K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

ASSIGNMENT-2

Course Name: Programming for Problem Solving Using Python

Assignment Title: Contact Book

Student Name : SAKSHAM SHARMA

Roll No. : 2501940007

Program : MCA (AI & ML)

Semester : 1st

Session : 2025-26

Faculty Name : Ms. Neha Kaushik

Date of Submission : 18/11/2025

Contact Book Program

This project is a **Python-based Contact Management System** that enables users to store and manage essential contact information such as **Name**, **Phone Number**, and **Email Address**.
The application uses **CSV** and **JSON** files for persistent data storage and supports structured error logging through exception handling to ensure reliability.

Objectives:

1. Learn and implement **file handling** using CSV and JSON in Python.
2. Perform **CRUD operations** (Create, Read, Update, Delete) on contact records.
3. Understand and apply **exception handling** and structured **error logging**.
4. Practice **modular and structured programming** in a console-based application.
5. Build a functional Python project demonstrating real-world problem-solving skills

Contact_manager.py

CODE:

```
# Name- Saksham Sharma

# Date- 15/11/2025

# Project Title: Contact Book
```

#Code Starts

```
import csv
import json
import datetime
import os

CSV_FILE = "contacts.csv"
JSON_FILE = "contacts.json"
ERROR_LOG = "error_log.txt"
FIELDS = ["name", "phone", "email"]

try:
    open(ERROR_LOG, "a", encoding="utf-8").close()
except Exception:
    # if touching the file fails we can't call log_error yet (not defined), so print as fallback
    print("Warning: Could not create or access error_log.txt. Logging may fail at runtime.")

def log_error(operation, error_msg):
    try:
        with open(ERROR_LOG, "a", encoding="utf-8") as log:
            timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            log.write(f"[{timestamp}] Operation: {operation} | Error: {error_msg}\n")
    except Exception:
        # last-resort: print to console if logging fails
        print("Failed to write to error log.")

def welcome():
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("          WELCOME TO CONTACT BOOK          ")
```

```
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("This tool helps you manage contacts using CSV and JSON files.")
    print("You can add, view, search, update, and delete contacts in this program\n")

def add_contact():

    try:
        name = input("Enter Name: ").strip()
        if not name:
            print("Name cannot be empty.")
            return

        phone = input("Enter Phone No. : ").strip()
        email = input("Enter Email Address: ").strip()

        # Check for duplicate (case-insensitive)
        existing = []
        if os.path.isfile(CSV_FILE):
            with open(CSV_FILE, "r", newline="", encoding="utf-8") as f:
                reader = csv.DictReader(f)
                for row in reader:
                    existing.append(row.get("name", "").strip())

        if any(row_name.lower() == name.lower() for row_name in existing):
            print(f"A contact named '{name}' already exists.")
            choice = input("Do you want to add a duplicate? (y/N): ").strip().lower()
            if choice != "y":
                print("Add cancelled.")
                return
```

```
        contact = {"name": name, "phone": phone, "email": email}
        # Append (create file & header if needed)
        with open(CSV_FILE, "a", newline="", encoding="utf-8") as file:
            writer = csv.DictWriter(file, fieldnames=FIELDS)
            if file.tell() == 0:
                writer.writeheader()
            writer.writerow(contact)
        print(f"\nContact: '{name}' added successfully.\n")

    except Exception as e:
        print("Error adding contact. See error log.")
        log_error("Add Contact", str(e))

def show_contact():
    try:
        if not os.path.isfile(CSV_FILE):
            print("\nNo contact file found — add contacts first.\n")
            return

        with open(CSV_FILE, "r", newline="", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            contacts = list(reader)

        if not contacts:
            print("\nNo contacts found — add some contact first.\n")
            return

        print("\n===== CONTACT LIST =====")
```

```
        print(f"{'Name':<20}{'Phone':<15}{'Email'}")
        print("-" * 50)
        for c in contacts:
            print(f"{c.get('name',''):<20}{c.get('phone',''):<15}{c.get('email','')}")
        print("-" * 50)

    except Exception as e:
        print("Error displaying contacts. See error log.")
        log_error("View Contacts", str(e))

def search_contact(name):
    try:
        if not os.path.isfile(CSV_FILE):
            print("No contacts found — add some first.")
            return

        with open(CSV_FILE, "r", newline="", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            matches = [c for c in reader if c.get("name", "").strip().lower() == name.strip().lower()]

        if matches:
            c = matches[0]
            print("\nContact Found:")
            print(f"Name: {c.get('name')}\nPhone: {c.get('phone')}\nEmail: {c.get('email')}\n")
        else:
            print("\nContact not found — try a different name.\n")
    except Exception as e:
        print("Error searching contact. See error log.")
        log_error("Search Contact", str(e))
```

```
def update_contact(name):
    try:
        if not os.path.isfile(CSV_FILE):
            print("No contacts found — add some first.")
            return

        contacts = []
        found = False
        with open(CSV_FILE, "r", newline="", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            for c in reader:
                if c.get("name", "").strip().lower() == name.strip().lower():
                    found = True
                    print("\nEnter new details (leave blank to keep existing):")
                    new_phone = input(f"New Phone ({c.get('phone','')}): ").strip() or c.get("phone",
"")
                    new_email = input(f"New Email ({c.get('email','')}): ").strip() or c.get("email", "")
                    c["phone"], c["email"] = new_phone, new_email
                contacts.append(c)

        if not found:
            print("\nContact not found.\n")
            return

        # Write back updated list
        with open(CSV_FILE, "w", newline="", encoding="utf-8") as file:
            writer = csv.DictWriter(file, fieldnames=FIELDS)
            writer.writeheader()
            writer.writerows(contacts)
```

```
        print(f"\nContact '{name}' updated successfully.\n")

    except Exception as e:
        print("Error updating contact, See error log.")
        log_error("Update Contact", str(e))

def delete_contact(name):
    try:
        if not os.path.isfile(CSV_FILE):
            print("No contacts found — add some first.")
            return

        contacts = []
        found = False
        with open(CSV_FILE, "r", newline="", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            for c in reader:
                if c.get("name", "").strip().lower() == name.strip().lower():
                    found = True
                else:
                    contacts.append(c)

        if not found:
            print("\nContact not found — nothing deleted.\n")
            return

        with open(CSV_FILE, "w", newline="", encoding="utf-8") as file:
            writer = csv.DictWriter(file, fieldnames=FIELDS)
            writer.writeheader()
```

```
            writer.writerows(contacts)
        print(f"\nContact '{name}' deleted successfully.\n")

    except Exception as e:
        print("Error deleting contact. See error log.")
        log_error("Delete Contact", str(e))

def export_to_json():
    try:
        if not os.path.isfile(CSV_FILE):
            print("No contacts to export.")
            return

        with open(CSV_FILE, "r", newline="", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            contacts = list(reader)

        with open(JSON_FILE, "w", encoding="utf-8") as json_file:
            json.dump(contacts, json_file, indent=4)
        print("\n Contacts exported to contacts.json\n")

    except Exception as e:
        print("Error exporting to JSON. See error log.")
        log_error("Export to JSON", str(e))

def import_from_json():
    try:
        if not os.path.isfile(JSON_FILE):
            print("No JSON file found.")
```

```
            return

        with open(JSON_FILE, "r", encoding="utf-8") as file:
            contacts = json.load(file)

        # Basic validation
        valid = [c for c in contacts if isinstance(c, dict) and c.get("name")]
        if not valid:
            print("No valid contacts found in JSON.")
            return

        print(f"Found {len(valid)} contact(s) in JSON.")
        choice = input("Replace CSV with JSON contacts? (y/N): ").strip().lower()
        if choice == "y":
            with open(CSV_FILE, "w", newline="", encoding="utf-8") as file:
                writer = csv.DictWriter(file, fieldnames=FIELDS)
                writer.writeheader()
                writer.writerows(valid)
            print("CSV replaced with JSON contacts.")
        else:
            print("Import cancelled (no changes made).")

    except Exception as e:
        print("Error importing from JSON. See error log.")
        log_error("Import from JSON", str(e))

def main():
    welcome()
    while True:
```

```
print("\nOptions:")
print("1. Add Contact")
print("2. View Contacts")
print("3. Search Contact")
print("4. Update Contact")
print("5. Delete Contact")
print("6. Export to JSON")
print("7. Import from JSON")
print("8. Exit")

choice = input("\nEnter your choice: ").strip()

if choice == "1":
    add_contact()
elif choice == "2":
    show_contact()
elif choice == "3":
    name = input("Enter name to search: ").strip()
    if name:
        search_contact(name)
    else:
        print("Search name cannot be empty.")
elif choice == "4":
    name = input("Enter name to update: ").strip()
    if name:
        update_contact(name)
    else:
        print("OOPS!!, Name cannot be empty.")
elif choice == "5":
```

```
            name = input("Enter name to delete: ").strip()
            if name:
                delete_contact(name)
            else:
                print("OOPS!!, Name cannot be empty.")
        elif choice == "6":
            export_to_json()
        elif choice == "7":
            import_from_json()
        elif choice == "8":
            print("\nExiting Contact Book, Goodbye!")
            break
        else:
            print("Your choice doesn't exist, please try again.")

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

Output:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          WELCOME TO CONTACT BOOK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This tool helps you manage contacts using CSV and JSON files.
You can add, view, search, update, and delete contacts in this program


Options:
1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Export to JSON
7. Import from JSON
8. Exit

Enter your choice: 1
Enter Name: Saksham Sharma
Enter Phone No. : 9910040025
Enter Email Address: sak@gmail.com

Contact: 'Saksham Sharma' added successfully.
```

```
Options:
1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Export to JSON
7. Import from JSON
8. Exit

Enter your choice: 2

===== CONTACT LIST =====
Name                Phone           Email
-------------------------------------------------
Saksham Sharma      9910040025      sak@gmail.com
Uday Garg           8974561230      udg@gmail.com
Piyush chokker      9874561230      piy@gmail.com
Rohan Kwatra        8974561233      roh@gmail.com
-------------------------------------------------
```

```
Enter your choice: 3
Enter name to search: Saksham Sharma

Contact Found:
Name: Saksham Sharma
Phone: 9910040025
Email: sak@gmail.com
```

```
Enter your choice: 5
Enter name to delete: Rohan Kwatra

Contact 'Rohan Kwatra' deleted successfully.
```

```
Options:
1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Export to JSON
7. Import from JSON
8. Exit

Enter your choice: 6

 Contacts exported to contacts.json


Options:
1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Export to JSON
7. Import from JSON
8. Exit

Enter your choice: 7
Found 3 contact(s) in JSON.
Replace CSV with JSON contacts? (y/N): N
Import cancelled (no changes made).
```

```
Options:
1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Export to JSON
7. Import from JSON
8. Exit

Enter your choice: 8

Exiting Contact Book, Goodbye!
```

contacts.csv

name,phone,email

Saksham Sharma,9910040025,sak@gmail.com

Uday Garg,8974561230,udg@gmail.com

Piyush chokker,9874561230,piy@gmail.com

Contacts.json


```
[
  {
    "name": "Saksham Sharma",
    "phone": "9910040025",
    "email": "sak@gmail.com"
  },
  {
    "name": "Uday Garg",
    "phone": "8974561230",
    "email": "udg@gmail.com"
  },
  {
    "name": "Piyush chokker",
    "phone": "9874561230",
    "email": "piy@gmail.com"
  }
]
```


Github Repository Link:

https://github.com/Saksham6453/Python-Assignment-2