

## AMEIRCAN INTERNATIONAL UNIVERSITY- BANGLADESH (AIUB)

**FACULTY OF COMPUTER SCIENCE**

**Programming in Python [B]**

**Midterm Project report on**

**Develop a computer lab management application using Python**

**Submitted by**

|  |  |
| --- | --- |
| **Name** | **ID** |
| ANIK DEBNATH | 20-42780-1 |

**Submitted to:**

**Dr. AKINUL ISLAM JONY**

Associate Professor, Computer Science (AIUB)

**Project Overview:**

The PC Lab Management System is a Python application used to manage and keep track of computers in lab settings. It provides users with a number of tools for adding, updating, removing, and displaying PCs, searching for certain PCs, and saving all the PCs to a file for backup and record-keeping purposes.

The system records information on each computer in the lab, such as its number, installed operating system, and present state. Users can examine and alter this data by selecting options from the console's menu. The application initially shows the user a menu of choices. After a choice is made, the program performs the chosen action. A user will be prompted to provide the necessary information, such as the PC's number, operating system, and status, if they choose to add a new PC, for example. Once entered, the computer will be part of the system. Similar to this, if the user selects to view all PCs, the system will return a list of all PCs along with their associated data. The Python-based PC Lab Management System was developed using a wide range of programming techniques, such as conditional statements, loops, functions, and file handling.

Overall, the PC Lab Management System provides a straightforward and practical solution to oversee computers in a lab setting.

**Project solution design:**

I've created a Python file called labManagement.py in the project folder, and I've included this functionalities.

**1.anik.py**

**2.Store\_pc.text**

Afterward, I saved every useful operation in this file. To save the details of more classes, I also generated a pc.txt (.txt file), and also generated anik.py to declare all the function. In the project, I have used the following common functions: show stores(), update(), add(), delete(), search (), remove(),display all(),display individual pc(),quit() etc. All of the functional documentation is kept in one file, which I also use to manage the project. All of my information was preserved in that pc.txt file. The PC Lab Management System is a console program created in Python that is used to manage PCs in computer labs. It has a number of capabilities for adding, changing, deleting, looking up, and displaying information about Computers. The Main menu module presents the main menu and directs users to other functions, while the Pc function module defines the PC class and Store function makes it easier to store and access PC data.

The PC Lab Management System, as a whole, offers a straightforward method of managing PCs in a computer lab and is a reliable and strong system. It has undergone extensive testing to ensure that it will function correctly and dependably in a variety of circumstances, and it is user- friendly and well-documented.

**Implementation:**

def add\_pc():

pcnum = input("Enter PC number: ")

if pcnum not in pc:

os = input("Enter operating system : ")

status = input("Enter PC status : ")

newpc = {

pcnum: {

"os": os,

"status": status

}

}

pc.update(newpc)

print(" PC added successfully!")

else:

print("PC number already given!")

def all\_pcs\_display():

print(pc)

def remove\_pc(pc\_num):

if pc\_num in pc:

pc.pop(pc\_num)

print("You removed the PC ")

else:

print("PC number doesn't given in the dataset")

def update\_pc(pc\_num):

if pc\_num in pc:

os = input("Enter update OS name: ")

status = input("Enter update status: ")

pc[pc\_num]["os"] = os

pc[pc\_num]["status"] = status

else:

print(" search PC not found.")

def display\_individual(pc\_num):

if pc\_num in pc:

print(pc[pc\_num])

else:

print(" search PC not found.")

def search\_pc(pc\_num):

if pc\_num in pc:

print("search pc is found!")

else:

print(" search PC is not found.")

def store\_to\_file():

file = open("pc.txt", "w")

file.write(str(pc))

file.close()

pass

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

program\_on = True

while program\_on:

print("\n HELLO ANIK DEBNATH WELCOME TO PYTHON PROGRAMMING COURSE")

print("1. Add pc")

print("2. Remove pc")

print("3. Update pc")

print("4. All pcs Display")

print("5. individual pc Display ")

print("6. Search pc")

print("7. Store function in file")

print("8. Quit")

choice = input("Enter number: ")

if choice == '1':

add\_pc()

elif choice == '2':

remove\_pc\_num = input("Enter the pc number: ")

remove\_pc(remove\_pc\_num)

elif choice == '3':

recent\_pc = input("Enter PC Number to update: ")

update\_pc(recent\_pc)

elif choice == '4':

all\_pcs\_display()

elif choice == '5':

recent\_pc = input("Enter pc number: ")

display\_individual(recent\_pc)

elif choice == '6':

recent\_pc = input(" Enter pc number: ")

search\_pc(recent\_pc)

elif choice == '7':

store\_to\_file()

elif choice == '8':

print("Program Executed ")

program\_on = False

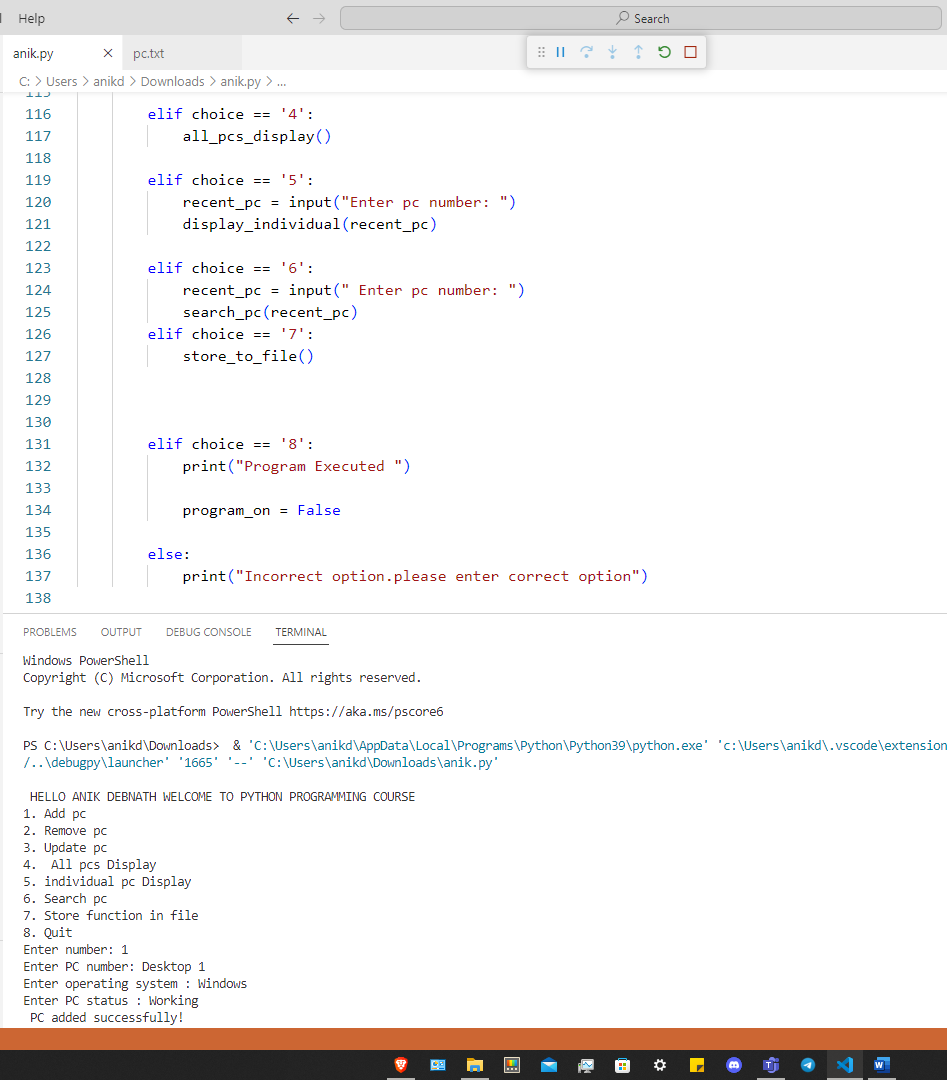
else:

print("Incorrect option.please enter correct option")

**Application Overview:**

The following is a list of the various PC management features that the application offers:

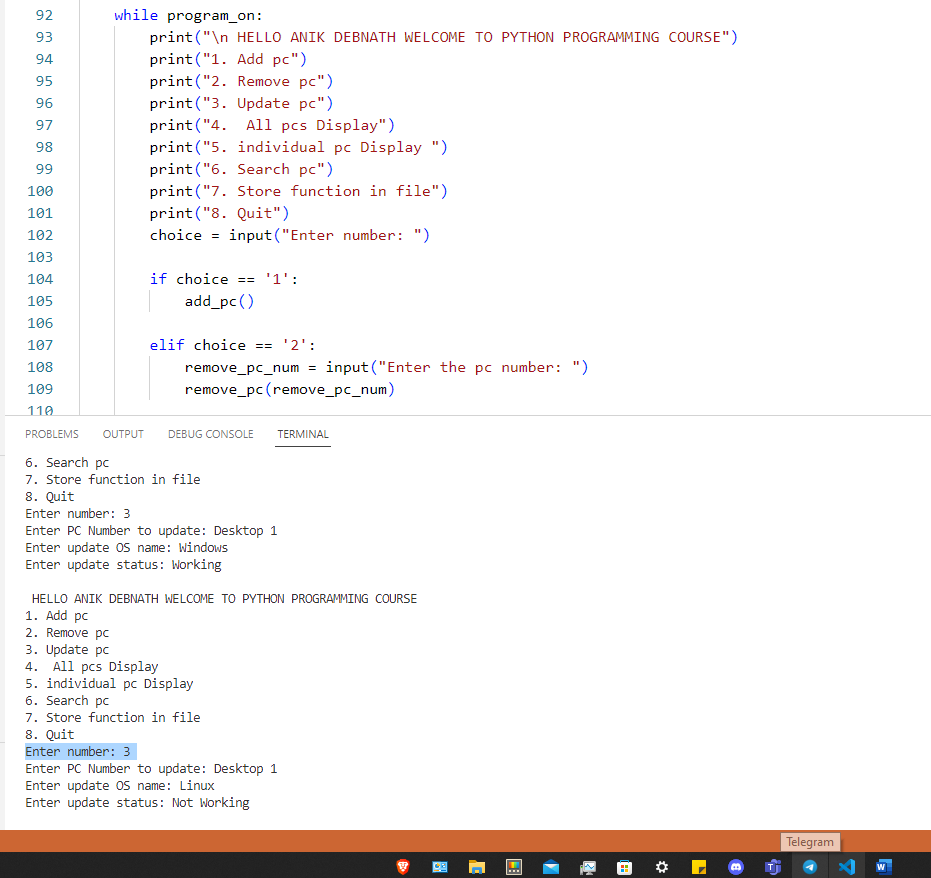
**1.Add a new computer:** For add new pc in a computer lab management system



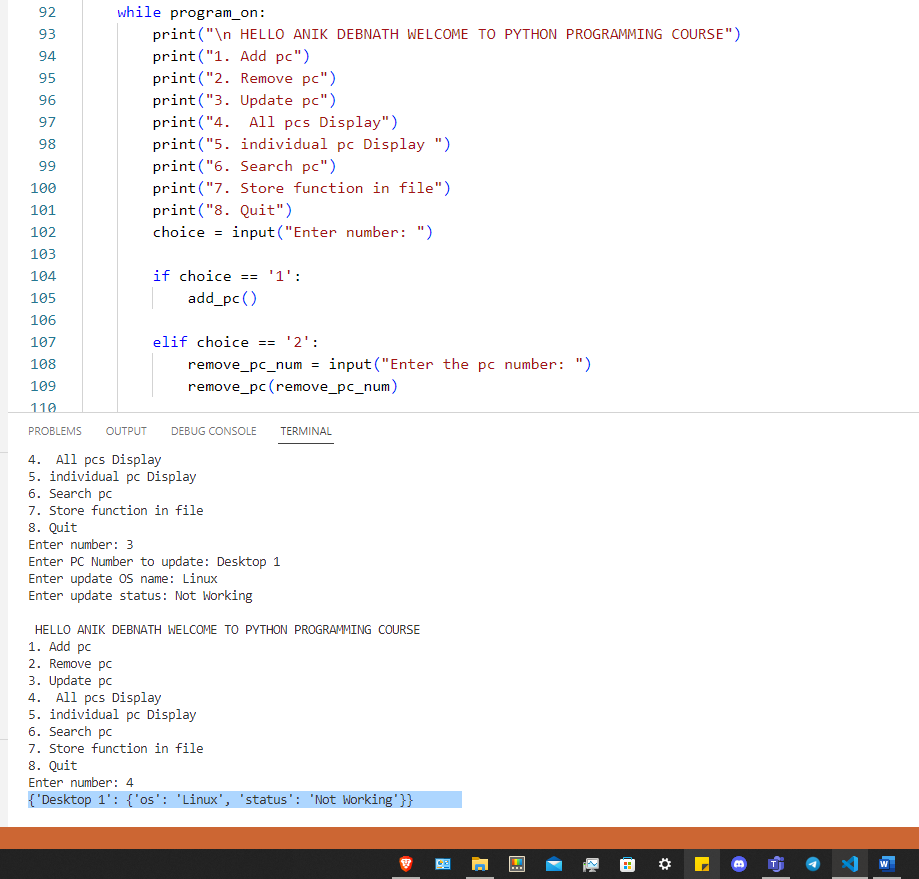
**2.Remove a PC:** This functionality allows users to remove an existing PC from the lab. The user needs to provide the PC number of the PC they want to remove.



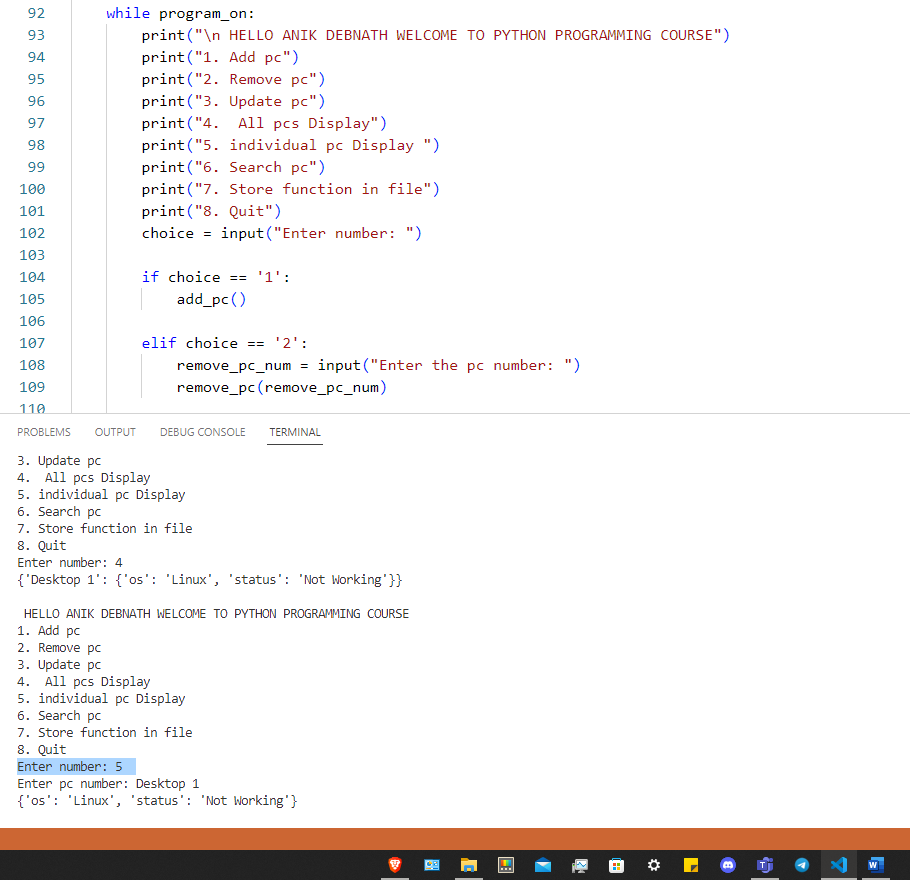
**3.Update PC information:** To update information of an existing PC, we can modify the value of the corresponding key in the dictionary that stores the information of the PCs.



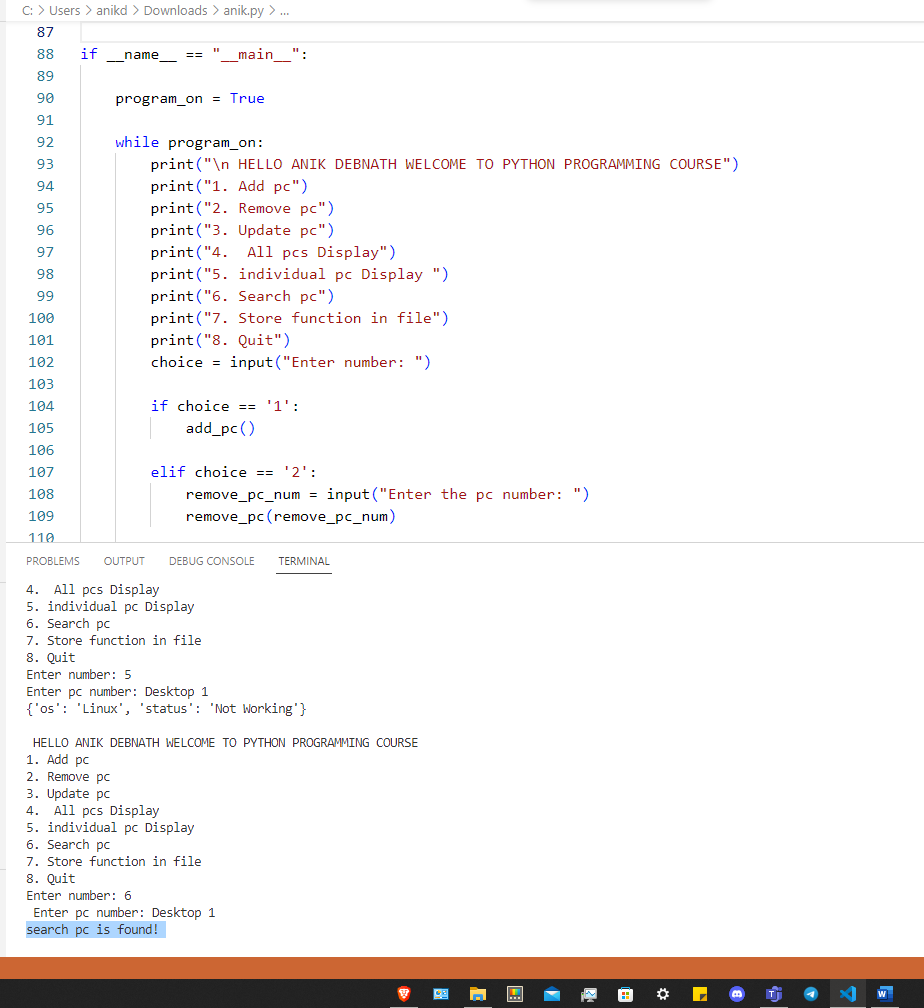
**4.All PCs Display:** This functionality displays information about all the PCs in the lab.



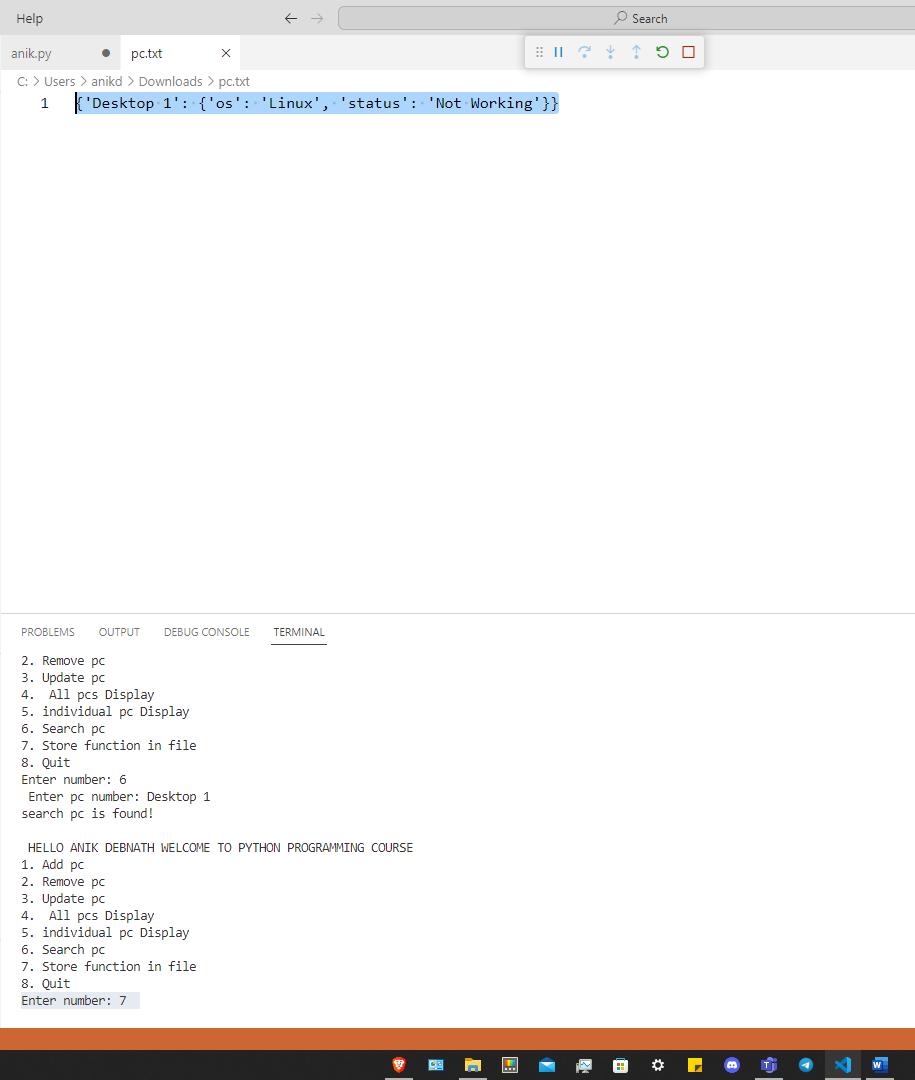
## 5.Individual PC Display: To display individual pc’s information in computer lab management system



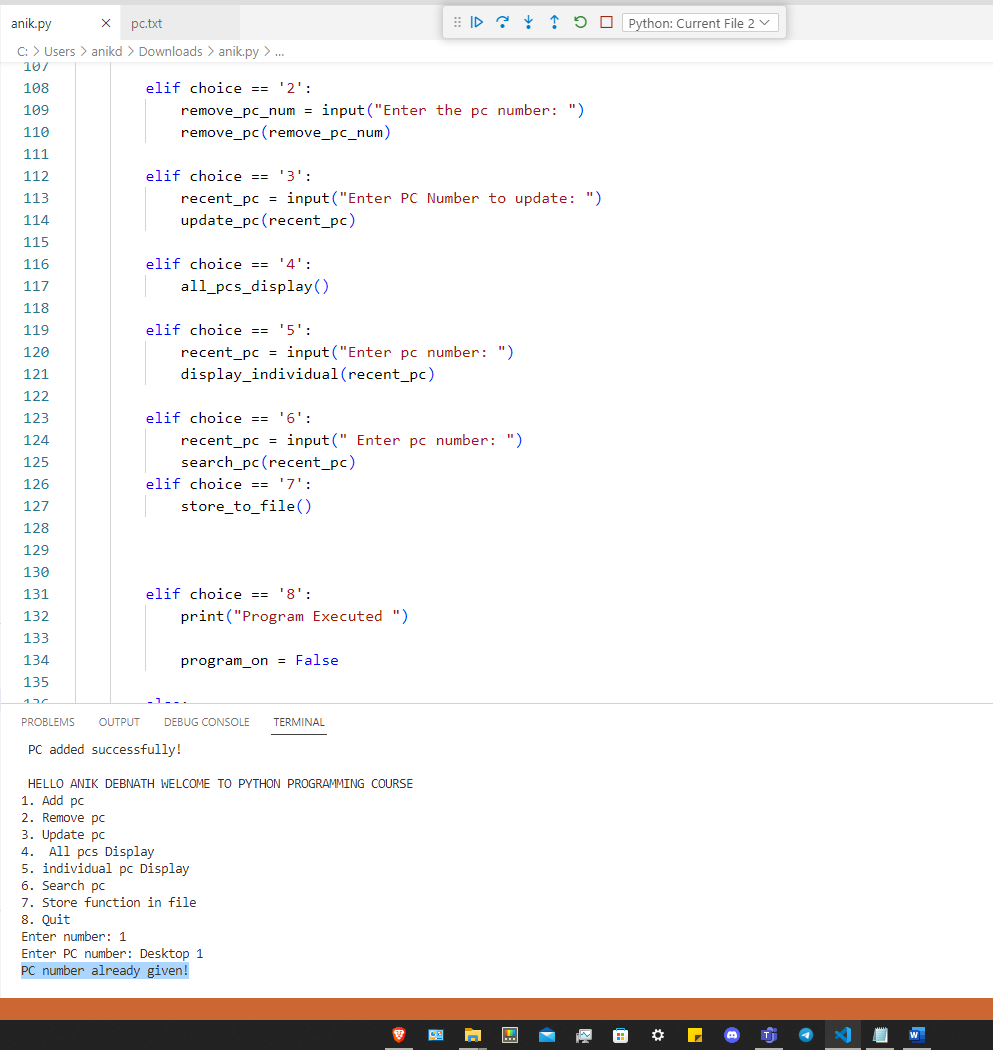
**6.Search PC:** To display a specific pc’s information



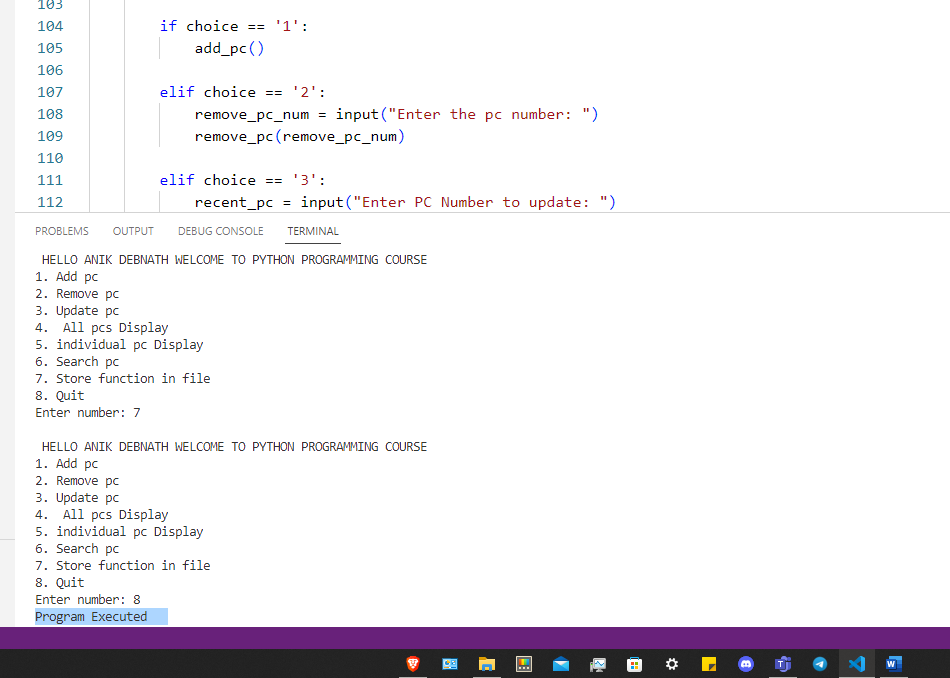
**7.Store function in file:** It store all the PC available in the application into a text file if user wants to maintain a physical copy in his hard drive.



**8.Checking functionality:** The application checks to see if the PC number is already in use when a user adds a new PC. If the PC number already exists in the dataset, the application prompts the user to either remove the PC from the lab or change the information on the existing PC:



**9.Quit:** For executing the computer lab management system application:



**Conclusion**: Developing a computer lab management application in Python involves designing a user-friendly interface, implementing computer availability and usage management.