File Edit View Navigate Code Refactor Run Tools VCS Window Help main_menu.py - acc_ctrl.py

Class 12 Computer Project > acc_ctrl.py >

acc_ctrl.py

```python
import mysql.connector as conn2
from time import sleep
from validate_email import validate_email

# -----------------------------------------------------------------------
'''This function will help the administrator to add new users.
   Here, Username entered will be accepted only if same username is already not present in database.
   Secondly, here email entered will be accepted only if it really exists
   (i.e) the given email-address has an MX record and SMTP server port. Disposable emails will
   not be accepted here. Module used here is 'validate_email' '''

def Add_new():
    mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon2.cursor(buffered=True)
    no_user = int(input('Enter number of user you want to add: '))
    for i in range(no_user):
        input_username = input("    Enter Login ID for the user: ")
        cursor.execute(f"SELECT username FROM accounts WHERE username = '{input_username}'")
        data = cursor.fetchone()
        if not data:
            input_password = input("    Enter Password of the user: ")
            input_name = input("    Enter name of the user: ")

            def ent():
                input_email = input("    Enter email of the user: ")
                cursor.execute(f"SELECT * FROM accounts WHERE email = '{input_email}'")
                data2 = cursor.fetchone()
                if data2 != None:
                    print("E-mail Address Already Registered.")
                    ent()
```

```
30                  ent()
31              else:
32                  print("Validating Details, please wait.....")
33                  try:
34                      is_valid = validate_email(input_email)
35                      if is_valid:
36                          cursor.execute(
37                              f"INSERT INTO accounts (username, passwd, name_u, email) VALUES('{input_username}', '{input_password}', "
38                              f"'{input_name}', '{input_email}')")
39                          mycon2.commit()
40                          print("Registering")
41                          sleep(2)
42                          print(f"Successfully Added User '{input_username}'")
43                      except:
44                          print("Some Error Occurred.")
45
46          ent()
47      else:
48          print("User already in database.")
49          continue
50      mycon2.close()
51      return
52
53  # ---------------------------------------------------------------------
54  '''This function will allow the administrator to update the every detail of the user like name, password
55     username and email-address. Here email entered will be accepted only if it really exists (i.e) the given
56     email-address has an MX record and SMTP server port. Disposable emails will not be accepted here.
57     Modules used here are 'validate_email' and 'mysql.connector' '''
58
59  def Update_user():
```

```python
def Update_user():
    mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
    cursor = mycon2.cursor(buffered=True)
    no_user = int(input('Enter number of user you want to update: '))
    count = 0
    while count < no_user:
        user_name = str(input("    Enter the username of the user you want to update: "))
        cursor.execute("SELECT * FROM accounts WHERE username = '{}'".format(user_name))
        data = cursor.fetchone()
        if data == None:
            print(f"No user like '{user_name}' exists in database.")  # Checking the wrong Input
            Update_user()
        elif data[0] == 1:
            print("Can't Update user. The specified username is Administrator.")
        else:
            print("     1 -> Update the user's username \n",
                  "    2 -> Update the user's password \n"
                  "     3 -> Update the user's name \n"
                  "     4 -> Update the user's email address")
            choice = int(input("Enter your choice (1 to 4): "))
            if choice == 1:
                new_username = str(input(f"    Enter the new username of {user_name}: "))
                cursor.execute("SELECT * FROM accounts WHERE username = '{}'".format(new_username))
                data2 = cursor.fetchone()
                if data2 == None:
                    cursor.execute(f"UPDATE accounts SET username = '{new_username}' WHERE username = '{user_name}'")
                    mycon2.commit()  # Updating the username
                    print("Successfully Updated")
                else:
                    print("Username Already Exists. \n User Not Updated :(")
```

```
89          elif choice == 2:
90              new_password = str(input(f"    Enter the new password of the {user_name}: "))
91              cursor.execute(f"UPDATE accounts SET passwd = '{new_password}' WHERE username = '{user_name}'")
92              mycon2.commit()  # Updating the user
93              print("Successfully Updated")
94          elif choice == 3:
95              new_name = str(input(f"    Enter the new name of the {user_name}: "))
96              cursor.execute(f"UPDATE accounts SET name_u = '{new_name}' WHERE username = '{user_name}'")
97              mycon2.commit()  # Updating the user
98              print("Successfully Updated")
99          elif choice == 4:
100             new_mail = str(input(f"    Enter the new email address of the {user_name}: "))
101             cursor.execute(f"SELECT email FROM accounts WHERE email = '{new_mail}'")
102             data3 = cursor.fetchone()
103             if data3 == None:
104                 is_valid = validate_email(new_mail)
105                 if is_valid:
106                     cursor.execute(f"UPDATE accounts SET email = '{new_mail}' WHERE username = '{user_name}'")
107                     mycon2.commit()  # Updating the user
108                     print("Successfully Updated")
109                 else:
110                     print("Invalid Email Address. User not updated :(")
111             else:
112                 print("Email Address Already registered. \n User Not updated :(")
113         else:
114             print("Wrong Input")
115             Update_user()
116     count += 1
117 mycon2.close()
118 return
```

acc_ctrl.py

```
120     # ------------------------------------------------------------------------
121     ''' This function will help administrator to delete any user he/she wants from database
122         by entering his/her username if it exists. But admin can't delete the administrator account
123         Module used here is 'mysql.connector' '''
124
125     def Del_user():
126         mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
127         cursor = mycon2.cursor(buffered=True)
128         no_user = int(input("Enter number of accounts you want to delete: "))
129         count = 0
130         while count < no_user:
131             user_name = str(input("    Enter the username of the user you want to delete: "))
132             cursor.execute(f"SELECT * FROM accounts WHERE username = '{user_name}'")
133             data = cursor.fetchone()
134             if data == None:
135                 print(f"No user like '{user_name}' exists in database.")  # Checking the Wrong Input
136                 Del_user()
137             elif data[0] == 1:
138                 print("Can't Delete user. The specified username is Administrator.")
139             else:
140                 cursor.execute(f"DELETE FROM accounts where username = '{user_name}'")
141                 mycon2.commit()  # Deleting the user
142                 sleep(2)
143                 print("Successfully Deleted.")
144             count += 1
145         mycon2.close()
146         return
147
```

```
148  # ---------------------------------------------------------------------
149  ''' This function will allow administrator to view every detail of every user in database.
150      Module used here is mysql.connector '''
151
152  def All_u():
153      mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
154      cursor = mycon2.cursor(buffered=True)
155      cursor.execute("SELECT * FROM accounts")
156      data = cursor.fetchall()
157      count = 0
158      for row in data:
159          tp = row
160          count += 1
161          print()
162          print(f"          User {count}")
163          print("      ID             : ", tp[0])
164          print("      Username       : ", tp[1])
165          print("      Password       : ", tp[2])
166          print("      Name           : ", tp[3])
167          print("      E-mail Address : ", tp[4])
168      mycon2.close()
169      return
170
171  # ---------------------------------------------------------------------
172  '''This function will help administrator to view details of the user by entering
173     his/her username if it exists in database.'''
174
175  def View_user():
176      mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
177      cursor = mycon2.cursor(buffered=True)
```

```
177         cursor = mycon2.cursor(buffered=True)
178         num = int(input("Enter the number of users of which you want details of: "))
179         count = num
180         while count != 0:
181             user_name = str(input("    Enter the username of the user: "))
182             cursor.execute(f"SELECT * FROM accounts WHERE username = '{user_name}'")
183             data = cursor.fetchone()
184             if not data:
185                 print("Username specified is invalid. Enter Again.")
186                 print()
187             else:
188                 print(f"   ID:            {data[0]}")
189                 print(f"   Username:      {data[1]}")
190                 print(f"   Name:          {data[3]}")
191                 print(f"   Email-Address: {data[4]}")
192                 print()
193                 count -= 1
194         return
195
196     # ---------------------------------------------------------------------
197     '''Sign_Up function will allow any new user running the software to register himself in database.
198        For this, they have to create a username, password, give their name and enter their email address which
199        really exists as it will be checked then only the user will be registered.
200        Module used here is 'validate_email' and 'mysql.connector' '''
201
202     def Sign_Up():
203         mycon2 = conn2.connect(host="localhost", user="root", password="Rinshu@03", database="book_shop")
204         cursor = mycon2.cursor(buffered=True)
205         input_username = input("            Create Username (Login ID) : ")
206         cursor.execute(f"SELECT * FROM accounts WHERE username = '{input_username}'")
```

```
206     cursor.execute(f"SELECT * FROM accounts WHERE username = '{input_username}'")
207     data = cursor.fetchone()
208     if data == None:
209         input_password = input("            Create Password: ")
210         input_name = input("           Enter your name: ")
211 
212         def ent():
213             input_email = input("            Enter your email address: ")
214             cursor.execute(f"SELECT * FROM accounts WHERE email = '{input_email}'")
215             data2 = cursor.fetchone()
216             if data2 != None:
217                 print("E-mail Address Already Registered.")
218                 ent()
219             else:
220                 print("Validating Details, please wait.....")
221                 is_valid = validate_email(input_email)
222                 if is_valid:
223                     cursor.execute(
224                         f"INSERT INTO accounts (username, passwd, name_u, email) VALUES('{input_username}', "
225                         f"'{input_password}', '{input_name}', '{input_email}')")
226                     mycon2.commit()
227                     print("Registering")
228                     sleep(2)
229                     print(f"Successfully Added User '{input_username}'")
230         ent()
231     else:
232         print("User Already In Database.")
233         print("Try Again")
234         Sign_Up()
235     mycon2.close()
236     return
237 
```