



# PROJECT REPORT

**Submitted By,**

**Anik Sen**

**ID : 20-42138-1**

**Section : B**

**Date of Submission: 5<sup>th</sup> March, 2023**

## ***Project overview***

This project is about computer lab management system. It is the console-based application.

First of all I tried to show a menu bar for user. That is why user can see all the functionality and feature of the application by running it. There is total eight options for the user like a user can see all the pc with functionality, can see the specific pc functionality, add a pc, remove a pc, update info of a existing pc, store the information, make a text file on hard disk.

In this application, tried to store the operating system installed in the pc, status of this PC that working or not working and pc number as well. User can add a pc but if there is exist a pc with a same number then user will get a reminder that it is not possible to use the number. User can easily update any records of existing pc and remove it also.

If a user wants to store the records in his hard disk then he/she will easily do it but for this case user have to type a name for the file which will save as a text file.

Users can exit the program by choosing the "quit" option, and the program will be structured around a menu that lists all of the alternatives. The entire piece of code will be developed with a clear structure and appropriate documentation to ensure that it is well-organized and follows to best practices.

## ***Project solution design***

I used the concept of dictionary for keep the records of pc. After user action I convert it to json format and keep it in a json file. In the very beginning, records are stored in a dictionary where key is the pc number other records are value against this key.

When the application has finished executing, a menu will appear, allowing the user to choose any option from the available options at once. Users can return to the menu after completing each job.

To display information about all the PCs in the lab, the application will iterate over the dictionary and display the PC number, operating system, and status for each PC.

To display all the information for a particular PC, the user will be prompted to enter the PC number. The application will then display the operating system and status for that PC.

The application will ask the user to input the PC number, operating system, and status in order to add a new PC. The program will determine whether the PC number is already included in the dictionary before adding it. If so, the program will ask the user if they want to change the current Computer or take it out of the lab. The new PC will be entered into the dictionary if not.

The user will be required to provide the PC number and the updated information (operating system and/or status) in order to update information for an existing PC. The information in the dictionary will then be updated by the program.

The user will be invited to enter a file name to store all the PC data in a text file. The dictionary will then be added to the file by the application.

Each method features a "back to menu" button so that users may always return to the menu after completing a task.

## Implementation

## JsonModule.py

```
import json

jsonFileName='E:\\Python\\MID Project\\PcList.json'

def load():
    with open(jsonFileName) as f:
        FileContent=json.load(f)
    return FileContent
```

## index.py

```
import json
from jsonModule import *

jsonFileName='E:\\Python\\MID Project\\PcList.json'
PC_List={}

class LAB :

    def __init__(self,PC_List) :
        self.PC_List=PC_List


    def main(self):
        print("\n\t\t\t~~~~~ Menu ~~~~~\n")
        print("\t|-----")
        print("\t| 1. Show all PC")
        print("\t| 2. Search a PC")
        print("\t| 3. Add PC")
        print("\t| 4. Update PC")
        print("\t| 5. Remove PC")
```

```

print("\t| 6. Quit")
print("\t|-----")

check=int(input("--> Select an option <--\n"))

while check<1 or check>6:
    check=int(input("\n\nInvalid input! Try again.: "))

if check==1:
    self.showAllPc()
elif check==2:
    self.showPc()
elif check==3:
    self.addPc()
elif check==4:
    check=input("Press the PC No. you want to update: ")
    self.updatePc(check)
elif check==5:
    check=input("Press the PC No. you want to remove: ")
    self.removePc(check)
else:
    print("Logged Out Successfully")
    return

def addPc(self):
    """For add new PC in Lab"""
    pc_no = input("PC no: ")

    try:

        if pc_no in load():
            print("PC number already exists.\n-----\nPress\n")
            print("1. Modify existing PC")
            print("2. Remove existing PC")
            print("3. Back")

            check = int(input("\nEnter your choice: "))

            while check<1 or check>3:
                check=input("Invalid input.! Please Try Again. ")

            if check == 1:

```

```

        self.updatePc(pc_no)

    elif check == 2:
        self.removePc(pc_no)

    elif check == 3:
        self.main()

elif(pc_no.isdigit()):
    os = input("Installed operating system : ")
    status = input("Status : ")
    self.PC_List[pc_no] = {"OS": os, "STATUS": status}
# temporary taken in a list.
    print("PC added successfully.")
    self.storePc(pc_no)

else:
    print("\t----- Numerical input expected. -----\\n")
    print("\\n-----\\nPress\\n")
    print("1. Go back to menu")
    print("2. Try Again")
    check=int(input())
    while check!=1 and check!=2:
        check=input("Invalid input.! Please Try Again. ")
    if check==1:
        self.main()
    else:
        self.addPc()
check=int(input("\\n\\nPress 0 to go back to Menu: "))
while check!=0:
    check=int(input("Invalid input.! Please Try Again. "))
if check==0:
    self.main()

except FileNotFoundError:
    print("File Not Found")

def storePc(self,pc_no):
    """For storing PC in Lab"""
    filename = 'PcList.json'
    with open(jsonFileName,'r') as f:
        FileContent=json.load(f)

```

```

        with open(jsonFileName, 'w') as f:
            json.dump(PC_List, f)

        check = int(input("\n\nPress 1 to save info in a text file \n\tOR\nPress 0
to go back to menu : "))
        while check != 0 or check != 1:
            if check == 0:
                self.main()
            else:
                filename = input("Write File Name: ")
                filename += '.txt'
                filename = 'E:\\Python\\MID Project\\'+filename

                try:
                    with open(filename, 'a') as file:
                        for pc_no, pc_function in FileContent.items():
                            file.write(f"PC number:{pc_no}\n", )
                            file.writelines(f"OS:{pc_function['OS']}\n", )
                            file.writelines(f"STATUS:{pc_function['STATUS']}\n", )

                            file.write("\n")

                    print("\nPC details stored in file\n")
                    check = int(input("To see the stored details press 1:

"))

                    if check == 1:
                        with open(filename) as file:
                            for line in file:
                                print(line)
                except FileNotFoundError:
                    print("Sorry, file not found")

                check = int(input("\n\nPress 0 to go back to Menu: "))
                if check == 0:
                    self.main()

def removePc(self, pc_no):
    """used for remove particular pc"""
    with open(jsonFileName) as f:
        FileContent = json.load(f)

    if pc_no not in FileContent:
        print("PC is not found.")

```

```

else:
    del FileContent[pc_no]
    with open(jsonFileName, 'w') as f:
        json.dump(FileContent, f)
    print("PC removed successfully.")

check=int(input("\n\nPress 0 to go back to Menu: "))
if check==0:
    self.main()

def updatePc(self, pc_no):
    """used for update particular pc"""

    with open(jsonFileName) as f:
        FileContent=json.load(f)

    if pc_no not in FileContent:
        print("PC is not found.")

    else:
        os = input("Installed operating system in the PC: ")
        status = input("Status: ")
        FileContent[pc_no]["OS"] = os
        FileContent[pc_no]["STATUS"] = status

        with open(jsonFileName, 'w') as f:
            json.dump(FileContent, f)

        print("PC information updated successfully.")

    check=int(input("\n\nPress 0 to go back to Menu: "))
    if check==0:
        self.main()

def showAllPc(self):
    """used for show the details of all pc """

    print("\n\n*****Records*****\n")
    with open(jsonFileName) as f:
        FileContent=json.load(f)

    if not FileContent:

```

```

        print("No PC found.")
    else:
        for pc_no,pc_function in FileContent.items():
            print(f"PC number:{pc_no}", )
            print(f"Operating system: {pc_function['OS']}" )
            print(f"Status: {pc_function['STATUS']}")
            print()

    check=int(input("\n\nPress 0 to go back to Menu: "))
    if check==0:
        self.main()

def showPc(self):
    """used for search particular PC"""
    print("\nSearch for a PC info\n")
    pc_number = input("Enter PC number: ")
    with open(jsonFileName) as f:
        FileContent=json.load(f)
    if pc_number in FileContent:
        for pc_no,pc_function in FileContent.items():
            if pc_no==pc_number:
                print(f"PC number:{pc_no}", )
                print(f"Operating system: {pc_function['OS']}" )
                print(f"Status: {pc_function['STATUS']}")
                print()
    else:
        print("\nPC not found.")

    check=input("\n\nPress 0 to go back to Menu: ")
    if check=='0':
        self.main()

```

```

obj=LAB(PC_List)
obj.main()

```

**PcList.json**

```

{"10": {"OS": "Windows", "STATUS": "Working"}, "12": {"OS": "Linux", "STATUS": "Working"}, "13": {"OS": "Linux", "STATUS": "Not Working"}}

```

```

MID Project > {} PcListjson > ...
1 | [{"10": {"OS": "Windows", "STATUS": "Working"}, "12": {"OS": "Linux", "STATUS": "Working"}, "13": {"OS": "Linux", "STATUS": "Not Working"}}]

```



## Application Overview

- When a user wants to add a new pc but it already exists in lab. So user can remove it by the system.

```
PS E:\Python> python -u "e:\Python\MID Project\index.py"

~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC  |
| 5. Remove PC  |
| 6. Functionality of PC |
| 7. Quit      |
|-----|

--> Select an option <--
3
PC no: 10
PC number already exists.
-----
Press

1. Modify existing PC
2. Remove existing PC
3. Back

Enter your choice: 2
PC removed successfully.
```

- User can add PC details.

```
Press 0 to go back to Menu: 0

~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC  |
| 5. Remove PC  |
| 6. Functionality of PC |
| 7. Quit      |
|-----|

--> Select an option <--
3
PC no: 10
Installed operating system : Windows
Status : Working
PC added successfully.

Press 1 to save info in a text file
OR
Press 0 to go back to menu : 0
```

- User can add PC record by this time generate a text file on HDD.

```
          ~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC   |
| 5. Remove PC   |
| 6. Functionality of PC |
| 7. Quit        |
|-----|

--> Select an option <--
3
PC no: 13
Installed operating system : Linux
Status : Not Working
PC added successfully.

Press 1 to save info in a text file
OR
Press 0 to go back to menu : 1
Write File Name: record

PC details stored in file
```

- This is how a .txt file generate

```
MID Project > record.txt

1  PC number:10
2  OS:Windows
3  STATUS:Working
4
5  PC number:12
6  OS:Linux
7  STATUS:Working
8
9  PC number:13
10 OS:Linux
11 STATUS:Not Working
12
13
```

- User can see the records of PC in lab

```
PS E:\Python> python -u "e:\Python\MID Project\index.py"

          ~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC   |
| 5. Remove PC   |
| 6. Functionality of PC |
| 7. Quit        |
|-----|

--> Select an option <--
1

*****Records*****

PC number:10
Operating system: Windows
Status: Working

PC number:12
Operating system: Linux
Status: Working

PC number:13
Operating system: Linux
Status: Not Working
```

➤ *User can update PC details*

```
Press 0 to go back to Menu: 0

          ~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC   |
| 5. Remove PC   |
| 6. Functionality of PC |
| 7. Quit        |
|-----|

--> Select an option <--
4
Press the PC No. you want to update: 10
Installed operating system in the PC: Windows
Status: Not Working
PC information updated successfully.

Press 0 to go back to Menu: 0
```

➤ *User can search a particular PC*

```
PS E:\Python> python -u "e:\Python\MID Project\index.py"

~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC   |
| 5. Remove PC   |
| 6. Quit        |
|-----|

--> Select an option <--
2

Search for a PC info

Enter PC number: 10
PC number:10
Operating system: Windows
Status: Not Working

Press 0 to go back to Menu: █
```

➤ *Logging out*

```
PS E:\Python> python -u "e:\Python\MID Project\index.py"

~~~~~ Menu ~~~~~

|-----|
| 1. Show all PC |
| 2. Search a PC |
| 3. Add PC      |
| 4. Update PC   |
| 5. Remove PC   |
| 6. Quit        |
|-----|

--> Select an option <--
6
Logged Out Successfully
PS E:\Python> █
```

-----