

## Md Neel

# Project: Contact Management System

### Contact Management System Program Execution Order Step by Step

#### 1. **main.py** (Entry Point)

- The execution starts here.
  - This script initializes the program, loads contacts from the file (save\_file.py), and displays the menu.
  - Based on the user's choice, it calls the appropriate module to perform the required action.
- 

#### 2. **save\_file.py** (Load Contacts)

- As the first step, main.py calls the load\_contacts() function from save\_file.py.
  - This function reads the contacts.csv file (or contacts.txt) to load saved contacts into the dictionary structure in memory.
  - If the file is empty or invalid, it handles the issue gracefully (e.g., by skipping invalid lines or initializing an empty dictionary).
- 

#### 3. **User's Menu Selection in main.py**

- After loading contacts, the program displays the menu to the user.
  - The user selects one of the following options:
    1. **Add Contact:** Calls add\_contact() from add\_contact.py.
    2. **View Contacts:** Calls view\_contacts() from view\_contacts.py.
    3. **Delete Contact:** Calls delete\_contact() from delete\_contact.py.
    4. **Exit:** Terminates the program.
- 

#### 4. **add\_contact.py** (If User Selects "Add Contact")

- The add\_contact() function is executed when the user chooses to add a new contact.

- This function:
    - Validates user input for the name, phone, email, and address.
    - Ensures no duplicate names or phone numbers exist.
    - Updates the contacts dictionary with the new contact.
    - Calls the `save_contacts()` function from `save_file.py` to save the updated contacts to the file.
- 

#### 5. **view\_contacts.py** (If User Selects "View Contacts")

- The `view_contacts()` function is executed when the user wants to see all saved contacts.
  - It reads the contacts dictionary and formats the data into a clean table.
  - The formatted data is displayed in the terminal.
- 

#### 6. **delete\_contact.py** (If User Selects "Delete Contact")

- The `delete_contact()` function is executed when the user wants to remove a contact.
  - This function:
    - Prompts the user to enter the name of the contact to delete.
    - Searches the contacts dictionary for the matching name.
    - If found, deletes the contact and updates the dictionary.
    - Calls `save_contacts()` from `save_file.py` to save the updated data back to the file.
- 

#### 7. **save\_file.py** (Save Contacts)

- Whenever a modification is made to the contacts (e.g., adding or deleting), the `save_contacts()` function from `save_file.py` is called.
- This function writes the current state of the contacts dictionary back to the file (`contacts.csv`).