American International University- Bangladesh
Faculty of Science and Technology

Project Title: Develop a computer lab management application using Python

Mid Project Report
Spring 2022-2023

Course Name: PROGRAMMING IN PYTHON
Section: B
Faculty Name: AKINUL ISLAM JONY

Submitted By:

Student Name	Student Id
Partha Malakar	**20-42908-1**

Project overview:

The PC lab management application is a console-based program that allows users to keep records of all the PCs in a lab. The application provides features such as adding new PCs, updating existing PC information, removing PCs, displaying all or individual PC information, searching for a particular PC, and storing PC information in a text file. The application will have a menu-based interface that allows users to select the desired action.

Data Structure:

To maintain the records of the PCs in the lab, we need a data structure to store and manage this information. A suitable data structure for this task could be a dictionary or a list of dictionaries. Each dictionary would represent a single PC and contain the PC's number, operating system, and status information.

Adding functionality:

The 'adding functionality' of the PC lab management application allows users to add new PCs to the lab. When adding a new PC, the user will be prompted to enter the PC's number, operating system, and status information. The application will then create a new dictionary to store this information and add it to the list of PCs.

Updating functionality:

The 'updating functionality' of the PC lab management application allows users to update the information of an existing PC. To update an existing PC, the user will be prompted to enter the PC's number and then choose which information they want to update (operating system or status). Once the user has entered the new information, the application will update the corresponding dictionary in the list of PCs.

Remove functionality:

The 'remove functionality' of the PC lab management application allows users to remove an existing PC from the lab. To remove a PC, the user will be prompted to enter the PC's number, and then the corresponding dictionary will be removed from the list of PCs.

All PC display functionality:

The 'all PC display functionality' of the PC lab management application allows users to display information about all the PCs in the lab. When this option is selected from the menu, the application will display the PC number, operating system, and status information for all the PCs in the list.

Individual PC display functionality:

The 'individual PC display functionality' of the PC lab management application allows users to display all the information about a specific PC in the lab. When this option is selected from the menu, the user will be prompted to enter the PC's number, and then the corresponding dictionary will be displayed with its PC number, operating system, and status information.

Search functionality:
The 'search functionality' of the PC lab management application allows users to search for a specific PC in the lab. When this option is selected from the menu, the user will be prompted to enter the PC's number. If the PC is already in the list, the corresponding dictionary will be displayed. If the PC is not in the list, the user will be prompted to add the new PC to the lab.

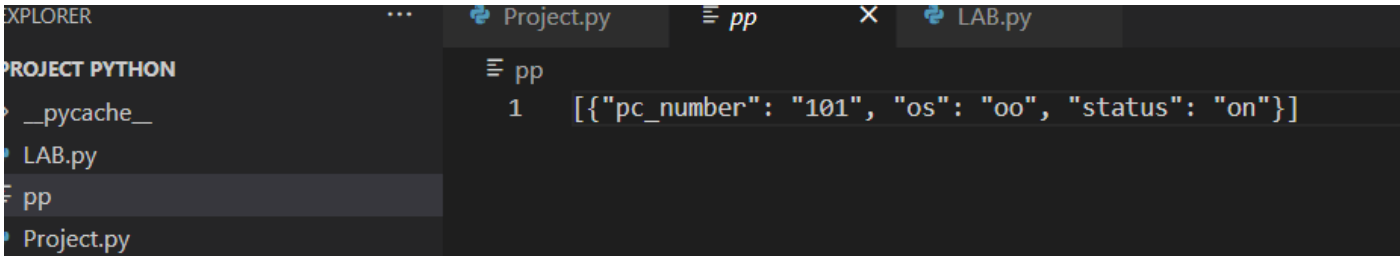
Checking functionality:
The 'checking functionality' of the PC lab management application ensures that each PC in the lab has a unique PC number. When adding a new PC, the application will check if a PC with the same PC number already exists in the lab. If it does, the application will prompt the user to modify the existing PC's information, remove the PC from the lab, or take no action.

Store functionality:
The 'store functionality' of the PC lab management application allows users to store all the PC information in a text file for safekeeping. When this option is selected from the menu, the user will be prompted to enter a filename. The application will then write the PC information to a text file with the given filename.

In summary, the PC lab management application provides an efficient and user-friendly way to manage and maintain records of the PCs in a lab. With features such as adding, updating, and removing PCs, displaying information about all or individual PCs, searching for a specific PC, and storing PC information in a text file, the application ensures that lab administrators can keep track of their PCs easily and effectively.

Project solution design:

1. User Interface: We can start by creating a simple menu-based user interface using the built-in input() function. The menu should contain options to add a new PC, update an existing PC, remove a PC, display information about all PCs, display information about a specific PC, search for a PC, and quit the application.
2. Data Structure: As mentioned earlier, we can use a list of dictionaries to store information about each PC in the lab.



3. Core Functions: We will need several core functions to implement the functionality of the application, such as add_pc(), update_pc(), remove_pc(), display_all_pcs(), display_pc(), search_pc(), and find_pc_index().
4. Validation: We will need to add validation to ensure that the user inputs are in the correct format and that the program does not crash if the user enters invalid inputs.

5. File Handling: Finally, we can add file handling functionality to allow the user to store all the PC information in a text file if they want to maintain a physical copy. We can use the built-in open() function to create and write to a text file.

```
70
71      def store_pcs(self, filename):
72          with open(filename, 'w') as f:
73              json.dump([p.__dict__ for p in self.pcs], f)
74          print("**********PCs stored successfully.**********")
75
76      def load_pcs(self, filename):
77          with open(filename, 'r') as f:
78              data = json.load(f)
79          self.pcs = [PC(p['pc_number'], p['os'], p['status']) for p in data]
80          print("*******PCs loaded successfully.*********")
```

Overall, this project can be implemented in a modular fashion by breaking it down into several functions and using a menu-based user interface to provide easy access to the application's functionality.

Implementation:

```
import json

class PC:
    def __init__(self, pc_number, os, status):
        self.pc_number = pc_number
        self.os = os
        self.status = status

class Lab:
    def __init__(self):
        self.pcs = []

    def add_pc(self, pc):
        for p in self.pcs:
            if p.pc_number == pc.pc_number:
                choice = input("PC with same number already exists. Do you want to modify
(M), remove (R), or take no action (N)?")
                if choice.lower() == 'm':
                    p.os = pc.os
                    p.status = pc.status
                    return "--********PC information modified successfully.---*******"
                elif choice.lower() == 'r':
                    self.pcs.remove(p)
                    return "*********PC removed successfully.**********"
                else:
                    return "*********No action taken.************"
        self.pcs.append(pc)
        return "*********PC added successfully.**********"
```

```
    def remove_pc(self, pc_number):
        for p in self.pcs:
            if p.pc_number == pc_number:
                self.pcs.remove(p)
                return "******************PC removed successfully.**************"
        return "**********PC not found.****"

    def update_pc(self, pc):
        for p in self.pcs:
            if p.pc_number == pc.pc_number:
                p.os = pc.os
                p.status = pc.status
                return "***********PC information modified successfully.**********"
        return "PC not found."

    def display_all_pcs(self):
        for p in self.pcs:
            print("PC Number:", p.pc_number)
            print("Operating System:", p.os)
            print("Status:", p.status)
            print()

    def display_pc_info(self, pc_number):
        for p in self.pcs:
            if p.pc_number == pc_number:
                print("PC Number:", p.pc_number)
                print("Operating System:", p.os)
                print("Status:", p.status)
                return
        print("PC not found.")

    def search_pc(self, pc_number):
        for p in self.pcs:
            if p.pc_number == pc_number:
                print("PC Number:", p.pc_number)
                print("Operating System:", p.os)
                print("Status:", p.status)
                return
        pc = PC(pc_number, input("Enter operating system: "), input("Enter status: "))
        self.add_pc(pc)
        print("********PC added successfully.*********")

    def store_pcs(self, filename):
        with open(filename, 'w') as f:
            json.dump([p.__dict__ for p in self.pcs], f)
        print("**************PCs stored successfully.**********")

    def load_pcs(self, filename):
```

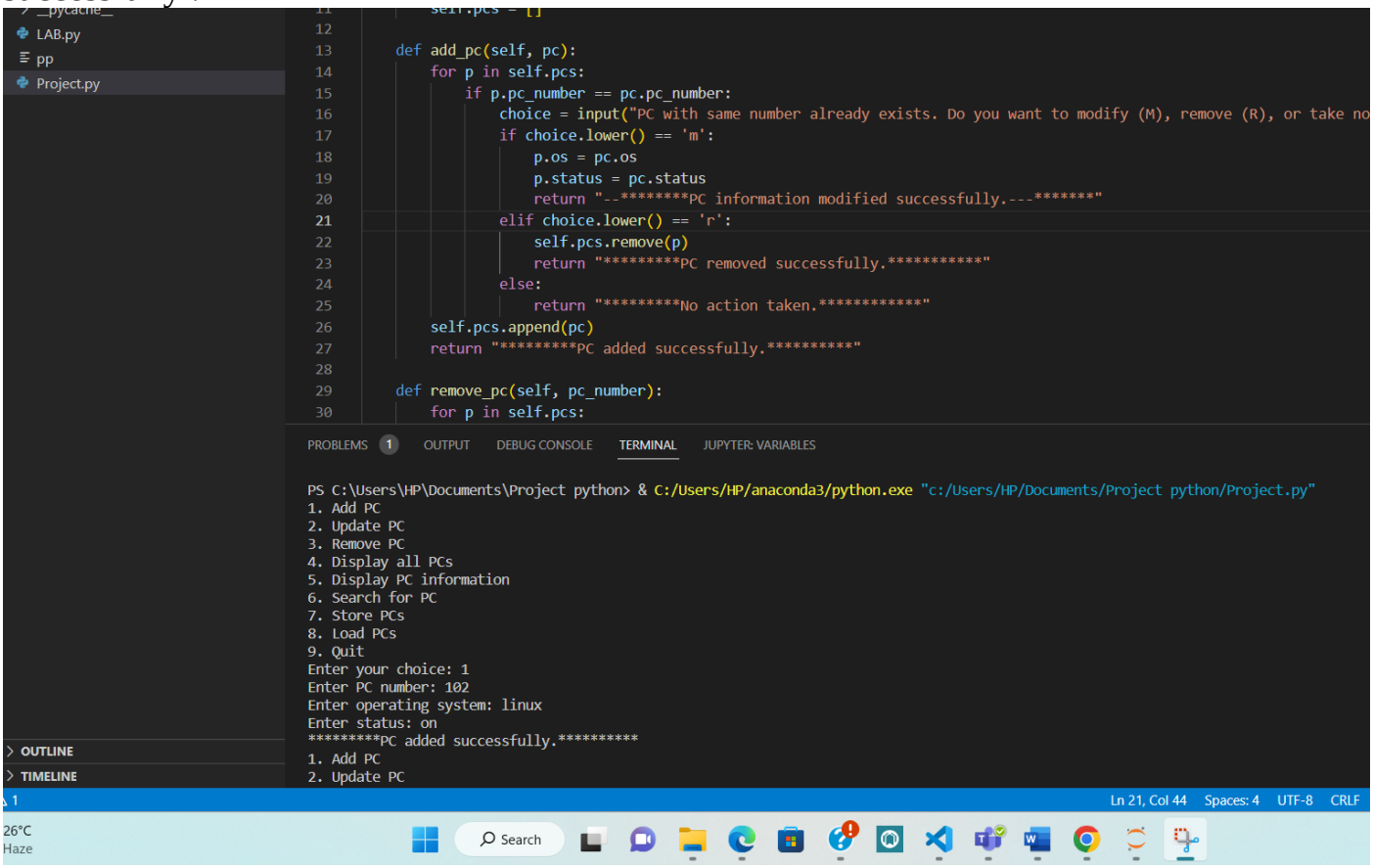
```
        with open(filename, 'r') as f:
            data = json.load(f)
        self.pcs = [PC(p['pc_number'], p['os'], p['status']) for p in data]
        print("*******PCs loaded successfully.*********")


def main():
    lab = Lab()
    while True:
        print("1. Add PC")
        print("2. Update PC")
        print("3. Remove PC")
        print("4. Display all PCs")
        print("5. Display PC information")
        print("6. Search for PC")
        print("7. Store PCs")
        print("8. Load PCs")
        print("9. Quit")
        choice = input("Enter your choice: ")
        if choice == '1':
            pc_number = input("Enter PC number: ")
            os = input("Enter operating system: ")
            status = input("Enter status: ")
            pc = PC(pc_number, os, status)
            print(lab.add_pc(pc))
        elif choice == '2':
            pc_number = input("Enter PC number: ")
            os = input("Enter operating system: ")
            status = input("Enter status: ")
            pc = PC(pc_number, os, status)
            print(lab.update_pc(pc))
        elif choice == '3':
            pc_number = input("Enter PC number: ")
            print(lab.remove_pc(pc_number))
        elif choice == '4':
            lab.display_all_pcs()
        elif choice == '5':
            pc_number = input("Enter PC number: ")
            lab.display_pc_info(pc_number)
        elif choice == '6':
            pc_number = input("Enter PC number: ")
            lab.search_pc(pc_number)
        elif choice == '7':
            filename = input("Enter file name: ")
            lab.store_pcs(filename)
        elif choice == '8':
            filename = input("Enter file name: ")
            lab.load_pcs(filename)
        elif choice == '9':
```

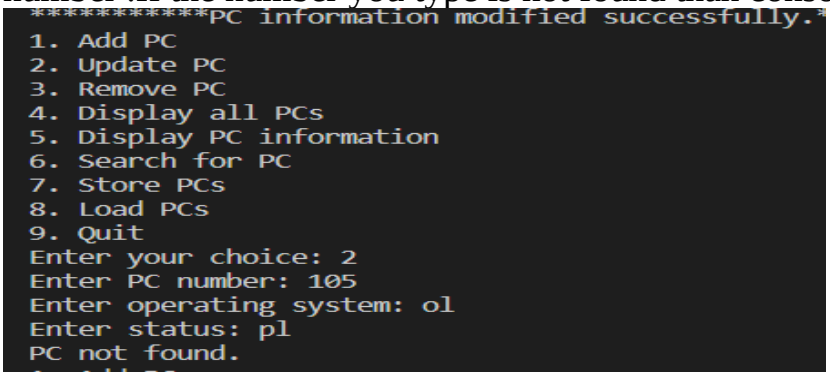
```
            print("GOOD Bye.using again please.")
            break
        else:
            print("Invalid choice. Please enter a number from 1 to 9.")
main()
```

Application Overview:

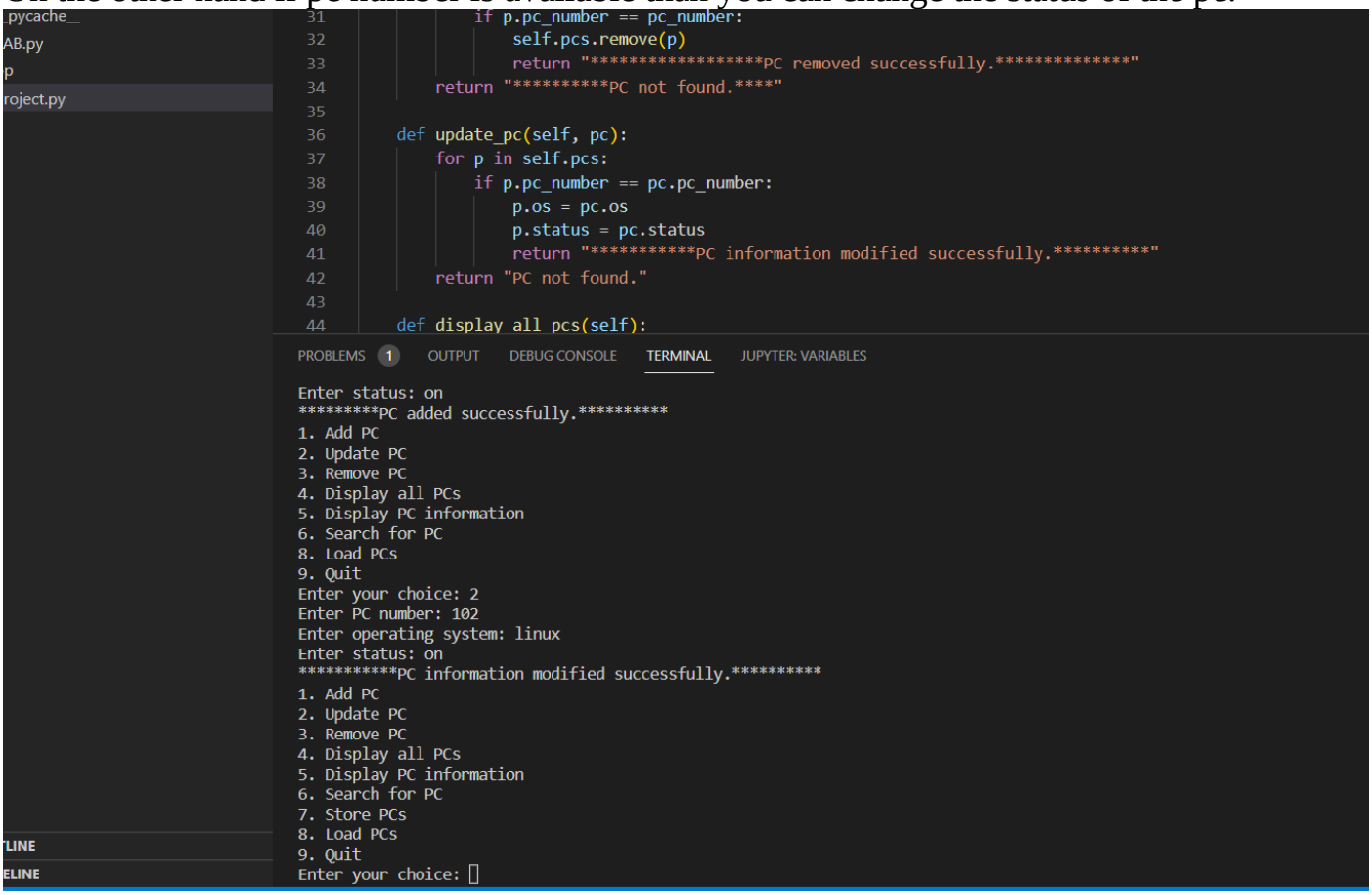
Adding functionality: To add a new pc type 1, then give the input .If input is correct pc added successfully .



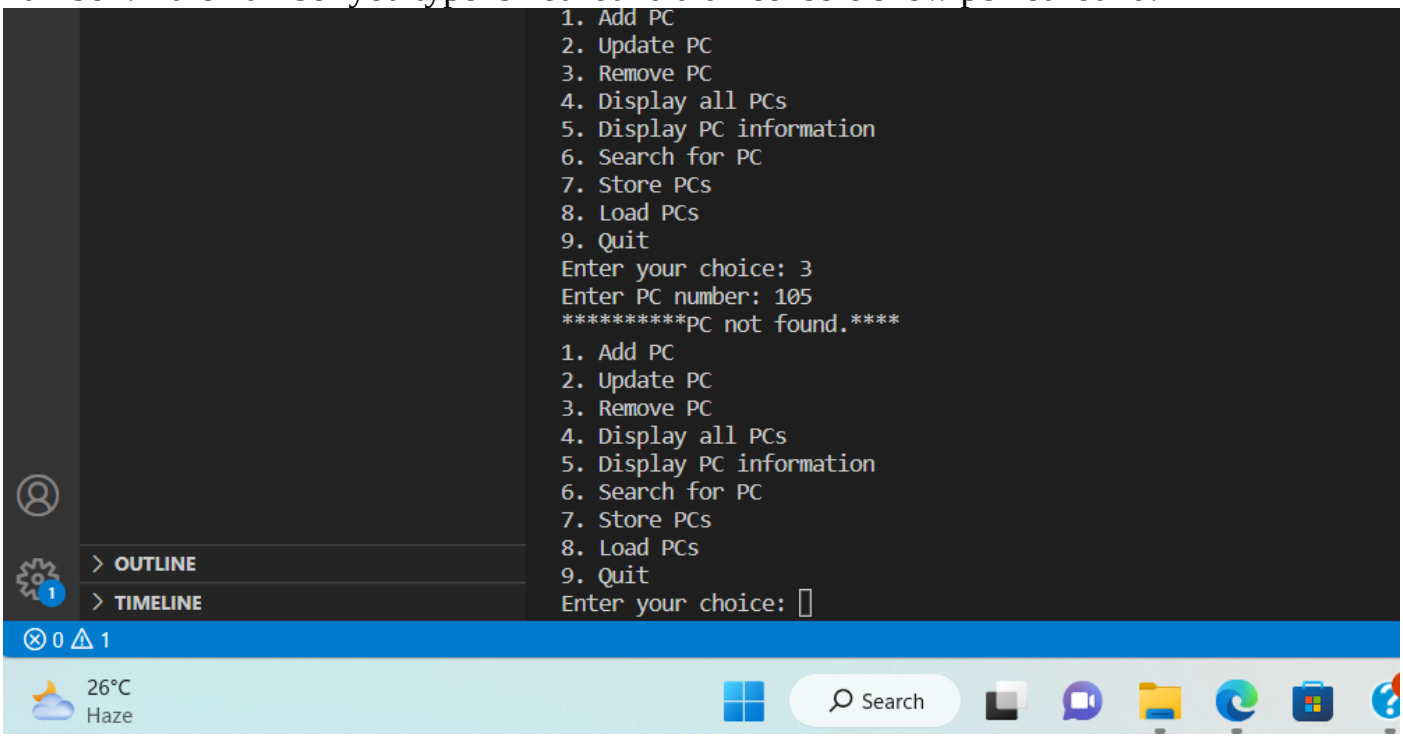
Update functionality: To update a pc information first enter choice 2.Then enter the pc number .If the number you type is not found than console show pc not found.



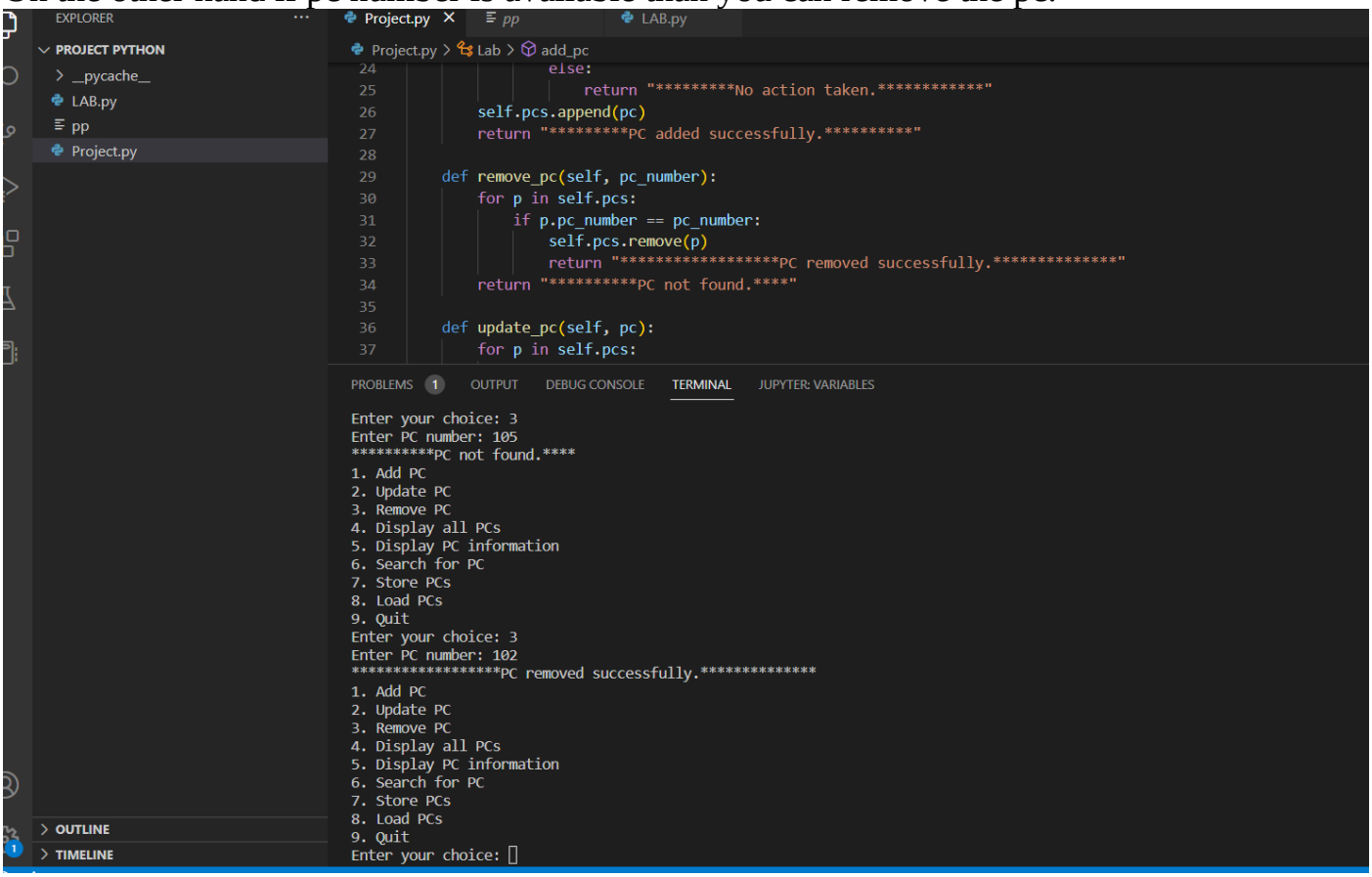
On the other hand if pc number is available than you can change the status of the pc.



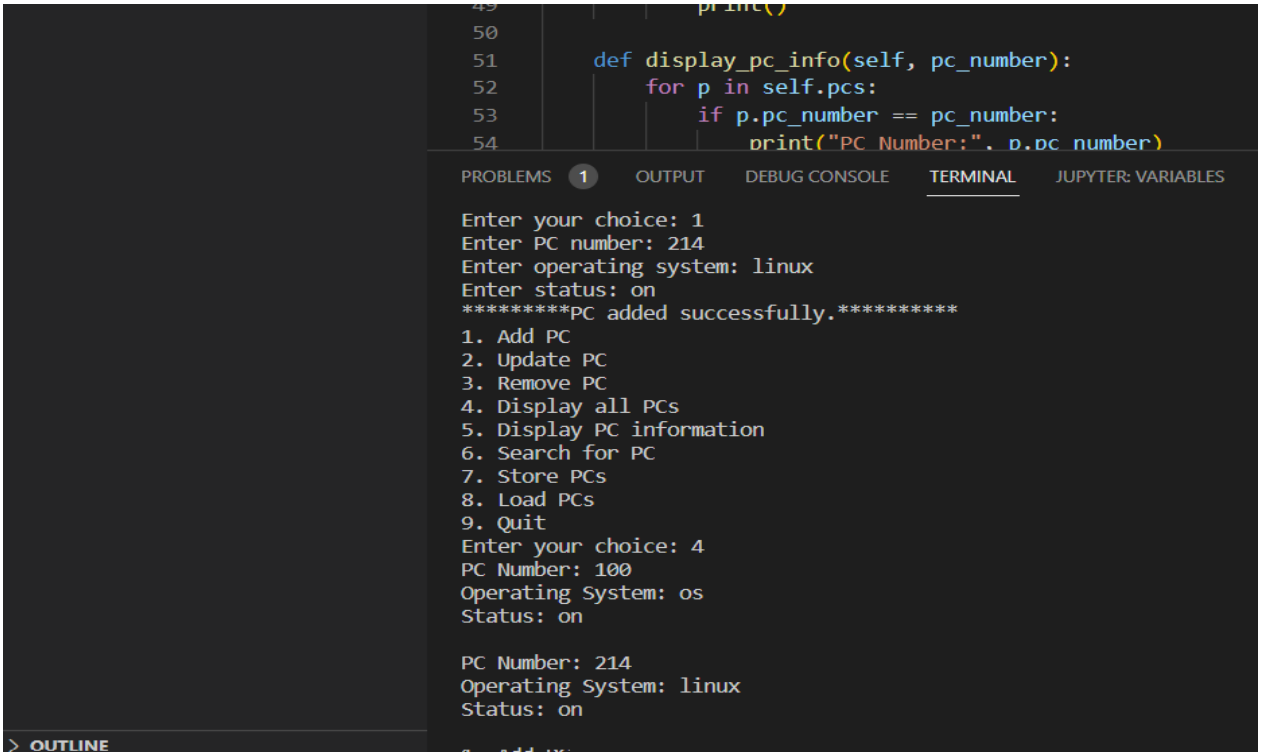
Remove functionality: To remove a pc information first enter choice 3.Then enter the pc number .If the number you type is not found than console show pc not found.



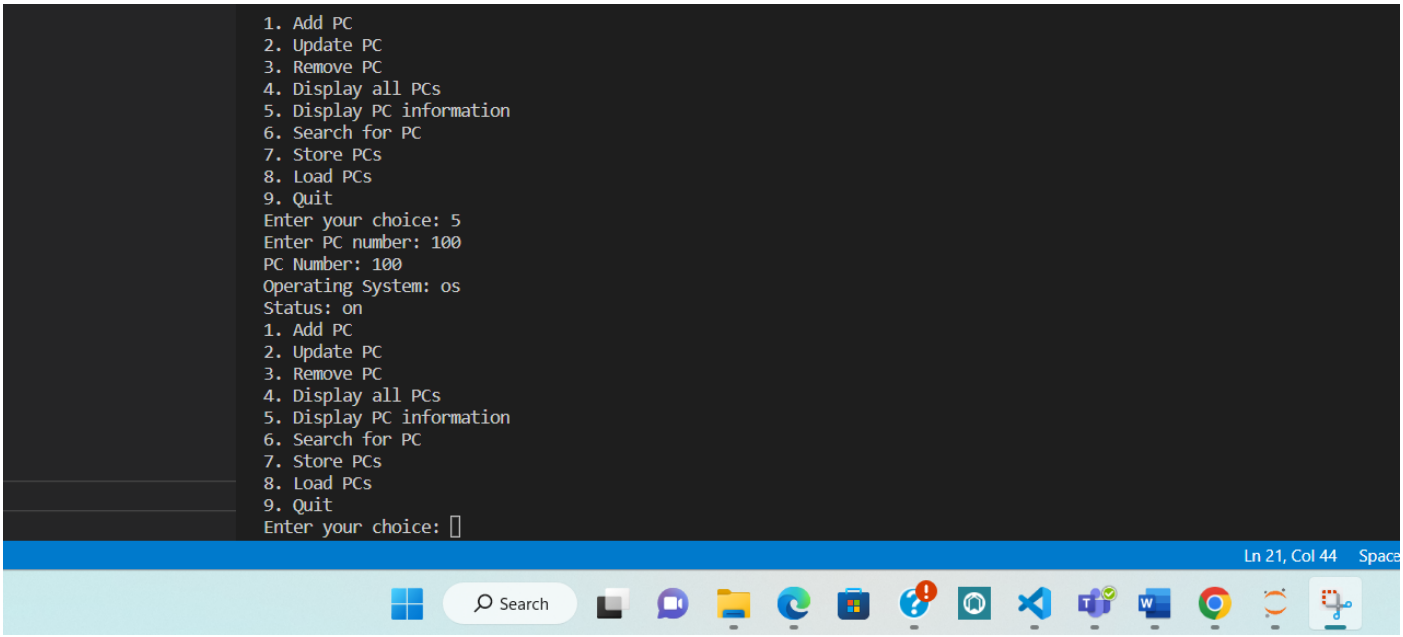
On the other hand if pc number is available than you can remove the pc.



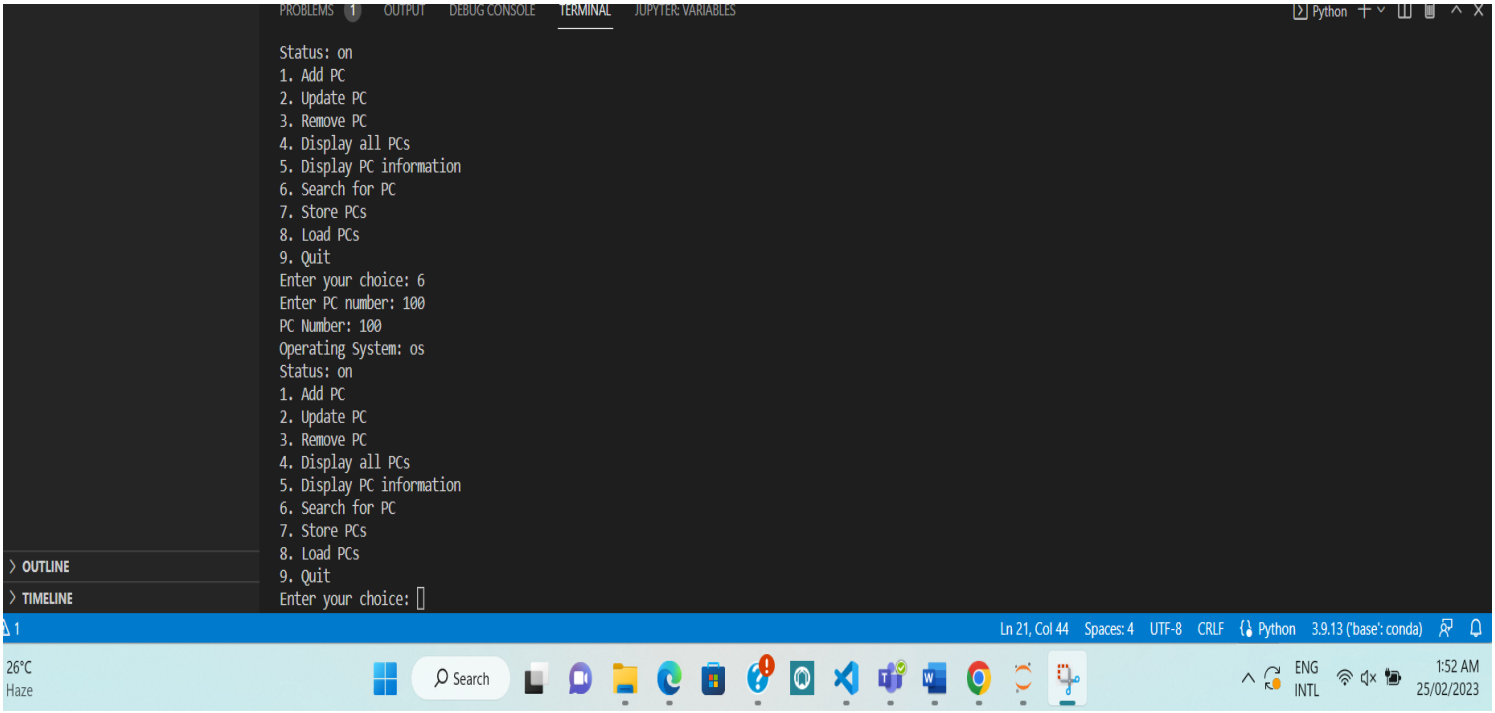
Display All pc functionality: To display all pcs information first enter choice 4.



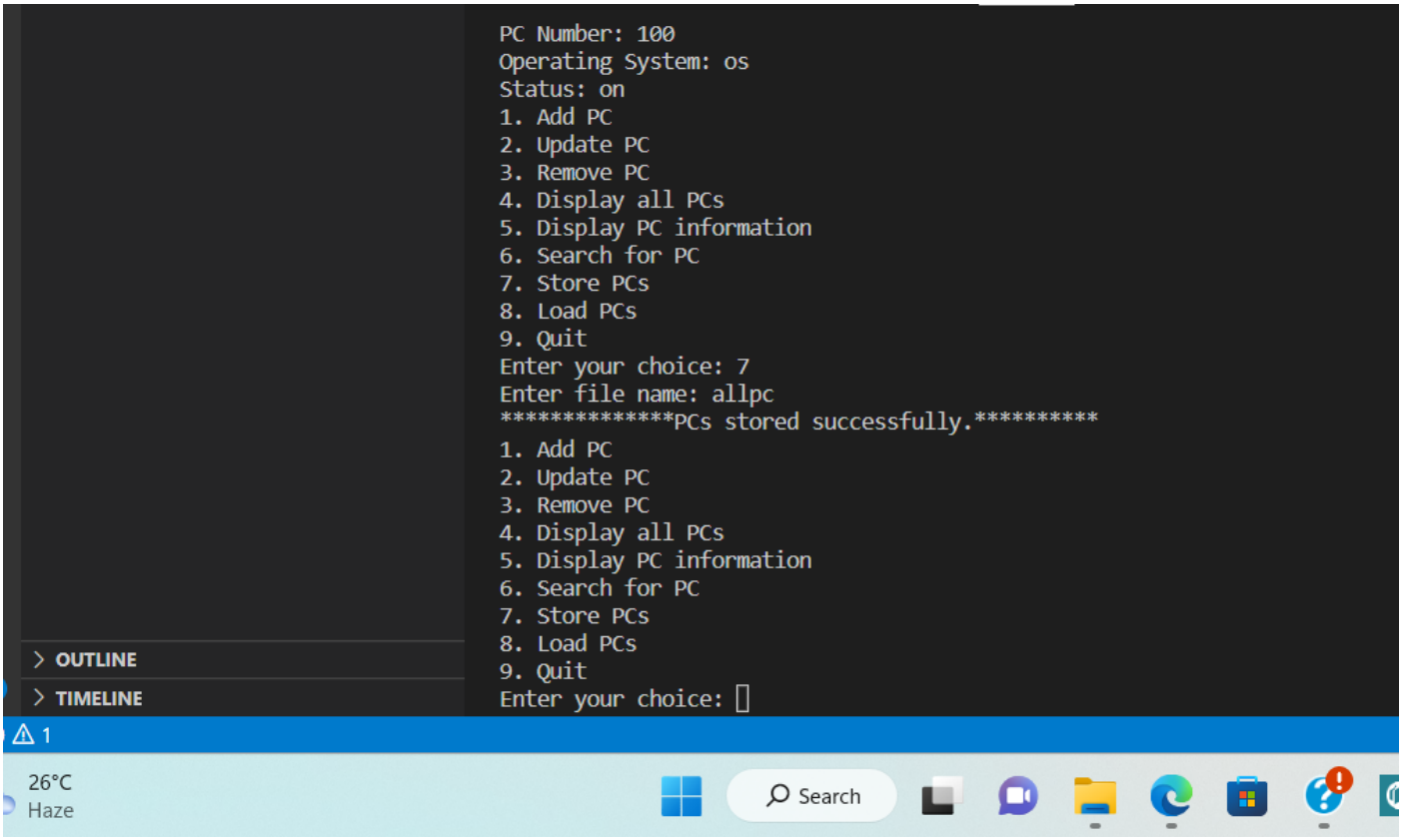
Display a pc functionality: To display a pc information first enter choice 5.Then enter the pc number if pc exists it show the information.

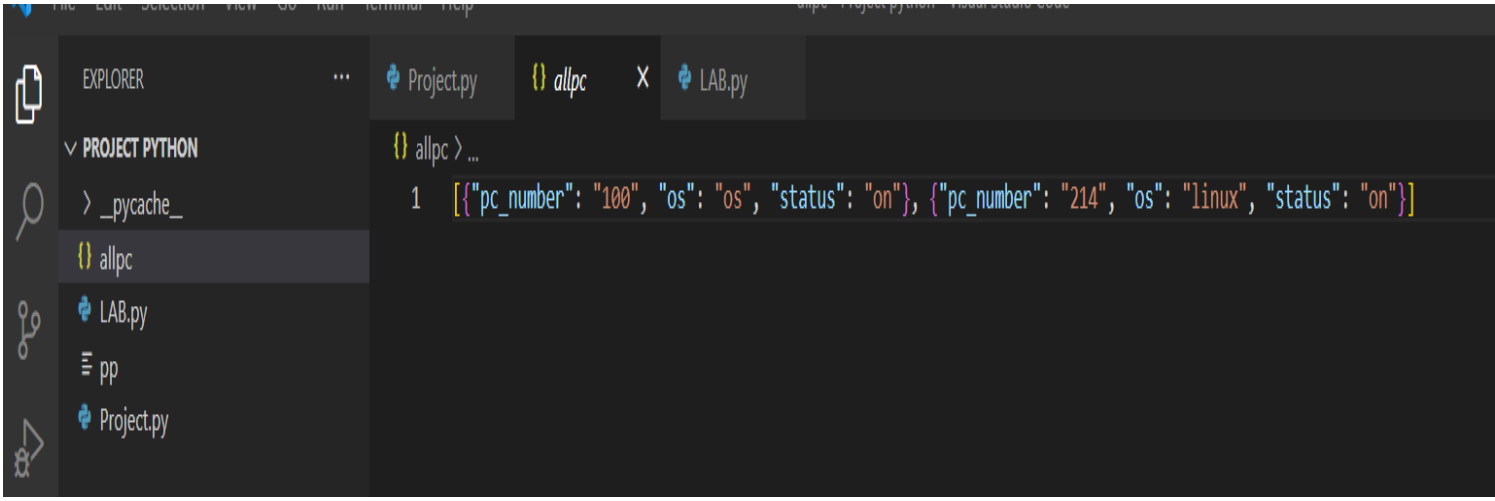


Search pc functionality: To search a pc information first enter choice 6.Then enter the pc number if pc exists it show the information.



Store functionality: To Store all pc information first enter choice 7.The 'store functionality' of the PC lab management application allows users to store all the PC information in a text file for safekeeping. When this option is selected from the menu, the user will be prompted to enter a filename.





Quit functionality: To exit the program enter your choice 9.