**Contact Book**

**VITyarthi Project Report**

***Submitted By***

**Manan Sharma (25BCE10074)**

* **Project Title**

Contact Book using python:

A simple Python based application to store, search, update and manage contacts using JSON file storage.

* **Problem Definition**

In the digital era maintaining personal or professional contacts manually in notebooks or scattered across different apps leads to data loss, inconsistency and difficulty in quickly retrieving information.  
To address this, a simple, lightweight, offline contact management system is needed that allows users to:

* Add contacts
* Store details securely
* Search and update information
* Delete unwanted records
* Load/save data persistently without databases

This project provides a solution using Python and JSON file handling.

* **Requirement Analysis**

**Functional Requirements**

1. The system must allow users to add a new contact.
2. It must display all saved contacts.
3. It must allow searching for a specific contact.
4. Users must be able to update existing contact details.
5. It must allow deleting a contact.
6. The application must load/s persist data in a JSON file.

**Non – Functional Requirements**

1. Easy to use
2. Should store data persistently
3. Should handle incorrect user inputs
4. Should run on any OS with Python installed
5. Should be modular and follow structured programming

**Software Requirements**

1. Python 3.x
2. JSON module (built-in)
3. OS module (built-in)

* **Top – Down Design / Modularization**

**Main Module**

1. Main menu
2. User choice handling

**Sub Modules**

1. load\_contacts()
2. save\_contacts()
3. add\_contacts()
4. view\_contacts()
5. search\_contact()
6. update\_contact()
7. delete\_contact()

* **Algorithms (Pseudocode)**

**Algorithm 1: Load Contacts**

START

IF contacts.json exists THEN

OPEN file in read mode

READ JSON data into dictionary

ELSE

RETURN empty dictionary

END IF

RETURN contacts

END

**Algorithm 2: Add Contact**

START

INPUT name, phone, email

IF name is empty THEN

DISPLAY "Name required"

ELSE

ADD entry to contacts dictionary

END IF

END

**Algorithm 3: Search Contact**

START

INPUT name to search

IF name in contacts THEN

DISPLAY contact details

ELSE

DISPLAY "Not found"

END IF

END

**Algorithm 4: Update Contact**

START

INPUT name to update

IF name exists THEN

ASK for new phone

ASK for new email

UPDATE only if new values provided

ELSE

DISPLAY "Not found"

END IF

END

**Algorithm 5: Delete Contact**

START

INPUT name to delete

IF name exists THEN

REMOVE entry from dictionary

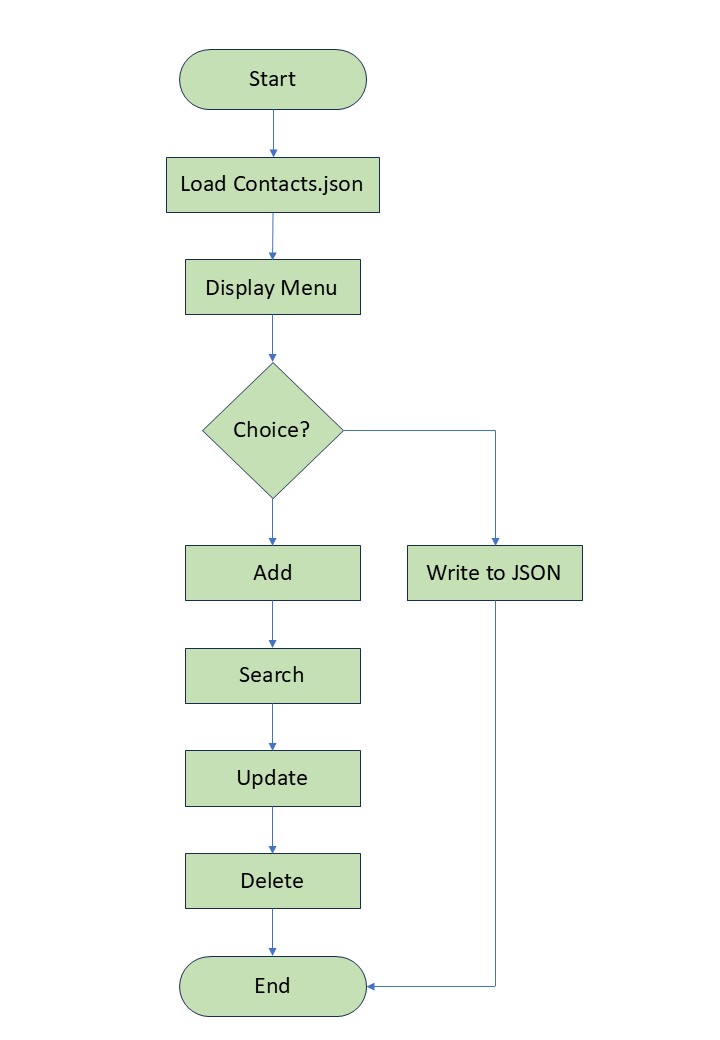
ELSE

DISPLAY "Not found"

END IF

END

* **Flowchart**



* **Screen Recording:**

