TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A REPORT ON

Use of function in python programming

SUBMITTED BY:

SUDIN MAHARJAN (081BEL084)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & 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:

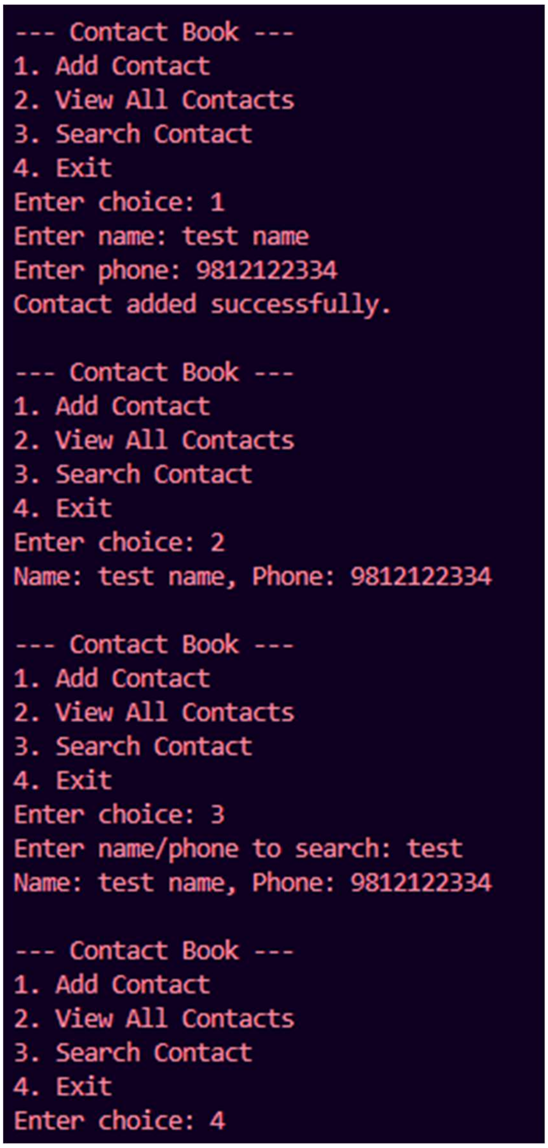
contacts.txt (stores contact info: Name, Phone)

CODE:

```
import os


FILENAME = "contacts.txt"


def add_contact(name, phone):
    try:
        with open(FILENAME, "a") as f:
            f.write(f"{name},{phone}\n")
        print("Contact added successfully.")
    except Exception as e:
        print("Error while adding contact:", e)


def view_contacts():
    try:
        with open(FILENAME, "r") as f:
            contacts = f.readlines()
            if not contacts:
                print("No contacts found.")
            for line in contacts:
                name, phone = line.strip().split(",")
                print(f"Name: {name}, Phone: {phone}")
    except FileNotFoundError:
        print("No contacts file found.")
    except Exception as e:
        print("Error while reading contacts:", e)

def search_contact(keyword):
    try:
        with open(FILENAME, "r") as f:
            found = False
            for line in f:
                name, phone = line.strip().split(",")
                if keyword.lower() in name.lower() or keyword in phone:
                    print(f"Name: {name}, Phone: {phone}")
                    found = True
            if not found:
                print("No matching contact found.")
    except FileNotFoundError:
        print("No contacts file found.")

while True:
    print("\n--- Contact Book ---")
    print("1. Add Contact")
    print("2. View All Contacts")
    print("3. Search Contact")
    print("4. Exit")
    choice = input("Enter choice: ")
```

```
if choice == "1":
    n = input("Enter name: ")
    p = input("Enter phone: ")
    add_contact(n, p)
elif choice == "2":
    view_contacts()
elif choice == "3":
    k = input("Enter name/phone to search: ")
    search_contact(k)
elif choice == "4":
    break
else:
    print("Invalid choice")
```

OUTPUT:



```
--- Contact Book ---
1. Add Contact
2. View All Contacts
3. Search Contact
4. Exit
Enter choice: 1
Enter name: test name
Enter phone: 9812122334
Contact added successfully.

--- Contact Book ---
1. Add Contact
2. View All Contacts
3. Search Contact
4. Exit
Enter choice: 2
Name: test name, Phone: 9812122334

--- Contact Book ---
1. Add Contact
2. View All Contacts
3. Search Contact
4. Exit
Enter choice: 3
Enter name/phone to search: test
Name: test name, Phone: 9812122334

--- Contact Book ---
1. Add Contact
2. View All Contacts
3. Search Contact
4. Exit
Enter choice: 4
```

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

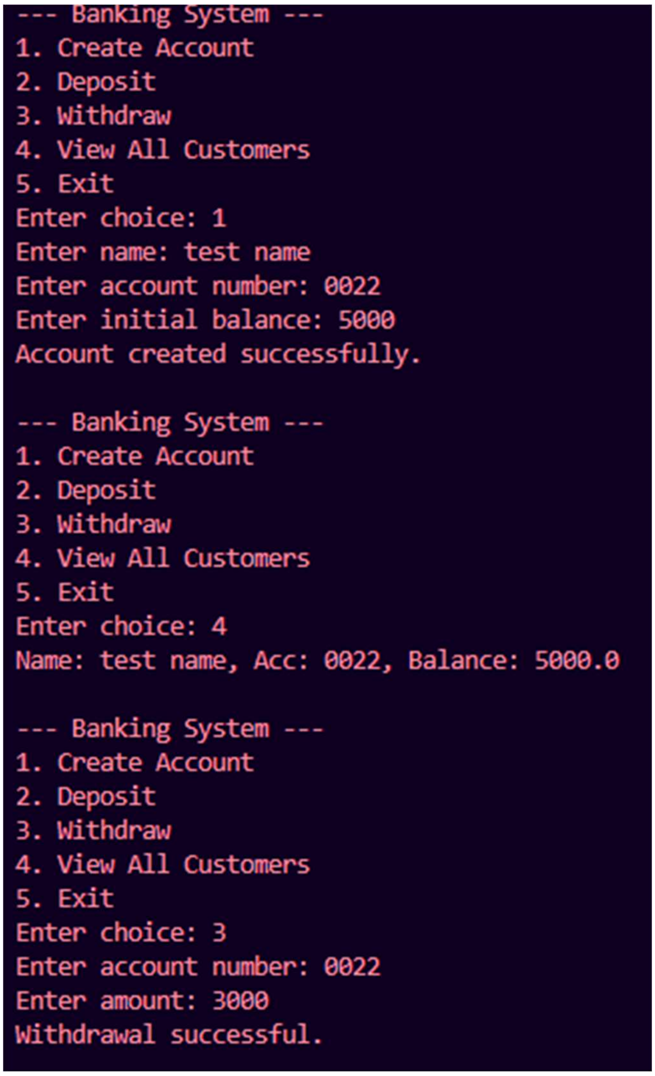
CODE:

```
import os
from datetime import datetime

CUSTOMER_FILE = "customers.txt"
TRANSACTION_FILE = "transactions.txt"

def load_customers():
    customers = {}
    try:
        with open(CUSTOMER_FILE, "r") as f:
            for line in f:
                name, acc, bal = line.strip().split(",")
                customers[acc] = {"name": name, "balance": float(bal)}
    except FileNotFoundError:
        pass
    return customers

def save_customers(customers):
    with open(CUSTOMER_FILE, "w") as f:
        for acc, data in customers.items():
            f.write(f"{data['name']},{acc},{data['balance']}\n")

def log_transaction(acc, action, amount):
    with open(TRANSACTION_FILE, "a") as f:
        f.write(f"{datetime.now()}, {acc}, {action}, {amount}\n")
```

```
def deposit(customers, acc, amount):
    if acc in customers:
        customers[acc]["balance"] += amount
        log_transaction(acc, "Deposit", amount)
        print("Deposit successful.")
    else:
        print("Account not found.")

def withdraw(customers, acc, amount):
    if acc in customers:
        if customers[acc]["balance"] >= amount:
            customers[acc]["balance"] -= amount
            log_transaction(acc, "Withdraw", amount)
            print("Withdrawal successful.")
        else:
            print("Insufficient balance.")
    else:
        print("Account not found.")

def create_account(customers, name, acc, bal):
    if acc in customers:
        print("Account already exists.")
    else:
        customers[acc] = {"name": name, "balance": bal}
        print("Account created successfully.")

customers = load_customers()
```

```
while True:
    print("\n--- Banking System ---")
    print("1. Create Account")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. View All Customers")
    print("5. Exit")
    choice = input("Enter choice: ")

    if choice == "1":
        n = input("Enter name: ")
        a = input("Enter account number: ")
        b = float(input("Enter initial balance: "))
        create_account(customers, n, a, b)

    elif choice == "2":
        a = input("Enter account number: ")
        amt = float(input("Enter amount: "))
        deposit(customers, a, amt)

    elif choice == "3":
        a = input("Enter account number: ")
        amt = float(input("Enter amount: "))
        withdraw(customers, a, amt)

    elif choice == "4":
        for acc, data in customers.items():
            print(f"Name: {data['name']}, Acc: {acc}, Balance: {data['balance']}")
    elif choice == "5":
        save_customers(customers)
        break

    else:
        print("Invalid choice")
```

OUTPUT:



```
--- Banking System ---
1. Create Account
2. Deposit
3. Withdraw
4. View All Customers
5. Exit
Enter choice: 1
Enter name: test name
Enter account number: 0022
Enter initial balance: 5000
Account created successfully.

--- Banking System ---
1. Create Account
2. Deposit
3. Withdraw
4. View All Customers
5. Exit
Enter choice: 4
Name: test name, Acc: 0022, Balance: 5000.0

--- Banking System ---
1. Create Account
2. Deposit
3. Withdraw
4. View All Customers
5. Exit
Enter choice: 3
Enter account number: 0022
Enter amount: 3000
Withdrawal successful.
```