



INFORMATICS
INSTITUTE OF
TECHNOLOGY

Foundation Certificate for Higher Education

Module: DOC334 Introduction to Programming 2

Module Leader: Mr. Sudarshana Welihinda

Type of Assignment: Individual Coursework

Submission Date: 21st of August 2022

Topic: Class Attendance

Student ID	Name
20211291	Hashim Kalam

Contents

Problem Statement	4
Problem Understanding	4
Program Python Code	6
For Inserting Student Details from the First Table	23
For Entering the Attendance for Students.....	26
For Updating Student Details from the First Table	29
For Deleting Student Details from the First Table.....	31
For Viewing All Student Details.....	33
For Viewing All Attendance Details.....	35
For Viewing Attendance Details By Student Number.....	37

Tables of Figures

Figure 1: Test Case 1 → For Inserting Student Details from the First Table	24
Figure 2: Test Case 2 → For Inserting Student Details from the First Table	24
Figure 3: Test Case 3 → For Inserting Student Details from the First Table	25
Figure 4: Test Case 4 → For Inserting Student Details from the First Table	25
Figure 5: Test Case 1 → For Entering The Attendance For Students	27
Figure 6: Test Case 2 → For Entering The Attendance For Students	28
Figure 7: Test Case 1 → For Updating Student Details from the First Table	29
Figure 8: Test Case 2 → For Updating Student Details from the First Table	30
Figure 9: Test Case 1 → For Deleting Student Details from the First Table	31
Figure 10: Test Case 2 → For Deleting Student Details from the First Table	32
Figure 11: Test Case 1 → For Viewing All Student Details from the First Table	34
Figure 12: Test Case 2 → For Viewing All Student Details from the First Table	34
Figure 13: Test Case 1 → For Viewing All Attendance Details	36
Figure 14: Test Case 2 → For Viewing All Attendance Details	36
Figure 18: Test Case 1 → For Viewing Attendance Details By Student Number.....	38

Problem Statement

Designing and developing a fully transactional console-based application which has the mentioned functionalities, and they are the program operator must be able to key in new student details that includes the basic information which is the student number, first name and last name. You are also free to insert any relevant information for this entity. A minimum of 3 and a maximum of 5 students can be in the table. The second functionality is that the program operator must be able to insert a date and the attendance for that particular date. The date could be recorded as a string if intended. Lastly for this functionality, the application should record the attendance based on the available students in the system. The third functionality is that the program operator must be able to insert, update, delete students from the first table as they wish. The fourth and final functionality is, the program operator must be able to view all student details, all attendance details, and if only attendance of a selected student required separately!

Problem Understanding

It is required to develop a python program to input student details like their student number, first name and last name. There can't be more than 5 students and less than 3 students in the table! It also should have the feature that could the attendance to the available students in the table. The data must be inputted by the user for the marking of the attendance. The program must also be able to make alteration to the first table/student details. Alterations like inserting, updating, and deleting the student details. Finally, the program must have the feature to view all the student details, view all attendance details and only the attendance of a selected student with the user's choice in preferably in a tabular format.

Packages

Two types of packages were used: MySQL connector python and tabulate!

MySQL connector python – enables the python program to access to the created MySQL databases, using an API that is compliant with the Python Database API Specification v2. 0 (PEP 249). Used at the start of the program to connect the program to the database which is crucial for this project.

Installation → Typed "cmd" in the search bar, ran as administrator, hit Enter to open the command line. Then I typed “pip install mysql-connector-python” in the command line and hit Enter to install python mysql connector to the default Python installation.

Tabulate – enables logical representation of numeric data in rows and columns to facilitate comparison and statistical analysis. Used at the start, only for getting the data in a tabulate format when needed.

Installation → Typed "cmd" in the search bar, ran as administrator, hit Enter to open the command line. Then I typed “pip install tabulate” in the command line and hit Enter to install tabulate to the default Python installation.

[Actually, at the start, I faced some problems when installing the mysql-connector-python. I tried many ways to solve the SSL certificate problem but couldn't. So, at the start I installed the MySQL connector with the below mentioned link.

<https://cdn.mysql.com//Downloads/Connector-Python/mysql-connector-python-8.0.30-windows-x86-64bit.msi>

But later I managed to solve the problem and then I installed the mysql-connector again, but it said the requirement is satisfied so then I installed the tabulate by the above-mentioned installation process.]

Program Python Code

```
import mysql.connector
from tabulate import tabulate

# code for viewing student details
def viewStudentDetails():

    # Opens database connection with a dictionary
    conDict = {'host':'localhost',
               'database':'db_student_attendance',
               'user':'root',
               'password':''}

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

    # Prepare a cursor object using cursor() method
    cursor = db.cursor()

    cursor.execute("SELECT * FROM student_info")

    data = cursor.fetchall()
    print(tabulate(data, headers=[
        "\nStudent No", "\nFirst Name", "\nLast Name" ]))

    db.close()

# code to add the attendance to the available students in the first table
def addAttendance():

    try:
```

```

# Opens database connection with a dictionary
conDict = {'host':'localhost',
           'database':'db_student_attendance',
           'user':'root',
           'password':''}

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

# Prepare a cursor object using cursor() method
cursor = db.cursor()

mySQLText = "SELECT studentNo,fName FROM student_info"

cursor.execute(mySQLText)

data = cursor.fetchall()

while True:

    print("Attendance Of ",cursor.rowcount," Students Needed To Be Added. yes/no?: ",
end=' ')

    goNext = input()
    print()

    viewStudentDetails()
    print()

    if (goNext == "yes"):

        date = input("Please Enter The Date For The Attendance: ")

```

```

print("\nStudent Attendance(Absent or Present)")

i = 0

while i < cursor.rowcount:
    print(data[i][1], end=" ")
    atd = input()

    cursor1 = db.cursor()

    mySQLText1 = "INSERT INTO keeping_track (studentNo,date,attendance)
VALUES (%s,%s,%s)"

    cursor1.execute(mySQLText1, (data[i][0], date, atd))

    db.commit()
    i += 1

print("\nAttendance Added\n")
main()

elif goNext == 'no':
    main()

else:
    print("\nEnter Either 'yes' or 'no' and try again!\n")
    addAttendance()

except:
    print("ERROR\n")

```



```
db.close()
```

```
def ask_again():
```

```
    askAgain = input("Do You Want to Add One Last Time(yes or no)? ")
```

```
    print()
```

```
    if askAgain == "yes":
```

```
        # Opens database connection with a dictionary
```

```
        conDict = {'host':'localhost',
```

```
                   'database':'db_student_attendance',
```

```
                   'user':'root',
```

```
                   'password':''}
```

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

```
        # Prepare a cursor object using cursor() method
```

```
        cursor = db.cursor()
```

```
        ustudentNo = int(input("Type Student Number : "))
```

```
        while(ustudentNo<0):
```

```
            print("The Student Number cannot be Negative. Please Try Again!")
```

```
            studentNo = int(input("Type student Number : "))
```

```
        ufName = input("Type First Name : ")
```

```
        while (len(ufName)>18):
```

```
            print("Too Many Characters, Maximum 18 Characters! Try Again.")
```

```
            ufName = input("Type First Name : ")
```

```

ulName = input("Type Last Name : ")
while (len(ulName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ulName = input("Type Last Name : ")

mySQLText = "INSERT INTO student_info (studentNo, fName, lName) VALUES
(%s,%s,%s)"
myValues = (ustudentNo, ufName, ulName)
cursor.execute(mySQLText, myValues)

db.commit()
print(cursor.rowcount, "Record Added\n")
db.close()
main()

elif askAgain == "no":
    print()
    main()

else:
    print("Enter Either 'yes' or 'no'!")
    ask_again()

# code to view necessary details
def view():

    # menu for option 4
    print(" 1. View all Student Details\n"
          " 2. View all Attendance Details\n")

```

```
" 3. View only Attendance of a Selected Student\n"
" 4. Return to the Main Menu\n")
```

```
choiceNo3 = int(input("Enter the Preferred Option(1, 2, 3 or 4) : "))
```

```
if (choiceNo3 == 1):
```

```
    viewStudentDetails()
    print()
    view()
```

```
elif (choiceNo3 == 2):
```

```
    # Opens database connection with a dictionary
    conDict = {'host':'localhost',
               'database':'db_student_attendance',
               'user':'root',
               'password':''}
```

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

```
    # Prepare a cursor object using cursor() method
    cursor = db.cursor()
```

```
    cursor.execute("SELECT * FROM keeping_track")
```

```
    data = cursor.fetchall()
    print(tabulate(data, headers=[
        "\nStudent No", "\nDate", "\nAttendance" ]))
```

```
    print()
```

```

db.close()
view()

elif (choiceNo3 == 3):

    # Opens database connection with a dictionary
    conDict = {'host':'localhost',
               'database':'db_student_attendance',
               'user':'root',
               'password':''}

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

    # Prepare a cursor object using cursor() method
    cursor = db.cursor()
    print()

    ustudentNo = input("Type the Students Number to get their Attendance : ")
    print()

    cursor.execute("SELECT * FROM keeping_track WHERE studentNo = " + ustudentNo)

    data = cursor.fetchall()
    print(tabulate(data, headers=[
        "\nStudent No", "\nDate", "\nAttendance" ]))

    print()
    db.close()
    view()

elif (choiceNo3 == 4):

```

```

print()
main()

else:
    print("\nEnter Either 1, 2, 3, or 4! Try Again\n")
    view()

# code for updating, inserting and deleting student details
def sub():

    choiceNo2 = input("Which one?; Insert, Update or Delete Students? Else, Type 'Back' to go to
the Main Menu: ")
    print()

    if (choiceNo2 == "Insert"):

        # Opens database connection with a dictionary
        conDict = {'host':'localhost',
                    'database':'db_student_attendance',
                    'user':'root',
                    'password':''}

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

        # Prepare a cursor object using cursor() method
        cursor = db.cursor()

        ustudentNo = int(input("Type Student Number : "))
        while(ustudentNo<0):
            print("The Student Number Cannot Be Negative. Please Try Again!")
            ustudentNo = int(input("Type Student Number : "))

```

```

ufName = input("Type First Name : ")
while (len(ufName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ufName = input("Type First Name : ")

ulName = input("Type Last Name : ")
while (len(ulName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ulName = input("Type Last Name : ")

mySQLText = "INSERT INTO student_info (studentNo, fName, lName) VALUES (%s,
%s, %s)"
myValues = (ustudentNo, ufName, ulName)
cursor.execute(mySQLText, myValues)

db.commit()

print(cursor.rowcount, "Record Added\n")

db.close()
main()

elif (choiceNo2 == "Update"):

# Opens database connection with a dictionary
conDict = {'host':'localhost',
           'database':'db_student_attendance',
           'user':'root',
           'password':''}

```

```

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

# Prepare a cursor object using cursor() method
cursor = db.cursor()

ustudentNo = input("Type Student Number : ")

ufName = input("Type First Name : ")
while (len(ufName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ufName = input("Type First Name : ")

ulName = input("Type last name : ")
while (len(ulName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ulName = input("Type Last Name : ")

updTxt = "UPDATE student_info SET fName = '" + ufName + "', lName = '" + ulName + "'
WHERE studentNo = " + ustudentNo
cursor.execute(updTxt)

db.commit()

print(cursor.rowcount, "Record Updated\n")

db.close()
main()

elif (choiceNo2 == "Delete"):

# Opens database connection with a dictionary

```

```

conDict = {'host':'localhost',
           'database':'db_student_attendance',
           'user':'root',
           'password':''}

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

# Prepare a cursor object using cursor() method
cursor = db.cursor()

ustudentNo = input("Type a student number to delete : ")

cursor.execute("DELETE FROM student_info WHERE studentNo = " + ustudentNo + "")

db.commit()

print(cursor.rowcount, "Record Deleted\n")

db.close()
main()

elif (choiceNo2 == "Back"):
    main()

else:
    print("Enter either 'Insert', 'Update', 'Delete' Or 'Back'\n")
    sub()

# code for inserting the student details
def newDetails():

```



```
t1Stu = 0
```

```
# Opens database connection with a dictionary
```

```
conDict = {'host':'localhost',  
           'database':'db_student_attendance',  
           'user':'root',  
           'password':''}
```

```
while t1Stu<3:
```

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

```
    # Prepare a cursor object using cursor() method
```

```
    cursor = db.cursor()
```

```
    ustudentNo = int(input("Type student number : "))
```

```
    while(ustudentNo<0):
```

```
        print("The Student Number Cannot Be Negative. Please Try Again!")
```

```
        ustudentNo = int(input("Type student number : "))
```

```
    ufName = str(input("Type first name : "))
```

```
    while (len(ufName)>18):
```

```
        print("Too Many Characters, Maximum 18 Characters! Try Again.")
```

```
        ufName = str(input("Type first name : "))
```

```
    ulName = input("Type last name : ")
```

```
    while (len(ulName)>18):
```

```
        print("Too Many Characters, Maximum 18 Characters! Try Again.")
```

```
        ulName = input("Type last name : ")
```

```

mySQLText = "INSERT INTO student_info (studentNo, fName, lName) VALUES
(%s,%s,%s)"

myValues = (ustudentNo, ufName, ulName)
cursor.execute(mySQLText, myValues)

db.commit()
print(cursor.rowcount, "Record Added\n")
db.close()
ttlStu += 1

addMore = input("Do You Want Enter More Student Details (yes or no)? ")
print()

if addMore == "yes":

    howMany = int(input("How Many More Student Details Do You Want To Add? 1 or 2?"))
    print()

    if howMany == 1:

        # Opens database connection with a dictionary
        conDict = {'host':'localhost',
                    'database':'db_student_attendance',
                    'user':'root',
                    'password':''}

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

        # Prepare a cursor object using cursor() method
        cursor = db.cursor()

```

```

ustudentNo = int(input("Type Student Number : "))
while(ustudentNo<0):
    print("The Student Number Cannot Be Negative. Please Try Again!")
    ustudentNo = int(input("Type Student Number : "))

ufName = input("Type First Name : ")
while (len(ufName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ufName = input("Type First Name : ")

ulName = input("Type Last Name : ")
while (len(ulName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ulName = input("Type Last Name : ")

mySQLText = "INSERT INTO student_info (studentNo, fName, lName) VALUES
(%s,%s,%s)"
myValues = (ustudentNo, ufName, ulName)
cursor.execute(mySQLText, myValues)

db.commit()
print(cursor.rowcount, "Record Added\n")
db.close()

ask_again()
print()
main()

elif (howMany == 2):

    while howMany > 0:

```

```

# Opens database connection with a dictionary
conDict = {'host':'localhost',
           'database':'db_student_attendance',
           'user':'root',
           'password':''}

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

# Prepare a cursor object using cursor() method
cursor = db.cursor()

ustudentNo = int(input("Type Student Number : "))
while(ustudentNo<0):
    print("The Student Number Cannot Be Negative. Please Try Again!")
    ustudentNo = int(input("Type Student Number : "))

ufName = input("Type First Name : ")
while (len(ufName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ufName = input("Type First Name : ")

ulName = input("Type Last Name : ")
while (len(ulName)>18):
    print("Too Many Characters, Maximum 18 Characters! Try Again.")
    ulName = input("Type Last Name : ")

mySQLText = "INSERT INTO student_info (studentNo, fName, lName) VALUES
(%s,%s,%s)"

myValues = (ustudentNo, ufName, ulName)
cursor.execute(mySQLText, myValues)

```

```

        db.commit()
        print(cursor.rowcount, "Record Added\n")
        db.close()

        howMany-=1
        main()
    else:
        print("You Can Not Enter Details Of More Than 5 Students In Total!")

    else:
        main()

# code for the main menu
def main():

    print("STUDENT ATTENDANCE\n")

    print("1. Key In New Details.")
    print("2. Entering The Attendance.")
    print("3. Insert, Update, Delete Students Details From The First Table.")
    print("4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.")
    print("5. Exit\n")

    choice = int(input("Enter Your Preferred Option : "))
    print()

    if (choice == 1):
        newDetails()

```

```
elif (choice == 2):
    addAttendance()

elif (choice == 3):
    sub()

elif (choice == 4):
    view()

elif (choice == 5):
    exit()

else:
    print("Invalid Input! Please Enter Either 1, 2, 3, 4 or 5.\n")
    main()

main()
```

For Inserting Student Details from the First Table

Test Cases #	Input		Expected Output		Actual Output	Remarks
	Student no.	First name	Last name			
1	-5	-	-	The Student Number Cannot Be Negative. Please Try Again!	The Student Number Cannot Be Negative. Please Try Again!	Test Case Pass
2	1234	hjoncwon ecjnnweo cnwiocni wen	-	Too Many Characters, Maximum 18 Characters! Try Again.	Too Many Characters, Maximum 18 Characters! Try Again.	Test Case Pass
3	1234	John	qwerta sdfgzx cvbnht y	Too Many Characters, Maximum 18 Characters! Try Again.	Too Many Characters, Maximum 18 Characters! Try Again.	Test Case Pass
4	1234	John	Doe	1 Record Added	1 Record Added	Test Case Pass

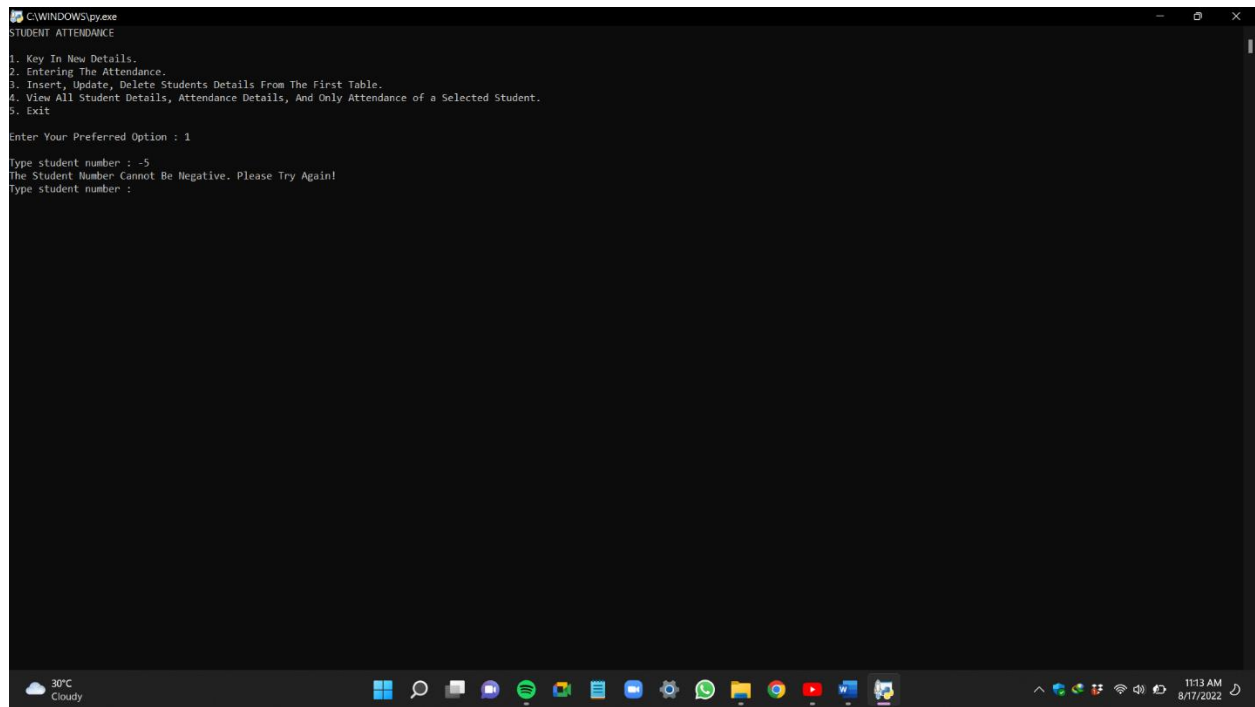


Figure 1: Test Case 1 → For Inserting Student Details from the First Table

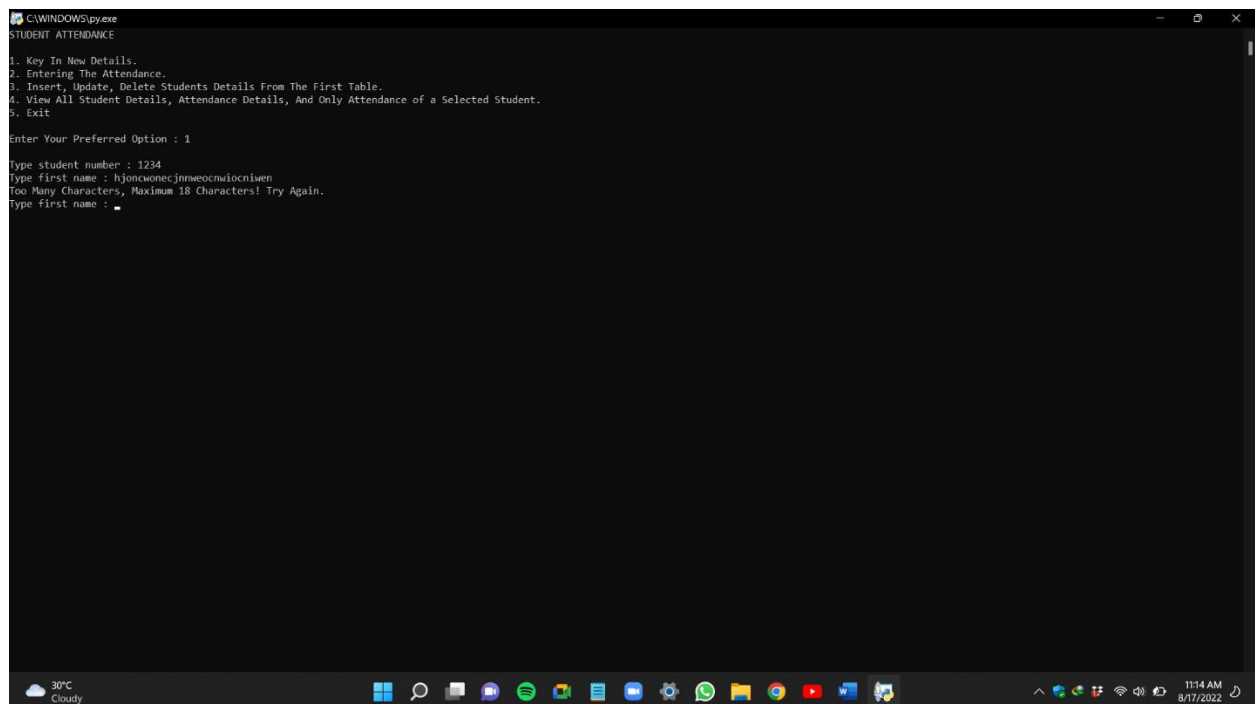


Figure 2: Test Case 2 → For Inserting Student Details from the First Table

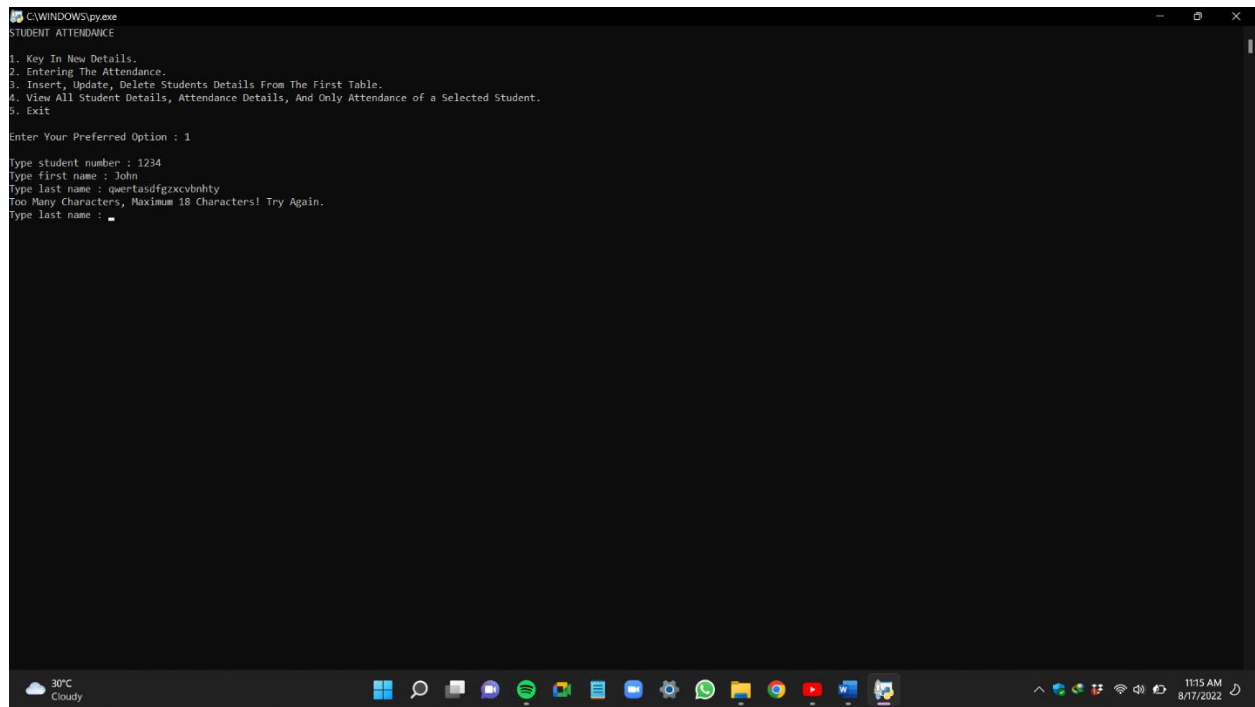


Figure 3: Test Case 3 → For Inserting Student Details from the First Table

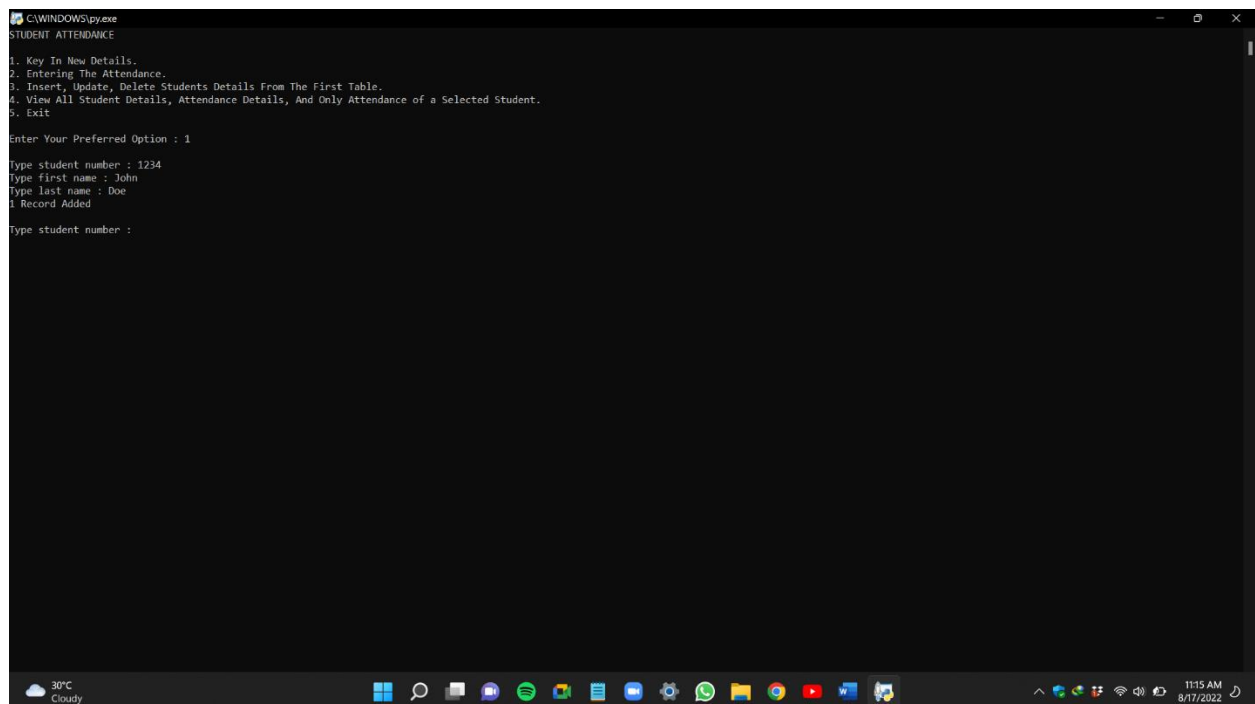


Figure 4: Test Case 4 → For Inserting Student Details from the First Table

For Entering the Attendance for Students

Test Cases #	Input	Expected Output			Actual Output	Remarks
	Step wise					
1	2; yes; 8/15/2021; Present; Present; Absent; Present; Absent.	Attendance Added			Attendance Added	Test Case Pass
2	2; no	Student No ----- 1234 4321 1241 2143 2131	First Name ----- John Malik Nial Larry Nazim	Last Name ----- DOe Zak Hor Khan Kalm	*same as Expected Output*	
STUDENT ATTENDANCE						
1. Key In New Details.						
2. Entering The Attendance.						
3. Insert, Update, Delete Students Details From The First Table.						
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.						

		<p>5. Fetch All The Entered New Details In A Tabular Format</p> <p>6. Exit</p> <p>Enter Your Preferred Option :</p>		
--	--	---	--	--

```

C:\WINDOWS\py.exe
STUDENT ATTENDANCE
1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 2

Attendance Of 5 Students Needed To Be Added. yes/no?: yes

Student No  First Name  Last Name
-----
1234  John  Doe
4321  Malik  Zak
1241  Nial  Hor
2143  Larry  Khan
2131  Nazim  Kalm

Please Enter The Date For The Attendance: 8/15/2021

Student Attendance(Absent or Present)
John Present
Malik Present
Nial Absent
Larry Present
Nazim Absent

Attendance Added

STUDENT ATTENDANCE
1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option :

```

Figure 5: Test Case 1 → For Entering The Attendance For Students

```
Select C:\WINDOWS\py.exe
STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 2

Attendance Of 5 Students Needed To Be Added. yes/no?: no

Student No  First Name  Last Name
-----
1234 John      Doe
4321 Malik    Zak
1241 Nial     Hor
2143 Larry    Khan
2131 Nazim    Kalm

STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option :
```

Figure 6: Test Case 2 → For Entering The Attendance For Students

For Updating Student Details from the First Table

Test Cases #		Input			Expected Output	Actual Output	Remarks
	Step wise	Student no.	First name	Last name			
1	3; 12; Zayn; Malik	12	Zayn	Malik	0 Record Updated	0 Record Updated	Test Case Pass
2	3; 1234; Hashim; Kalam	1234	Hashim	Kalam	1 Record Updated	1 Record Updated	Test Case Pass

```

C:\WINDOWS\py.exe
STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 3

Which one?; Insert, Update or Delete Students? Else, Type 'Back' to go to the Main Menu: Update

Type Student Number : 12
Type First Name : Zayn
Type last name : Malik
0 Record Updated

STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 3
  
```

Figure 7: Test Case 1 → For Updating Student Details from the First Table

```
C:\WINDOWS\py.exe
STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 3

Which one?, Insert, Update or Delete Students? Else, Type 'Back' to go to the Main Menu: Update

Type Student Number : 1234
Type First Name : Hashim
Type last name : Kalam
1 Record Updated

STUDENT ATTENDANCE

1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option :
```

Figure 8: Test Case 2 → For Updating Student Details from the First Table

For Deleting Student Details from the First Table

Test Cases #		Input	Expected Output	Actual Output	Remarks
	step wise	Student no			
1	3; 12	12	0 Record Deleted	0 Record Deleted	Test Case Pass
2	3; 1234	1234	1 Record Deleted	1 Record Deleted	Test Case Pass

```

C:\WINDOWS\py.exe
STUDENT ATTENDANCE
1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit
Enter Your Preferred Option : 3
Which one? Insert, Update or Delete Students? Else, Type 'Back' to go to the Main Menu: Delete
Type a student number to delete : 12
0 Record Deleted
STUDENT ATTENDANCE
1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details From The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit
Enter Your Preferred Option : _
  
```

Figure 9: Test Case 1 → For Deleting Student Details from the First Table

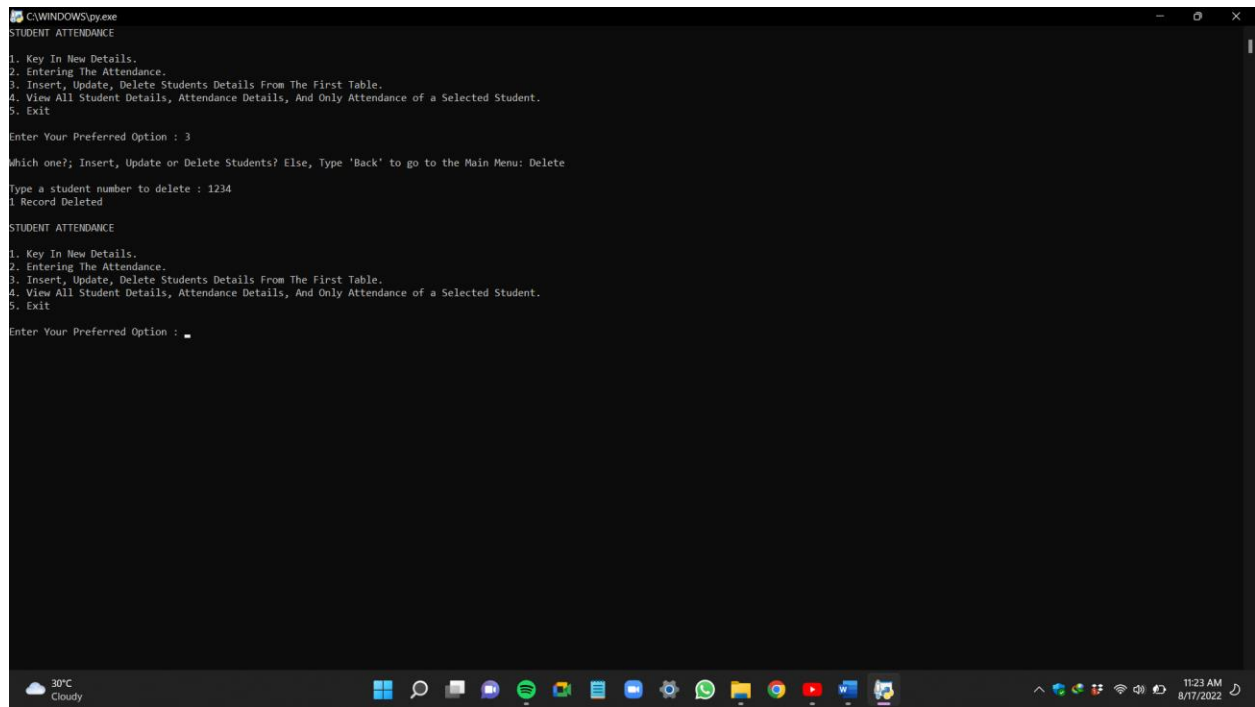


Figure 10: Test Case 2 → For Deleting Student Details from the First Table

For Viewing All Student Details

Test Cases #	Input	Expected Output	Actual Output	Remarks
	Step wise			
1	4; 8	Enter Either 1, 2, 3, or 4! Try Again 1. View all Student Details 2. View all Attendance Details 3. View only Attendance of a Selected Student 4. Return to the Main Menu Enter the Preferred Option(1, 2, 3 or 4) :	*same as Expected Output*	Test Case Pass
2	4; 1	Student No First Name Last Name ----- ----- ----- 4321 Malik Zak 1241 Nial Hor 2143 Larry Khan 2131 Nazim Kalm 1. View all Student Details 2. View all Attendance Details 3. View only Attendance of a Selected Student 4. Return to the Main Menu Enter the Preferred Option(1, 2, 3 or 4) :	*same as Expected Output*	Test Case Pass

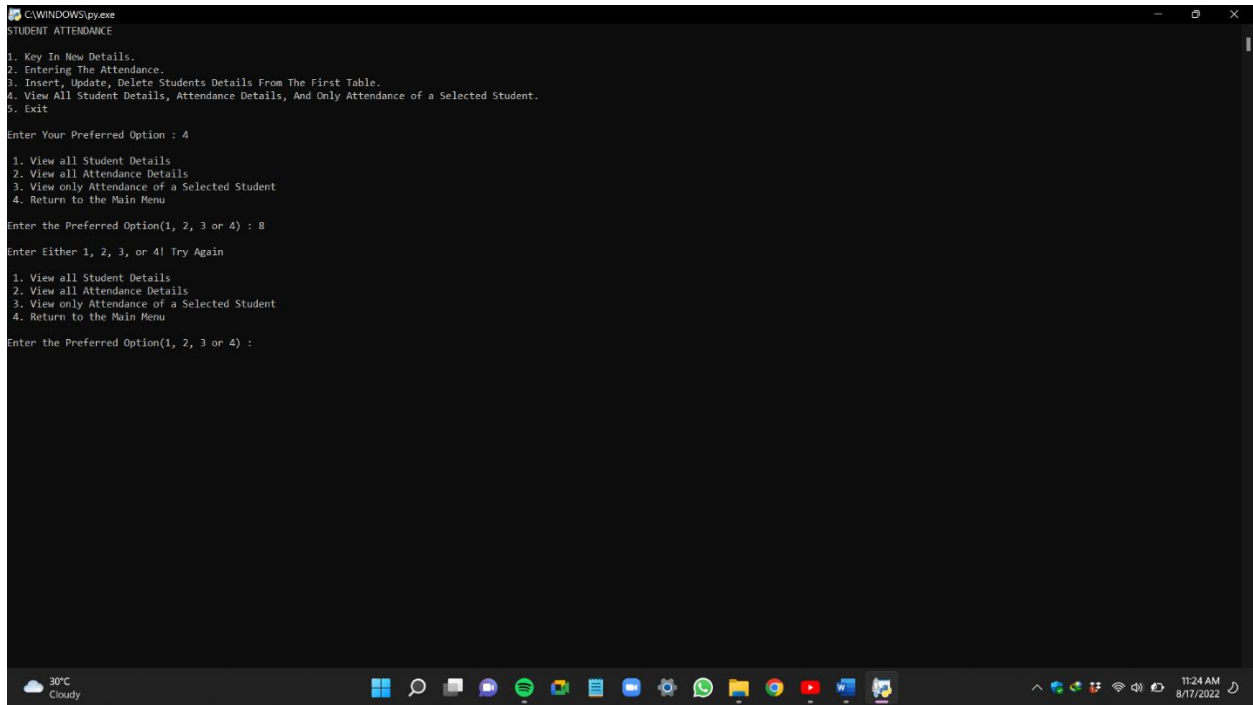


Figure 11: Test Case 1 → For Viewing All Student Details from the First Table

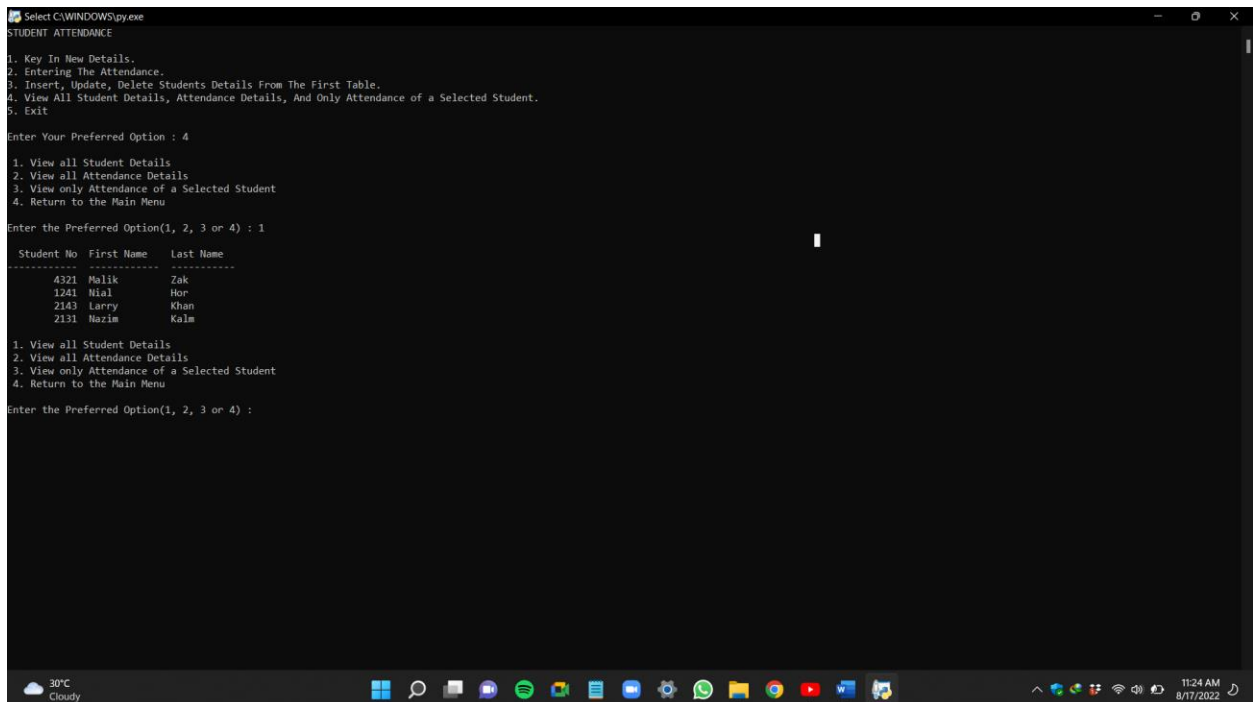
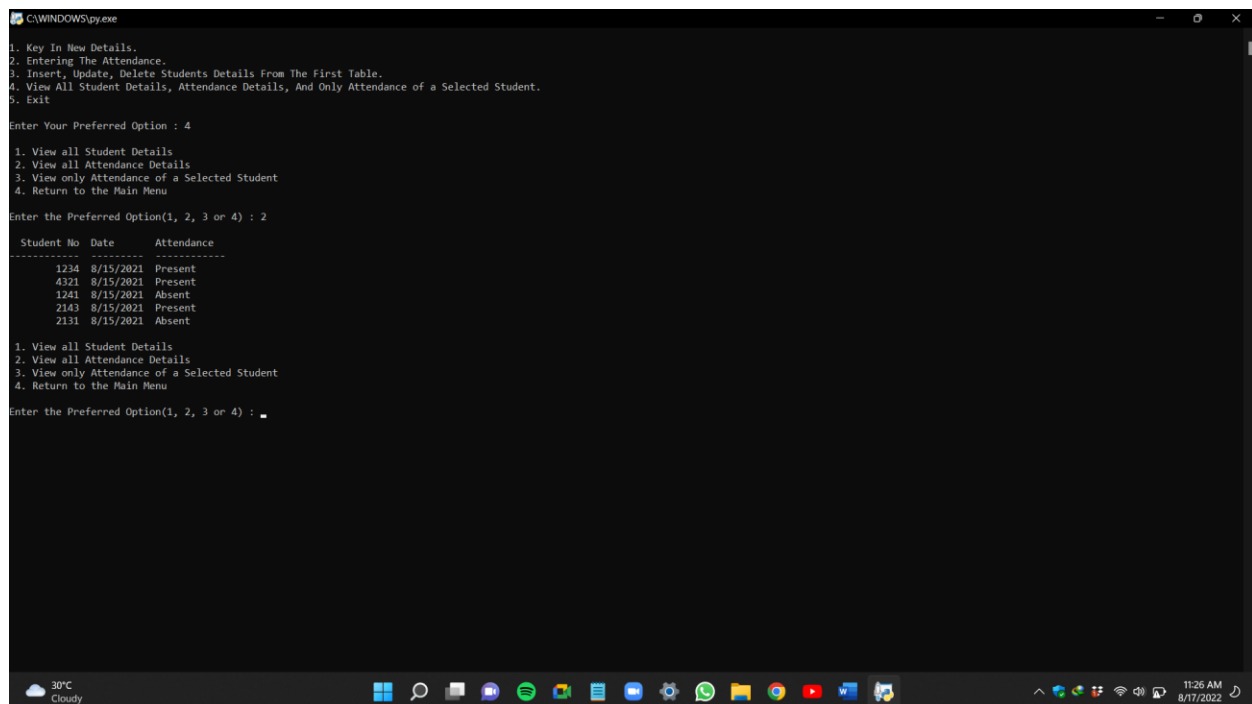
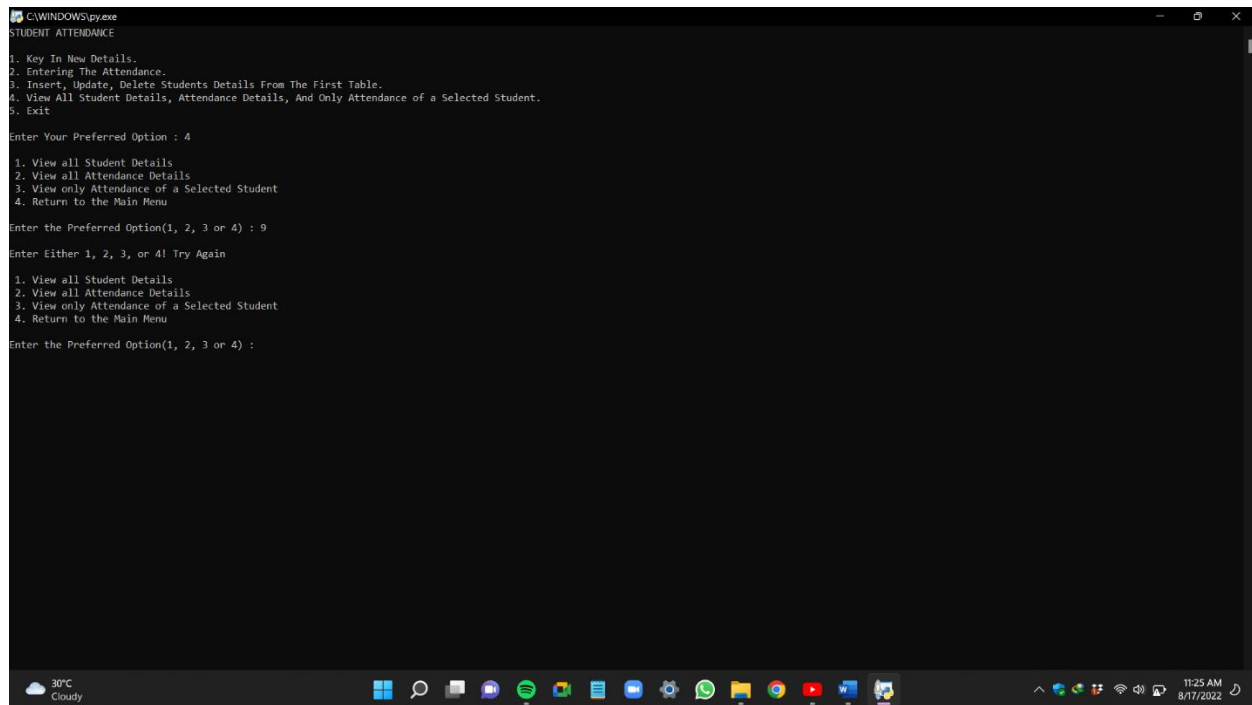


Figure 12: Test Case 2 → For Viewing All Student Details from the First Table

For Viewing All Attendance Details

Test Cases #	Input	Expected Output	Actual Output	Remarks																					
	Step wise																								
1	4; 9	Enter Either 1, 2, 3, or 4! Try Again 1. View all Student Details 2. View all Attendance Details 3. View only Attendance of a Selected Student 4. Return to the Main Menu Enter the Preferred Option(1, 2, 3 or 4) :	*same as Expected Output*	Test case pass																					
2	4; 2	<table><tr><td>Student No</td><td>Date</td><td>Attendance</td></tr><tr><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>1234</td><td>8/15/2021</td><td>Present</td></tr><tr><td>4321</td><td>8/15/2021</td><td>Present</td></tr><tr><td>1241</td><td>8/15/2021</td><td>Absent</td></tr><tr><td>2143</td><td>8/15/2021</td><td>Present</td></tr><tr><td>2131</td><td>8/15/2021</td><td>Absent</td></tr></table> 1. View all Student Details 2. View all Attendance Details 3. View only Attendance of a Selected Student 4. Return to the Main Menu Enter the Preferred Option(1, 2, 3 or 4) :	Student No	Date	Attendance	-----	-----	-----	1234	8/15/2021	Present	4321	8/15/2021	Present	1241	8/15/2021	Absent	2143	8/15/2021	Present	2131	8/15/2021	Absent	*same as Expected Output*	Test case pass
Student No	Date	Attendance																							
-----	-----	-----																							
1234	8/15/2021	Present																							
4321	8/15/2021	Present																							
1241	8/15/2021	Absent																							
2143	8/15/2021	Present																							
2131	8/15/2021	Absent																							



For Viewing Attendance Details By Student Number

Test Cases #	Input	Expected Output			Actual Output	Remarks
	Step wise	Student number				
1	4; 3; 2143	2143	Student No	Date	Attendance	*same as Expected Output*
			-----	-----	-----	
			2143	8/15/2021	Present	
			1. View all Student Details			
			2. View all Attendance Details			
			3. View only Attendance of a Selected Student			
			4. Return to the Main Menu			
			Enter the Preferred Option(1, 2, 3 or 4)			
			:			

```
C:\WINDOWS\py.exe
1. Key In New Details.
2. Entering The Attendance.
3. Insert, Update, Delete Students Details from The First Table.
4. View All Student Details, Attendance Details, And Only Attendance of a Selected Student.
5. Exit

Enter Your Preferred Option : 4

1. View all Student Details
2. View all Attendance Details
3. View only Attendance of a Selected Student
4. Return to the Main Menu

Enter the Preferred Option(1, 2, 3 or 4) : 3

Type the Students Number to get their Attendance : 2143

Student No Date Attendance
-----
2143 8/15/2021 Present

1. View all Student Details
2. View all Attendance Details
3. View only Attendance of a Selected Student
4. Return to the Main Menu

Enter the Preferred Option(1, 2, 3 or 4) :
```

Figure 15: Test Case 1 → For Viewing Attendance Details By Student Number