**NALLA MALLA REDDY**

**ENGINEERING COLLEGE**

**DOCUMENT**

**ON**

**Simple contact**

**management system in Java**

**Name:M.HARISH**

**Roll No:22B61A05B4**

**Section:CSE-3B**

**Project Title: Simple contact management system in Java**

**1. Create the ContactManager.java file**

Create a Java file named ContactManager.java to handle adding, editing, and deleting contacts.

**Here's an example implementation of ContactManager.java:**

import java.util.ArrayList;

import java.util.Scanner;

class Contact {

String name;

String phoneNumber;

String email;

public Contact(String name, String phoneNumber, String email) {

this.name = name;

this.phoneNumber = phoneNumber;

this.email = email;

}

@Override

public String toString() {

return "Name: " + name + ", Phone: " + phoneNumber + ", Email: " + email;

}

}

public class ContactManager {

private ArrayList<Contact> contacts;

private Scanner scanner;

public ContactManager() {

contacts = new ArrayList<>();

scanner = new Scanner(System.in);

}

public void addContact() {

System.out.println("Enter contact name:");

String name = scanner.nextLine();

System.out.println("Enter contact phone number:");

String phone = scanner.nextLine();

System.out.println("Enter contact email:");

String email = scanner.nextLine();

Contact newContact = new Contact(name, phone, email);

contacts.add(newContact);

System.out.println("Contact added successfully!");

}

public void editContact() {

System.out.println("Enter the name of the contact to edit:");

String name = scanner.nextLine();

for (Contact contact : contacts) {

if (contact.name.equals(name)) {

System.out.println("Enter new phone number:");

contact.phoneNumber = scanner.nextLine();

System.out.println("Enter new email:");

contact.email = scanner.nextLine();

System.out.println("Contact updated successfully!");

return;

}

}

System.out.println("Contact not found!");

}

public void deleteContact() {

System.out.println("Enter the name of the contact to delete:");

String name = scanner.nextLine();

for (int i = 0; i < contacts.size(); i++) {

if (contacts.get(i).name.equals(name)) {

contacts.remove(i);

System.out.println("Contact deleted successfully!");

return;

}

}

System.out.println("Contact not found!");

}

public void displayContacts() {

if (contacts.isEmpty()) {

System.out.println("No contacts available.");

return;

}

System.out.println("Contacts List:");

for (Contact contact : contacts) {

System.out.println(contact);

}

}

public static void main(String[] args) {

ContactManager manager = new ContactManager();

while (true) {

System.out.println("\n1. Add Contact");

System.out.println("2. Edit Contact");

System.out.println("3. Delete Contact");

System.out.println("4. Display Contacts");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

Scanner scanner = new Scanner(System.in);

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

manager.addContact();

break;

case 2:

manager.editContact();

break;

case 3:

manager.deleteContact();

break;

case 4:

manager.displayContacts();

break;

case 5:

System.out.println("Exiting...");

return;

default:

System.out.println("Invalid option. Try again.");

}

}

}

}

**2. Add the file to Git**

To track your project files with Git, first initialize the repository (if not already done):

*git init*

Now add ContactManager.java to Git:

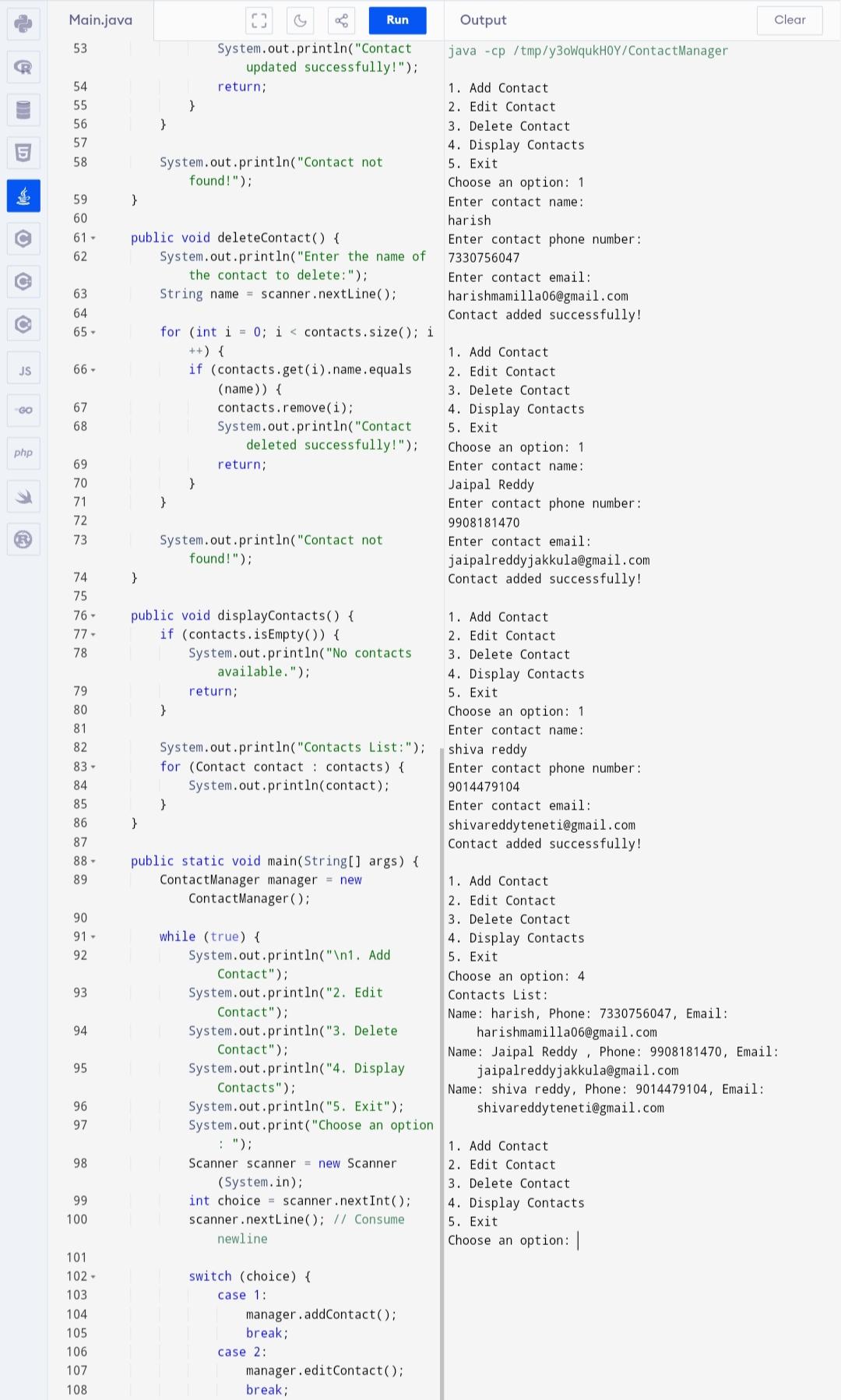
git add ContactManager.java

**3. Commit the changes**

After staging the file, you can commit it:

*git commit -m "Create contact management system"*

**OUTPUT:**

****