TRIBHUVAN UNIVERSITY
Institute of Engineering
Pulchowk Campus

LAB REPORT 4:

Submitted By:
Sashank Bhattarai
081BEL075

Submitted To:
Department of Computer Engineering

Project 1

Create a CLI (Command Line Interface) contact book that allows users to:

- Add a new contact (append to file)
- View all contacts (read from file)
- Search for a contact (read and filter)
- Handle file-related exceptions (e.g., file not found)

File Used:

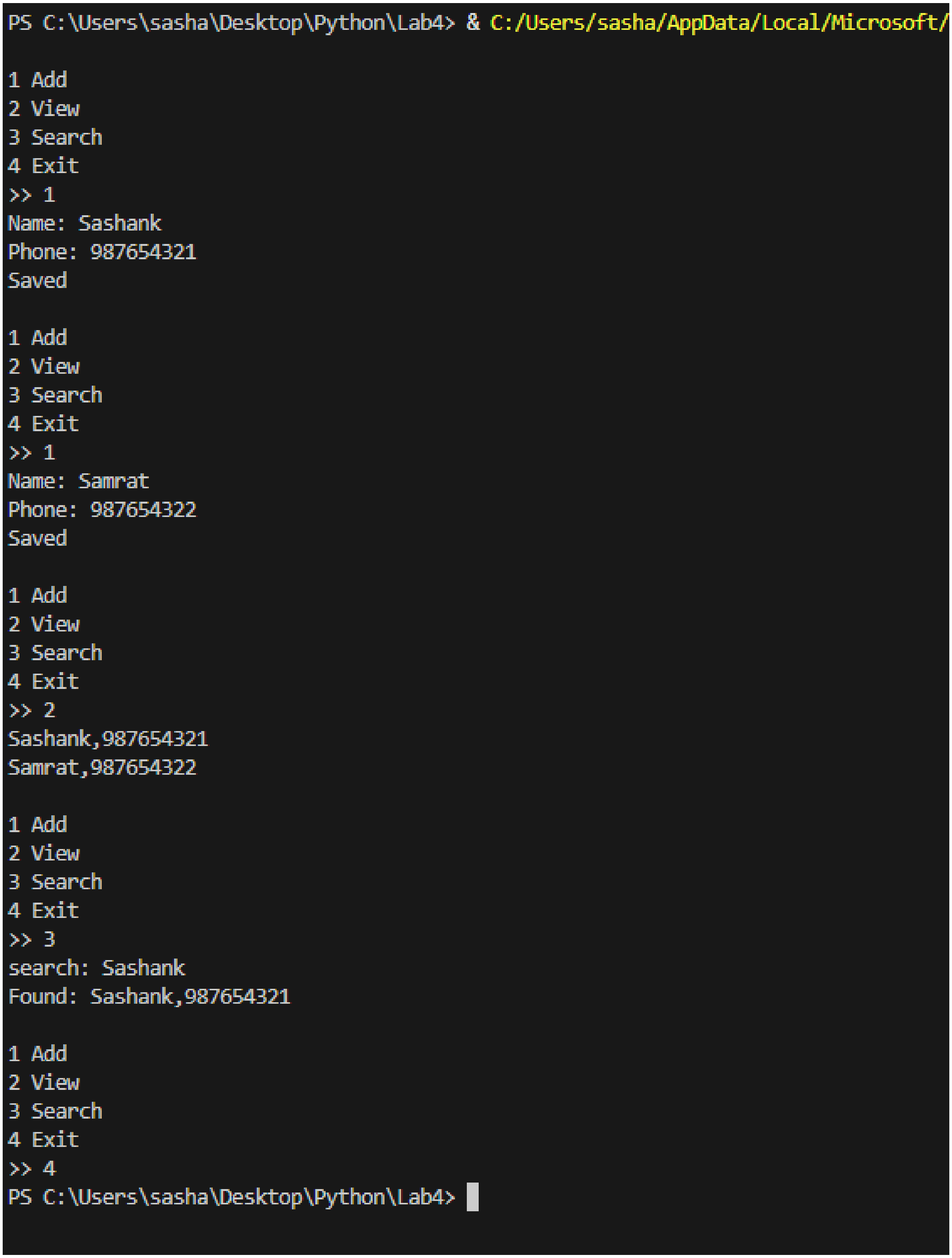
contacts.txt (stores contact info: Name, Phone)

PROGRAM:

```
def add():
    try:
        name = input("Name: ")
        phone = input("Phone: ")
        with open("contacts.txt", "a") as f:
            f.write(name + "," + phone + "\n")
        print("Saved")
    except Exception as e:
        print("Error while adding:", e)


def view():
    try:
        with open("contacts.txt", "r") as f:
            lines = f.readlines()
            if len(lines) == 0:
                print("No contacts found")
            else:
                for l in lines:
                    print(l.strip())
    except FileNotFoundError:
        print("contacts.txt not found")

def search():
    try:
        x = input("search: ")
        found = False
        with open("contacts.txt", "r") as f:
            for l in f:
                if x in l:
                    print("Found:", l.strip())
                    found = True
        if not found:
            print("No contact found")
    except FileNotFoundError:
        print("contacts.txt not found")

while True:
    print("\n1 Add\n2 View\n3 Search\n4 Exit")
```

```
c = input(">> ")
if c == "1":
    add()
elif c == "2":
    view()
elif c == "3":
    search()
elif c == "4":
    break
else:
    print("Invalid choice")
```

OUTPUT:



```
PS C:\Users\sasha\Desktop\Python\Lab4> & C:/Users/sasha/AppData/Local/Microsoft/

1 Add
2 View
3 Search
4 Exit
>> 1
Name: Sashank
Phone: 987654321
Saved

1 Add
2 View
3 Search
4 Exit
>> 1
Name: Samrat
Phone: 987654322
Saved

1 Add
2 View
3 Search
4 Exit
>> 2
Sashank,987654321
Samrat,987654322

1 Add
2 View
3 Search
4 Exit
>> 3
search: Sashank
Found: Sashank,987654321

1 Add
2 View
3 Search
4 Exit
>> 4
PS C:\Users\sasha\Desktop\Python\Lab4>
```

Project 2

Create a simple banking system that:

- Stores customer info in a file
- Allows deposits and withdrawals using functions
- Updates customer balance
- Logs all transactions in a separate file
- Handles exceptions gracefully

Files Used:

customers.txt — stores customer records in the format:

Name,AccountNumber,Balance

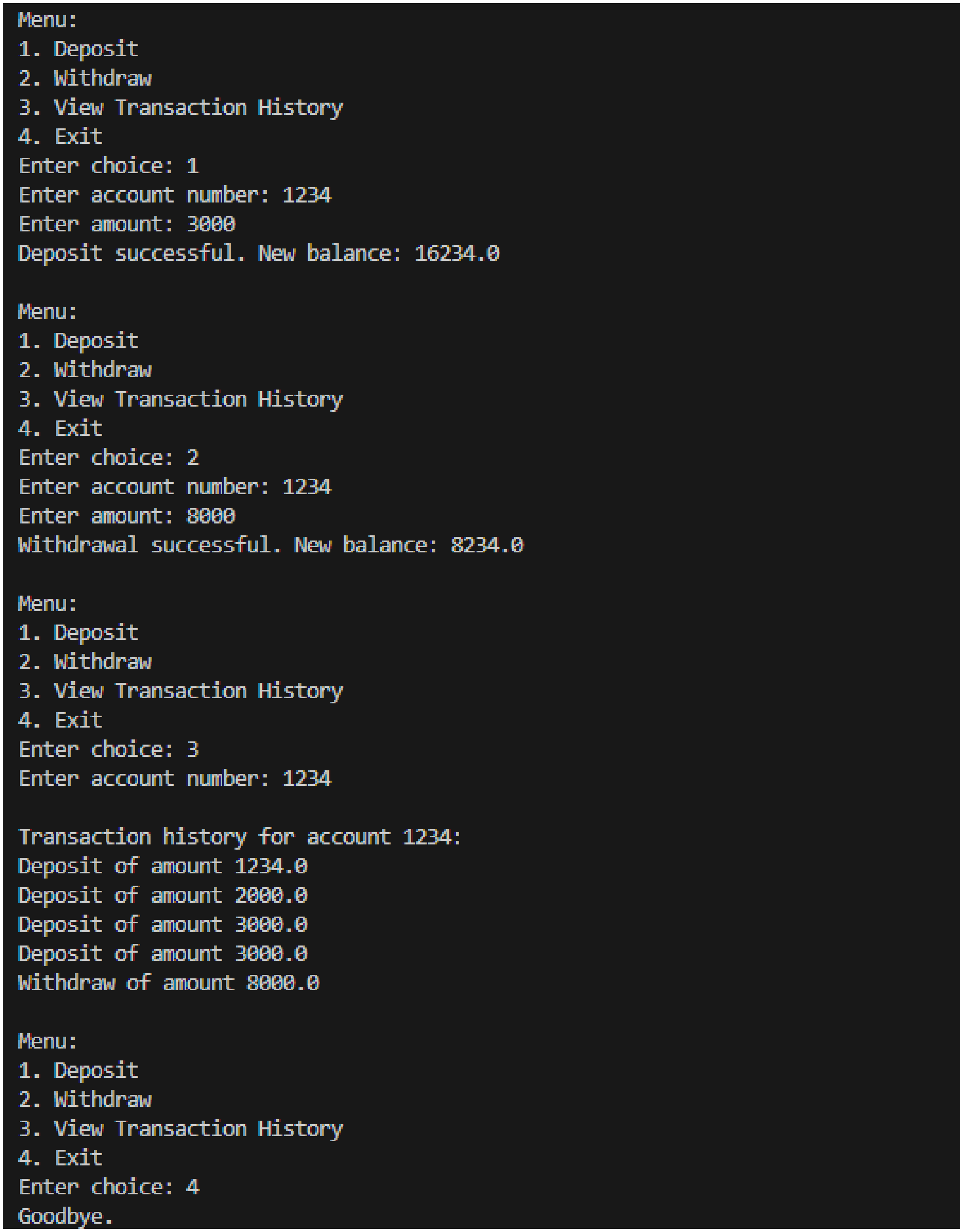
transactions.txt — appends every deposit or withdrawal record with timestamp

PROGRAM:

```
# Initialize customer file
def initialize():
    try:
        with open("customers.txt", "r") as f:
            if f.readline().strip() == "":
                raise Exception("Empty file")
    except:
        with open("customers.txt", "w") as f:
            f.write("Sashank,1234,7000\n")
            f.write("Rahul,5678,8000\n")
            f.write("Samrat,9101,1000\n")

# Load customers into dictionary
def load_customers():
    customers = {}
    try:
        with open("customers.txt", "r") as f:
            for line in f:
                parts = line.strip().split(",")
                if len(parts) == 3:
                    name, acc, bal = parts
                    try:
                        customers[acc.strip()] = [name.strip(), float(bal)]
                    except ValueError:
                        continue
    except:
        print("Error loading customers.")
    return customers

# Save customers back to file
def save_customers(customers):
    try:
        with open("customers.txt", "w") as f:
            for acc, (name, bal) in customers.items():
                f.write(f"{name},{acc},{bal}\n")
    except:
        print("Error saving customers.")

# Log transaction (no timestamp)
def log_transaction(acc, type_, amount):
    try:
        with open("transactions.txt", "a") as f:
            f.write(f"{acc},{type_},{amount}\n")
    except:
        print("Error logging transaction.")

# Show transaction history for an account
def show_history(acc):
    found = False
    try:
        with open("transactions.txt", "r") as f:
            print(f"\nTransaction history for account {acc}:")
            for line in f:
                parts = line.strip().split(",")
                if len(parts) == 3 and parts[0] == acc:
                    print(f"{parts[1]} of amount {parts[2]}")
                    found = True
        if not found:
            print("No transactions found.")
    except:
        print("Error reading transaction history.")

# Deposit function
def deposit(customers, acc, amount):
    if acc in customers:
        customers[acc][1] += amount
        log_transaction(acc, "Deposit", amount)
        print(f"Deposit successful. New balance: {customers[acc][1]}")
    else:
        print("Account not found.")

# Withdraw function
def withdraw(customers, acc, amount):
    if acc in customers:
        if customers[acc][1] >= amount:
            customers[acc][1] -= amount
            log_transaction(acc, "Withdraw", amount)
            print(f"Withdrawal successful. New balance: {customers[acc][1]}")
        else:
            print("Insufficient balance.")
    else:
        print("Account not found.")
```

```
# Main program
initialize()
customers = load_customers()

while True:
    print("\nMenu:")
    print("1. Deposit")
    print("2. Withdraw")
    print("3. View Transaction History")
    print("4. Exit")
    choice = input("Enter choice: ").strip()

    if choice in ["1", "2"]:
        acc = input("Enter account number: ").strip()
        if acc not in customers:
            print("Account not found.")
            continue
        try:
            amount = float(input("Enter amount: ").strip())
            if amount <= 0:
                print("Amount must be greater than 0.")
                continue
        except ValueError:
            print("Invalid amount.")
            continue

        if choice == "1":
            deposit(customers, acc, amount)
        else:
            withdraw(customers, acc, amount)

    elif choice == "3":
        acc = input("Enter account number: ").strip()
        if acc not in customers:
            print("Account not found.")
        else:
            show_history(acc)

    elif choice == "4":
        save_customers(customers)
        print("Goodbye.")
        break
    else:
        print("Invalid choice.")
```

OUTPUT:



```
Menu:
1. Deposit
2. Withdraw
3. View Transaction History
4. Exit
Enter choice: 1
Enter account number: 1234
Enter amount: 3000
Deposit successful. New balance: 16234.0

Menu:
1. Deposit
2. Withdraw
3. View Transaction History
4. Exit
Enter choice: 2
Enter account number: 1234
Enter amount: 8000
Withdrawal successful. New balance: 8234.0

Menu:
1. Deposit
2. Withdraw
3. View Transaction History
4. Exit
Enter choice: 3
Enter account number: 1234

Transaction history for account 1234:
Deposit of amount 1234.0
Deposit of amount 2000.0
Deposit of amount 3000.0
Deposit of amount 3000.0
Withdraw of amount 8000.0

Menu:
1. Deposit
2. Withdraw
3. View Transaction History
4. Exit
Enter choice: 4
Goodbye.
```