## Secure User Profile Management System (SUPMS) With Python Source Code .py

# Database connection details  
import mysql.connector  
  
conn\_obj = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="67808033",  
 database="Python\_Project")  
cur\_obj = conn\_obj.cursor()  
  
# create this function when you design password in registration part  
import re  
  
#1  
def strong\_password(password):  
 if (len(password) >= 6 and  
 re.search(r'[A-Z]', password) and  
 re.search(r'[a-z]', password) and  
 re.search(r'[0-9]', password) and  
 re.search(r'[\W\_]', password)):  
 return True  
 return False  
  
#2  
def strong\_userid(user\_id):  
 if (len(user\_id) >= 3 and  
 re.search(r'[A-Z]', user\_id) and  
 re.search(r'[0-9]', user\_id)):  
 return True  
 return False  
  
#3  
# Define function data\_entry\_sql  
def data\_entry\_sql(full\_name, address, phone\_number, user\_id, password):  
 # Build the query with user-provided name using LIKE operator  
 sql = "INSERT INTO cust\_details (full\_name,address,phone\_number,user\_id,password) VALUES (%s, %s, %s, %s, %s)"  
 data = (full\_name, address, phone\_number, user\_id, password)  
  
 try:  
 cur\_obj.execute(sql, data)  
 print(full\_name, ", Now You are Successfully Registered")  
 conn\_obj.commit()  
 except mysql.connector.Error as e:  
 print("Error retrieving data from MySQL:", e)  
 conn\_obj.rollback()  
  
#4  
# Define function data\_retrieve  
def data\_retrieve(user\_id):  
 # Build the query with user-provided name using LIKE operator  
 # select \* from students\_details WHERE Roll\_no=1;  
 query = f"select \* from cust\_details where user\_id=\'{user\_id}\'"  
  
 try:  
 cur\_obj.execute(query)  
 result = cur\_obj.fetchone()  
 conn\_obj.commit()  
 except mysql.connector.Error as e:  
 print("Error retrieving data from MySQL:", e)  
 conn\_obj.rollback()  
 return result  
  
  
  
#5  
def login():  
 user\_id = input("Enter your user id - ") #if you want you can keep user\_id validation by strong\_userid()  
 password = input("Enter your password - ")  
 user\_details = data\_retrieve(user\_id)  
 if user\_details:  
 actuall\_password = user\_details[-1]  
 if actuall\_password == password:  
 print("log in successfull")  
 else:  
 print("Wrong entry , log in failed")  
  
 else:  
 print("you are not registered, Please Register Yourself Below")  
 cust\_registration()  
 print("Now Log In")  
 login()  
  
  
  
#6  
def cust\_registration():  
 full\_name = input("Enter your full name - ").upper()  
 address = input("Enter your address - ")  
  
 while True:  
 phonr\_number = input("Enter your phone number - ")  
 if phonr\_number.isdigit() and len(phonr\_number) == 10:  
 break  
 else:  
 print("wrong entry")  
  
 cur\_obj.execute("SELECT \* FROM cust\_details WHERE phone\_number = %s", (phonr\_number,))  
 if cur\_obj.fetchone():  
 print("already registered .. Please Log in")  
 return # single return uporer function ke aar egote na dewar jonno use hoy  
  
 while True: # eta different  
 user\_id = input("Make your user id - ")  
 if not strong\_userid(user\_id):  
 print("Username must include uppercase and digit.")  
 continue  
  
 cur\_obj.execute("SELECT \* FROM cust\_details WHERE user\_id = %s", (user\_id,))  
 if cur\_obj.fetchone():  
 print("This user ID is already used. Please try something unique.")  
 else:  
 break  
  
 while True:  
 password = input("Make the password -")  
 if strong\_password(password):  
 break  
 else:  
 print("Password must include uppercase, lowercase, digit, and special character.")  
  
 data\_entry\_sql(full\_name, address, phonr\_number, user\_id, password)  
 print("Now Log In")  
 login()  
  
  
  
#7  
def Data\_Retrieve(phone\_number, cust\_id):  
 # Build the query with user-provided name using LIKE operator  
 # select \* from students\_details WHERE Roll\_no=1;  
  
 # Query = f"select \* from cust\_details where user\_id=\'{user\_id}\'"  
 # Query = f"select \* from cust\_details where phone\_number=\'{phone\_number}\' and cust\_id=\'{cust\_id}\'"  
 Query = f"select \* from cust\_details where phone\_number={phone\_number} and cust\_id={cust\_id}"  
  
 try:  
 cur\_obj.execute(Query)  
 Result = cur\_obj.fetchone()  
 conn\_obj.commit()  
 except mysql.connector.Error as e:  
 print("Error retrieving data from MySQL:", e)  
 conn\_obj.rollback()  
 if Result:  
 cust\_id, full\_name, address, phone\_number, user\_id, password = Result  
 print("Customer ID - ", cust\_id)  
 print("Customer Name - ", full\_name)  
 print("Customer Address - ", address)  
 print("Customer Contact Number - ", phone\_number)  
 else:  
 print("No data found")  
  
#8  
def profile\_view() :  
 while True:  
 # user\_id = input("Enter Your user ID - ")  
 phone\_number = input("enter your phone number - ")  
  
 # if strong\_userid(user\_id) :  
 if phone\_number.isdigit() and len(phone\_number) == 10:  
 break  
 else:  
 print("WRONG ENTRY")  
  
 while True:  
  
 cust\_id = input("Enter your customer id - ")  
 if cust\_id.isdigit():  
 break  
 else:  
 print("wrong format, customer id should be in digit")  
  
 Data\_Retrieve(phone\_number, cust\_id)  
  
#9  
def update\_profile() :  
 user\_id = input("Enter your user ID - ") # here we can nott use login(), cause login() function:Only prints "Log in successful" if credentials match.Does not return any user details.  
 password = input("Enter your password - ")  
  
 user\_details = data\_retrieve(user\_id)  
 if not user\_details:  
 print("User ID not found. Please register.")  
 return  
  
 stored\_password = user\_details[-1]  
 if password != stored\_password:  
 print("Incorrect password. Access denied.")  
 return  
 # here we taking agin userid & password to verify the credentials correct or not .. If correct, we are able to get and use user\_details for updating. If incorrect, to exit  
  
 print("Login successful. You can now update your details.")  
 print("Leave a field blank if you don't want to change it.")  
 print()  
 full\_name =input("Enter your new full name - ") .upper()  
 address = input("Enter your new address - ")  
  
 while True:  
 phone\_number = input("Enter new phone number Otherwise leave blank to keep unchanged - ")  
 # 2 1  
 if not phone\_number or phone\_number.isdigit() and len(phone\_number) ==10 :  
 # 1 2  
 # phone\_number==" "(blank) Otherwise Phone\_number == not blank, filled new phone number by user  
 break  
 else:  
 print("Invalid phone number.")  
  
 while True :  
 new\_password = input("Enter Your new password Otherwise leave blank to keep unchanged - ")  
 # 2 1  
 if new\_password == "" : #1  
 new\_password = stored\_password  
 break  
 elif strong\_password(new\_password) : #2  
 break  
 else:  
 print("Password must include uppercase, lowercase, digit, and special character.")  
  
 # knowing the position of old values  
 cust\_id=user\_details[0]  
 old\_name = user\_details[1]  
 old\_address = user\_details[2]  
 old\_phone\_number = user\_details[3]  
  
 #prepare new/updated values  
 updated\_name = full\_name or old\_name  
 # If the user typed a new name (full\_name is not empty), then updated\_name becomes that new name(full\_name). else the user left it blank, then updated\_name keeps the old name  
 updated\_address = address or old\_address  
 updated\_phone\_number = phone\_number or old\_phone\_number  
 qry = "UPDATE cust\_details SET full\_name = %s, address = %s, phone\_number = %s, password = %s WHERE cust\_id = %s"  
 vlu = (updated\_name,updated\_address,updated\_phone\_number,new\_password,cust\_id)  
 try:  
 cur\_obj.execute(qry, vlu)  
 conn\_obj.commit()  
 print("Details updated successfully.")  
 except mysql.connector.Error as e:  
 print("Error updating data:", e)  
 conn\_obj.rollback()  
  
  
#10  
def delete\_account() :  
 user\_id=input("Enter The User ID - ")  
 password = input("Enter Your Password - ")  
 user\_dtls=data\_retrieve(user\_id)  
 if not user\_dtls :  
 print("You are not registered user")  
 return  
  
 strd\_pswrd=user\_dtls[-1]  
 if password != strd\_pswrd :  
 print("You have entered worng password")  
 return  
 #verify entered credentials correct or not  
  
 print("ACCOUNT DELETE !")  
 print()  
 user\_choice\_to\_delete=input("Are you sure want to Delete your account ? Type YES to confirm - ") .upper()  
 if user\_choice\_to\_delete == "YES" :  
 try:  
 cur\_obj.execute("DELETE FROM cust\_details WHERE user\_id = %s", (user\_id,))  
 conn\_obj.commit()  
 print("Your account has been successfully deleted.")  
 except mysql.connector.Error as e:  
 print("Error deleting account:", e)  
 conn\_obj.rollback()  
  
 else:  
 print("Account deletion cancelled.")

# Main Code

cust\_choice = input("1 - Log In \n2 - Register Yourself \n3 - Profile View \n4 - Update Your Profile \n5 - Delete Account \n Enter Your Choice - ")  
  
  
if cust\_choice == "1":  
 login()  
  
  
elif cust\_choice == "2":  
 # register yourself  
 cust\_registration()  
  
  
elif cust\_choice == "3":  
 profile\_view()  
  
  
elif cust\_choice == "4" :  
 update\_profile()  
 print("View Your Profile")  
 profile\_view()  
  
  
elif cust\_choice == "5" :  
 delete\_account()  
  
  
  
  
else:  
 print("invalid entry")  
  
conn\_obj.close()