VS CODE

'''

ADMIN FEATURES:

1.ADD STUDENT

2.DELETE STUDENT

3.UPDATE STUDENT

4.TIME TABLE

5.MARKS

6.ATTENDANCE

'''

from db import connect

import getpass

import hashlib

def login():

username = input("Enter username: ")

password = getpass.getpass("Enter password: ")

hashed\_password = hashlib.sha256(password.encode()).hexdigest()

con = connect()

cursor = con.cursor()

query = "SELECT \* FROM admins WHERE username = %s AND password = %s"

values = (username, hashed\_password)

cursor.execute(query, values)

admin = cursor.fetchone()

if admin:

print("Login successful!")

admin\_menu()

else:

print("Invalid username or password. Please try again.")

login()

def admin\_menu():

con = connect()

cursor = con.cursor()

print("""\nAdmin menu:

1.Add student

2.Update student

3.Reset student password

4.Update marks

5.Add marks

6.View student marks

7.View All students

8.Update Timetable

9.Add attendance

10.Update attendance

11.Logout""")

ch = int(input("Enter your choice: "))

if ch == 1:

add\_student()

elif ch == 2:

update\_student()

elif ch == 3:

reset\_student\_password()

elif ch == 4:

update\_marks()

elif ch == 5:

add\_marks()

elif ch == 6:

view\_student\_marks()

elif ch == 7:

view\_all\_students()

elif ch == 8:

update\_timetable()

elif ch == 9:

add\_attendance()

elif ch == 10:

update\_attendance()

elif ch == 11:

logout()

login()

else:

print("Invalid choice. Please try again.")

admin\_menu()

def add\_student():

con = connect()

cursor = con.cursor()

roll\_no = input("Enter Roll No: ")

name = input("Enter Name: ")

class\_name = input("Enter class: ")

section = input("Enter section: ")

password = "password@123"

email = input("Enter Email: ")

query = "INSERT INTO students(roll\_no, name, class, section, password, email) values(%s, %s, %s, %s, %s, %s)"

values = (roll\_no, name, class\_name, section, password, email)

cursor.execute(query, values)

con.commit()

print("Student added successfully.")

admin\_menu()

def update\_student():

con = connect()

cursor = con.cursor()

roll\_no = input("Enter Roll No of student to update: ")

name = input("Enter new Name: ")

class\_name = input("Enter new class: ")

section = input("Enter new section: ")

email = input("Enter new Email: ")

query = "UPDATE students SET name=%s, class=%s, section=%s, email=%s where roll\_no=%s"

values = (name, class\_name, section, email, roll\_no)

cursor.execute(query, values)

con.commit()

print("Student details updated successfully.")

admin\_menu()

def reset\_student\_password():

pass

def update\_marks():

con = connect()

cursor = con.cursor()

roll\_no = input("Enter Roll No of student to update marks: ")

subject = input("Enter subject: ")

marks = input("Enter Marks: ")

query = "UPDATE marks SET marks=%s WHERE roll\_no=%s AND subject=%s"

values = (marks, roll\_no, subject)

cursor.execute(query, values)

con.commit()

print("Marks updated successfully.")

admin\_menu()

def add\_marks():

con = connect()

cursor = con.cursor()

roll\_no = input("Enter Roll No of student to add marks: ")

subject = input("Enter subject: ")

marks = input("Enter Marks: ")

query = "insert into marks(roll\_no, subject, marks) values(%s, %s, %s)"

values = (roll\_no, subject, marks)

cursor.execute(query, values)

con.commit()

print("Marks added successfully.")

admin\_menu()

def view\_student\_marks():

con = connect()

cursor = con.cursor()

roll\_no = input("Enter Roll No of student to view student marks: ")

query = "SELECT \* from marks WHERE roll\_no = %s"

cursor.execute(query, (roll\_no,))

results = cursor.fetchall()

for row in results:

print(row)

admin\_menu()

def view\_all\_students():

con = connect()

cursor = con.cursor()

query = "SELECT \* FROM students"

cursor.execute(query)

results = cursor.fetchall()

for row in results:

print(row)

admin\_menu()

def update\_timetable():

pass

def add\_attendance():

con = connect()

cursor = con.cursor()

roll\_no = int(input("Enter roll number: "))

date = input("Enter date (YYYY-MM-DD): ")

status = input("Enter attendance status (Present/Absent): ")

query = "INSERT INTO attendance (roll\_no, date, status) VALUES (%s, %s, %s)"

values = (roll\_no, date, status)

try:

cursor.execute(query, values)

con.commit()

print("Attendance added successfully.")

except Exception as e:

con.rollback()

print("Error adding attendance:", str(e))

finally:

con.close()

admin\_menu()

def update\_attendance():

con = connect()

cursor = con.cursor()

roll\_no = int(input("Enter roll number: "))

date = input("Enter date (YYYY-MM-DD): ")

status = input("Enter new attendance status (Present/Absent): ").capitalize()

while status not in ['Present', 'Absent']:

status = input("Invalid status. Enter attendance status (Present/Absent): ").capitalize()

query = "SELECT \* FROM attendance WHERE roll\_no = %s AND date = %s"

cursor.execute(query, (roll\_no, date))

attendance\_record = cursor.fetchone()

if attendance\_record is None:

print("No attendance record found for the given roll number and date. Would you like to add attendance? (yes/no)")

choice = input().lower()

if choice == "yes":

add\_attendance()

else:

print("Attendance not updated.")

admin\_menu()

query = "UPDATE attendance SET status = %s WHERE roll\_no = %s AND date = %s"

values = (status, roll\_no, date)

try:

cursor.execute(query, values)

con.commit()

print("Attendance updated successfully.")

except Exception as e:

con.rollback()

print("Error updating attendance:", str(e))

finally:

con.close()

admin\_menu()

def logout():

print("Logging out...")

if \_name\_ == "\_main\_":

# Create admin account if not exists

con = connect()

cursor = con.cursor()

query = "CREATE TABLE IF NOT EXISTS admins (username VARCHAR(255), password VARCHAR(255))"

cursor.execute(query)

con.commit()

# Insert default admin credentials

query = "SELECT \* FROM admins"

cursor.execute(query)

admin = cursor.fetchone()

if not admin:

username = "admin"

password = hashlib.sha256("password".encode()).hexdigest()

query = "INSERT INTO admins (username, password) VALUES (%s, %s)"

values = (username, password)

cursor.execute(query, values)

con.commit()

login()