

**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** |
| Oishi Singh | 20-43067-1 |

**Submitted to:**

# Dr. AKINUL ISLAM JONY

Associate Professor, Computer Science (AIUB)

# Project Overview:

This project is about a computer lab management system using python. This is a console-based project and all

features are available in the screen and users will decide what he should do with this menu bar and he know

about what he wants to do with it. Each of pcs should have the information like pc number, operating system and status of the pc. The main features of this project these are:

• Add PC

• Remove PC

• Update PC

• Display all PCS

• Display individual PC

• Search PC

• Store all pcs to file

• Quit

Here, Add functionality for adding PCs, update functionality for updating existing PCs, and remove functionality to remove existing pcs, search functionality to locate existing pcs, search for a specific pc and display its contents, and finally quit functionality to exit the project. If the pc does not already exist in the database, the user must first add it. If the pc already exists in the database, the user must enter the pc's unique number. show all pcs functionality to display all pcs exiting, store pc functionality to save all pcs information pc. text file, quit function to exit the application.

# Overall, the PC Lab Management System provides a straightforward and efficient method for managing and monitoring computers in a lab setting.

# Project solution design:

The project involves developing a computer lab management system. I used a dictionary to store the data about PCs because I needed to store the key-value pair. And I used the list to store the dictionary's data. I used the Mid.py file to encourage user participation. I declared a dictionary called pc in the mid.py file that can accept all of these functionalities. I also created a text file called pc.txt to accompany it. I documented all practical operations and saved all PC information. First, I store the data in a dictionary called pc structure, and then I write an operation to display all of the pcs. Also, create computers and display all of them. It can also show one computer at a time by searching the computer number. Finally, it can save the entire data set as a text file on the device. If a computer is not found in the computer list by searching, we can create it and save it in the dictionary. Update procedures provide access to currently running computers. The PC Lab Management System as a whole provides an easy way to manage PCs in a computer lab and is a dependable 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:

# PC\_structure

# dictionary

PC = {

# "1": {

# "Operating system": "Windows",

# "PC status": "Enabled"

# },

# "2": {

# "Operating system": "Linux",

# "PC status": "Disabled"

# },

# "3": {

# "Operating system": "MacOS",

# "PC status": "Enabled"

# }

}

#ADD FUNCTIONALITY:

def add\_PC():

PCnum = input("What's the PC number? : ")

if PCnum not in PC:

os = input("What's the operating system? : ")

status = input("What's the PC status? : ")

newpc = {

PCnum: {

"os": os,

"status": status

}

}

PC.update(newpc)

print(PCnum + " number PC added successfully!")

else:

print("PC number: " + PCnum + " already exists in the database!")

pass

#ALL PCS DISPLAY FUNCTIONALITY:

def all\_PCS\_display():

for key in PC:

print("PC number " + key )

print("os :" + PC[key]["os"])

print("status :" + PC [key]["status"])

pass

#REMOVE FUNCTIONALITY:

def remove\_PC(PC\_num):

if PC\_num in PC:

PC.pop(PC\_num)

print("You have removed PC Number: " + PC\_num)

else:

print("PC number: " + PC\_num + " doesn't exists in the database!")

pass

#UPDATE FUNCTIONALITY:

def update\_PC(PC\_num):

if PC\_num in PC:

os = input("New operating system name: ")

status = input("New status: ")

PC[PC\_num]["os"] = os

PC[PC\_num]["status"] = status

else:

print("PC not found.")

pass

#INDIVIDUAL DISPLAY FUNCTIONALITY:

def display\_individual(PC\_num):

if PC\_num in PC:

print(PC[PC\_num])

else:

print("PC not found.")

pass

#SEARCH FUNCTIONALITY:

def search\_PC(PC\_num):

if PC\_num in PC:

print("PC no " + PC\_num + " is found!")

else:

print("PC not found.")

pass

#UPDATE FUNCTIONALITY:

def store\_to\_file():

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

for key in PC:

file.write("PC number " + key + ", os :" + PC[key]["os"] + ", status :" + PC[key]["status"] + "\n")

file.close()

pass

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

program\_on = True

while program\_on:

print("\n Hello! Oishi Welcome to Computer Lab Management System ")

print("1. Add New PC For Your Computer Lab")

print("2. Remove PC ")

print("3. Update PC ")

print("4. Display all PCS ")

print("5. Display individual PC ")

print("6. Search PC ")

print("7. Store all PCS to file")

print("8. Quit This Item")

choice = input("Press key: ")

if choice == '1':

add\_PC()

elif choice == '2':

remove\_PC\_num = input("Enter the PC Number, you want to delete: ")

remove\_PC(remove\_PC\_num)

elif choice == '3':

target\_PC = input("Which PC details do you want to update? PC Number: ")

update\_PC(target\_PC)

elif choice == '4':

all\_PCS\_display()

elif choice == '5':

target\_PC = input("Input PC Number: ")

display\_individual(target\_PC)

elif choice == '6':

target\_PC = input("PC num: ")

search\_PC(target\_PC)

elif choice == '7':

store\_to\_file()

elif choice == '8':

print("Program has executed ... ")

program\_on = False

else:

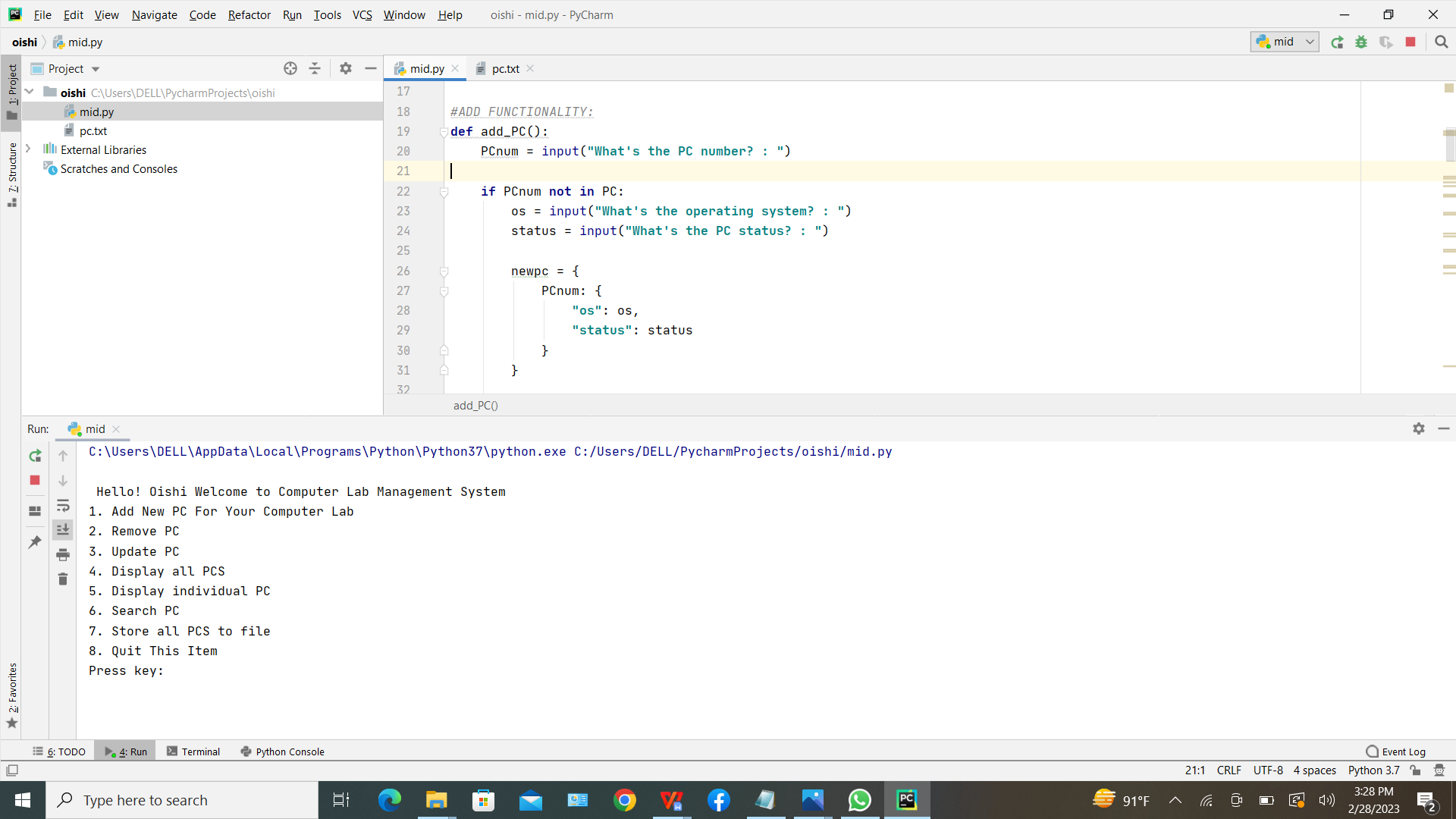
print("Invalid Option! Please Write your option number correctly only!")

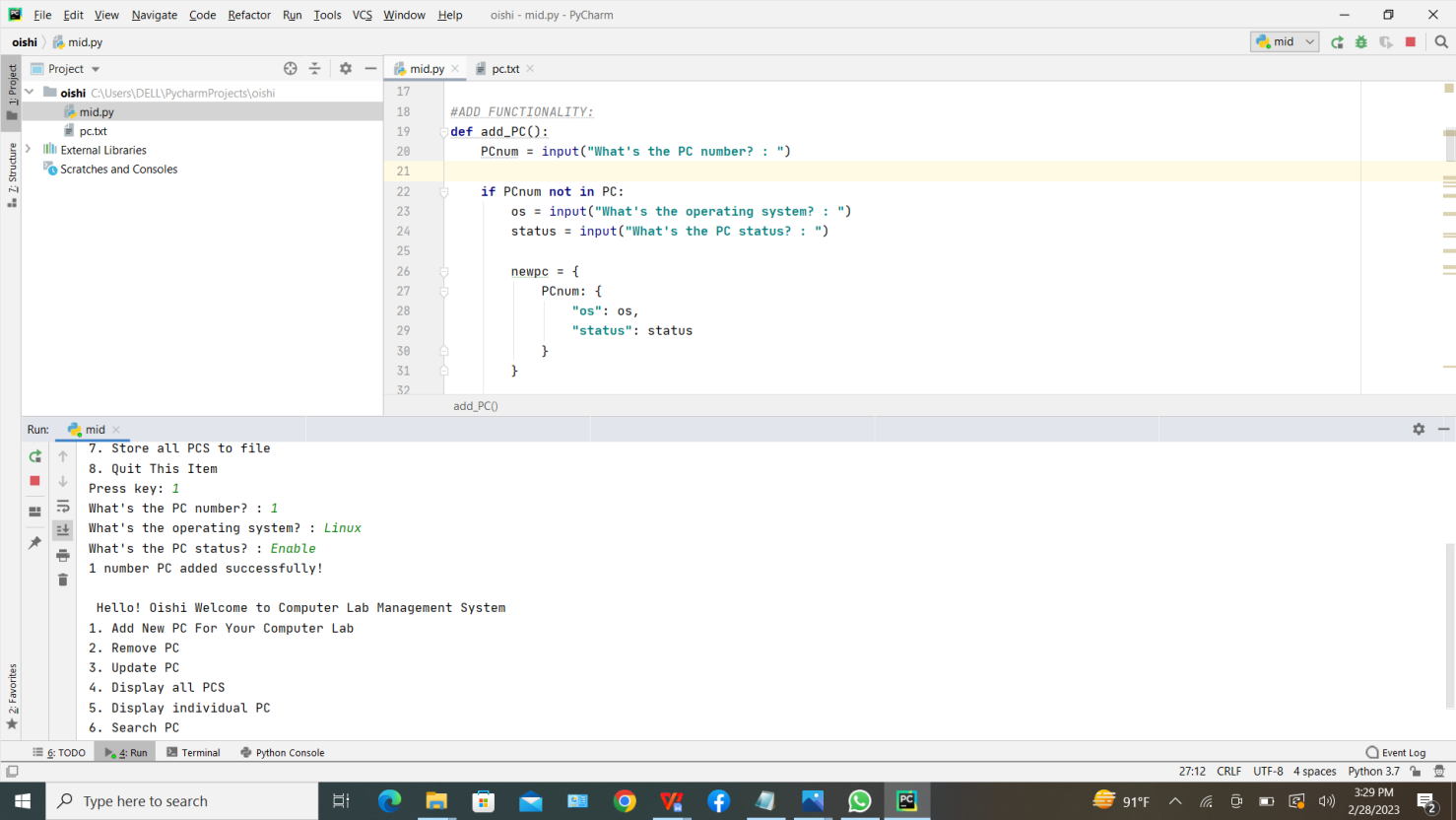
pass

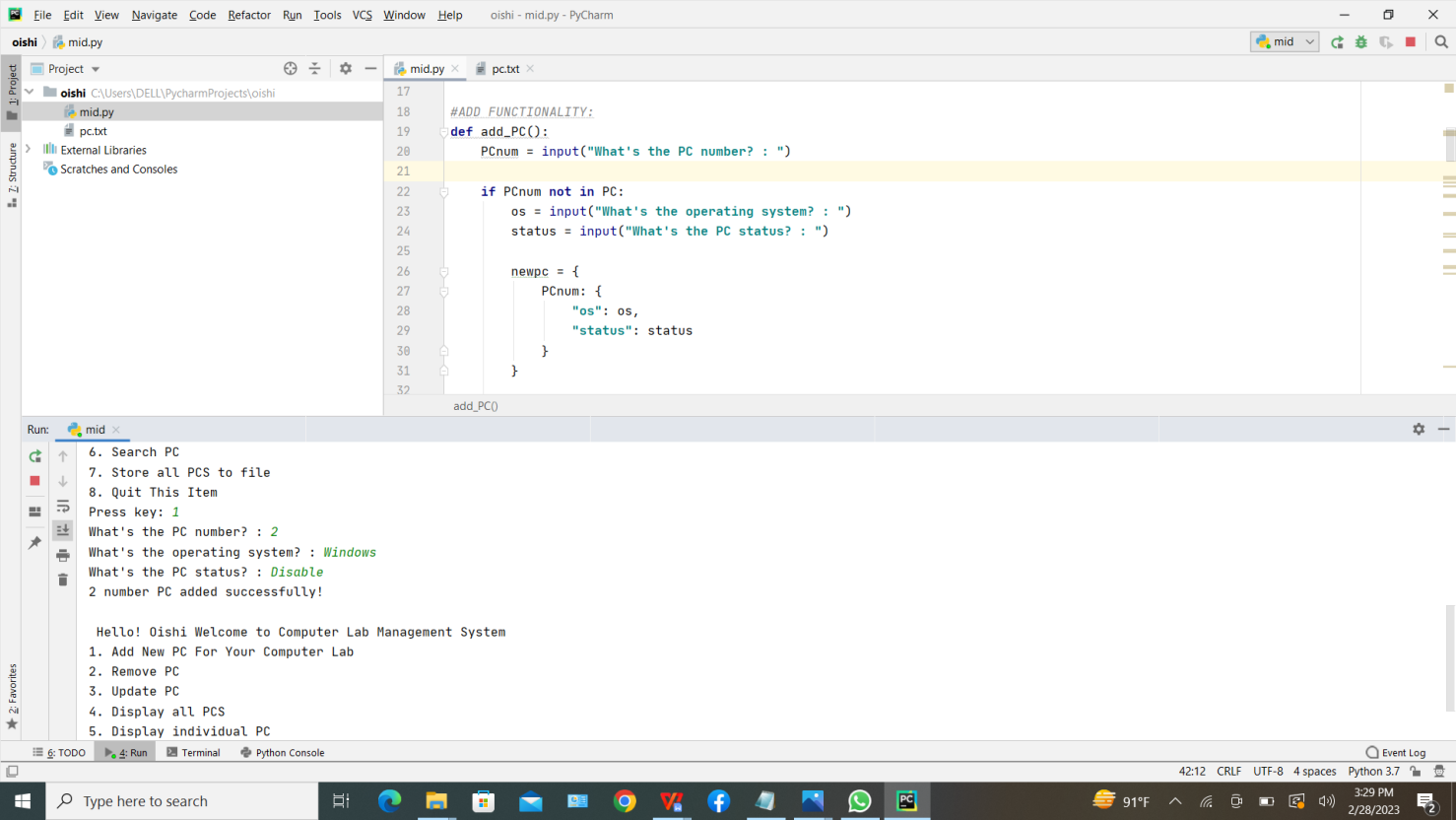
# Application Overview:

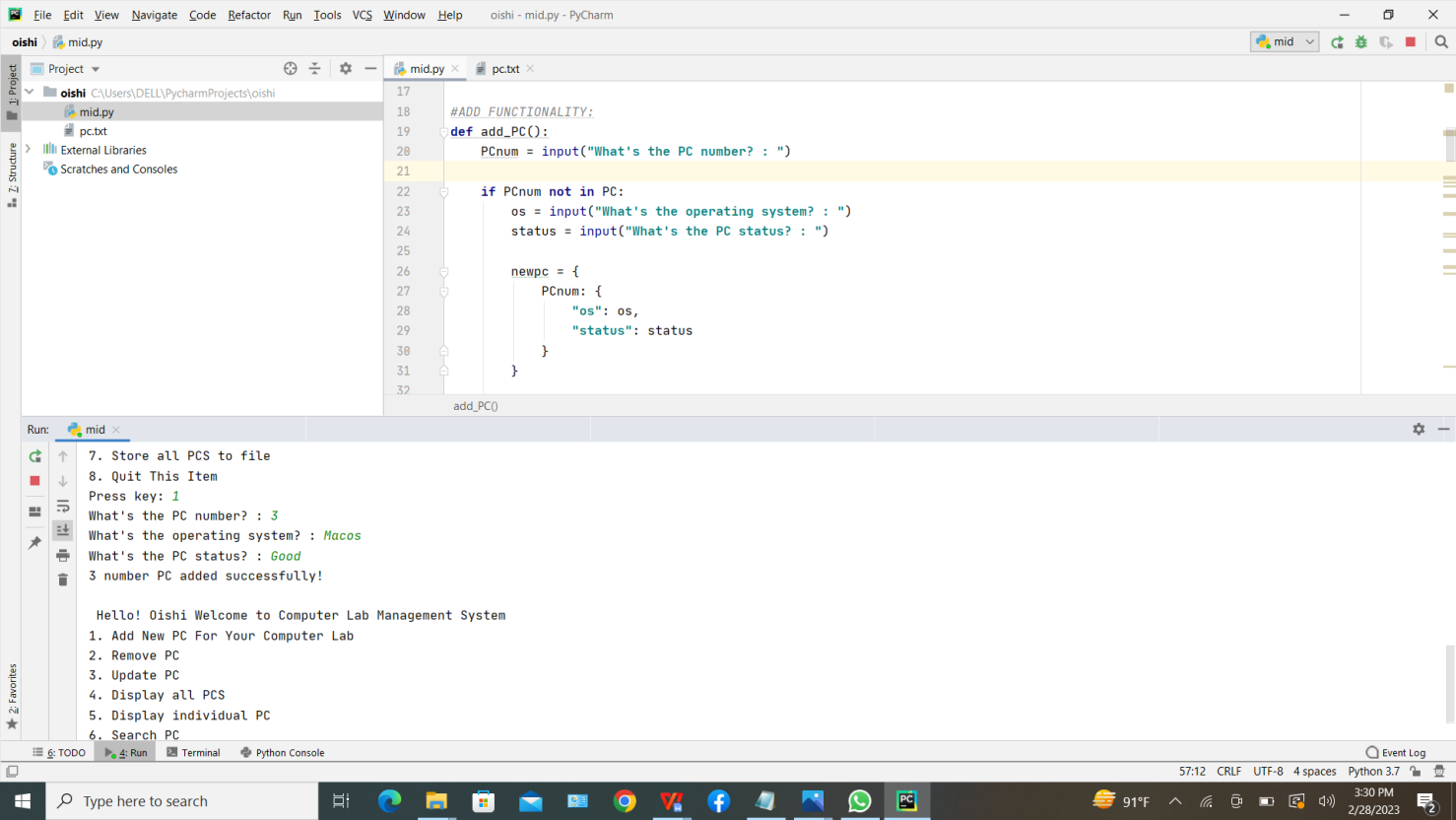
The following is a list of the various PC management features that the application offers:

1. **Add a new computer:** To add a new pc here I use a function that takes the name of Add a new PC as a parameter and adds it.

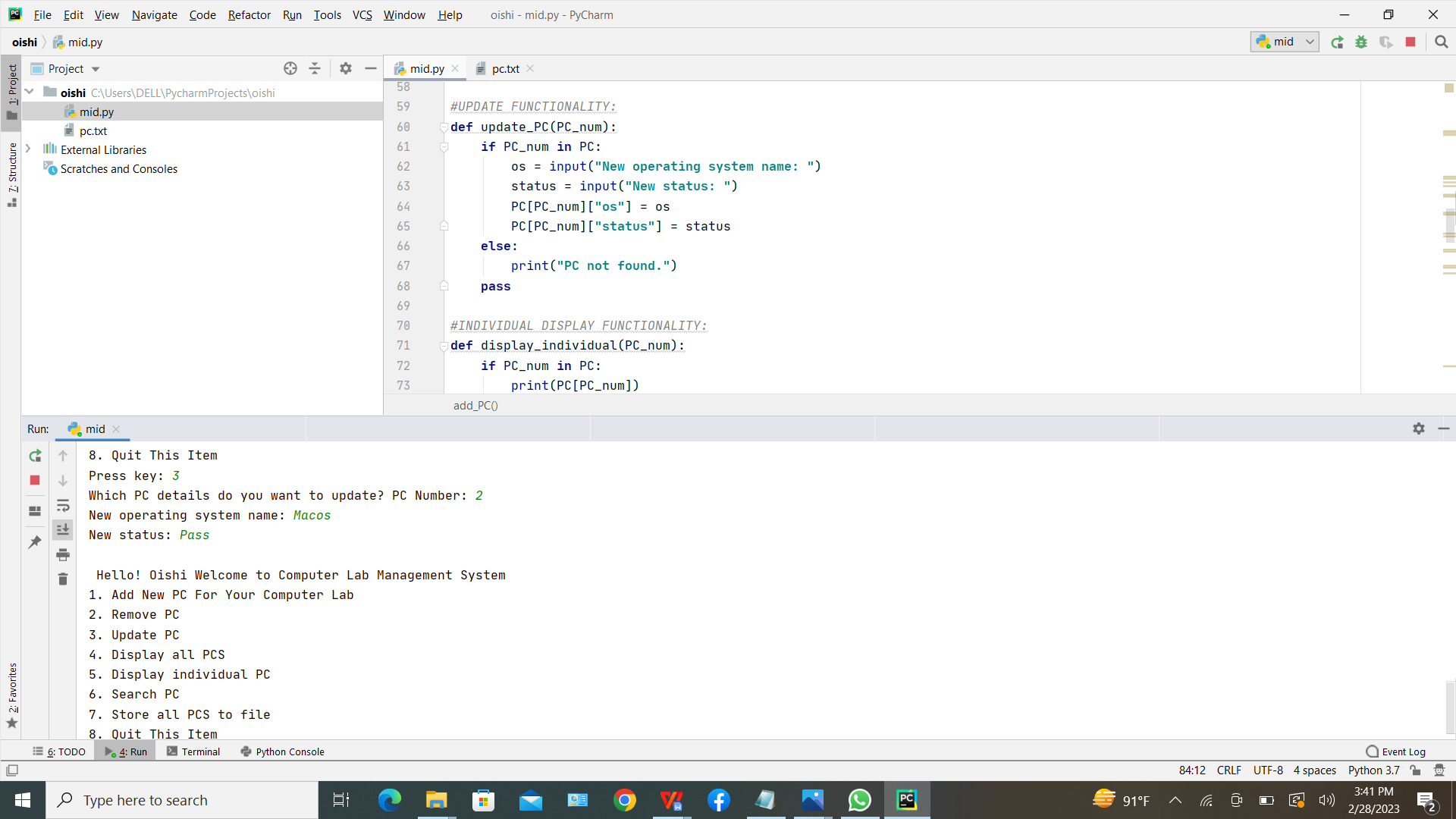




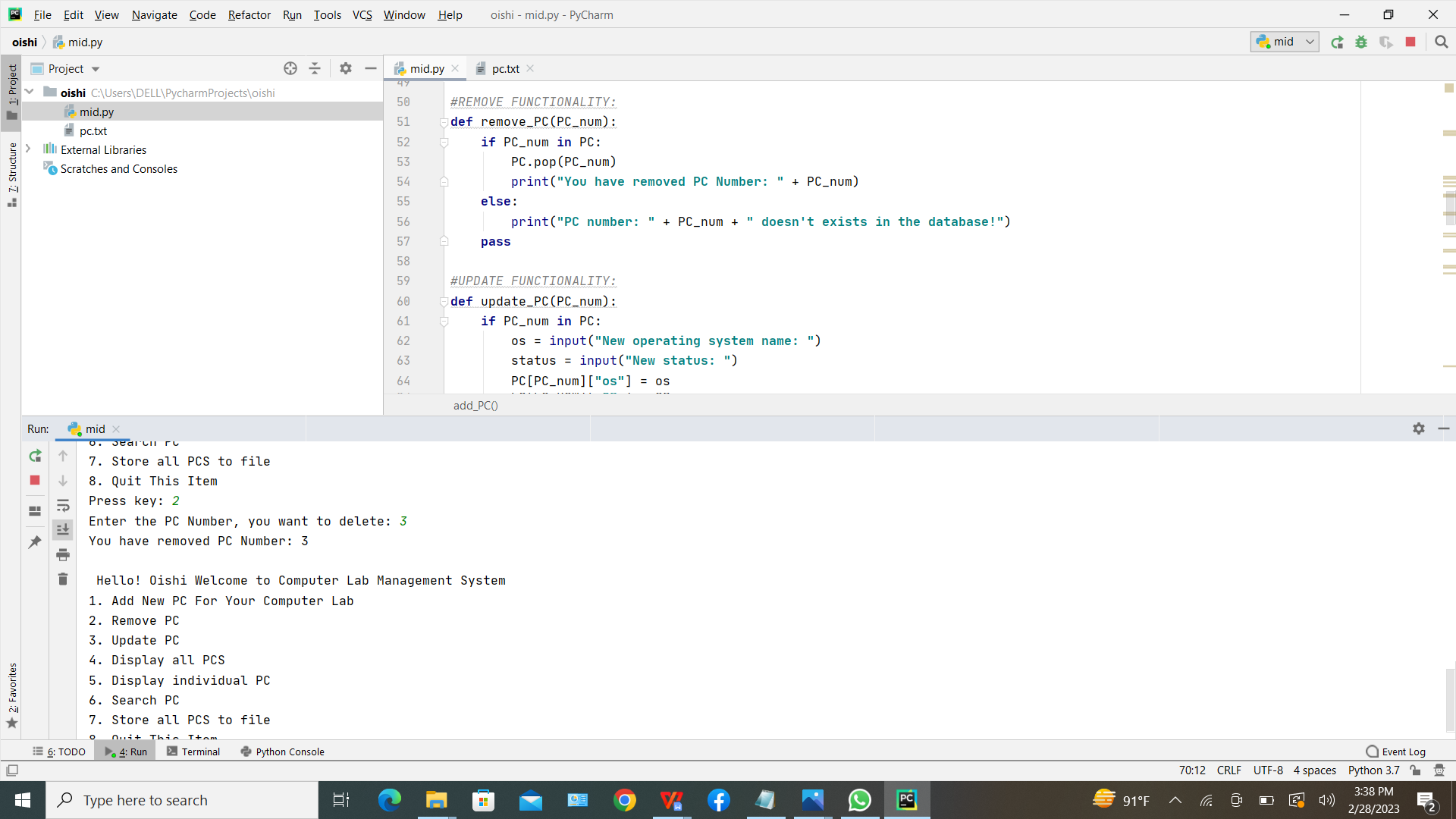




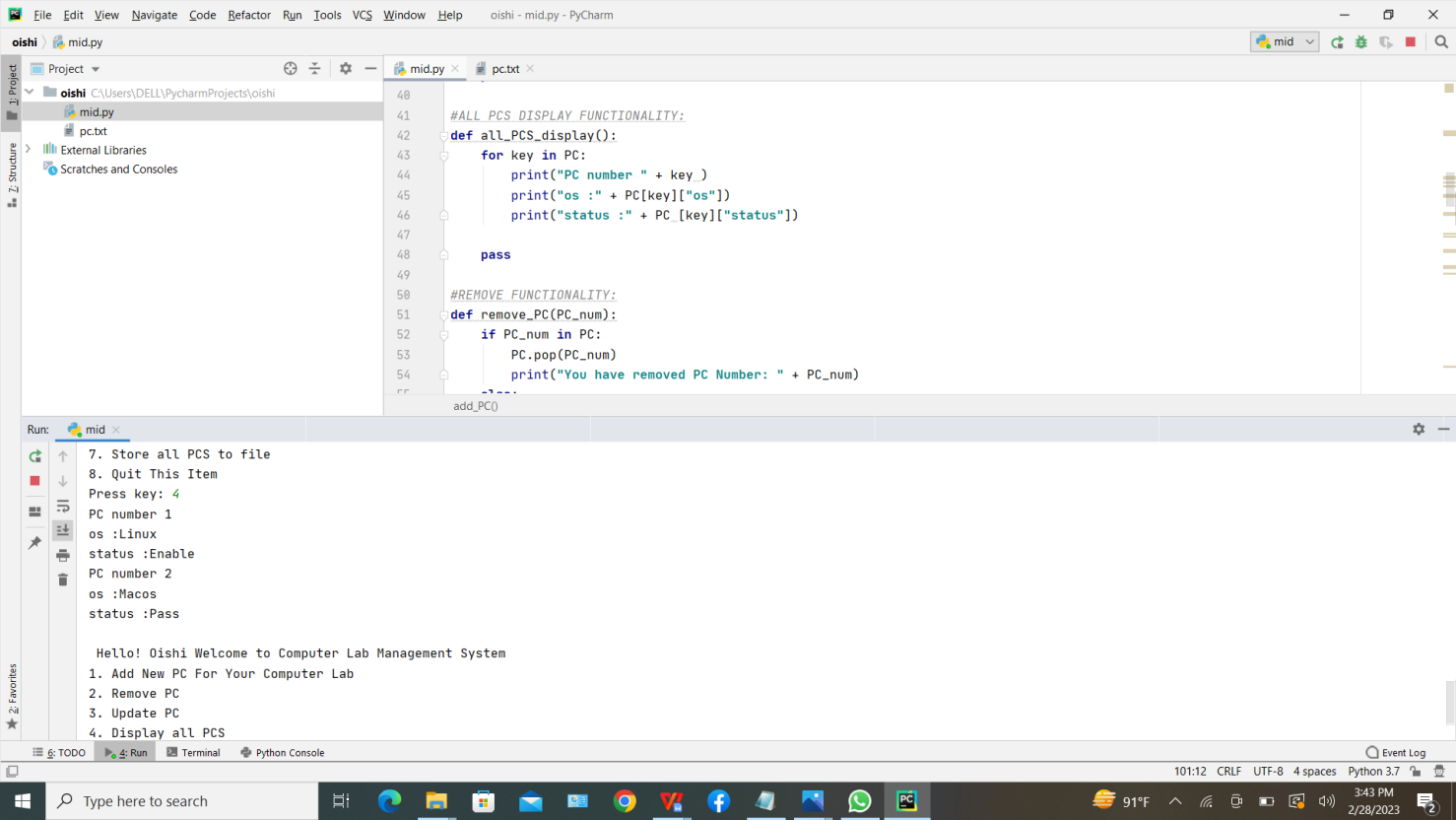
**2.Update PC information:** To update information of an existing PC, we can modify the value of the corresponding key in the dictionary that stores the information of the PCs.



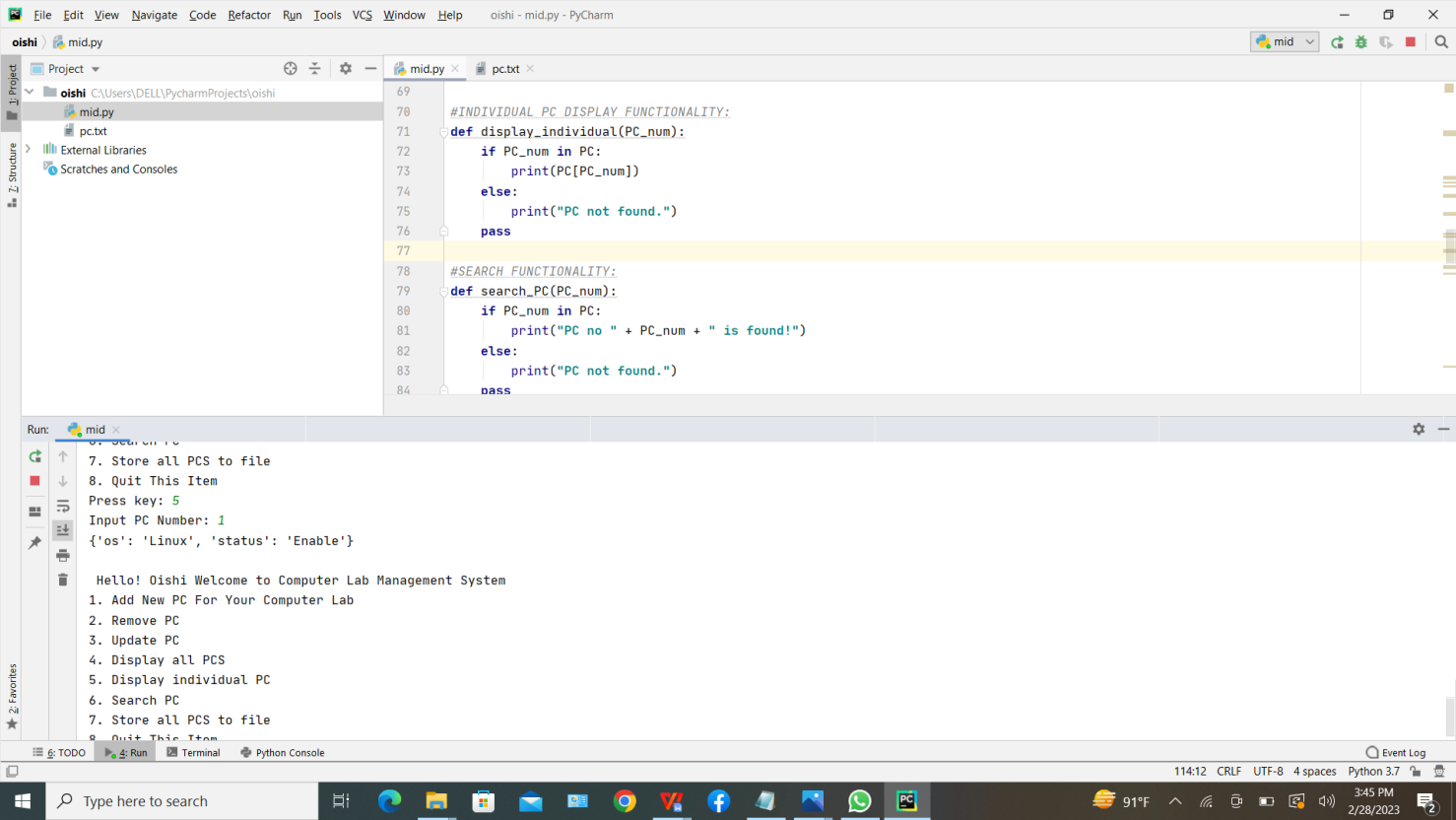
**3.Remove a PC:** This functionality allows users to remove an existing PC from the lab. The user needs to provide the PC number of the PC they want to remove.



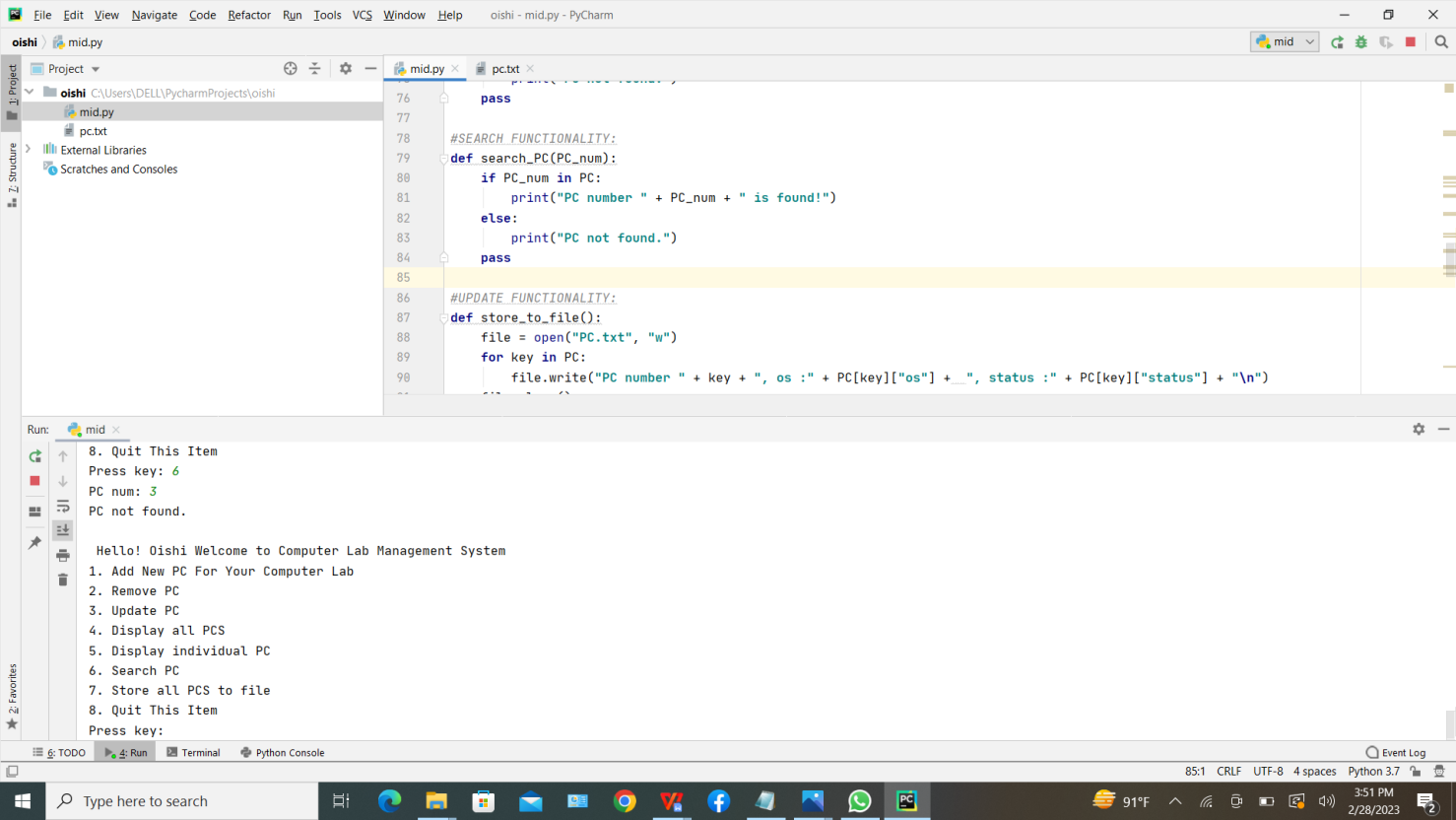
**4. Display all PCs:** This functionality displays information about all the PCs in the lab.

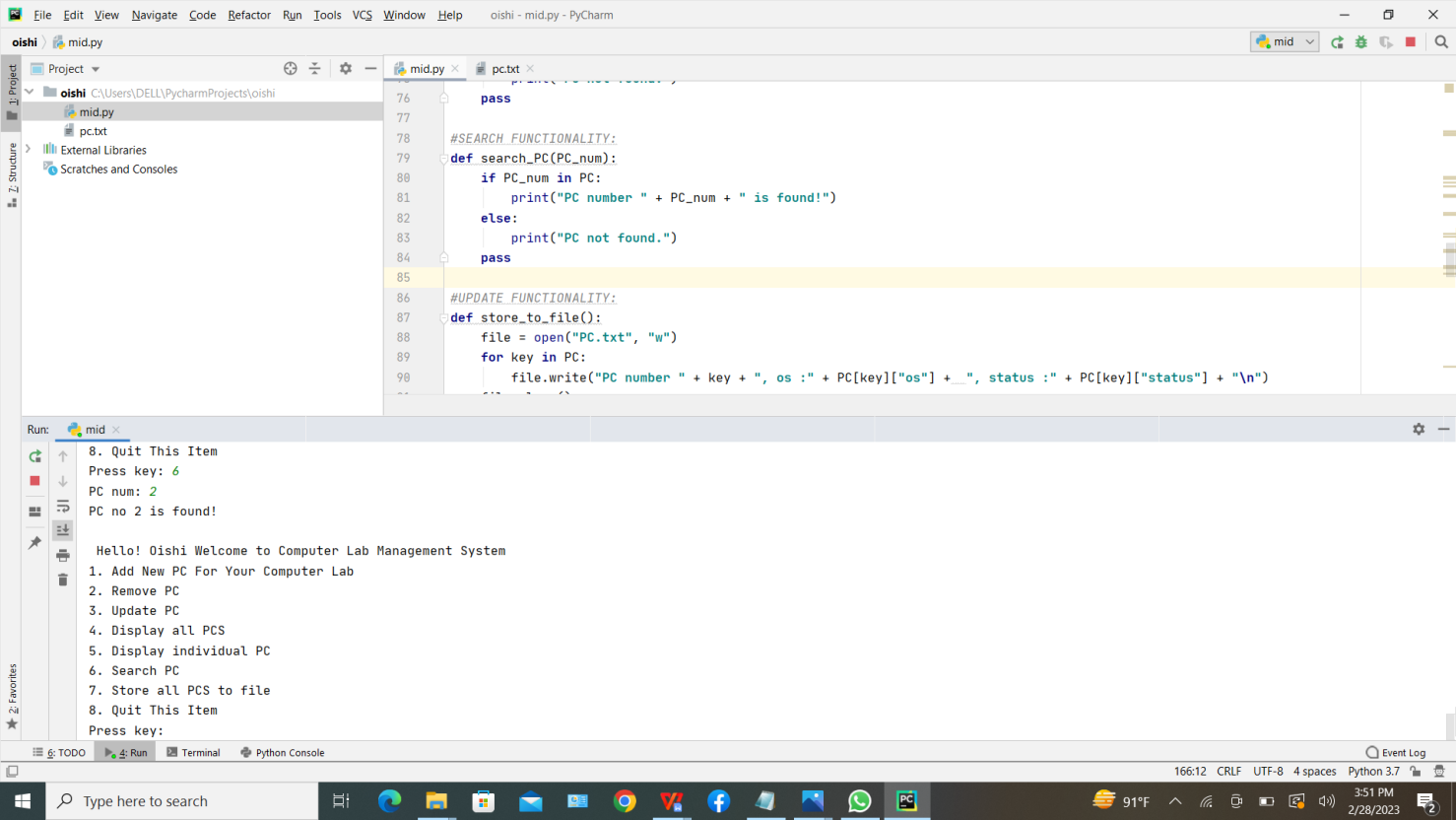


**5.Display Individual PC:** This functionality enables users to access all the data on a specific PC. The user must input the PC's number in order for it to be displayed.

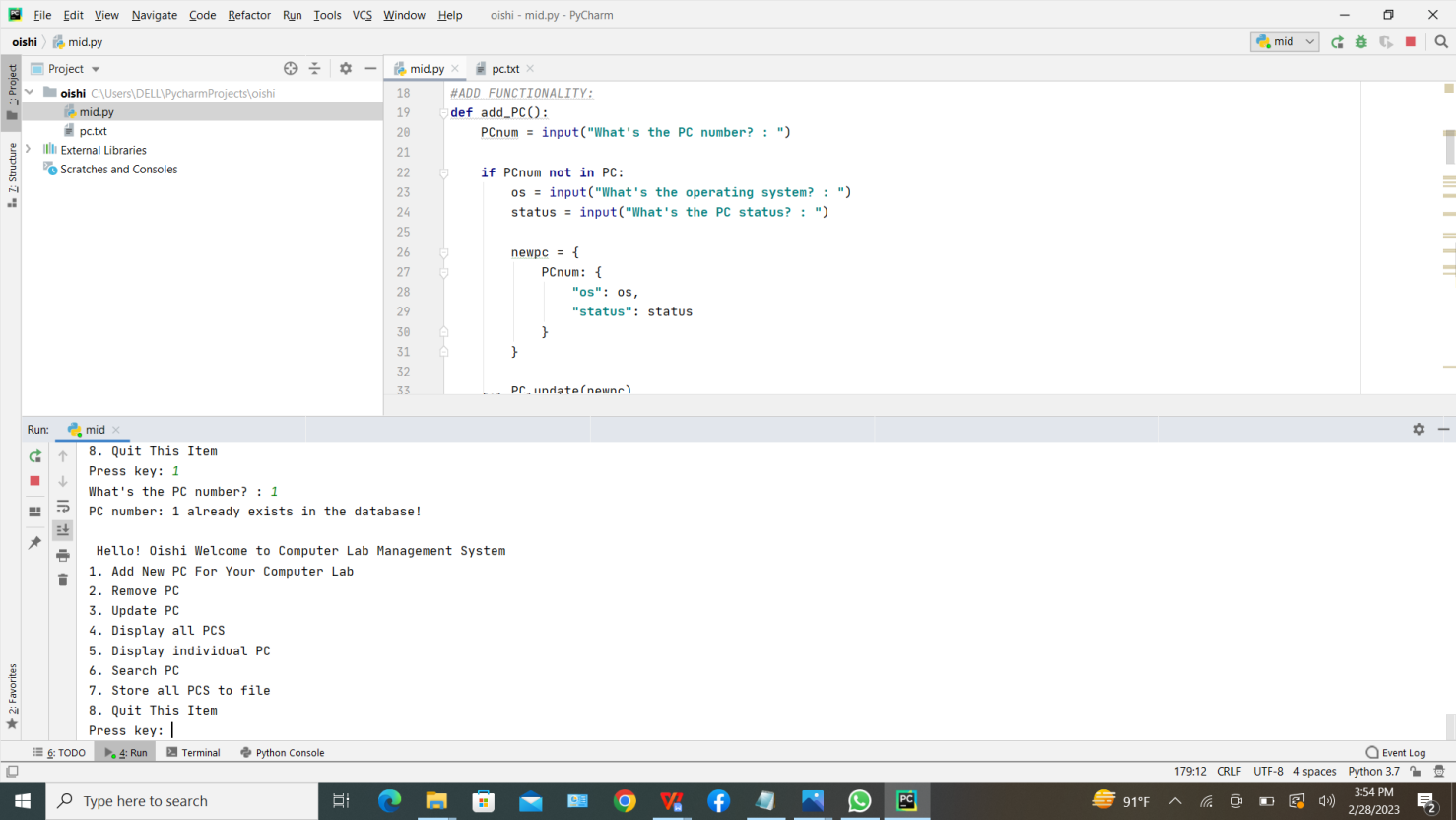


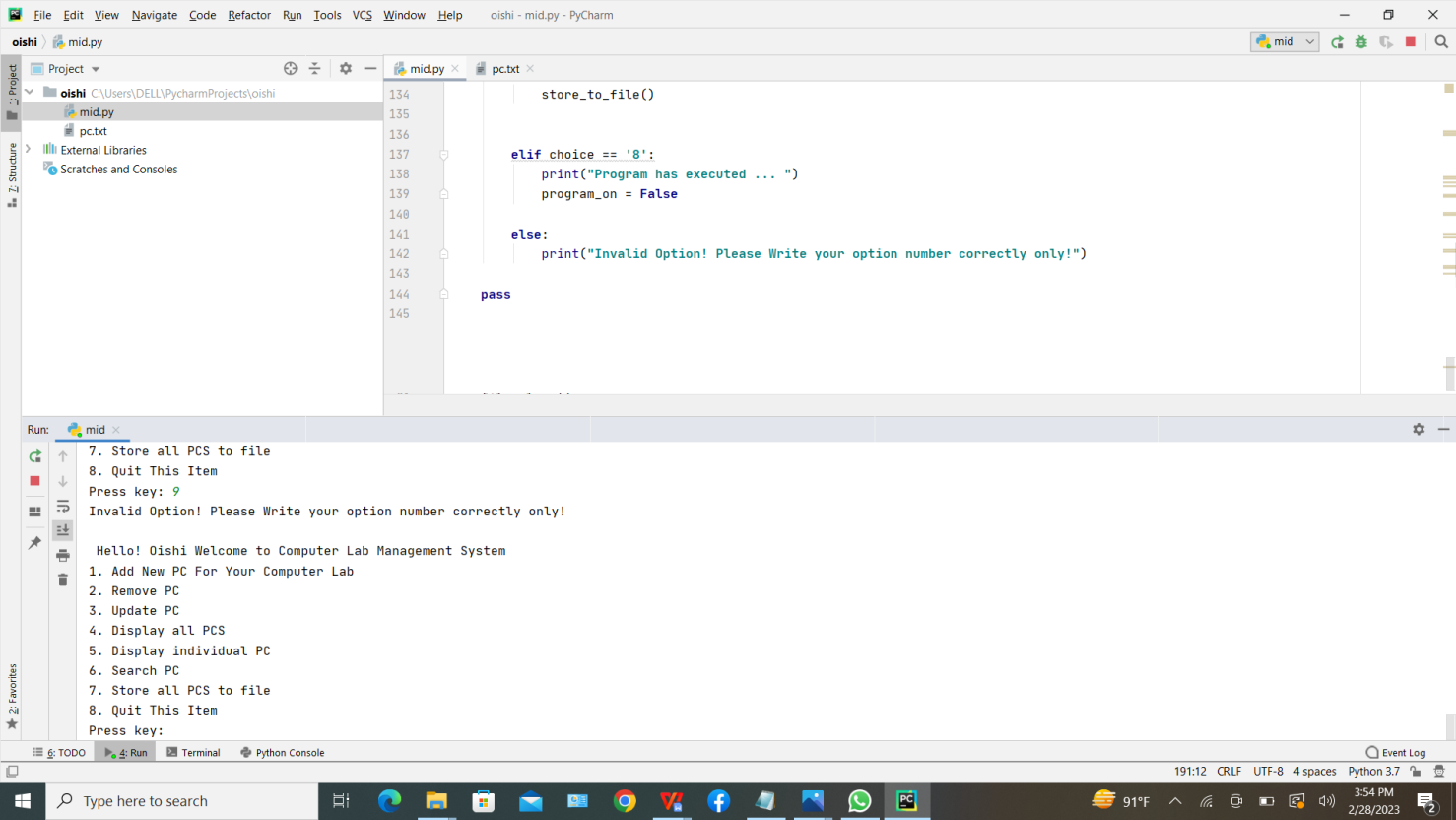
**6. Search for a PC:** Users can utilize this feature to look for a certain PC and see its details. The user will be asked to add the new PC to the lab if the PC is not already listed in the application.



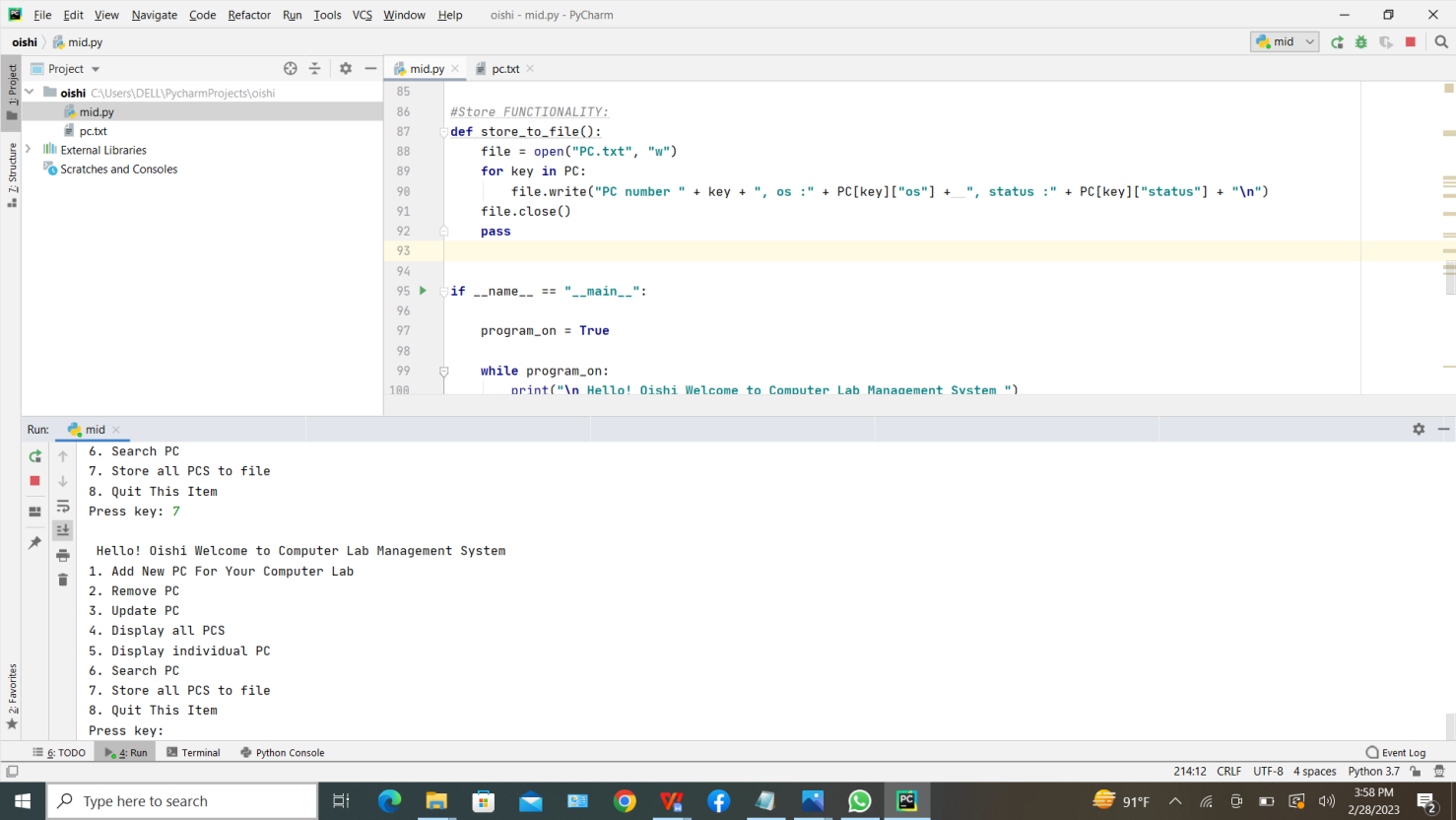


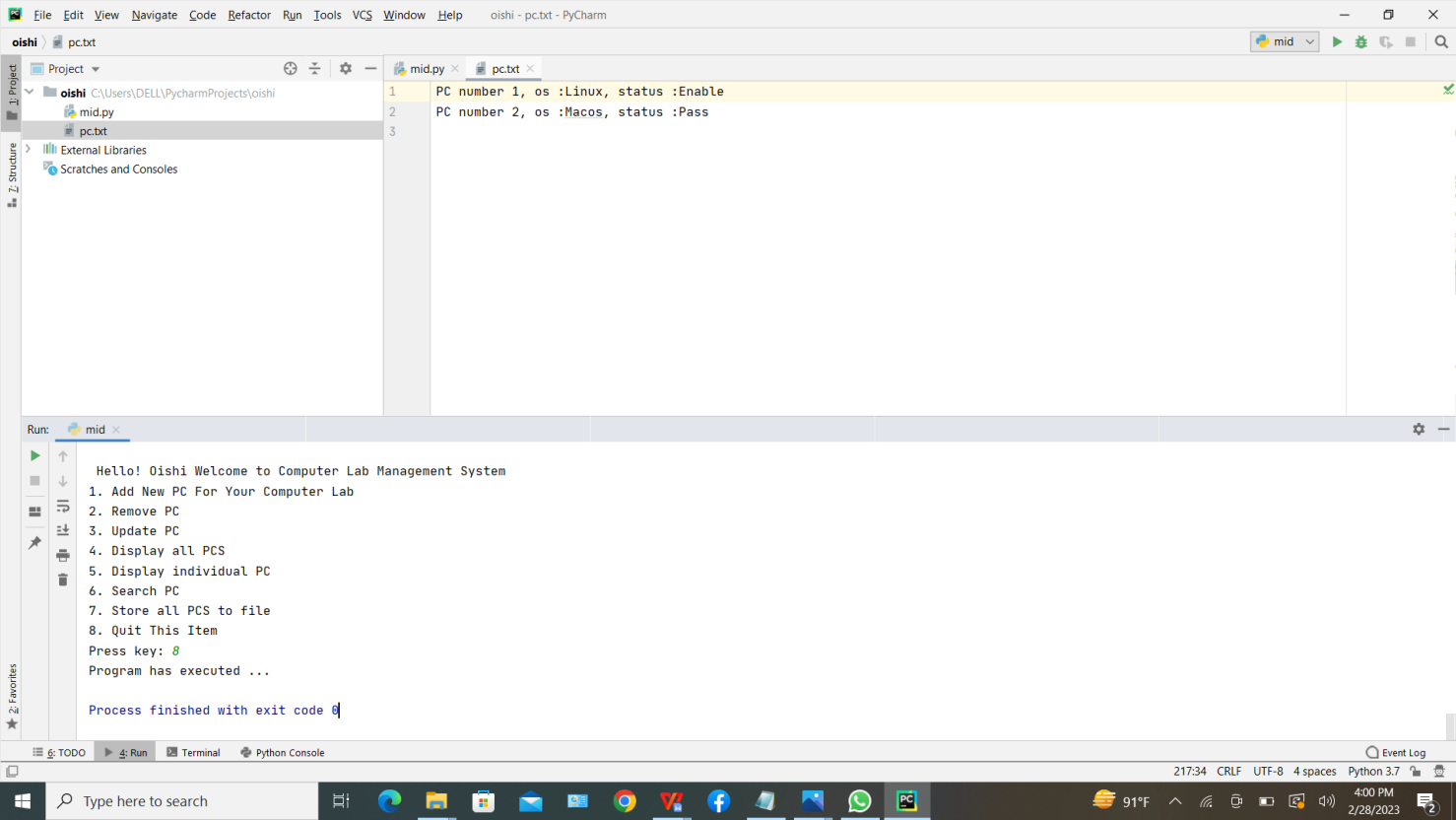
**7. Checking functionality:** The application checks to see if the PC number is already in use when a user adds a new PC. If the PC number already exists, the application prompts the user to either delete the PC from the lab or change the information on the existing PC.



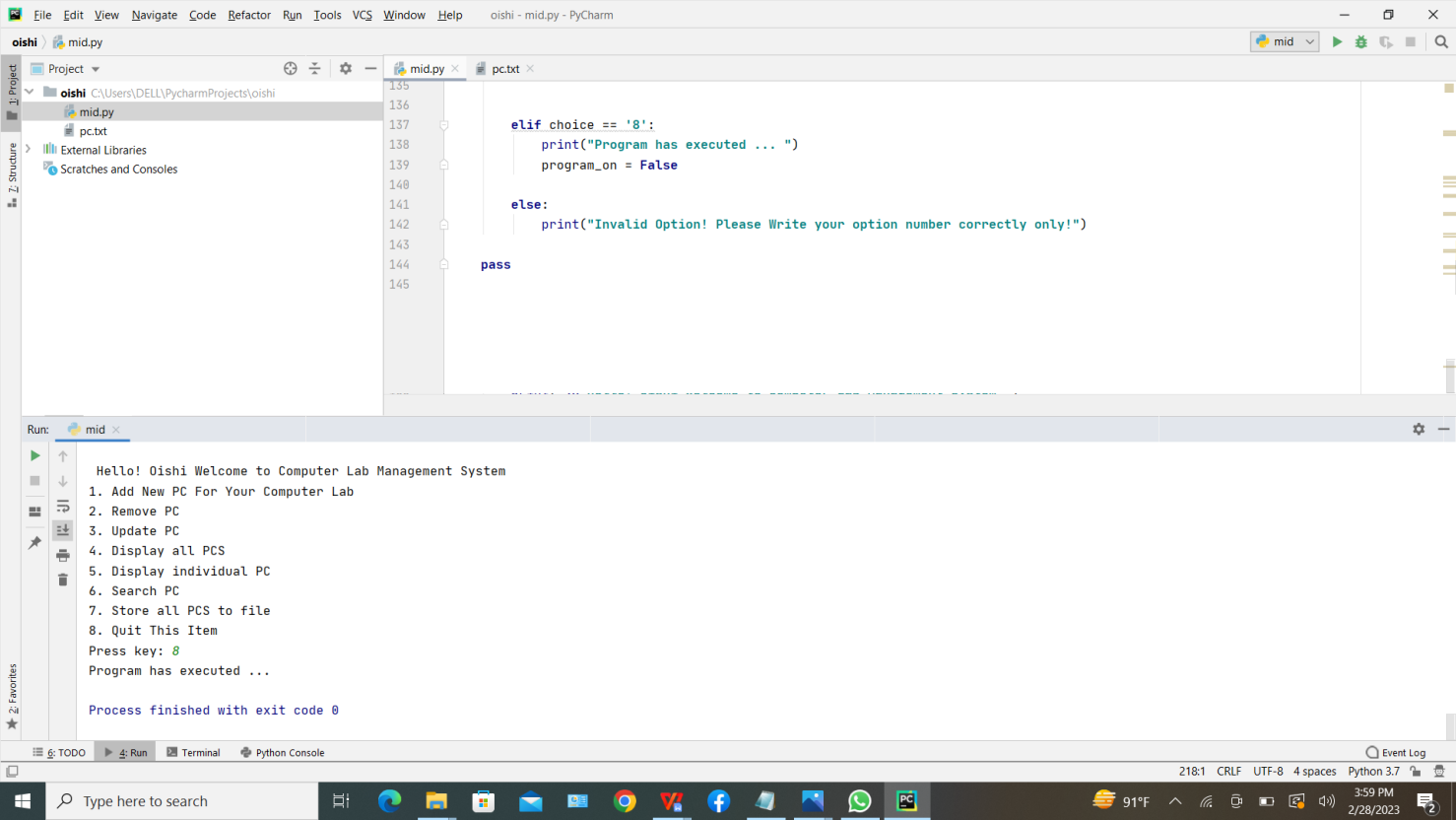


**8. Store functionality:** If a user wants to keep a physical copy on their hard drive, the application offers a store option that enables users to save all the Data available in the application into a text file.





**9. Quit:** The application offers a cancellation option.



**Conclusion**: The PC Lab Management System's goal is to make managing lab computers easier. This tool makes it simple to add, update, and remove PCs from the lab. Creating a user-friendly user interface, configuring machine availability, and managing computer usage are all part of creating a computer lab management program in Python.