

Student Management System

A Console-Based CRUD Application in Python



Lakshita Pareek

Introduction

- A software application designed to help educational institutions manage student data.
- Functions include tracking student enrollment, grades, attendance, and personal information.

Importance of Student Management System :

- Facilitates better organization and access to student records.
- Reduces administrative workload and errors.

Purpose

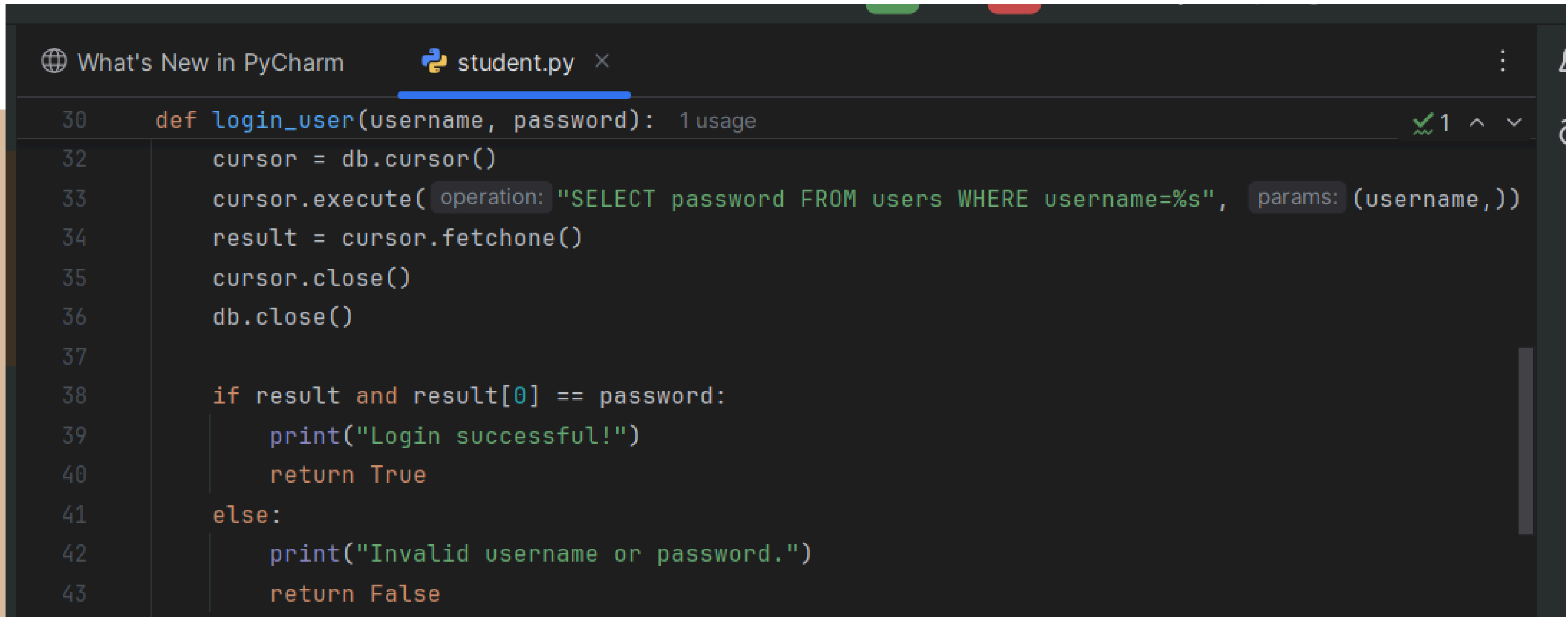
Goals of Developing the System:

- Efficiency: Streamline processes such as registration, record-keeping, and reporting.
- Centralization: Provide a single platform for all student-related data.
- Accessibility: Make it easy for authorized users to access and manage data.

Code Overview

- **Database Connection:** Establishes a connection to a MySQL database.
- **User Management:** Functions for registering and logging in users.
- **Student Management:** CRUD operations for managing student records.

Code Explanation - User Login

A screenshot of the PyCharm IDE interface. The top bar shows a tab for 'student.py' with a Python icon and a close button. Below the tab, the code editor displays a Python function 'login_user' with its implementation. The code is syntax-highlighted, with keywords in orange, strings in green, and identifiers in white. Line numbers 30 through 43 are visible on the left margin. The function logic involves connecting to a database, querying for a password by username, and comparing it to the provided password.

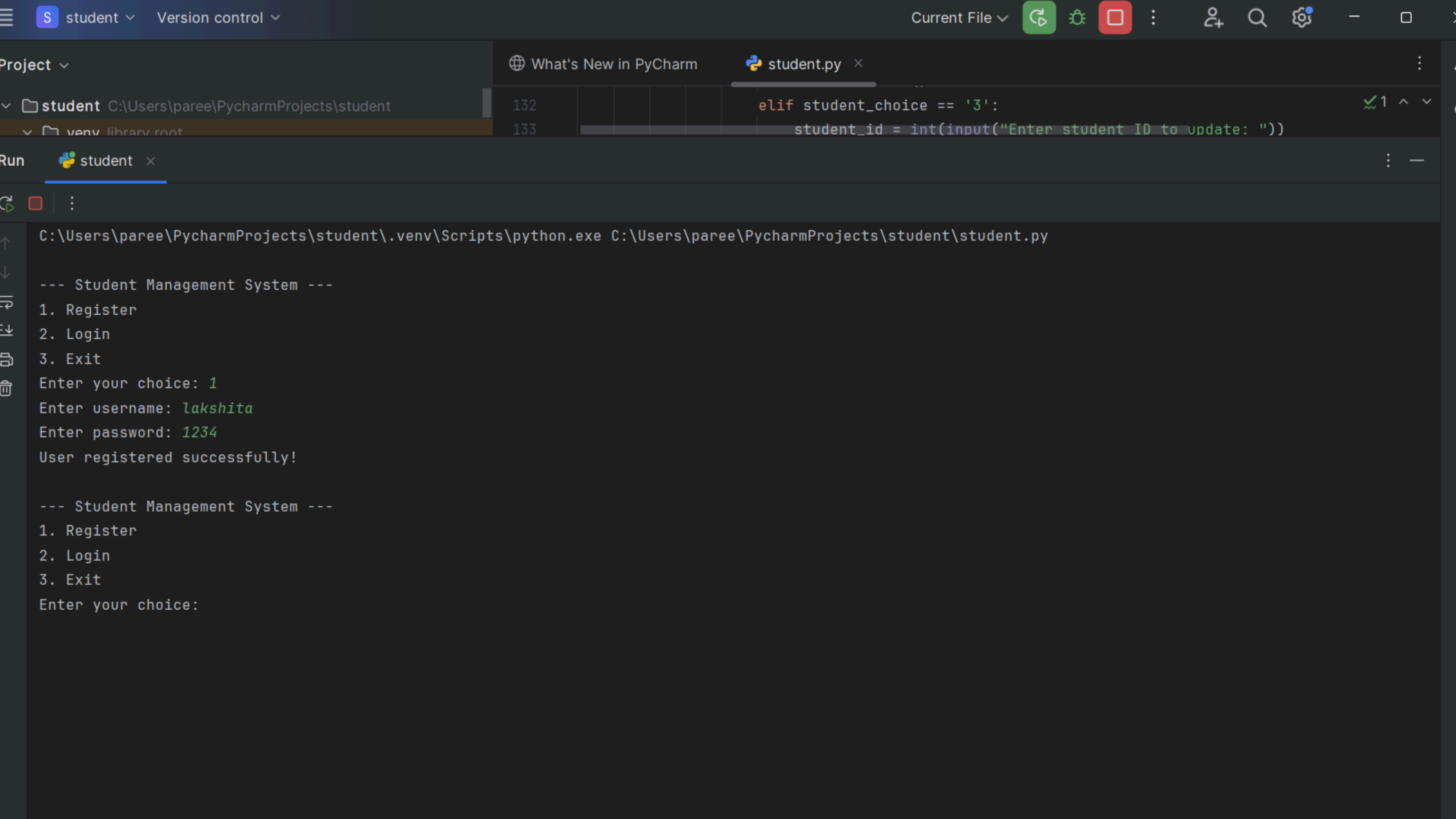
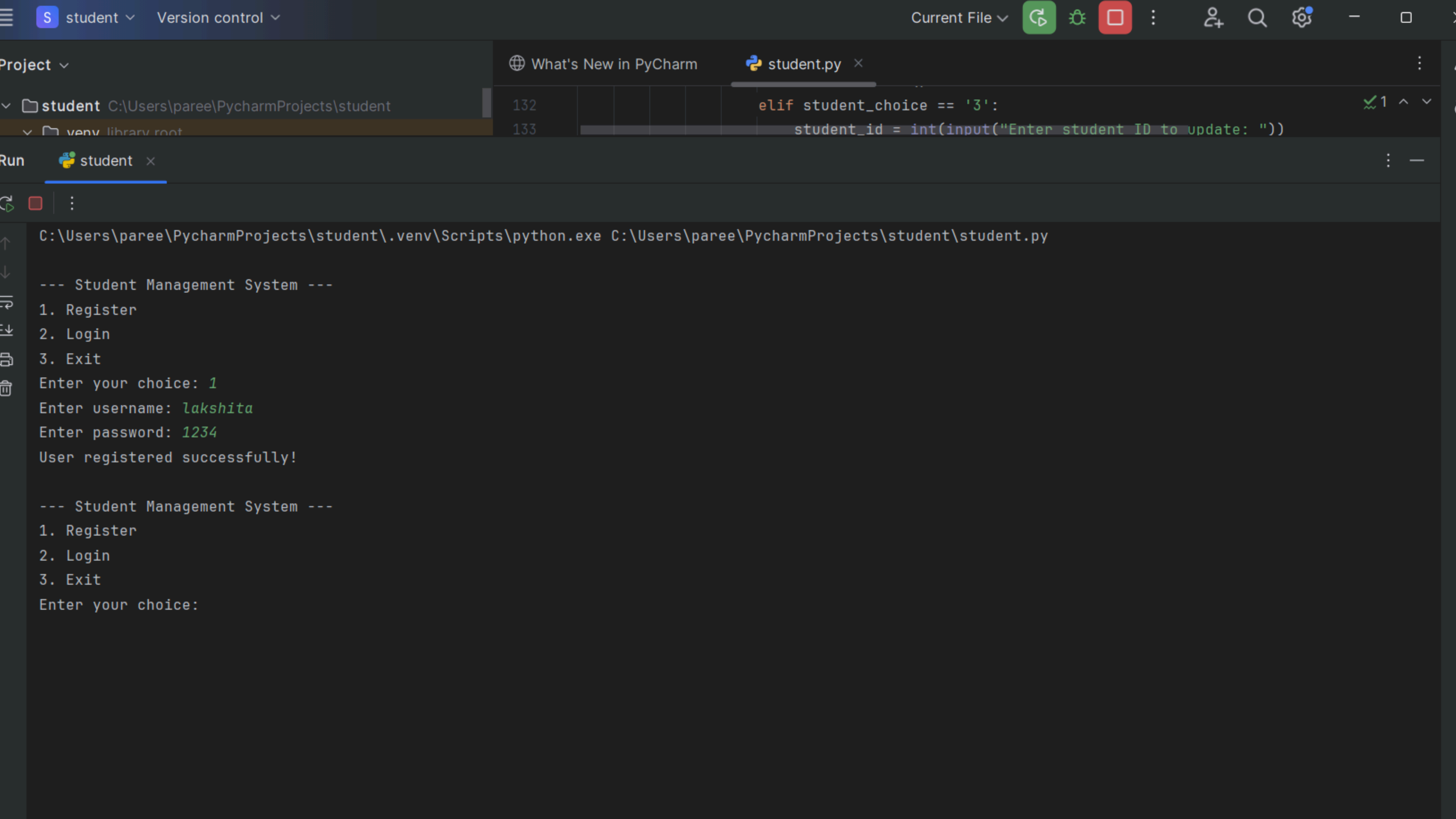
```
30 def login_user(username, password): 1 usage
31
32     cursor = db.cursor()
33     cursor.execute(operation: "SELECT password FROM users WHERE username=%s", params: (username,))
34     result = cursor.fetchone()
35     cursor.close()
36     db.close()
37
38     if result and result[0] == password:
39         print("Login successful!")
40         return True
41     else:
42         print("Invalid username or password.")
43         return False
```

Functionality:

- Checks user credentials against the database.
- Fetches the stored password for the given username and compares it.

Code Explanation - CRUD Operations for Students

- Adding a Student: Inserts a new student record into the students table.
- Viewing Students: Retrieves and displays all student records in a readable format.
- Updating a Student: Updates an existing student's information based on the student ID.
- Deleting a Student: Removes a student record from the database.



Project

student C:\Users\paree\PycharmProjects\student

venv library root

Run student

What's New in PyCharm

student.py

132 elif student_choice == '3':

133 student_id = int(input("Enter student ID to update: "))

Enter name: lakshita

Enter age: 21

Enter grade: 86

Student added successfully!

--- Student Management ---

1. Add Student

2. View Students

3. Update Student

4. Delete Student

5. Logout

Enter your choice: 1

Enter name: aayushi

Enter age: 22

Enter grade: 91

Student added successfully!

--- Student Management ---

1. Add Student

2. View Students

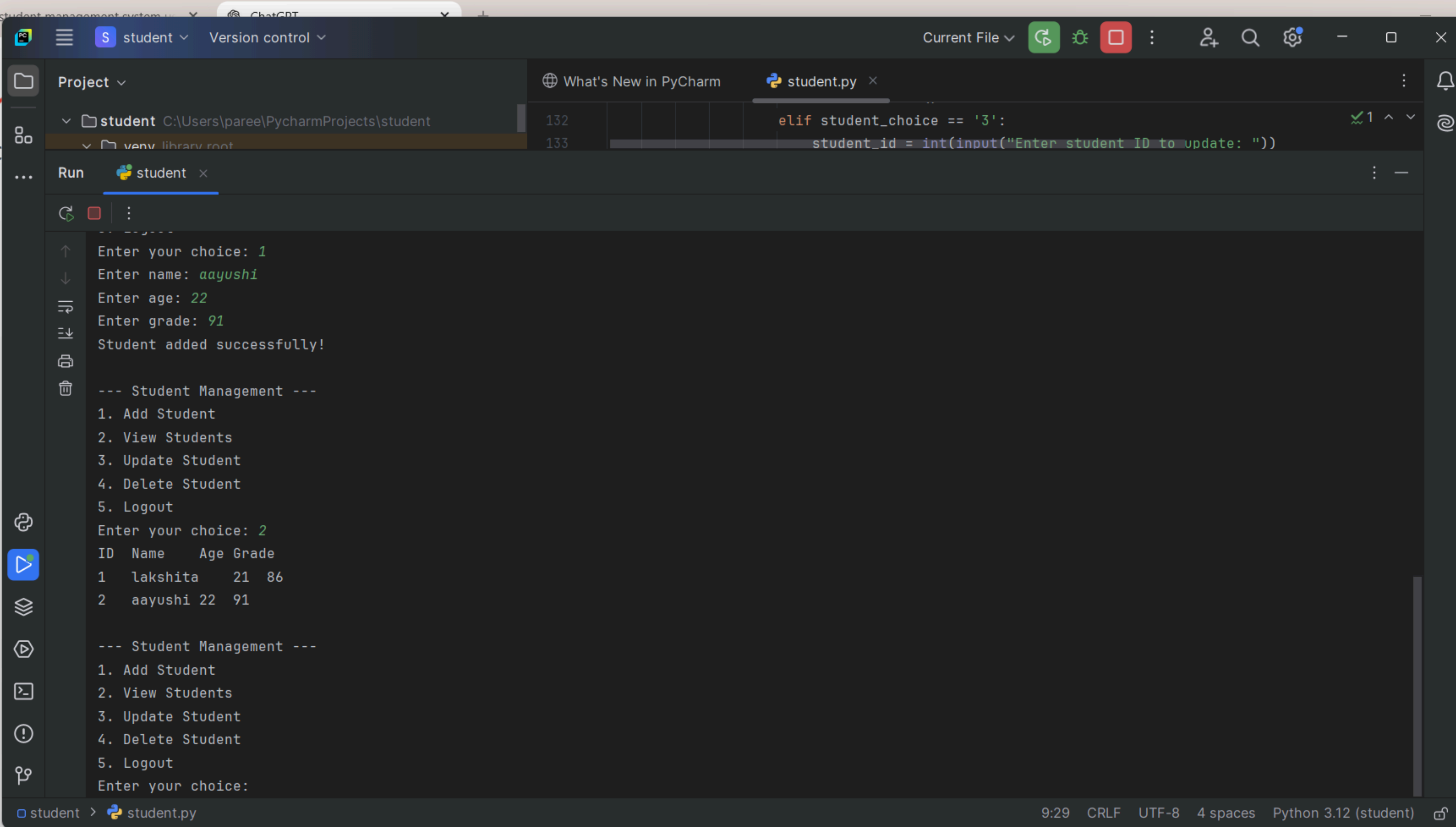
3. Update Student

4. Delete Student

5. Logout

Enter your choice:

student > student.py 9:29 CRLF UTF-8 4 spaces Python 3.12 (student)



PC

☰

S student ▾

Version control ▾

Current File ▾

⋮

—

□

×

Project ▾

What's New in PyCharm

student.py ×

⋮

🔔

MySQL Workbench

— □ ×

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

▼ sys

▶ Tables

▶ Views

▶ Stored Procedures

▶ Functions

Query 1 ×

SQLAdditions

Limit to 10

◀ ▶

Jump to ▾

12 grade VARCHAR(10)

13);

14 • SHOW TABLES ;

15 • SELECT * FROM students;

16 • SELECT * FROM users;

17

Result Grid

Filter Rows:

	id	username	password
▶	1	lakshita	1234
•	NULL	NULL	NULL

Result Grid

users 3 ×

Apply

Context Help

Snippets

Output

Action Output ▾

	#	Time	Action	Message	Duration / Fetch
✓	1	14:17:21	CREATE DATABASE student_management	1 row(s) affected	0.000 sec
✓	2	14:17:27	USE student_management	0 row(s) affected	0.000 sec
✓	3	14:18:09	USE student_management	0 row(s) affected	0.000 sec
✓	4	14:18:28	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, ...	0 row(s) affected	0.046 sec
✓	5	14:19:00	CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY,...	0 row(s) affected	0.047 sec
✗	6	14:25:37	SHOW TABLE students	Error Code: 1064. You have an error in your SQL syntax; check the manual...	0.016 sec
✓	7	14:27:12	SHOW TABLES	2 row(s) returned	0.000 sec / 0.000 sec
✓	8	14:28:29	USE student_management	0 row(s) affected	0.000 sec
✓	9	14:28:54	SELECT * FROM students LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Administration Schemas

Information

Schema: sys

1

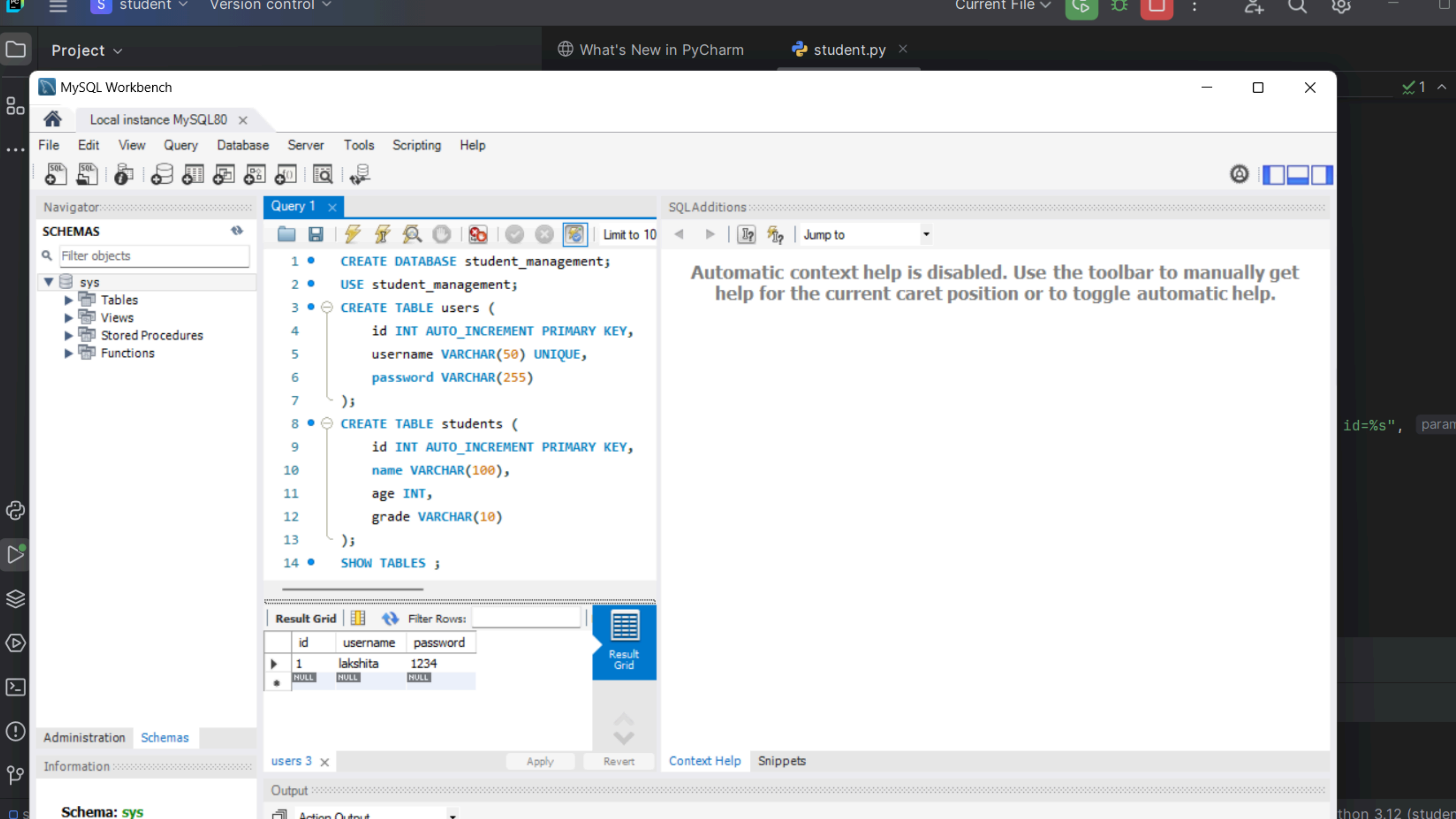
^ ▾

🔗

id=%s", params:

thon 3.12 (student)

🔗



Thank You

