**Project Description**

1. **Aim of the Project:**

The aim of this Python Address Book project is to create a digital address book application that allows users to efficiently manage their contacts. The application provides functionalities such as adding new contacts, deleting existing ones, searching for specific contacts, and displaying the entire contact list.

1. **Problem Statement:**

The task is to develop a Python program for managing a digital address book. The program should allow users to add, delete, search for, and display contacts stored in the address book. The objective is to create a user-friendly application that simplifies contact management tasks.

**Features to implement:**

* **Add contact:**

1. Users should be able to add new contacts to the address book by providing the contact's name, address, and phone number.
2. Each contact should be uniquely identified by their name.

* **Delete contact:**

1. Users should be able to delete existing contacts from the address book by specifying the name of the contact to be deleted.
2. If the specified contact does not exist, appropriate feedback should be provided.

* **Search contact:**

1. Users should be able to search for a specific contact by entering the contact's name.
2. If the contact is found, its details (name, address, and phone number) should be displayed.
3. If the contact is not found, appropriate feedback should be provided.

* **Display contact:**

1. Users should be able to view all contacts stored in the address book.
2. If the address book is empty, appropriate feedback should be provided.
3. **Project Description:**

* The project consists of a Python class named AddressBook, which encapsulates the functionalities related to contact management.
* The class includes methods for adding, deleting, searching for, and displaying contacts.
* Contact details are stored in a dictionary within the AddressBook class, where the contact name serves as the key, and a dictionary containing address and phone number serves as the value.
* The application provides feedback messages to users for various actions, ensuring a user-friendly experience.

1. **Functionalities:**

The provided code defines a simple address book functionality in Python. Here are the functionalities of the code:

****Initialization (**\_\_init\_\_**):****

Initializes the address book instance with an empty dictionary to store contacts.

****Add Contact (**add\_contact**):****

* 1. Allows adding a new contact to the address book.
  2. Takes parameters such as name, address, and phone number.
  3. Creates a new entry in the contacts dictionary with the contact's name as the key and their address and phone number as values.
  4. Prints a success message indicating that the contact has been added.

****Delete Contact (**delete\_contact**):****

* 1. Allows deleting an existing contact from the address book.
  2. Takes the name of the contact to be deleted as a parameter.
  3. Checks if the contact exists in the address book. If found, deletes the contact.
  4. Prints a success message if the contact is deleted, or a message indicating that the contact is not found if it does not exist.

****Search Contact (**search\_contact**):****

* 1. Allows searching for a contact by name in the address book.
  2. Takes the name of the contact to be searched as a parameter.
  3. Checks if the contact exists in the address book. If found, prints the contact's name, address, and phone number.
  4. Prints a message indicating that the contact is not found if it does not exist.

****Display Contacts (**display\_contacts**):****

* 1. Displays all contacts stored in the address book.
  2. Iterates through the contacts dictionary and prints the name, address, and phone number of each contact.
  3. If the address book is empty, prints a message indicating that the address book is empty.

****Example Usage:****

* 1. Creates an instance of the AddressBook class.
  2. Adds several contacts using the add\_contact method.
  3. Displays all contacts using the display\_contacts method.
  4. Searches for a specific contact using the search\_contact method.
  5. Deletes a contact using the delete\_contact method (in this case, attempting to delete a non-existent contact).

5**.Code Implementation:**

This code defines a class AddressBook with methods to add, delete, search, and display contacts. It also includes example usage where contacts are added to the address book, displayed, searched for, and deleted.

class AddressBook:

def \_\_init\_\_(self):

self.contacts = {}

def add\_contact(self, name, address, phone):

self.contacts[name] = {"address": address, "phone": phone}

print(f"Contact '{name}' added successfully.")

def delete\_contact(self, name):

if name in self.contacts:

del self.contacts[name]

print(f"Contact '{name}' deleted successfully.")

else:

print("Contact not found.")

def search\_contact(self, name):

print("Searching...........")

if name in self.contacts:

print(f"Name: {name}\nAddress: {self.contacts[name]['address']}\nPhone: {self.contacts[name]['phone']}")

else:

print("Contact not found.")

def display\_contacts(self):

if self.contacts:

print("\nContacts:")

for name, info in self.contacts.items():

print(f"Name: {name}, Address: {info['address']}, Phone: {info['phone']}")

else:

print("Address book is empty.")

# Example usage:

address\_book = AddressBook()

address\_book.add\_contact("PRASANTH", "267, PP SALAI, NAMAKKAL", "9874563210")

address\_book.add\_contact("KARUPPASAMY", "456 Elm St, RAJAPALAYAM", "6369060554",)

address\_book.add\_contact("KAVIN","420, PERIYAR NAGAR, KOVILPALAYAM", "9420420420")

address\_book.add\_contact("SABARI","69, KAMARAJAR NAGAR,SANTHOSAPALAYAM", "9421421421")

address\_book.display\_contacts()

print("\n")

address\_book.search\_contact("KARUPPASAMY")

print("\n")

address\_book.delete\_contact("PRASANTH")

print("\n")

**Output:**

Contact 'PRASANTH' added successfully.

Contact 'KARUPPASAMY' added successfully.

Contact 'KAVIN' added successfully.

Contact 'SABARI' added successfully.

Contacts:

Name: PRASANTH, Address: 267, PP SALAI, NAMAKKAL, Phone: 9874563210

Name: KARUPPASAMY, Address: 456 Elm St, RAJAPALAYAM, Phone: 6369060554

Name: KAVIN, Address: 420, PERIYAR NAGAR, KOVILPALAYAM, Phone: 9420420420

Name: SABARI, Address: 69, KAMARAJAR NAGAR,SANTHOSAPALAYAM, Phone: 9421421421

Searching...........

Name: KARUPPASAMY

Address: 456 Elm St, RAJAPALAYAM

Phone: 6369060554

Contact 'PRASANTH' deleted successfully.

6.**Results and Outcomes:**

* Four contacts were added successfully: "PRASANTH", "KARUPPASAMY", "KAVIN", and "SABARI".
* The display\_contacts method was called, printing all the contacts.
* The search\_contact method was called to find the details of "KARUPPASAMY", which were displayed.
* Then, the delete\_contact method was called to delete the contact "PRASANTH", which was successful.

7.**Conclusion:**

The conclusion of the provided code execution is as follows:

1. Four contacts are added to the address book: "PRASANTH", "KARUPPASAMY", "KAVIN", and "SABARI".
2. The display\_contacts method is called, showing all contacts with their details.
3. The search\_contact method is called to find the details of the contact named "KARUPPASAMY", which is successfully found and displayed.
4. The delete\_contact method is called to remove the contact named "PRASANTH" from the address book, which is successful.
5. The code execution concludes after these operations.