

Project 2

Contact Management System

Problem Statement:

This project is simple Contact Management System in Python that includes functionalities allow users to add new contacts by inputting details such as name, phone number, email address. This system will utilize lists, dictionaries, functions, file handling, and exception handling.

Requirements:

1. Programming Language: Python 3.x
2. Operating System: Windows, macOS, or Linux
3. Hardware Requirements: A computer with a minimum of 2GB RAM and 10GB free disk space

Proposed solution

The goal is to develop a Contact Management System (CMS) that allows users to efficiently manage their contacts. This system should enable the addition, updating, viewing, and deletion of contact information. Each contact will include essential details such as name, phone number, email address, and address.

Features:

1. Function to load contacts from a file
2. Function to save contacts to a file
3. Function to add a contact
4. Function to view all contacts
5. Function to update a contact
6. Function to delete a contact

Future scope:

1. GUI Improvements: Improve the graphical user interface to make it more intuitive and user-friendly.
2. Mobile App: Develop a mobile app version of the contact management system.

Name : Adhikesh R S

Conclusion:

Made a project which is a simple Contact Management System in Python that includes functionalities allow users to add new contacts by inputting details such as name, phone number, email address. This system will utilize lists, dictionaries, functions, file handling, and exception handling.

Code:

Task 1: Function to load contacts from a file

```
import json
import os
# File to store the contact data
CONTACTS_FILE = 'contacts_data.json'
# Function to load contacts from a file
def load_contacts():
    if not os.path.exists (CONTACTS_FILE):
        return {}
    try:
        with open (CONTACTS_FILE, 'r') as file:
            return json. load(file)
    except Exception as e:
        print(f"Error loading contacts data: {e}")
        return {}
```

Task 2: Function to save contacts to a file

```
# Function to save contacts to a file
def save_contacts(contacts) :
    try:
        with open(CONTACTS_FILE, 'w') as file:
            json.dump(contacts, file)
    except Exception as e:
        print(f"Error saving contacts data: {e}")
```

Name : Adhikesh R S

Task 3: Function to add a contact

```
# Function to add a contact
def add_contact(contacts, name, phone, email):
    if name in contacts:
        print("Contact already exists. Use update option to modify.")
    else:
        contacts [name] = {'phone': phone, 'email': email}
        save_contacts (contacts)
        print(f"Contact '{name}' added successfully!")
```

Task 4: Function to view all contacts

```
# Function to view all contacts
def view_contacts(contacts) :
    if not contacts:
        print("No contacts found.")
        return
    for name, details in contacts.items():
        print (f"Name: {name}, Phone: {details[ 'phone']}, Email: {details['email']}")
```

Task 5: Function to update a contact

```
# Function to update a contact
def update_contact(contacts, name, phone=None, email=None) :
    if name not in contacts:
        print (f"Contact '{name}' not found.")
        return
    if phone:
        contacts[name]['phone'] = phone
    if email:
        contacts[name]['email'] = email
    save_contacts (contacts)
    print (f"Contact '{name}' updated successfully!")
```

Task 6: Function to delete a contact

```
# Function to delete a contact
def delete_contact(contacts, name):
    if name not in contacts:
        print(f"Contact '{name}' not found.")
        return
    del contacts [name]
    save_contacts (contacts)
    print(f"Contact '{name}' deleted successfully!")
```

Name : Adhikesh R S

Task 7: Main function to run the Contact Management System

```
#Main function to run the Contact Management System
def main():
    contacts = load_contacts()
    while True:
        print("\nContact Management System")
        print("1. Add Contact")
        print("2. View Contacts")
        print("3. Update Contact")
        print("4. Dolete Contact")
        print("5. Exit")
        choice = input("Enter your choice: ")
        if choice == '1':
            name = input("Enter the contact name: ")
            phone = input("Enter the contact phone number: ")
            email = input("Enter the contact email: ")
            add_contact (contacts, name, phone, email)
        elif choice == '2':
            view_contacts(contacts)
        elif choice == '3':
            name = input("Enter the contact name to update: ")
            phone = input("Enter the new phone number (leave blank to keep current): ")
            email = input("Enter the new email (leave blank to keep current): ")
            update_contact(contacts, name, phone if phone else None, email if email else None)
        elif choice == '4':
            name = input("Enter the contact name to delete: ")
            delete_contact (contacts, name)
        elif choice == '5':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")

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

Output:

```
Contact Management System
1. Add Contact
2. View Contacts
3. Update Contact
4. Dolete Contact
5. Exit
Enter your choice: 1
Enter the contact name: adhikesh r s
Enter the contact phone number: 9947782802
Enter the contact email: adhikeshshambhu@gmail.com
Contact 'adhikesh r s' added successfully!

Contact Management System
1. Add Contact
2. View Contacts
3. Update Contact
4. Dolete Contact
5. Exit
Enter your choice: 2
Name: adhikesh r s, Phone: 9947782802, Email: adhikeshshambhu@gmail.com
```

Name : Adhikesh R S

Contact Management System

1. Add Contact
2. View Contacts
3. Update Contact
4. Delete Contact
5. Exit

Enter your choice: 3

Enter the contact name to update: adhikesh r s

Enter the new phone number (leave blank to keep current):

Enter the new email (leave blank to keep current):

Contact 'adhikesh r s' updated successfully!

Contact Management System

1. Add Contact
2. View Contacts
3. Update Contact
4. Delete Contact
5. Exit

Enter your choice: 4

Enter the contact name to delete: adhikesh r s

Contact 'adhikesh r s' deleted successfully!

Contact Management System

1. Add Contact
2. View Contacts
3. Update Contact
4. Delete Contact
5. Exit

Enter your choice: 5

Exiting the system. Goodbye!