Python Lab – 3

2447155 Sujay Sharma

Q1 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.

User_scan Module

```
import qrcode
from PIL import Image
from pyzbar.pyzbar import decode

# In-Memory Storage: Simulate a database using a list of dictionaries
user_records = []

# Lambda function to create a new user record
create_user = lambda name, email: {'name': name, 'email': email}

# Lambda function to insert the user record into the list
insert_user = lambda user: user_records.append(user)

# Lambda function to fetch all user records from the list
fetch_all_users = lambda: user_records

# Function to generate QR code from inputted data
def generate_qr_code(data):
    img = qrcode.make(data)
    img.save('imgqr.png')
    print("Image generated and saved as imgqr.png")

# Function to decode the QR code
def decode_qr_code(image_path):
```

```
    img = Image.open(image_path)
    decoded_data_raw = decode(img)
    if decoded_data_raw:
        decoded_data = decoded_data_raw[0].data.decode('utf-8')
        return decoded_data
    return ""

# Main file
def RegisterUserFromSmartScan(image_path):
    # Decode the SmartScan Code to extract user data
    user_data = decode_qr_code(image_path)

    # Split user data by newlines if multiple records are encoded
    records = user_data.split('\n')

    for record in records:
        try:
            # Extract name and email from the decoded data
            name, email = record.split(',')

            # Create a new user record using the lambda function
            new_user = create_user(name, email)

            # Insert the user record into the in-memory list
            insert_user(new_user)
        except ValueError:
```

```
        print(f"Skipping invalid record: {record}")

    # Print the list of all registered users
    print("Registered Users:")
    for user in fetch_all_users():
        print(f"Name: {user['name']}, Email: {user['email']}")
```

Register_user.py

```
import smartscan_registration_module as srm

# Defining the list to store the inputted data
user_data = []

print("***Enter the user data***")

choice = True
i = 1
# Loop to get as many records as you want
while choice:
    print("User ", i)
    name = input("Enter name: ")
    email = input("Enter email: ")
    data = f"{name},{email}"
    user_data.append(data)
```

```
    more = input("Do you want to add another user? (yes/no): ")
    # Using .lower() to account for case sensitivity
    if more.lower() == "yes":
        i += 1
    else:
        choice = False

# Concatenate all user data into a single string separated by newlines
all_user_data = "\n".join(user_data)
# Calling the functions to get the outputs
srm.generate_qr_code(all_user_data)

srm.RegisterUserFromSmartScan("imgqr.png")
```