SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

AN AUTONOMOUS INSTITUTION

APPROVED BY AICTE, AFFILIATED TO JNTUH

ACCREDITED BY NAAC-A GRADE, NBA (CSE, ECE, ME) & ISO 9001:2015 CERTIFIED

PROJECT NAME : PHONE BOOK

BATCH N0 : 12

S.NO	STUDENT NAME	-	H .T.NO
1.	Nama Rahul	-	22VE1A0439
2.	O Manivardhan	-	22VE1A0443
3.	Posani Divya Tej	-	22VE1A0448
4.	Rachkonda Yashasvi	-	22VE1A0449
5.	Sunchu Jyoshna	-	22VE1A0458

ABSTRACT :

- The PhoneBook is a software application that allows users to store and manage contact information, such as Names, Phone Numbers, email and Addresses.
- It can be used to Add,View all, Modify, Search, and Delete contacts in an organized manner.
- The program uses arrays, structures, and file handling to store and retrieve data.
- The user-friendly interface makes it easy for people of all ages to use and maintain their own personal phonebook.

MODULES :

- There are 5 Functions performed in this program :
    1. Add Contact
    2. List Contacts
    3. Edit Contact
    4. Search Contact
    5. Delete Contact

1. Add Contact

```
void add_contact(FILE *fp)
{
 struct Contact c;

 printf("Enter Name: ");
 scanf("%s", c.name);
 int len;
 do
  {
   printf("Enter Phone Number (10 digits): ");
   scanf("%s", c.phone);
   len = strlen(c.phone);
   if (len != 10)
    {
     printf("Invalid Entry. Phone number must be 10 digits.\n");
    } else
    {
     int i;
     for (i = 0; i < len; i++)
      {
       if (!isdigit(c.phone[i]))
        {
         printf("Invalid Entry. Phone number must only contain digits.\n");
         break;
        }
      }
     if (i == len) break;
    }
  } while (1);
 printf("Enter email: ");
 scanf("%s", c.email);
 printf("Enter Address: ");
 scanf("%s", c.address);

 fwrite(&c, sizeof(struct Contact), 1, fp);
}
```

2. List Contacts

```
void list_contacts(FILE *fp)
{
 struct Contact c;

 rewind(fp);
 while (fread(&c, sizeof(struct Contact), 1, fp) == 1)
  {
    printf("Name: %s\n", c.name);
    printf("Phone: %s\n", c.phone);
    printf("Email: %s\n", c.email);
    printf("Address: %s\n", c.address);
    printf("\n");
  }
}
```

3. Edit Contact

```
void edit_contact(FILE *fp)
{
 struct Contact c, new_c;
 char name[MAX_LEN];
 int found = 0;
 int cont;
 int len;

 printf("Enter Name of the Contact to Edit: ");
 scanf("%s", name);
 rewind(fp);
 while (fread(&c, sizeof(struct Contact), 1, fp) == 1)
 {
  if (strcmp(c.name, name) == 0)
  {
   found = 1;
   break;
  }
 }
 if (!found)
  {
   printf("Contact not found\n");
   return;
  }
 edit_menu();
 scanf("%d", &cont);
 switch (cont)
  {
   case 1:
    printf("Enter new Name: ");
    scanf("%s", new_c.name);
    strcpy(c.name, new_c.name);
    break;
   case 2:
    do
    {
     printf("Enter New Phone Number (10 digits): ");
     scanf("%s", new_c.phone);
     len = strlen(new_c.phone);
     if (len != 10)
     {
      printf("Invalid Entry. Phone number must be 10 digits.\n");
     } else
     {
      int i;
      for (i = 0; i < len; i++)
      {
       if (!isdigit(new_c.phone[i]))
        {
         printf("Invalid Entry. Phone number must only contain digits.\n");
         break;
        }
      }
      if (i == len) break;
     }
    } while (1);
    strcpy(c.phone, new_c.phone);
    break;
   case 3:
    printf("Enter new Address: ");
    scanf("%s", new_c.address);
    strcpy(c.address, new_c.address);
    break;
   default:
    printf("Invalid option");
    break;
  }
 fseek(fp, -1 * sizeof(struct Contact), SEEK_CUR);
 fwrite(&c, sizeof(struct Contact), 1, fp);
}
```

4. Search Contact

```
void search_contact(FILE *fp)
{
 struct Contact c;
 char name[MAX_LEN];
 int found = 0;

 printf("Enter name of the Contact to search: ");
 scanf("%s", name);
 rewind(fp);
 while (fread(&c, sizeof(struct Contact), 1, fp) == 1)
  {
   if (strcmp(c.name, name) == 0)
    {
     found = 1;
     break;
    }
  }
 if (!found)
  {
   printf("Contact not found\n");
   return;
  }
 printf("Name: %s\n", c.name);
 printf("Phone: %s\n", c.phone);
 printf("Email: %s\n", c.email);
 printf("Address: %s\n", c.address);
}
```

5. Delete Contact

```
void delete_contact(FILE *fp)
{
 struct Contact c;
 char name[MAX_LEN];
 int found = 0;

 printf("Enter Name of the contact to Delete: ");
 scanf("%s", name);
 FILE *temp_fp = fopen("temp.dat", "wb");
 rewind(fp);
 while (fread(&c, sizeof(struct Contact), 1, fp) == 1)
  {
   if (strcmp(c.name, name) != 0)
    {
     fwrite(&c, sizeof(struct Contact), 1, temp_fp);
    }
   else
    {
     found = 1;
    }
  }
 if (!found)
  {
   printf("Contact not found\n");
   fclose(temp_fp);
   return;
  }
 fclose(fp);
 fclose(temp_fp);
 remove(FILE_NAME);
 rename("temp.dat", FILE_NAME);
 fp = fopen(FILE_NAME, "r+b");
 printf("Contact Deleted Sucessfully\n");
}
```

Outputs :

```
***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 1
Enter Name: vikram
Enter Phone Number (10 digits): 7893688230
Enter email: vicky@gmail.com
Enter Address: Nagole

***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 2
Name: vikram
Phone: 7893688230
Email: vicky@gmail.com
Address: Nagole
```

```
***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 3
Enter Name of the Contact to Edit: vikram
Choose an Option to Edit
1.Name
2.Phone Numb
3.Address
3
Enter new Address: Uppal

***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 4
Enter name of the Contact to search: vikram
Name: vikram
Phone: 7893688230
Email: vicky@gmail.com
Address: Uppal
```

```
***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 5
Enter Name of the contact to Delete: vikram
Contact Deleted Sucessfully

***** Main Menu *****
1. Add Contact
2. List Contacts
3. Edit Contact
4. Search Contact
5. Delete Contact
6. Exit
**********************

Enter your Choice : 6
Exiting Phonebook App
```

FUTURE WORK :

A phonebook program can be used in a variety of settings where contact management is needed. Here are some common uses:

- Personal Use: An individual can use a phonebook program to store and manage their personal contacts, such as family, friends, and colleagues.
- Small Businesses: Small businesses can use a phonebook program to store and manage their customer and vendor information, making it easier to maintain a record of all their contacts.
- Non-Profit Organizations: Non-profit organizations can use a phonebook program to manage their donor, volunteer, and staff information, helping to keep track of all their key contacts.
- Educational Institutions: Educational institutions can use a phonebook program to manage their student, faculty, and staff information, making it easier to keep track of their community.
- Government Agencies: Government agencies can use a phonebook program to manage their employee and citizen information, helping to keep track of all the individuals they interact with.


thank
you