INFORMATICS INSTITUTE OF TECHNOLOGY

Foundation Certificate in Higher Education

Module : DOC 330 Designing Innovative Solutions

Module Leader : Mr.Welihinda

Assessment Type : Individual Coursework

Submission Date : 21/08/2022

Student ID: 20211608

Student Name: Tharindu D. Rathnayake

Abstract

This report briefly explains about the design and develops a program using python to manage student management system.

Acknowledgements

I would like to offer my heartiest gratitude to our module leader Mr. Sudharshan Welihinda and module lecture Mr.Nishan, Ms.Keerthiga for all guidance and all helps given to complete this module and assignment.

Table of Contents

List of figures

List of Table

1 Problem and Problem understanding of the Project

1.1 Problem

Create a console python 3.x program which will allow user to demonstrate a small program for classroom attendance.

1.2 Problem understanding

1. Create a menu with different options and allow user to choose which one user need to be done.
2. Ask for necessary inputs
3. If user inputs an invalid value, then display an error message.
4. After getting inputs that inputs need to store in database system.
5. If user wants to continue, he/she should be able to do so without exiting from the program.
6. When “Exit” option is selected program should terminate.

2 Algorithm

Step1 : Start

Step2 : Initiate the variables.

Step3 : Display options to user to choose

Step4 : Get the user input; user has to select one of option above. If user input invalid value display error message and ask again for valid input.

Step5 : After choosing which option ask user for inputs to selected option

Step6 : After getting valid inputs store it in database which base on the system.

Step7 : Ask user that he/she want to continue the program or want to exit.(Y/N)

Step8 : If user input “Y” program should run again. If user input “N” program should terminate.

.

3 Python Code

```
import os

import sys

import mysql.connector


# Define variables

option=0


#Menu For The Student Management System


def menu():

# Open database connetion

    conDict={'host':"localhost",

        'database':'student',

        'user':"root",

        'password':""}

    global db

    db=mysql.connector.connect(**conDict)
```

```
# Print Menu for the user


    print("|======== Welcome To Student Management System ========|")

    print("\nOption 1 : To View Student's List \nOption 2 : To view Attendance List \nOption 3 : To Add New Student detail  \nOption 4 : To Delete student detail \nOption 5 : To Update student detail \nOption 6 : To mark attendance\nOption 7 : To view selected student attendance")

    userInput=int(input("\nPlease Select An Above Option: "))

    while userInput<1 or userInput>7:

        print("\n Incorrect option number") #Error Message

        userInput = int(input("Please Select An Above Option: "))



    # ---------- Option 01 ---------- #


    if userInput==1:

        print("\nList of students : ")

        cursor=db.cursor()

        cursor.execute("SELECT * FROM student")

        data=cursor.fetchall()

        for item in data:

            print(item)

        db.close()


        again=input("\nYou want to go back to menu? (Y/N) : ")
```

```
    if again=="Y" or "y":
        menu()
    else:
        exit()

# ---------- Option 02 ---------- #
if userInput==2:
    print("\nList of attend students : ")
    cursor=db.cursor()
    cursor.execute("SELECT * FROM student_attend")
    data=cursor.fetchall()
    for item in data:
        print(item)
    db.close()

    again=input("\nYou want to go back to menu? (Y/N) : ")
    if again=="Y" or "y":
        menu()
    else:
        exit()
```

```
    # ---------- Option 03 ---------- #


    # Open database connetion
    if userInput==3:
            print("\nPlease enter following details")
            cursor=db.cursor()
            # Read user input
            stNo = int(input("\nType student Number: "))
            fName = input("Type first name : " )
            lName = input("type last name : ")
            dob = input("type date of birth : ")
            telephoneNo = int(input("Type telephone number: " ))


            # Execute SQL Query using execute() method
            mySQLText= "INSERT INTO student(stNo,fName,lName,dob,telephoneNo)
VALUES (%s,%s,%s,%s,%s)"
            myValues=(stNo,fName,lName,dob,telephoneNo)
            cursor.execute(mySQLText, myValues)
            # Commit the change
            db.commit()
            print("\nStudent information Added")
            db.close()
            again=input("\nYou want to go back to menu? (Y/N) : ")
            if again=="Y" or "y":
               menu()
```

```
        else:
            exit()


# ---------- Option 04 ---------- #
# Open database connetion
if userInput==4:
        print("\nPlease enter following details")
        cursor=db.cursor()
        stNo=input("\nEnter the student number to be removed :")
        # Execute SQL Query using execute() method
        cursor.execute("DELETE FROM student WHERE stNo="+ stNo +"")
        # Commit the change
        db.commit()
        print(cursor.rowcount,"Student information Deleted")
        db.close()
        again=input("\nYou want to go back to menu? (Y/N) : ")
        if again=="Y" or "y":
            menu()
        else:
            exit()
```

```
    # ---------- Option 05 ---------- #
    # Open database connetion
    if userInput==5:
        global option
        option=input("\nWhich part you want to update?
(stNo,fName,lName,dob,telephoneNo) : ")
    if option=="stNo":
        cursor=db.cursor()
        stNo=input("Type student number: ")
        newstNo=input("Enter new student number: ")
        updTxt="UPDATE student SET stNo='" + newstNo + "'WHERE stNo=" + stNo
        cursor.execute(updTxt)
        db.commit()
        print(cursor.rowcount,"Student number Updated")
        db.close()
    elif option=="fName":
        cursor=db.cursor()
        stNo=input("Type student number: ")
        fName=input("Enter student first name: ")
        updTxt="UPDATE student SET fName='" + fName + "'WHERE stNo=" + stNo
        cursor.execute(updTxt)
        db.commit()
        print(cursor.rowcount,"Student first name Updated")
        db.close()
```

```
    elif option=="lName":
        cursor=db.cursor()
        stNo=input("Type student number: ")
        lName=input("Enter student last name: ")
        updTxt="UPDATE student SET lName='" + lName + "'WHERE stNo=" + stNo
        cursor.execute(updTxt)
        db.commit()
        print(cursor.rowcount,"Student last name Updated")
        db.close()
    elif option=="dob":
        cursor=db.cursor()
        stNo=input("Type student number: ")
        dob=input("Enter student date of birth (Year/month/Day): ")
        updTxt="UPDATE student SET dob='" + dob + "'WHERE stNo=" + stNo
        cursor.execute(updTxt)
        db.commit()
        print(cursor.rowcount,"Student Date of birth Updated")
    elif option=="telephoneNo":
        cursor=db.cursor()
        stNo=input("Type student number: ")
        telephoneNo=input("Enter student telephone no: ")
        updTxt="UPDATE student SET telephoneNo='" + telephoneNo + "'WHERE stNo="
+ stNo
        cursor.execute(updTxt)
```

```
        db.commit()
        print(cursor.rowcount,"Student telephone number Updated")
        again=input("\nYou want to go back to menu? (Y/N) : ")
        if again=="Y" or "y":
            menu()
        else:
            exit()

    # ---------- Option 06 ---------- #
    # Open database connetion

    if userInput==6:
        cursor=db.cursor()
        cursor.execute("SELECT * FROM student")
        data=cursor.fetchall()
        for item in data:
            print(item)
            stNo=input("Type student number: ")
            attendDate=input("Enter date : ")
            attendance=input("Is this student attend? (Yes/No) :")
            # Execute SQL Query using execute() method
            mySQLText="INSERT INTO student_attend(stNo,attendDate,attendance)
VALUES (%s,%s,%s)"
            myValues=(stNo,attendDate,attendance)
            cursor.execute(mySQLText, myValues)
```

```
        # Commit the change
    db.commit()
    db.close()
    again=input("\nYou want to go back to menu? (Y/N) : ")
    if again=="Y" or "y":
       menu()
    else:
       exit()

# ---------- Option 07 ---------- #

if userInput==7:
    cursor=db.cursor()
    stNo=input("Enter the student number to find his/her attendance :")
    # Execute SQL Query using execute() method
    cursor.execute("SELECT * FROM student_attend WHERE stNo="+ stNo +"")
    data=cursor.fetchall()
    print("\nStudent attend detail(stNo,Attend Date,Attendance) :")
    for item in data:
        print(item)
    # Commit the change
    db.commit()
    db.close()
```

```
        again=input("\nYou want to go back to menu? (Y/N) : ")
        if again=="Y" or "y":
            menu()
        else:
            exit()

menu()
```

4 Screenshots of the program in various states

4.1 Menu

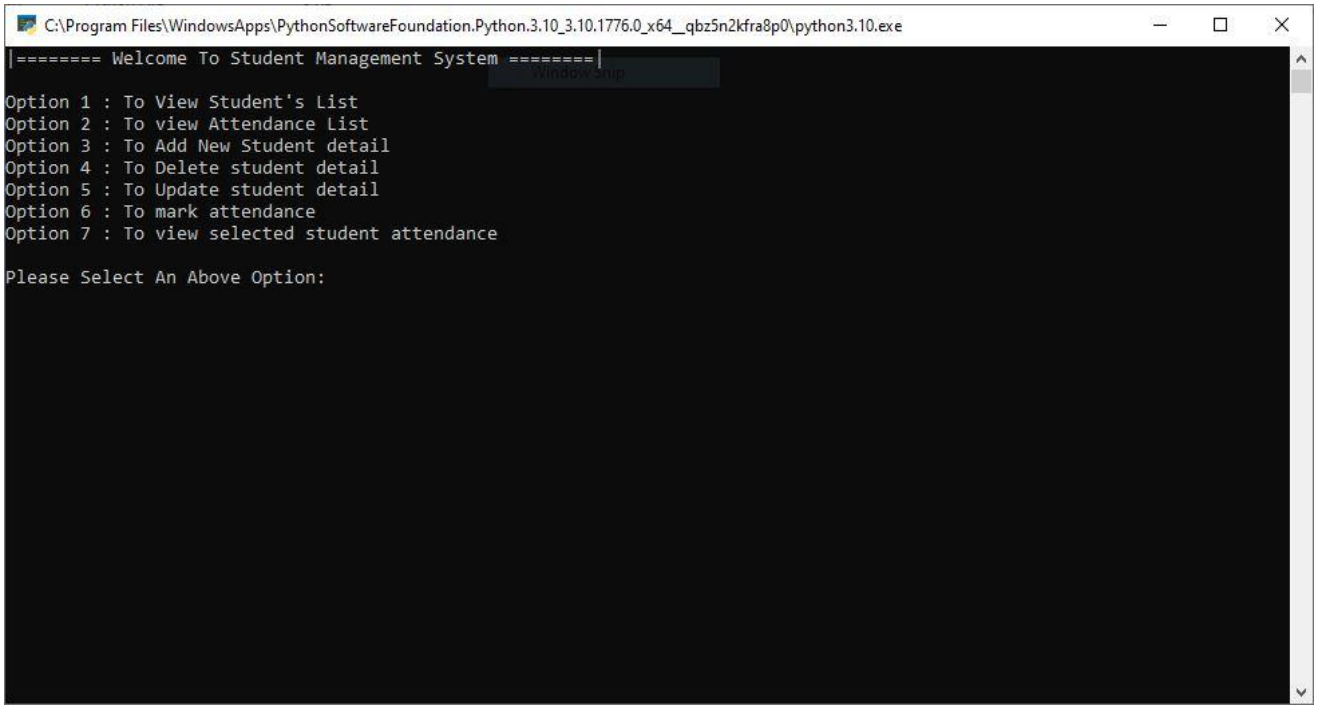

Figure 1 Main menu of program

4.2 View student's list

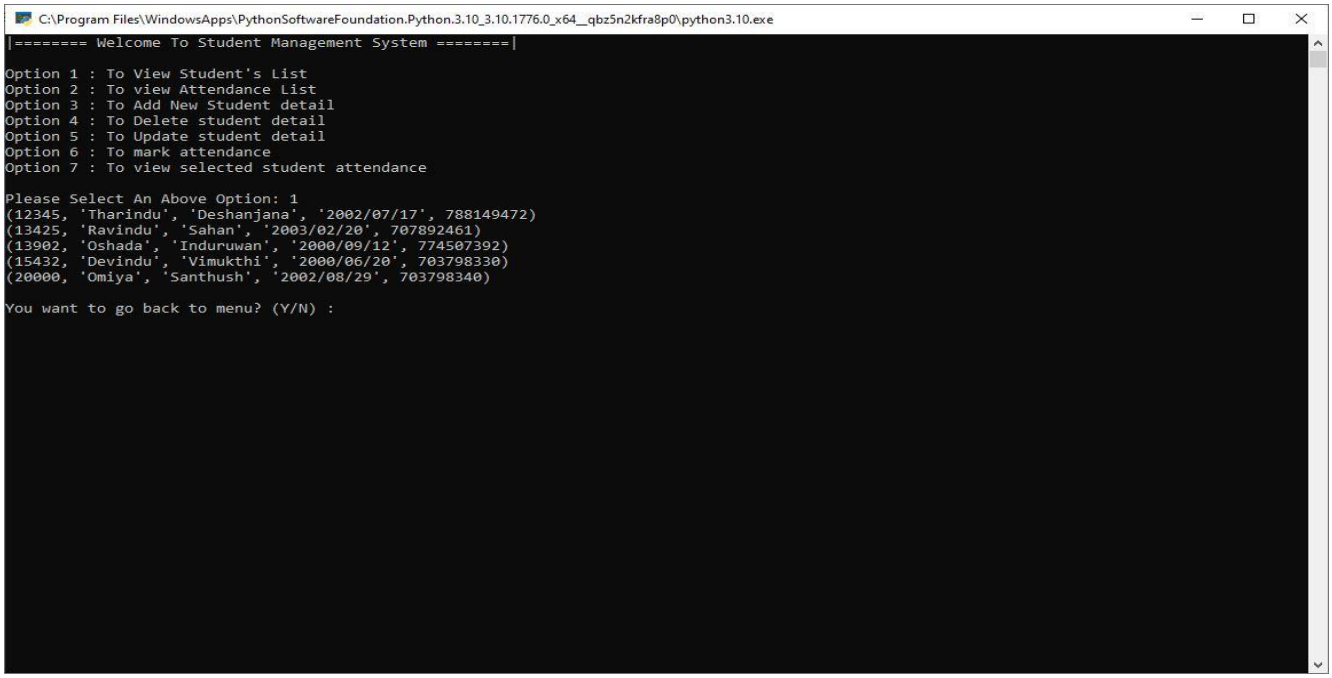

Figure 2 View student detail

4.3 View attendance list

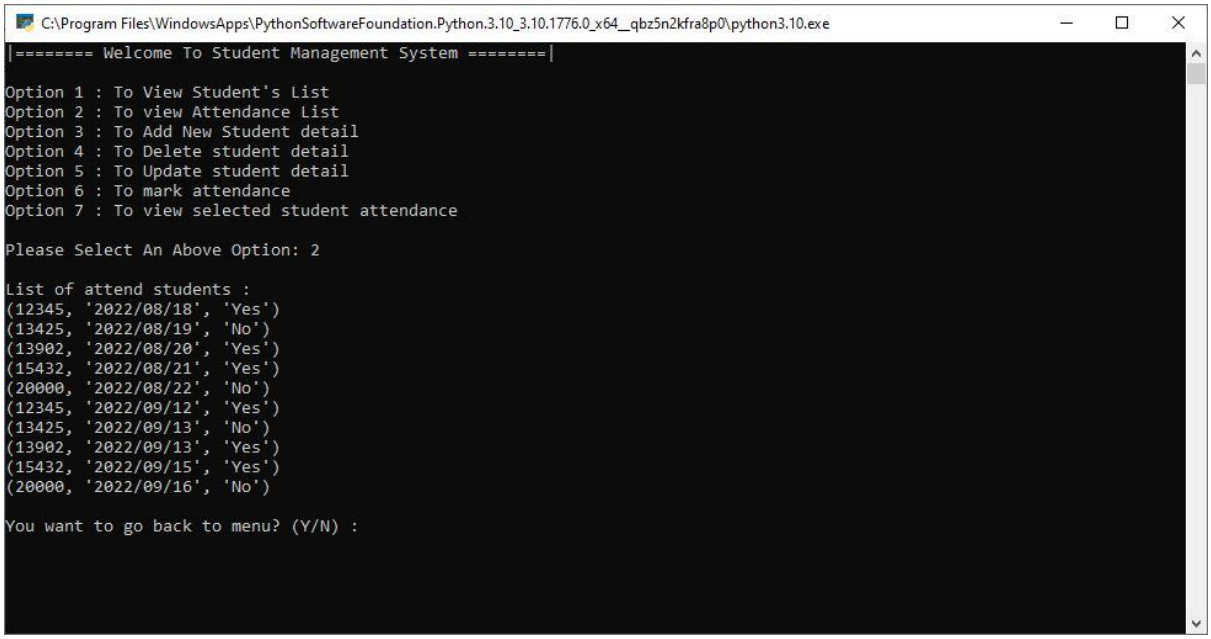

Figure 3 View attendance detail

stNo	attendDate	attendance
12345	2022/08/18	Yes
13425	2022/08/19	No
13902	2022/08/20	Yes
15432	2022/08/21	Yes
20000	2022/08/22	No
12345	2022/09/12	Yes
13425	2022/09/13	No
13902	2022/09/13	Yes
15432	2022/09/15	Yes
20000	2022/09/16	No

Figure 4 Attending list

4.4 Adding new student

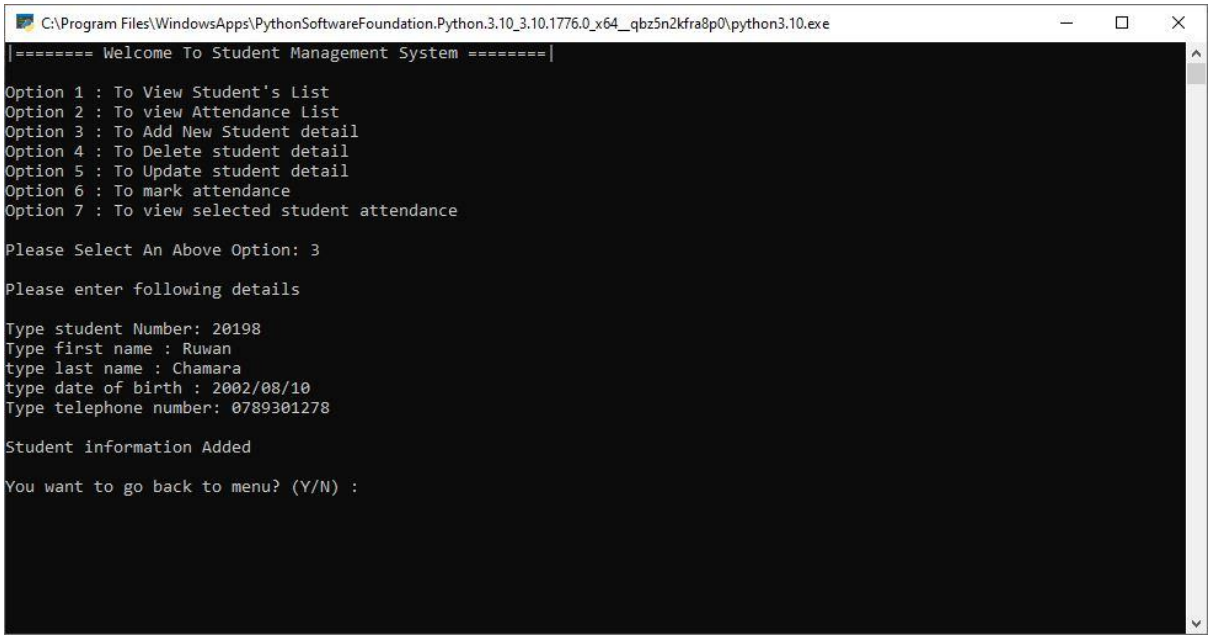

Figure 5 Adding new student detail

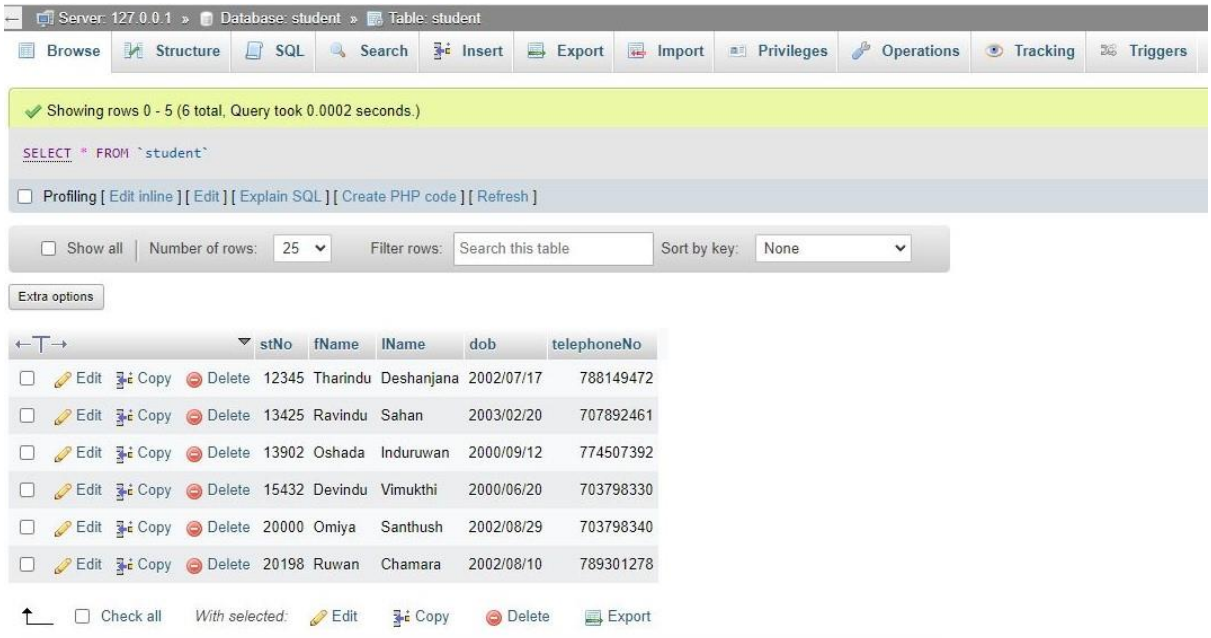

Figure 6 after console command 1

4.5 Deleting student detail

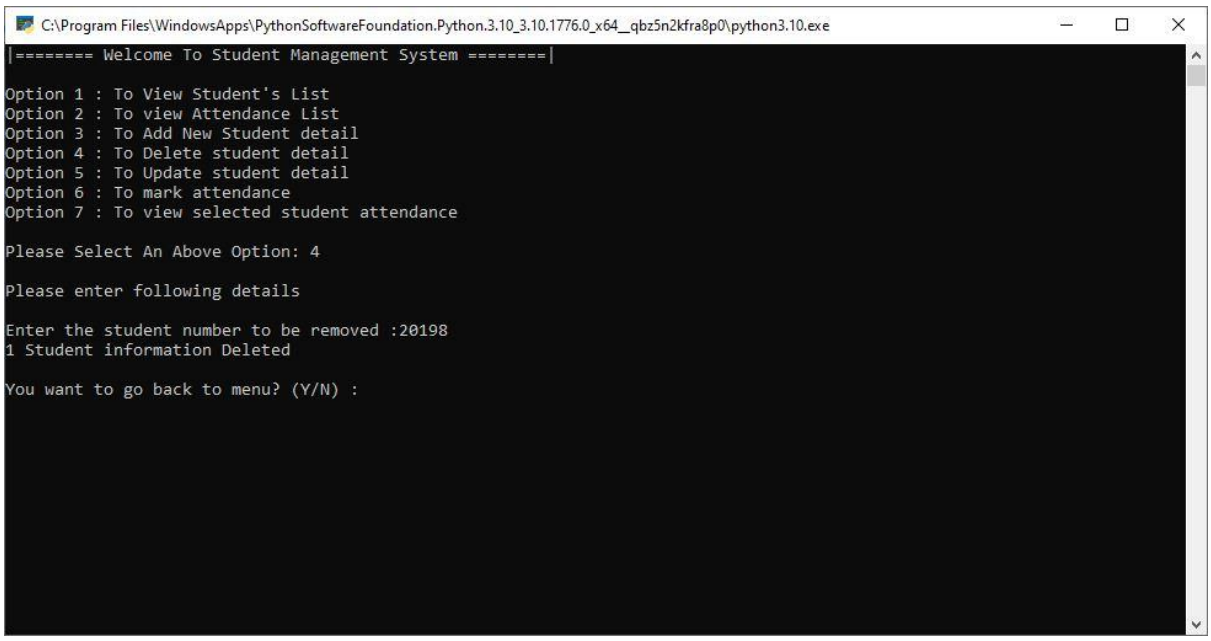

Figure 7 Deleting a student detail

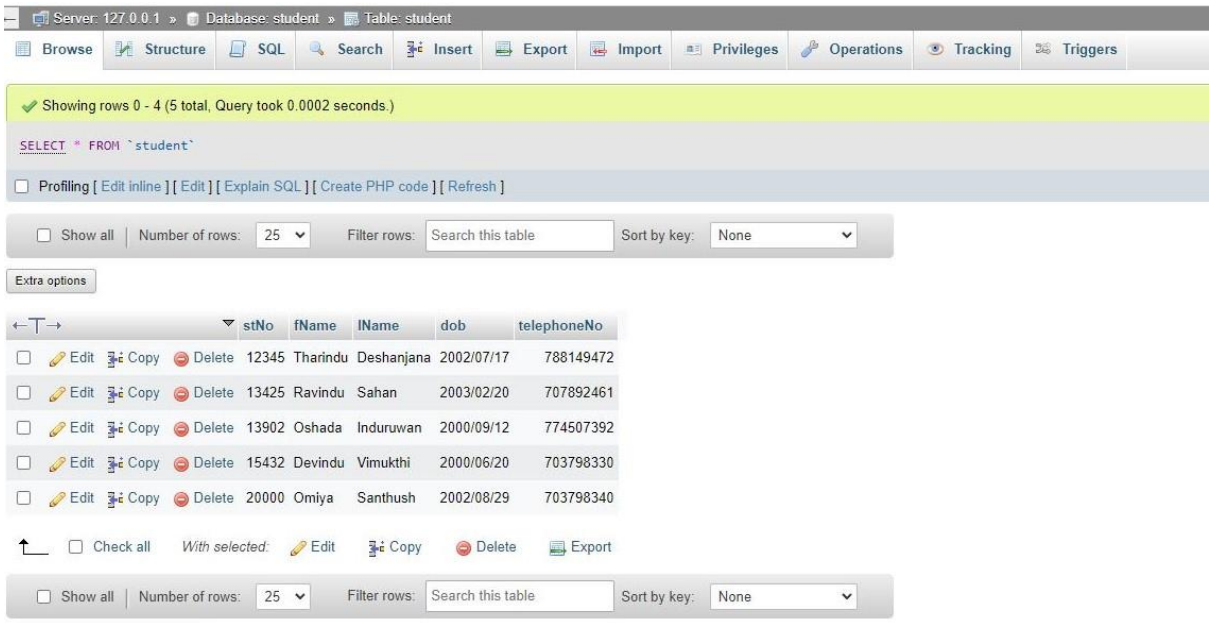

Figure 8 After console command 2

4.6 Updating student detail

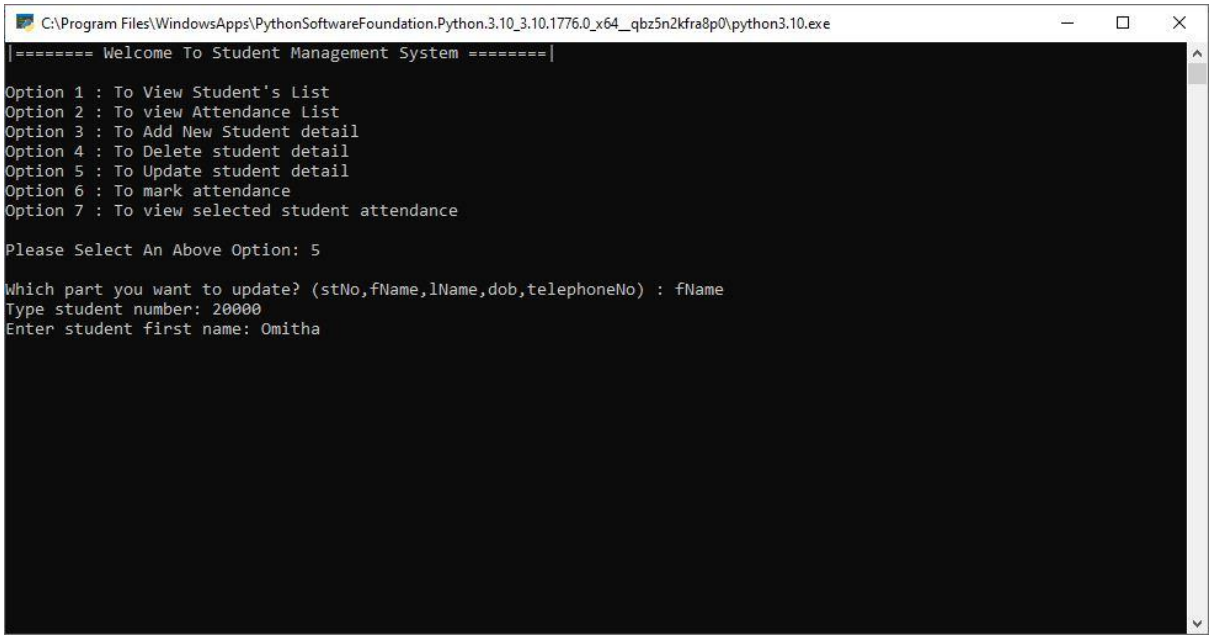

Figure 9 Updating student detail

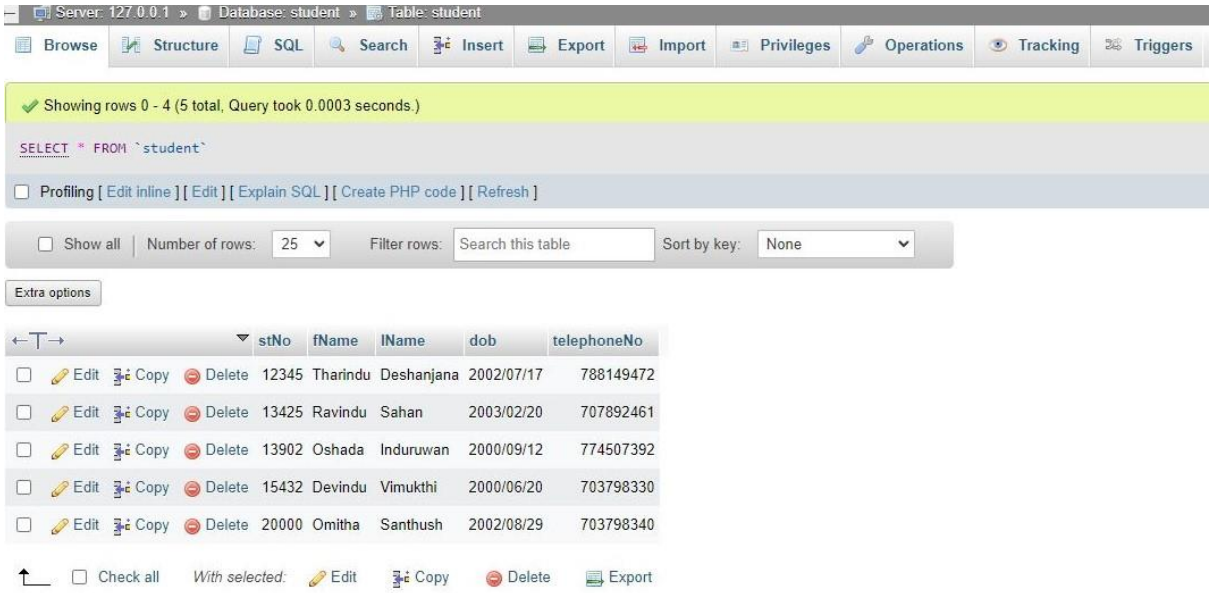

Figure 10 After console command 3

4.7 Marking attendance

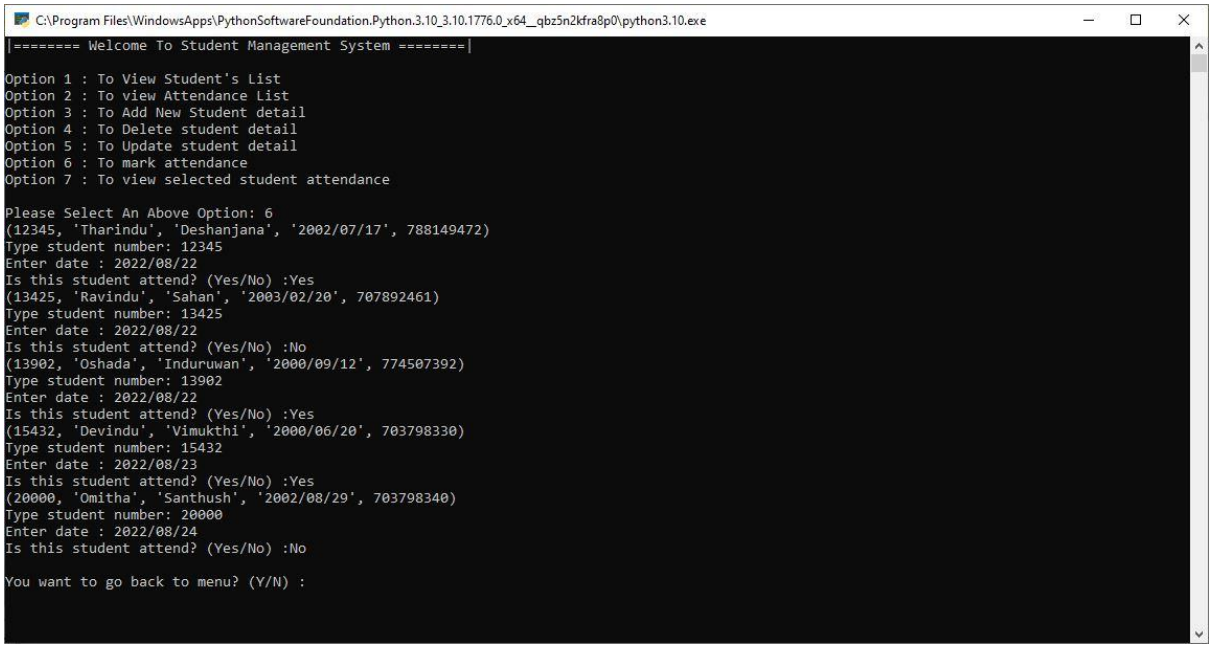

Figure 11 Marking attendance

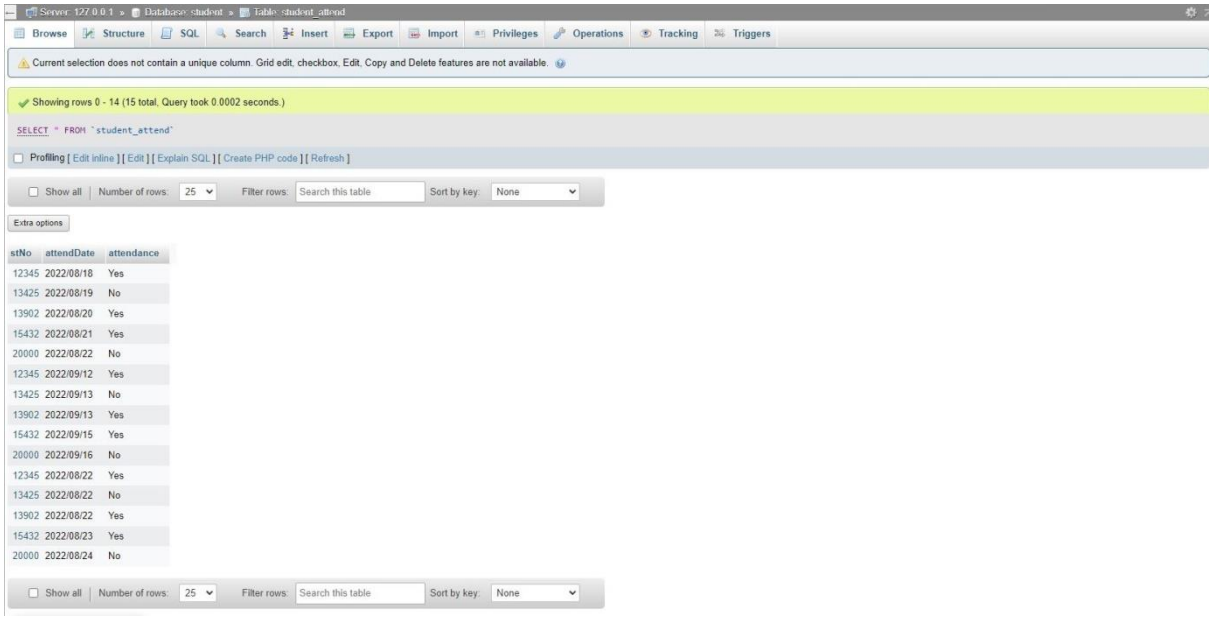

Figure 12 After console commands 4

4.8 View selected student attendance

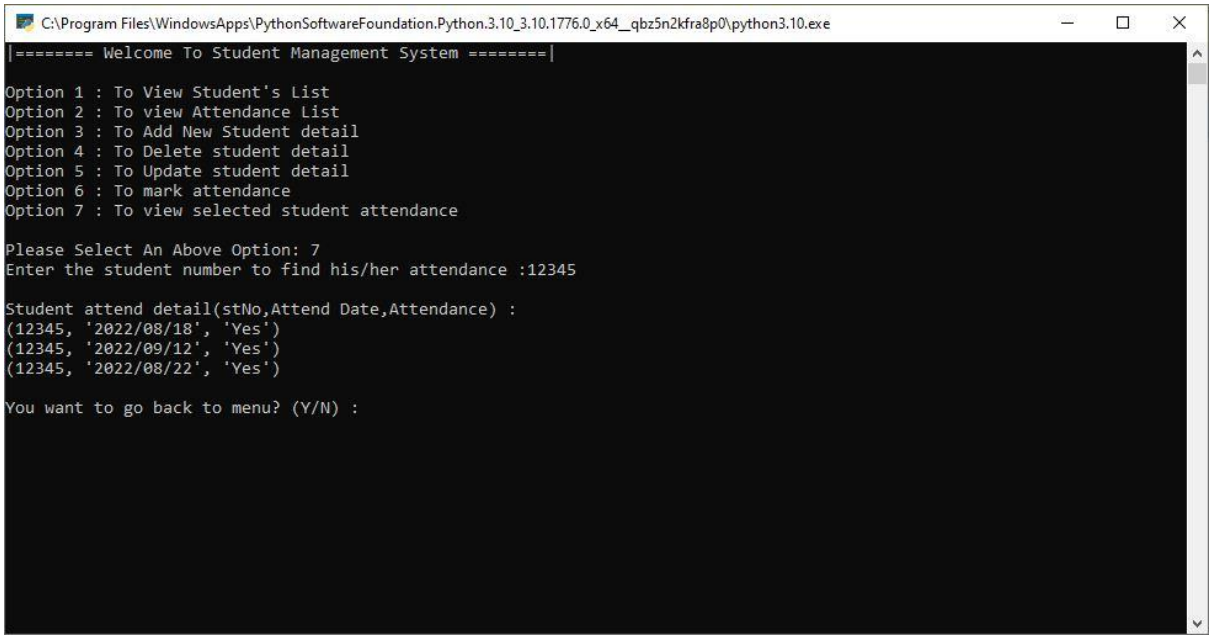

Figure 13 View selected student attendance

5 Test Case Table

Test Case	Input	Expected Output	Actual Output	Remarks
1	Input “1”	Display student’s list	Display student’s list	Test Pass
2	Input “2”	Display attendance list	Display attendance list	Test Pass
3	Input “3”, student number, First name, Last name, date of birth, telephone number	Store following user inputs in database	User input data stored successfully	Test Pass
4	Input “4”,input student number	Delete data from database according to the user input	Delete data from database according to the user input	Test Pass
5	Input “5”, input “fName” ,stNo and new fName	New First name of following students should be update	First name updated	Test Pass
6	Input “5”, input “lName” ,stNo and new lName	New Last name of following students should be update	Last name updated	Test Pass
7	Input “5”, input “dob” ,stNo and new date of birth	New Date of birth of following students should be update	Date of birth updated	Test Pass
8	Input “5”, input “tekephoneNo” ,stNo and new telephone number	New telephone number of following students should be update	Telephone number updated	Test Pass

9	Input “6”, student number, date, attendance	stNo,date and attendance should be store in attendance table	Following data stored in attendance table	Test Pass
10	Input “7” and student number	Attendance detail of entered stNo should be display	Display attendance details	Test Pass
11	Input “Y” to run program again	Program should run again	Program run again	Test Pass

Table 1 Test cases

6 Screenshots of test cases

6.1 Display student's list (Test case 01)

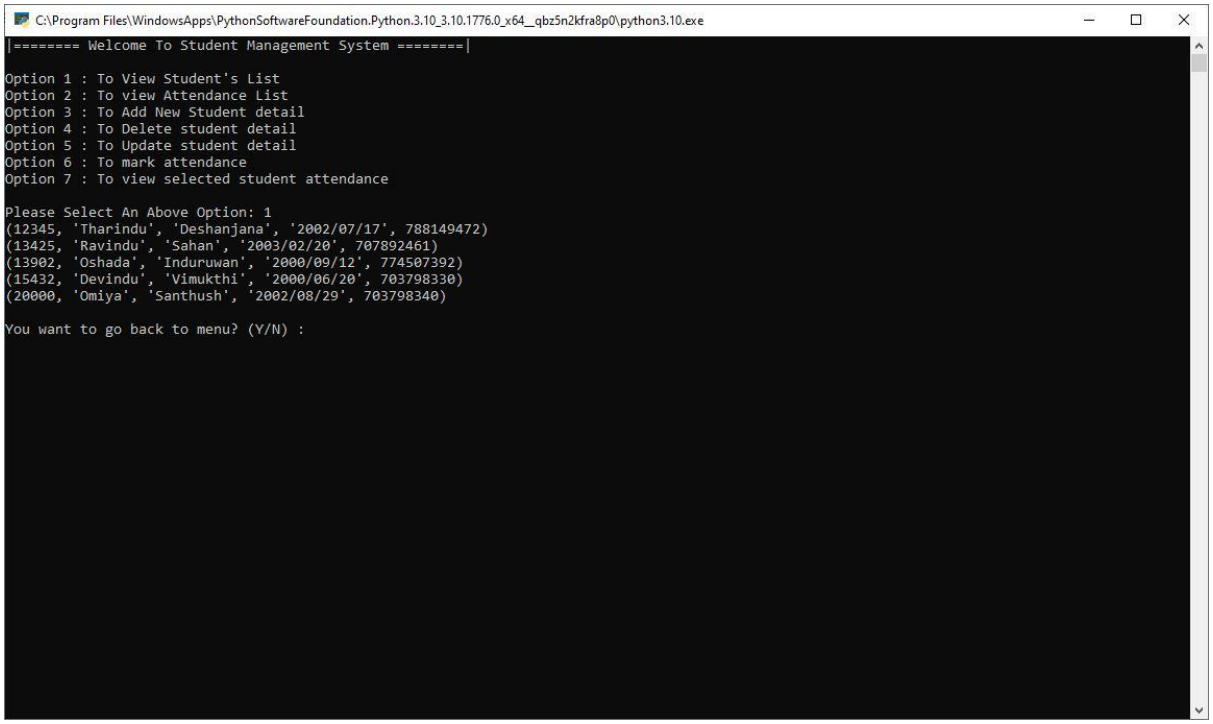

Figure 14 Test case 01 input

6.2 Display attendance list (Test case 02)

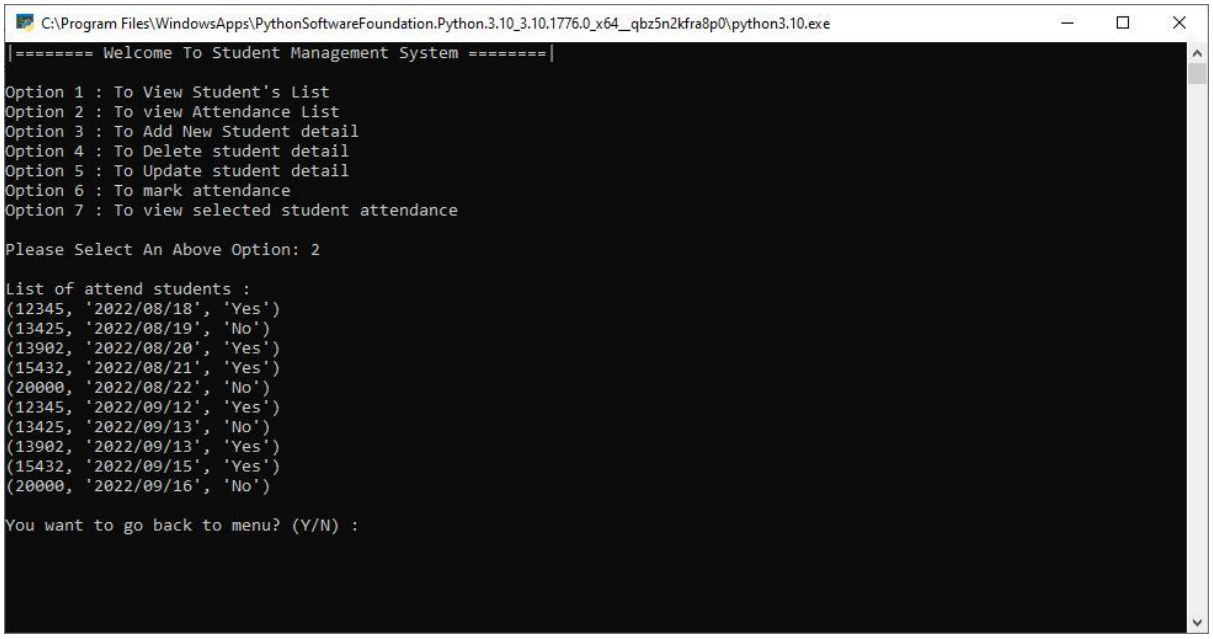

Figure 15 Test case 02 input

6.3 Store new student details (Test case 03)

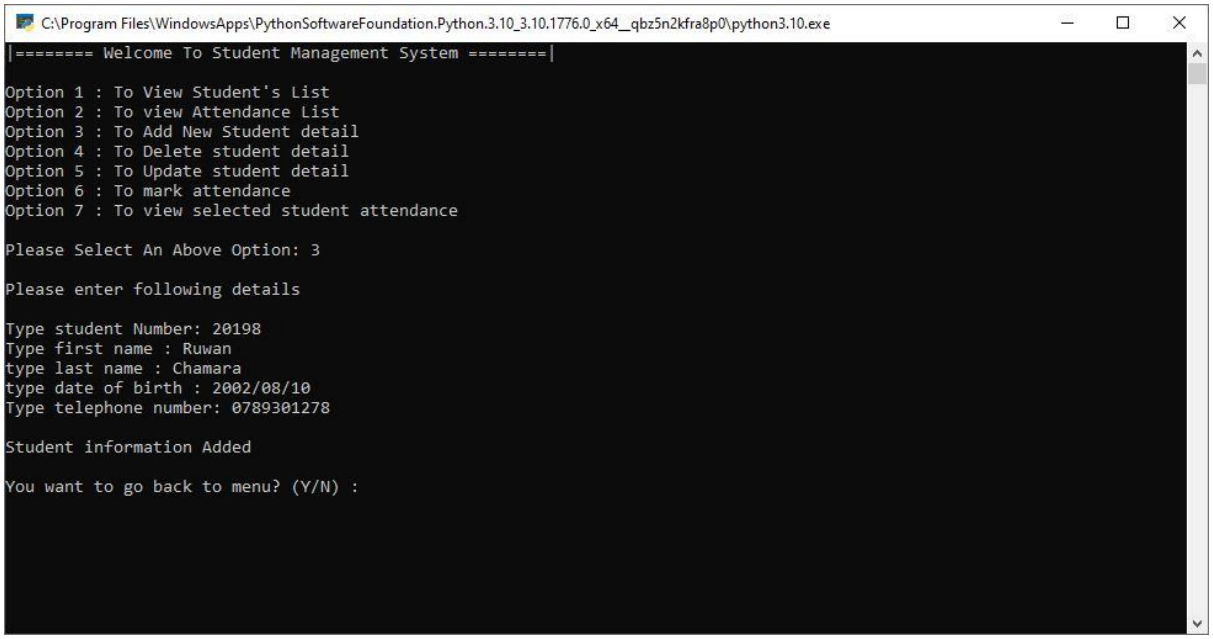

Figure 16 Test case 03 input

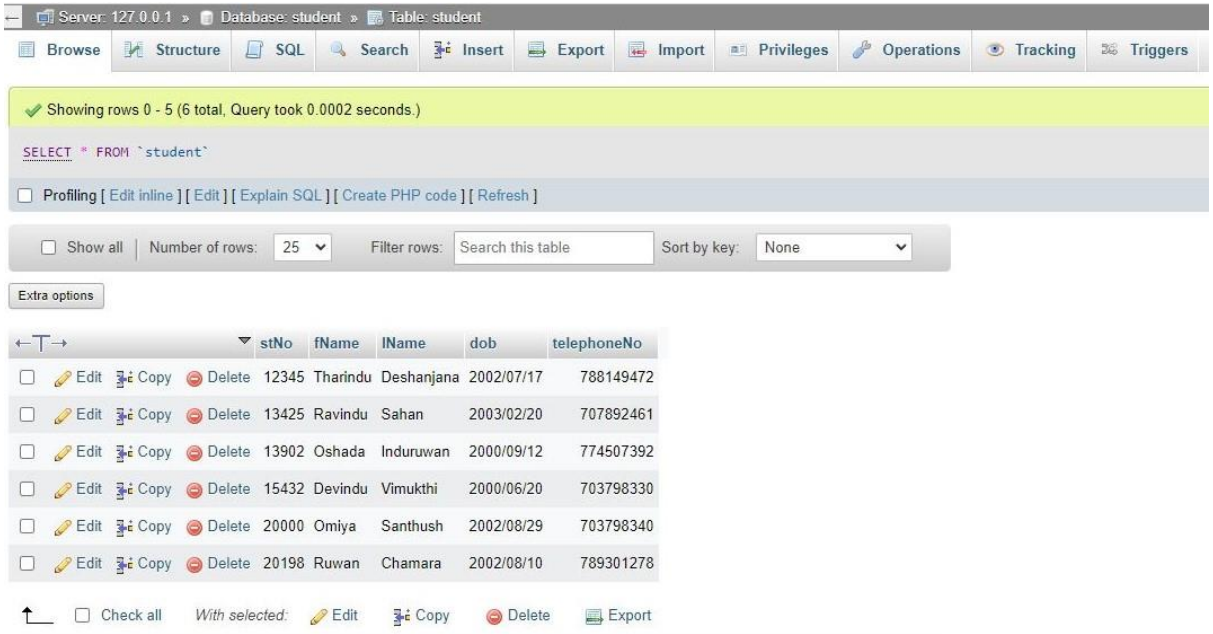

Figure 17 Test case 03 output

6.4 Delete data from database (Test case 04)

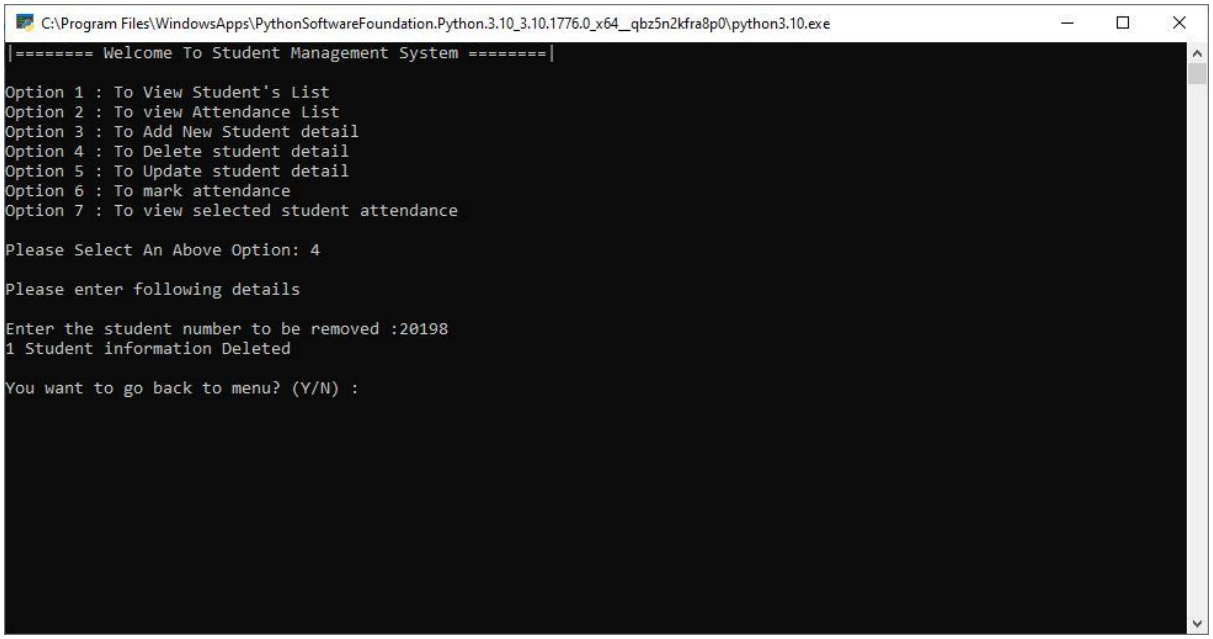

Figure 18 Test case 04 input

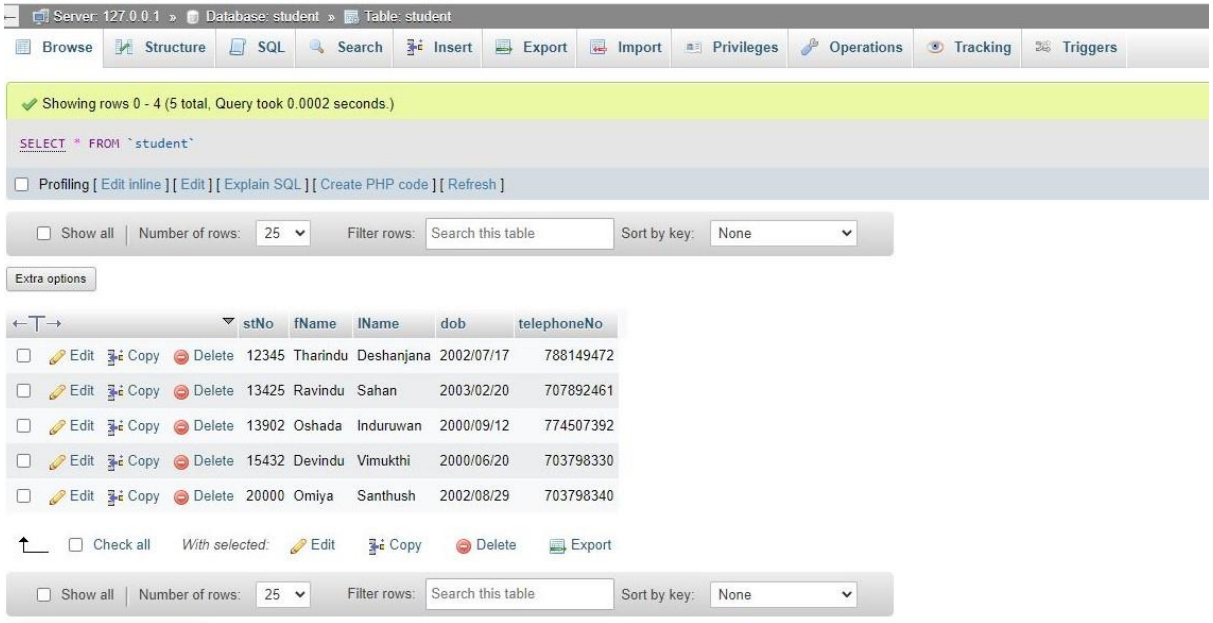

Figure 19 Test case 04 output

6.5 Updating first name of a student (Test case 05)

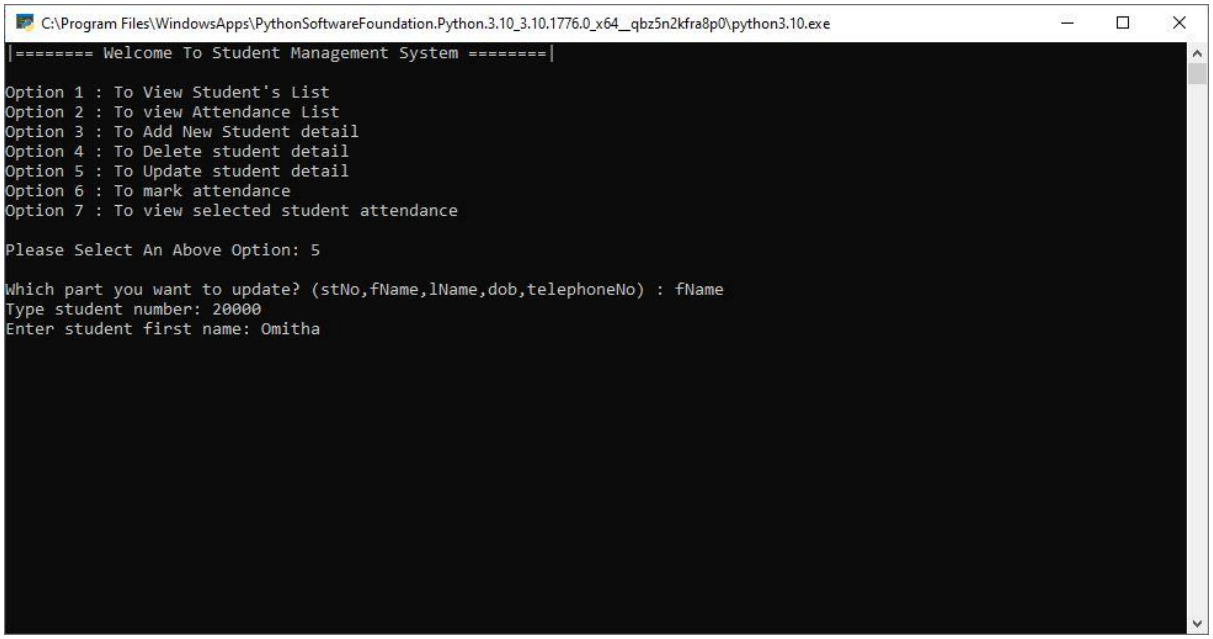

Figure 20 Test case 05 input

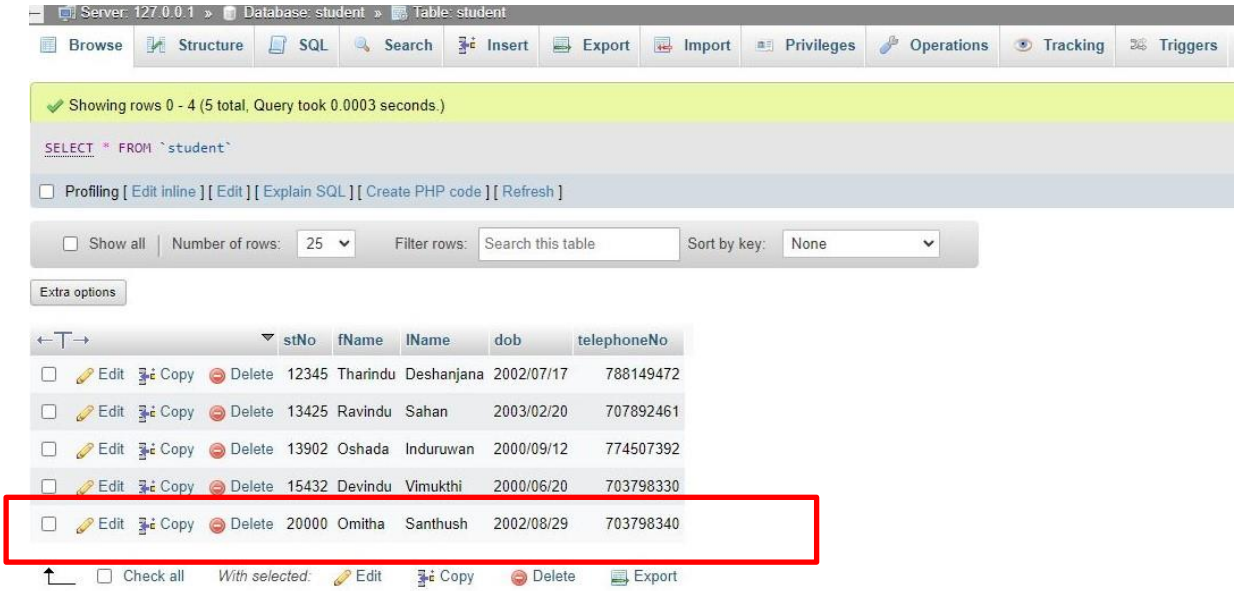

Figure 21 Test case 05 output

6.6 Updating last name of a student (Test case 06)

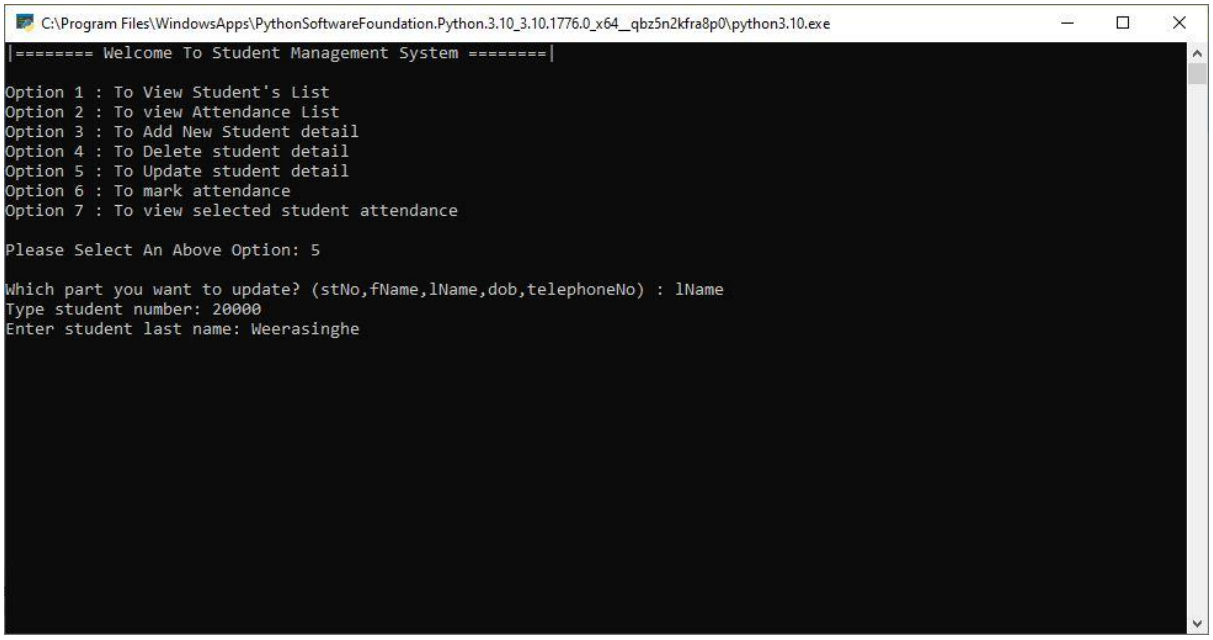

Figure 22 Test case 06 input

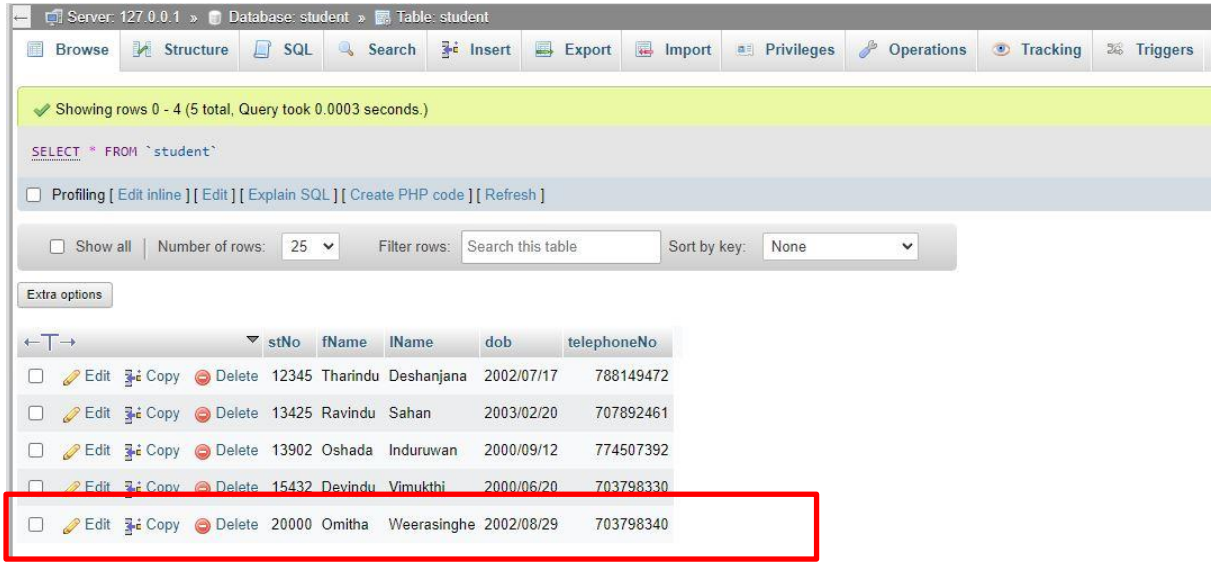

Figure 23 Test case 06 output

6.7 Updating date of birth of a student (Test case 07)

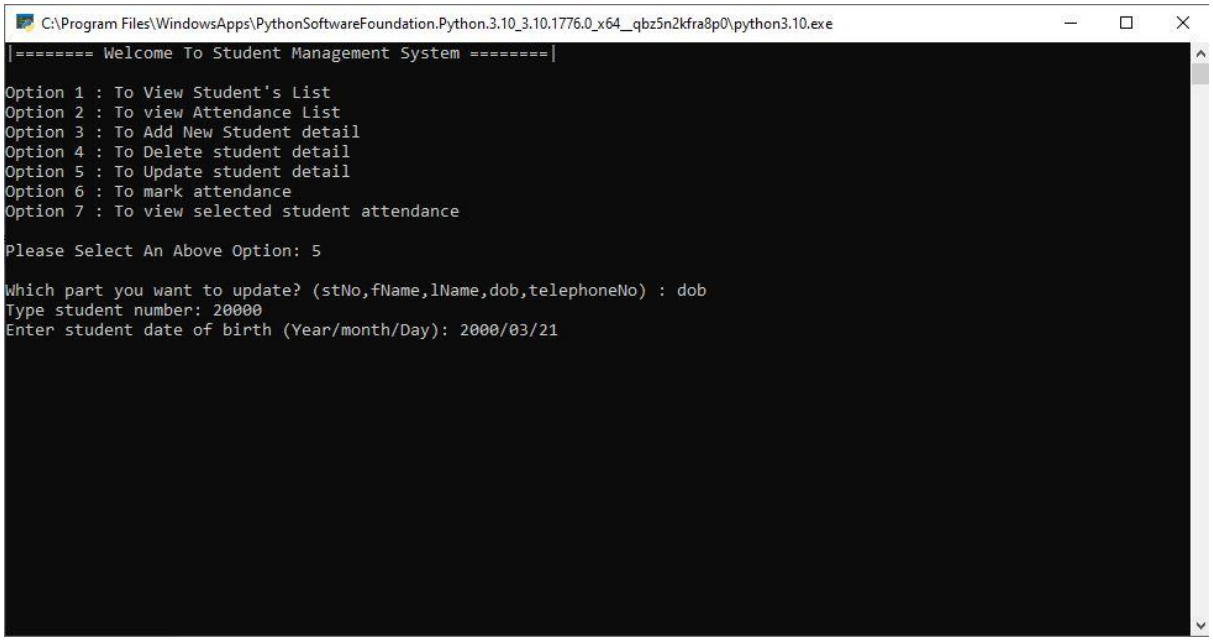

Figure 24 Test case 07 input

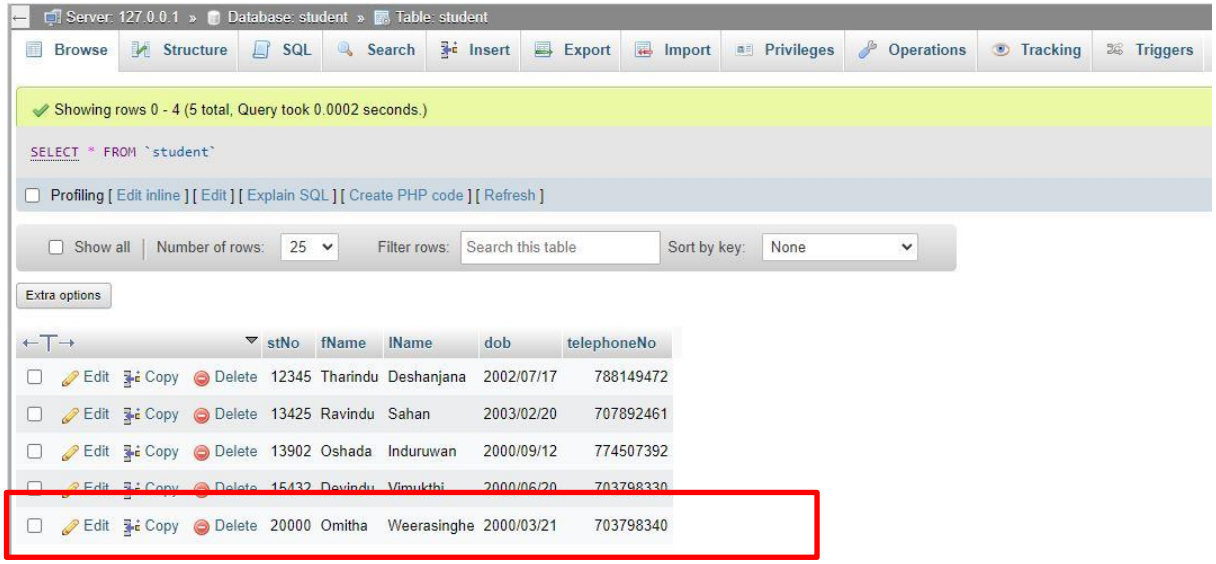

Figure 25 Test case 07 output

6.8 Updating telephone number of a student (Test case 08)

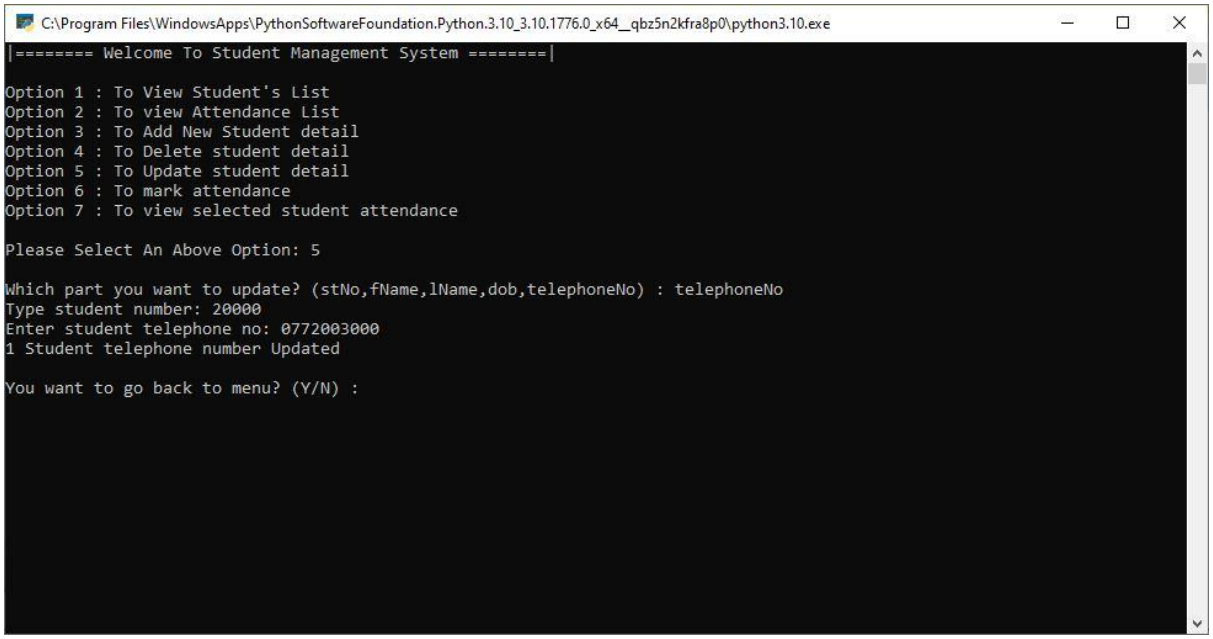

Figure 26 Test case 08 input

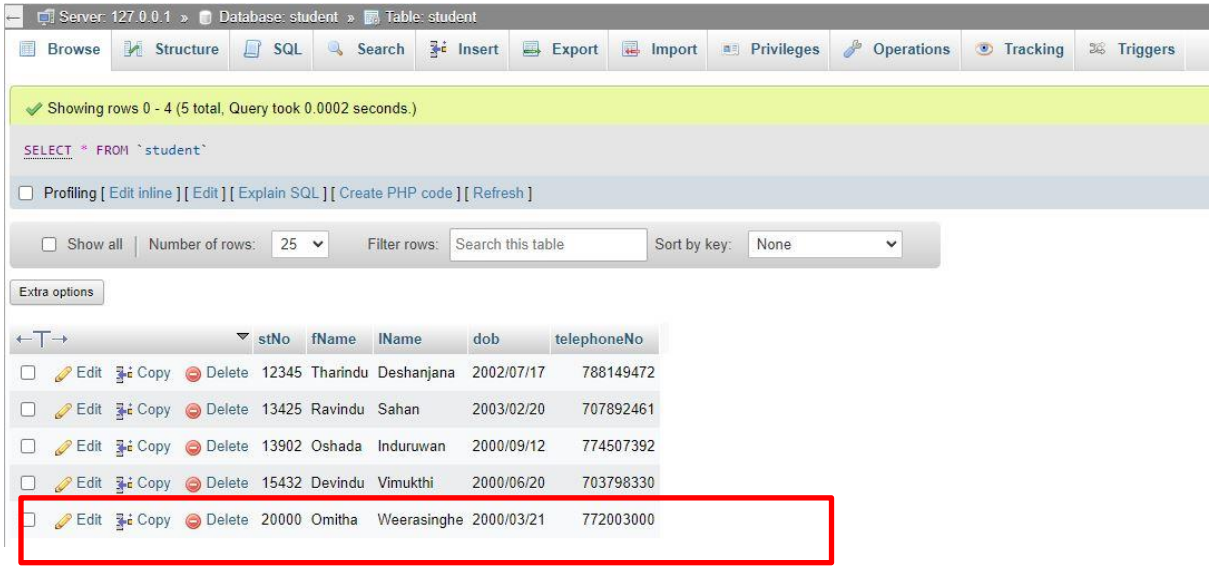

Figure 27 Test case 08 output

6.9 Marking attendance of students (Test case 09)

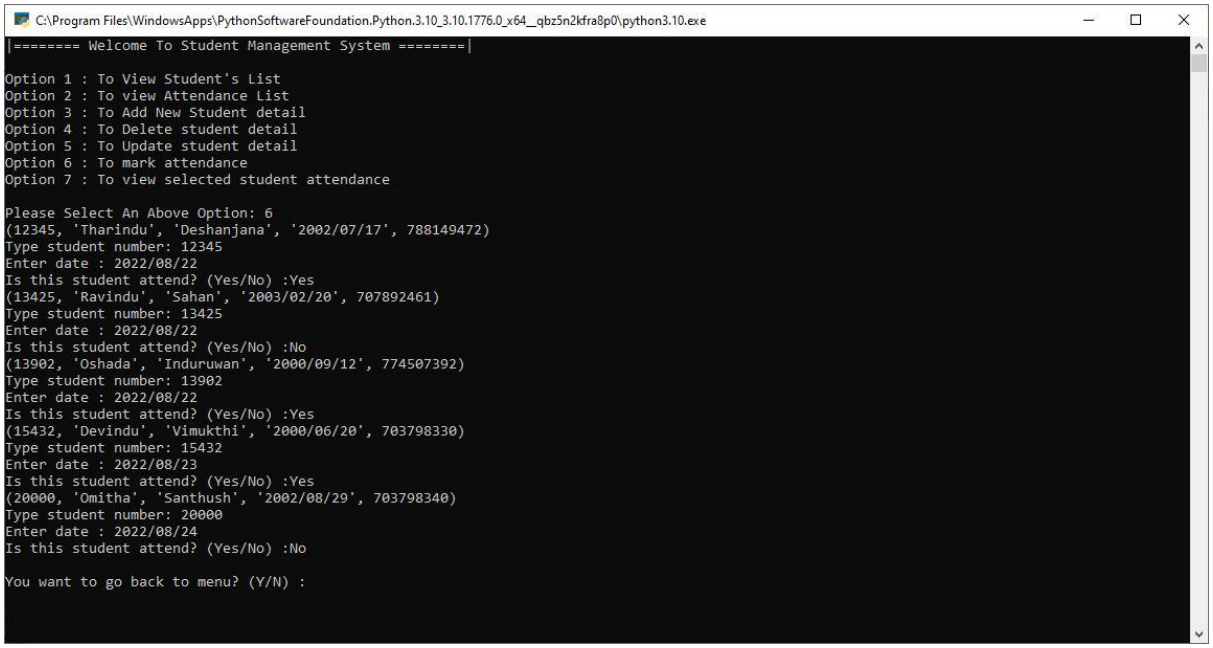

Figure 28 Test case 09 input

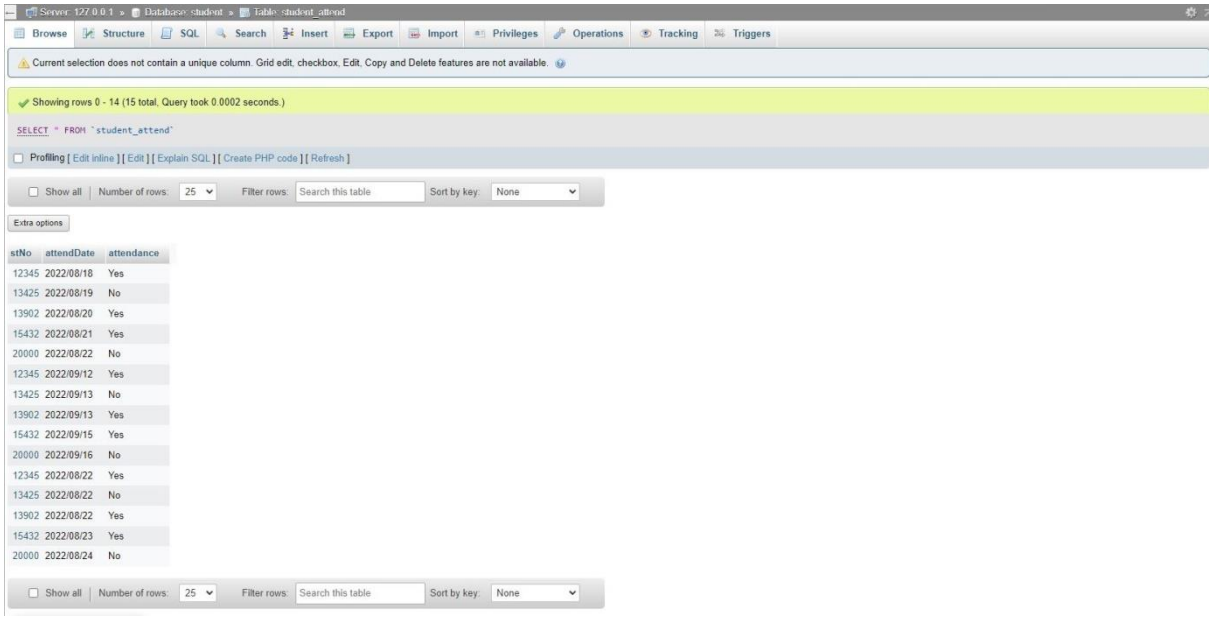

Figure 29 Test case 09 output

6.10 View attendance list of a selected student (Test case 10)

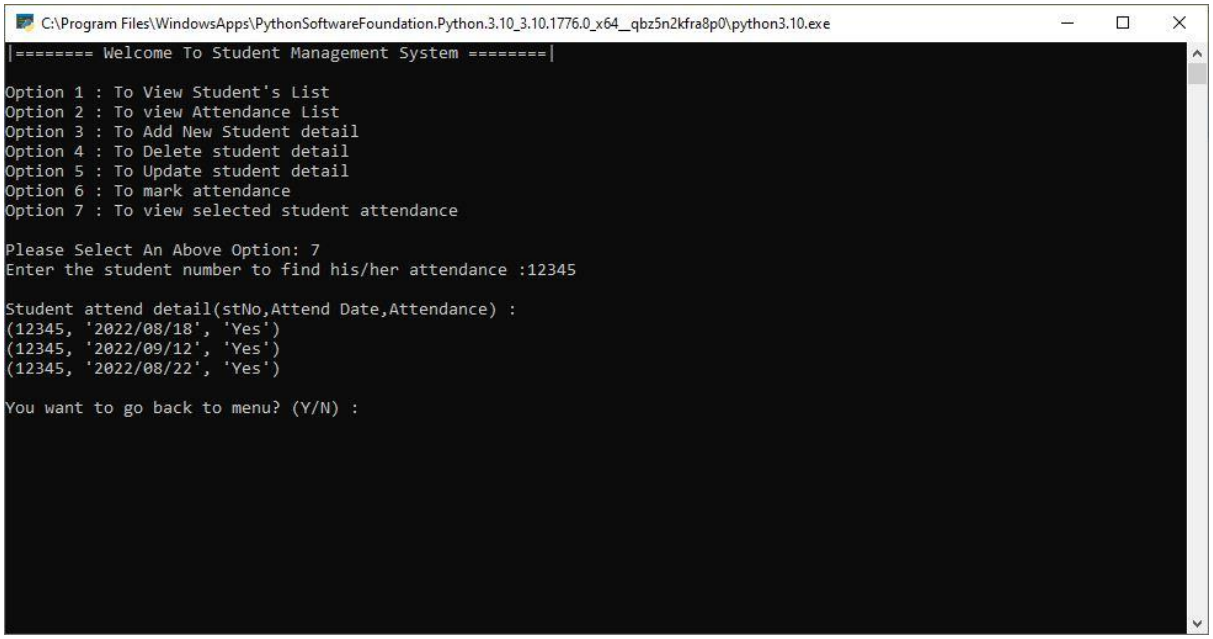

Figure 30 Test case 10 input

6.11 Run program again (Test case 11)

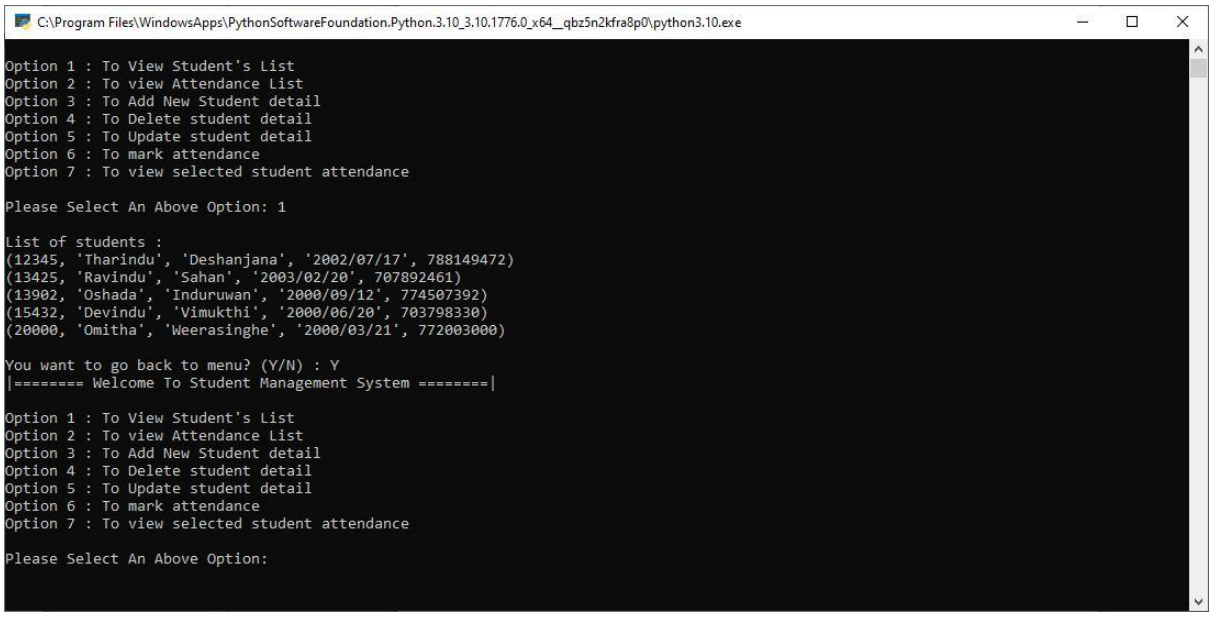

Figure 31 Test case 11