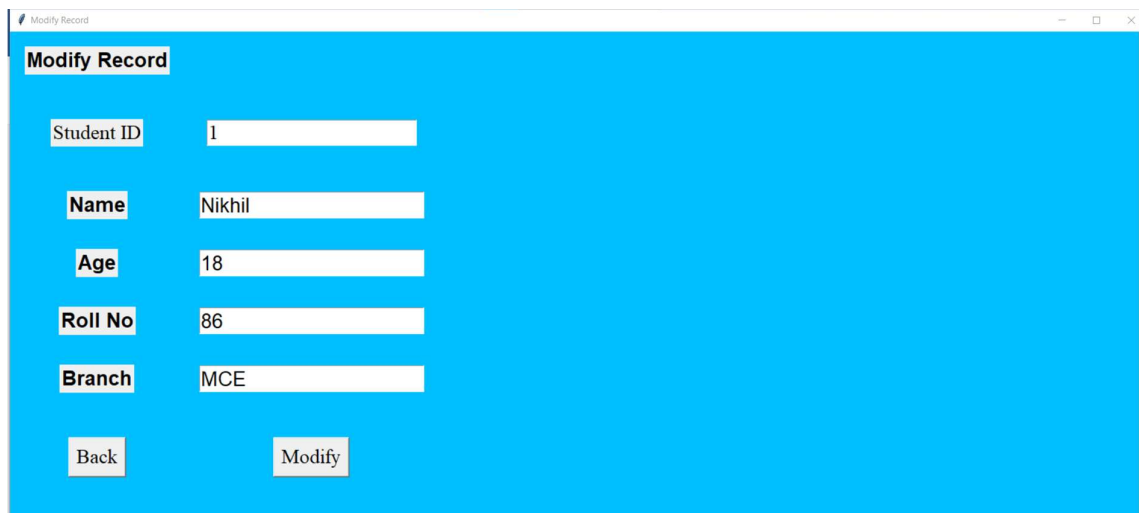
The screenshot shows a web application window titled "Modify Record". The window has a blue background. At the top left, there is a header "Modify Record". Below the header, there is a form with a label "Student ID" and an empty text input field. Below the input field, there are two buttons: "Back" and "Search And Modify".

If the entered ID exist, it shows details

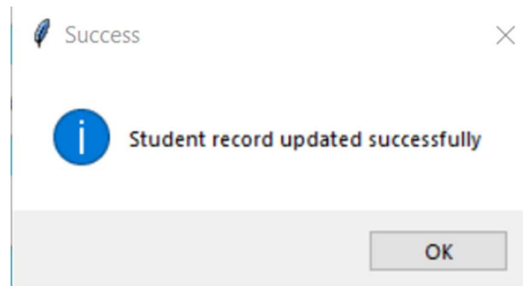


The screenshot shows the "Modify Record" window with the "Student ID" field containing the value "1". Below the input field, the details of the student are displayed in a form with labels and text input fields: "Name" (Nikhil Gupta), "Age" (18), "Roll No" (86), and "Branch" (MCE). At the bottom, there are two buttons: "Back" and "Modify".

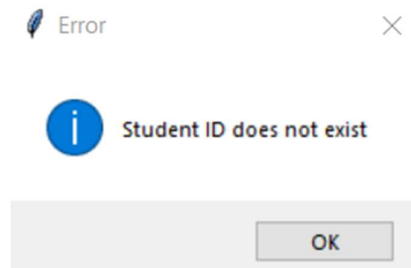
After updating the details,



The screenshot shows the "Modify Record" window with the "Student ID" field containing the value "1". Below the input field, the details of the student are displayed in a form with labels and text input fields: "Name" (Nikhil), "Age" (18), "Roll No" (86), and "Branch" (MCE). At the bottom, there are two buttons: "Back" and "Modify".



If the ID doesn't exist.

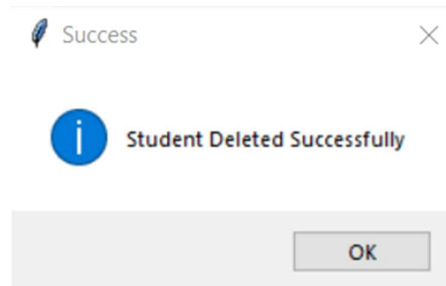


5. Delete Record

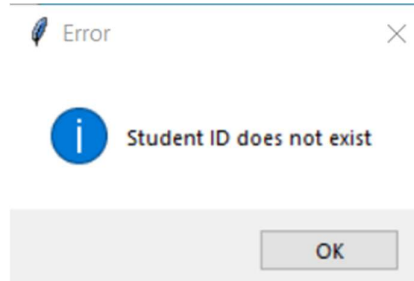
A screenshot of a web application window titled 'Delete Record'. The background is a solid bright blue. At the top left, the title 'Delete Record' is in white. Below it, there is a white input field labeled 'Student ID'. Underneath the input field are two white buttons: 'Back' and 'Search And Delete'.

If the entered ID exists, it shows details and provide option for deletion.

A screenshot of the same 'Delete Record' web application window. The background is blue. The 'Student ID' input field now contains the number '2'. Below it, several student details are displayed in white text on the blue background: 'Name' with the value 'Naman', 'Age' with '19', 'Roll No' with '85', and 'Branch' with 'MCE'. At the bottom, there are two white buttons: 'Back' and 'Delete'.



If the entered ID doesn't exist.



6. View All Records

This will show all the records of students.

The "View All Records" dialog box has a blue header bar with a feather icon, the text "View All Records", and a close button (X). Below the header is a table with the following data:

	ID	Name	Age	Roll No	Branch
1		Nikhil	18	86	MCE
3		Pradeep Sharma	20	98	MCE

Below the table is a grey button labeled "Back".

Conclusion

The code for the Student Record Management System has been implemented successfully. A lot of concepts learned during the lectures and classes such as functions, indentation, loops, inbuilt string functions, type conversions, etc. has been used. This helps me to get a deeper knowledge of all these concepts and also provides a hand on experience with these technologies. The most important thing that I learned during the course of making of this project is error handling as during the implementation, I faced a lot of error and exceptions. I learned more about the nature of MySQL queries, implementation of same functionality using different logics and also learn how to write clean code.

References

1. [Programming With Mosh](#)
2. [Codemy.com](#)
3. [freeCodeCamp.org](#)
4. [Python Documentation](#)
5. [Stack Overflow](#)