Contents

[Student Database Management System 2](#_Toc155091187)

[Introduction 2](#_Toc155091188)

[File Structure 2](#_Toc155091189)

[Dependencies 2](#_Toc155091190)

[Classes and Functions 2](#_Toc155091191)

[1. DBHelper Class 2](#_Toc155091192)

[2. Login Class 3](#_Toc155091193)

[3. AddStudent Class 3](#_Toc155091194)

[4. Window Class 4](#_Toc155091195)

[5. Main Block (if \_\_name\_\_ == '\_\_main\_\_':) 5](#_Toc155091196)

[How to Run 5](#_Toc155091197)

# Student Database Management System

## Introduction

This project is a Student Database Management System (SDMS) implemented in Python using the PyQt5 library for the graphical user interface (GUI) and SQLite for database management. The application allows users to add, search, delete, and export student records.

## File Structure

* main.py: The main script to run the application.
* sdms.db: SQLite database file to store student, faculty, and course details.
* user.png: Placeholder image for student profile pictures.

## Dependencies

* Python 3.x
* PyQt5
* SQLite3
* Openpyxl (for Excel export)

## Classes and Functions

### 1. DBHelper Class

\_\_init\_\_(self)

* Initializes the DBHelper class and establishes a connection to the SQLite database.
* Creates three tables in the database (student, faculty, and Course) if they don't exist.

addStudent(self, sid, Sname, dept, year, course\_a, course\_b, course\_c)

* Adds a new student to the student table with the provided details.
* Shows a success message upon successful addition; otherwise, shows an error message.

searchStudent(self, sid)

* Searches for a student with the given roll number (sid) in the student table.
* If found, retrieves the student's details and calls the showStudent function.
* Shows an error message if the student is not found.

deleteRecord(self, sid)

* Deletes a student record with the provided roll number (sid) from the student table.
* Shows a success message upon successful deletion; otherwise, shows an error message.

exportToCSV(self, filename='student\_records.csv')

* Exports all student records from the student table to a CSV file.
* Shows a success message upon successful export; otherwise, shows an error message.

exportToExcel(self, filename='student\_records.xlsx')

* Exports all student records from the student table to an Excel file.
* Shows a success message upon successful export; otherwise, shows an error message.

### 2. Login Class

\_\_init\_\_(self, parent=None)

* Initializes the Login class, creating a simple login dialog with a username, password, and login button.
* Connects the login button to the handleLogin function.

handleLogin(self)

* Checks if the entered username and password are empty and accepts the login if both are empty.
* Shows a warning if the login credentials are not empty.

### 3. AddStudent Class

\_\_init\_\_(self)

* Initializes the AddStudent class, creating a dialog for adding student details.
* Sets up UI elements such as labels, combo boxes, and buttons.
* Connects buttons to corresponding functions (addStudent, Close, and reset).

Close(self)

* Closes the dialog.

reset(self)

* Clears the input fields.

addStudent(self)

* Retrieves student details from input fields and combo boxes.
* Calls the addStudent function from DBHelper to add the student to the database.

### 4. Window Class

\_\_init\_\_(self)

* Initializes the main application window with buttons for entering, showing, and deleting student details.
* Connects buttons to corresponding functions (enterstudent, showStudentDialog, showDeleteDialog, and exportRecords).

exportRecords(self)

* Opens a file dialog for choosing the export file format (CSV or Excel).
* Calls the exportToCSV or exportToExcel function from DBHelper based on the chosen format.

exportToCSV(self, filename)

* Calls the exportToCSV function from DBHelper with the chosen filename.

exportToExcel(self, filename)

* Calls the exportToExcel function from DBHelper with the chosen filename.

enterstudent(self)

* Opens the AddStudent dialog to add a new student.

showStudentDialog(self)

* Opens a dialog to enter the roll number and view student details.

showDeleteDialog(self)

* Opens a dialog to enter the roll number and delete a student record.

showStudent(self)

* Retrieves the roll number from the input field and calls the searchStudent function from DBHelper to show student details.

deleteRecord(self)

* Retrieves the roll number from the input field and calls the deleteRecord function from DBHelper to delete a student record.

### 5. Main Block (if \_\_name\_\_ == '\_\_main\_\_':)

* Creates an instance of the QApplication.
* Opens a login dialog, and if login is successful, opens the main window (Window class).
* Starts the application event loop.

## How to Run

1. Ensure that Python and the required dependencies are installed.
2. Run the main.py script.
3. A login dialog will appear; enter credentials or leave them empty to proceed.
4. The main window will open with options to add, search, delete, and export student records.