

C Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_CONTACTS 100
6  #define MAX_NAME_LEN 50
7  #define MAX_PHONE_LEN 15
8  #define MAX_EMAIL_LEN 50
9  #define FILENAME "contacts.txt"
10
11 typedef struct
12 {
13     char name[MAX_NAME_LEN];
14     char phone[MAX_PHONE_LEN];
15     char email[MAX_EMAIL_LEN];
16 } Contact;
17
18 Contact contacts[MAX_CONTACTS];
19 int contactCount = 0;
20
21 // Function declarations
22 void loadContacts();
23 void saveContacts();
24 void addContact();
25 void displayContacts();
26 void searchContact();
27 void updateContact();
28 void menu();
29 void clearInputBuffer();
30
31 int main()
32 {
33     loadContacts();
34     menu();
35     saveContacts();
36     return 0;
37 }
38
39 void clearInputBuffer()
40 {
41     while (getchar() != '\n');
42 }
43
44 void loadContacts()
45 {
46     FILE *file = fopen(FILENAME, "r");
47     if (file == NULL) return;
48
49     contactCount = 0;
50     while (fscanf(file, "%[^,],%[^,],%[^\n]\n",
51                  contacts[contactCount].name,
52                  contacts[contactCount].phone,
```

```
53         contacts[contactCount].email) == 3)
54     {
55         contactCount++;
56         if (contactCount >= MAX_CONTACTS) break;
57     }
58     fclose(file);
59 }
60
61 void saveContacts()
62 {
63     FILE *file = fopen(FILENAME, "w");
64     if (file == NULL)
65     {
66         printf("Error saving contacts.\n");
67         return;
68     }
69
70     for (int i = 0; i < contactCount; i++)
71     {
72         fprintf(file, "%s,%s,%s\n",
73                 contacts[i].name,
74                 contacts[i].phone,
75                 contacts[i].email);
76     }
77     fclose(file);
78 }
79
80 void addContact()
81 {
82     if (contactCount >= MAX_CONTACTS)
83     {
84         printf("Phonebook is full.\n");
85         return;
86     }
87
88     Contact newContact;
89     printf("Enter name: ");
90     fgets(newContact.name, MAX_NAME_LEN, stdin);
91     newContact.name[strcspn(newContact.name, "\n")] = 0;
92
93     printf("Enter phone: ");
94     fgets(newContact.phone, MAX_PHONE_LEN, stdin);
95     newContact.phone[strcspn(newContact.phone, "\n")] = 0;
96
97     printf("Enter email: ");
98     fgets(newContact.email, MAX_EMAIL_LEN, stdin);
99     newContact.email[strcspn(newContact.email, "\n")] = 0;
100
101     contacts[contactCount++] = newContact;
102     printf("Contact added successfully!\n");
103 }
104
105 void displayContacts()
106 {
107     if (contactCount == 0)
```

```
108     {
109         printf("No contacts available.\n");
110         return;
111     }
112
113     for (int i = 0; i < contactCount; i++)
114     {
115         printf("\nContact #%d\n", i + 1);
116         printf("Name : %s\n", contacts[i].name);
117         printf("Phone: %s\n", contacts[i].phone);
118         printf("Email: %s\n", contacts[i].email);
119     }
120 }
121
122 void searchContact()
123 {
124     char query[50];
125     printf("Enter name or phone to search: ");
126     fgets(query, 50, stdin);
127     query[strcspn(query, "\n")] = 0;
128
129     int found = 0;
130     for (int i = 0; i < contactCount; i++)
131     {
132         if (strstr(contacts[i].name, query) || strstr(contacts[i].phone, query))
133         {
134             printf("\nMatch found:\n");
135             printf("Name : %s\n", contacts[i].name);
136             printf("Phone: %s\n", contacts[i].phone);
137             printf("Email: %s\n", contacts[i].email);
138             found = 1;
139         }
140     }
141
142     if (!found)
143     {
144         printf("No contact found with the given query.\n");
145     }
146 }
147
148 void updateContact()
149 {
150     char query[50];
151     printf("Enter name of contact to update: ");
152     fgets(query, 50, stdin);
153     query[strcspn(query, "\n")] = 0;
154
155     for (int i = 0; i < contactCount; i++)
156     {
157         if (strcmp(contacts[i].name, query) == 0)
158         {
159             printf("Enter new name: ");
160             fgets(contacts[i].name, MAX_NAME_LEN, stdin);
161             contacts[i].name[strcspn(contacts[i].name, "\n")] = 0;
162         }
163     }
164 }
```

```
163         printf("Enter new phone: ");
164         fgets(contacts[i].phone, MAX_PHONE_LEN, stdin);
165         contacts[i].phone[strcspn(contacts[i].phone, "\n")] = 0;
166
167         printf("Enter new email: ");
168         fgets(contacts[i].email, MAX_EMAIL_LEN, stdin);
169         contacts[i].email[strcspn(contacts[i].email, "\n")] = 0;
170
171         printf("Contact updated successfully.\n");
172         return;
173     }
174 }
175
176     printf("Contact not found.\n");
177 }
178 void menu()
179 {
180     int choice;
181     do
182     {
183         printf("\n--- Mobile Phonebook Menu ---\n");
184         printf("1. Add Contact\n");
185         printf("2. Display Contacts\n");
186         printf("3. Search Contact\n");
187         printf("4. Update Contact\n");
188         printf("5. Exit\n");
189         printf("Enter your choice: ");
190         scanf("%d", &choice);
191         clearInputBuffer();
192
193         switch (choice)
194         {
195             case 1:
196                 addContact();
197                 break;
198             case 2:
199                 displayContacts();
200                 break;
201             case 3:
202                 searchContact();
203                 break;
204             case 4:
205                 updateContact();
206                 break;
207             case 5:
208                 printf("Exiting and saving contacts...\n");
209                 break;
210             default:
211                 printf("Invalid choice. Please try again.\n");
212         }
213     }
214     while (choice != 5);
215 }
216
```