Personal Contact Management System (CMS)

A Contact Management System (CMS) allows users to add, search, update, delete, and list contacts efficiently using Python's core data structures like lists, dictionaries, and sets.

Features

Add a New Contact

View All Contacts

Search a Contact by Name or Phone

Update Contact Details

Delete a Contact

Prevent Duplicate Contacts

✅ Key Concepts Used:

Classes and Objects

Lists for dynamic contact storage

Dictionaries to structure individual contact data

Sets to prevent duplicate phone numbers

String manipulation and input validation

Menu-driven program for user interaction

Double-click (or enter) to edit

```
class Contact:
    def __init__(self, name, phone, email):
        self.name = name
        self.phone = phone
        self.email = email

    def __str__(self):
        return f'Name: {self.name}, Phone: {self.phone}, Email: {self.email}'


class ContactManagementSystem:
    def __init__(self):
        self.contacts = []  # List of Contact objects
        self.phone_index = set()  # To prevent duplicate phone numbers

    def add_contact(self, name, phone, email):
        if phone in self.phone_index:
            print("Contact with this phone number already exists!")
            return
        new_contact = Contact(name, phone, email)
        self.contacts.append(new_contact)
        self.phone_index.add(phone)
        print("Contact added successfully!")

    def view_contacts(self):
        if not self.contacts:
            print("No contacts available.")
            return
        print("\nAll Contacts:")
        for idx, contact in enumerate(self.contacts, 1):
            print(f"{idx}. {contact}")

    def search_contact(self, search_term):
        found = False
        for contact in se
            if search_term.lower() in contact.name.lower() or search_term in contact.phone:
                print(contact)
```

```
                found = True
        if not found:
            print("No matching contact found.")

    def update_contact(self, phone, new_name=None, new_phone=None, new_email=None):
        for contact in self.contacts:
            if contact.phone == phone:
                if new_phone and new_phone != phone and new_phone in self.phone_index:
                    print("New phone number already exists!")
                    return
                # Remove old phone from index if changing
                if new_phone and new_phone != phone:
                    self.phone_index.remove(phone)
                    self.phone_index.add(new_phone)
                    contact.phone = new_phone
                if new_name:
                    contact.name = new_name
                if new_email:
                    contact.email = new_email
                print("Contact updated successfully!")
                return
        print("Contact not found.")

    def delete_contact(self, phone):
        for contact in self.contacts:
            if contact.phone == phone:
                self.contacts.remove(contact)
                self.phone_index.remove(phone)
                print("Contact deleted successfully!")
                return
        print("Contact not found.")


def main():
    cms = ContactManagementSystem()

    while True:
        print("\nContact Management System")
        print("1. Add Contact")
        print("2. View All Contacts")
        print("3. Search Contact")
        print("4. Update Contact")
        print("5. Delete Contact")
        print("6. Exit")

        choice = input("Enter your choice (1-6): ")

        if choice == '1':
            name = input("Enter Name: ")
            phone = input("Enter Phone: ")
            email = input("Enter Email: ")
            cms.add_contact(name, phone, email)

        elif choice == '2':
            cms.view_contacts()

        elif choice == '3':
            search_term = input("Enter Name or Phone to Search: ")
            cms.search_contact(search_term)

        elif choice == '4':
            phone = input("Enter Phone of the Contact to Update: ")
            print("Leave fields empty if no change required.")
            new_name = input("Enter New Name: ")
            new_phone = input("Enter New Phone: ")
            new_email = input("Enter New Email: ")
            cms.update_contact(phone, new_name or None, new_phone or None, new_email or None)

        elif choice == '5':
            phone = input("Enter Phone of the Contact to Delete: ")
            cms.delete_contact(phone)

        elif choice == '6':
            print("Exiting Contact Management System.")
            break
```

```
            else:
                print("Invalid choice! Please try again.")


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

```
Contact Management System
1. Add Contact
2. View All Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Exit
Enter your choice (1-6): 1
Enter Name: RJ
Enter Phone: 9993671809
Enter Email: rahuljaincse51@gmail.com
Contact added successfully!

Contact Management System
1. Add Contact
2. View All Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Exit
Enter your choice (1-6): 2

All Contacts:
1. Name: RJ, Phone: 9993671809, Email: rahuljaincse51@gmail.com

Contact Management System
1. Add Contact
2. View All Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Exit
Enter your choice (1-6): 6
Exiting Contact Management System.
```