Program -3
Name: Mohsin Chunawala
Reg No: 2447218

Demonstrate Custom modules with functions

Your company uses **SmartScan Codes** to streamline user registration. You need to implement a system that reads user data from a SmartScan Code image and manages it using custom modules with lambda functions.

(a) Create a Python module named smartscan_registration_module.py that includes:

In-Memory Storage: Simulate a database using a list of dictionaries. Define lambda functions within the module for:

i. Creating a new user record.
ii. Inserting the user record into the list.
iii. Fetching all user records from the list.

SmartScan Code Scanning: Implement a function that reads and decodes the SmartScan Code. The SmartScan Code contains user information encoded as a comma-separated string in the format "name,email".

User Registration Function: Implement a function RegisterUserFromSmartScan that:

i. Uses the scanning function to extract user data.
ii. Uses the lambda functions to create and insert the user record into the in-memory list.
iii. Prints the list of all registered users after adding the new user.

(b) Place the above function in a separate module file and create another script to import this module and invoke the function within the script.

Evaluation Rubrics

Timely Submission - 2 Marks
Correctness&Clarity - 4 Marks
Complexity&Validation - 2 Marks
Viva Voice - 2 Marks

Submission Guidelines

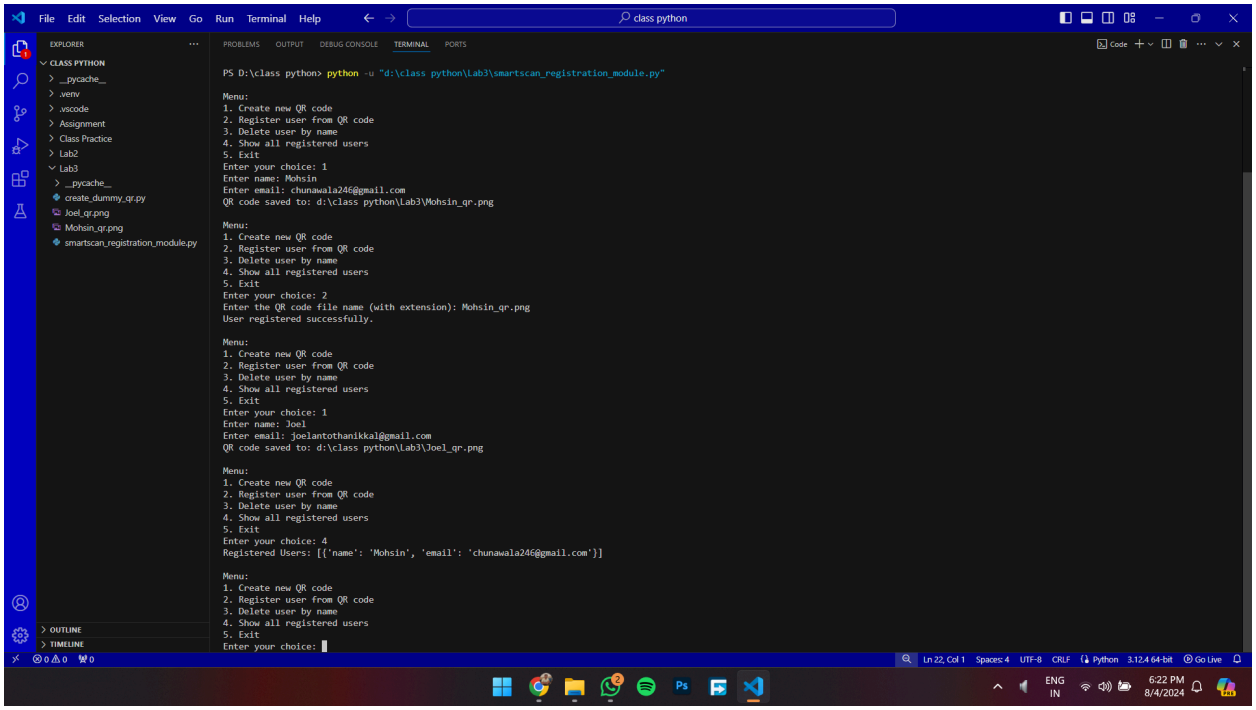
Generate the .pdf file for Program3 separately and save the file name with your register number followed by program No: Example: 2048501_P3

Upload the .pdf files in Google Classroom on or before the stated deadline.

```python
import qrcode
import os

def create_qr_code(name, email):
    # User data to encode in the QR code
    user_data = f"{name},{email}"

    # Create QR code
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )
    qr.add_data(user_data)
    qr.make(fit=True)

    # Create an image from the QR Code instance
    img = qr.make_image(fill='black', back_color='white')

    # Save the QR code image in the same folder as the script
    script_dir = os.path.dirname(os.path.abspath(__file__))
    img_name = f"{name.replace(' ', '_')}_qr.png"
    img_path = os.path.join(script_dir, img_name)
    img.save(img_path)

    print(f"QR code saved to: {img_path}")
    return img_path

if __name__ == "__main__":
    name = input("Enter name: ")
    email = input("Enter email: ")
    create_qr_code(name, email)
```

```python
import os
from pyzbar.pyzbar import decode
from PIL import Image

# In-Memory Storage
user_records = []

# Lambda functions
create_user_record = lambda name, email: {"name": name, "email": email}
insert_user_record = lambda record: user_records.append(record)
delete_user_record = lambda name: [user_records.remove(user) for user in
user_records if user['name'] == name]
fetch_all_user_records = lambda: user_records

# Function to scan and decode the SmartScan Code (QR Code)
def scan_qr_code(image_path):
    decoded_objects = decode(Image.open(image_path))
    for obj in decoded_objects:
        return obj.data.decode('utf-8')
    return None

# Function to register user from SmartScan (QR Code)
def RegisterUserFromSmartScan(image_path):
    user_data = scan_qr_code(image_path)
    if user_data:
        name, email = user_data.split(',')
        user_record = create_user_record(name, email)
        insert_user_record(user_record)
        print("User registered successfully.")
    else:
        print("No valid QR code found.")

# Menu-based interaction
def menu():
    while True:
        print("\nMenu:")
        print("1. Create new QR code")
        print("2. Register user from QR code")
        print("3. Delete user by name")
        print("4. Show all registered users")
```

```
        print("5. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            name = input("Enter name: ")
            email = input("Enter email: ")
            from create_dummy_qr import create_qr_code
            create_qr_code(name, email)

        elif choice == '2':
            file_name = input("Enter the QR code file name (with
extension): ")
            script_dir = os.path.dirname(os.path.abspath(__file__))
            img_path = os.path.join(script_dir, file_name)
            if os.path.exists(img_path):
                RegisterUserFromSmartScan(img_path)
            else:
                print("File not found.")

        elif choice == '3':
            name = input("Enter the name of the user to delete: ")
            delete_user_record(name)
            print("User deleted successfully.")

        elif choice == '4':
            all_users = fetch_all_user_records()
            print("Registered Users:", all_users)

        elif choice == '5':
            break

        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    menu()
```


File Edit Selection View Go Run Terminal Help
class python
EXPLORER
CLASS PYTHON
__pycache__
.venv
.vscode
Assignment
Class Practice
Lab2
Lab3
__pycache__
create_dummy_qr.py
Joel_qr.png
Mohsin_qr.png
smartscan_registration_module.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
PS D:\class python> python -u "d:\class python\Lab3\smartscan_registration_module.py"
Menu:
1. Create new QR code
2. Register user from QR code
3. Delete user by name
4. Show all registered users
5. Exit
Enter your choice: 1
Enter name: Mohsin
Enter email: chunawala246@gmail.com
QR code saved to: d:\class python\Lab3\Mohsin_qr.png
Menu:
1. Create new QR code
2. Register user from QR code
3. Delete user by name
4. Show all registered users
5. Exit
Enter your choice: 2
Enter the QR code file name (with extension): Mohsin_qr.png
User registered successfully.
Menu:
1. Create new QR code
2. Register user from QR code
3. Delete user by name
4. Show all registered users
5. Exit
Enter your choice: 1
Enter name: Joel
Enter email: joelantothanikkal@gmail.com
QR code saved to: d:\class python\Lab3\Joel_qr.png
Menu:
1. Create new QR code
2. Register user from QR code
3. Delete user by name
4. Show all registered users
5. Exit
Enter your choice: 4
Registered Users: [{'name': 'Mohsin', 'email': 'chunawala246@gmail.com'}]
Menu:
1. Create new QR code
2. Register user from QR code
3. Delete user by name
4. Show all registered users
5. Exit
Enter your choice:
Ln 22, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit Go Live
ENG IN
6:22 PM 8/4/2024