```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

struct Account {

char name[50];
char email[50];
char address[100];
char phone[15];
int acc_no;
int pin;
float balance;
};

int isValidName(char name[]) {
for (int i = 0; name[i] != '\0'; i++) {
if (!isalpha(name[i]) && name[i] != ' ')
return 0;
}
return 1;
}

int isValidPhone(char ph[]) {
if (strlen(ph) != 10) return 0;
for (int i = 0; i < 10; i++) {
if (!isdigit(ph[i])) return 0;
}
return 1;
}

void registerAccount() {
struct Account a;
FILE *fp = fopen("bankdata.dat", "ab");

printf("\n===== REGISTER =====\n");

while (1) {
printf("Enter Name       : ");
fgets(a.name, sizeof(a.name), stdin);
a.name[strcspn(a.name, "\n")] = 0;

if (isValidName(a.name)) {
printf("Valid Name\n");
break;
} else {
printf("Invalid Name! Only alphabets allowed.\n");
}
}

while (1) {
printf("Enter Email      : ");
```

```
45. fgets(a.email, sizeof(a.email), stdin);
46. a.email[strcspn(a.email, "\n")] = 0;

47. if (strstr(a.email, "@gmail.com")) {
48. printf("Valid Email\n");
49. break;
50. } else {
51. printf("Invalid email.\n");
52. }
53. }

54. printf("Enter Address     : ");
55. fgets(a.address, sizeof(a.address), stdin);
56. a.address[strcspn(a.address, "\n")] = 0;

57. while (1) {
58. printf("Enter Phone      : ");
59. fgets(a.phone, sizeof(a.phone), stdin);
60. a.phone[strcspn(a.phone, "\n")] = 0;

61. if (isValidPhone(a.phone)) {
62. printf("Valid Phone Number\n");
63. break;
64. } else {
65. printf("Invalid number.\n");
66. }
67. }

68. printf("Create Account No : ");
69. scanf("%d", &a.acc_no);
70. getchar();

71. printf("Create PIN       : ");
72. scanf("%d", &a.pin);
73. getchar();

74. a.balance = 0;

75. fwrite(&a, sizeof(a), 1, fp);
76. fclose(fp);

77. printf("\nRegistration Successful!\n");
78. }

79. int login(struct Account *a) {
80. int acc, pin;
81. FILE *fp = fopen("bankdata.dat", "rb");

82. if (!fp) {
83. printf("No accounts found!\n");
84. return 0;
85. }
```

```
86. printf("\n===== LOGIN =====\n");
87. printf("Account Number : ");
88. scanf("%d", &acc);
89. printf("PIN          : ");
90. scanf("%d", &pin);

91. while (fread(a, sizeof(*a), 1, fp) == 1) {
92. if (a->acc_no == acc && a->pin == pin) {
93. fclose(fp);
94. return 1;
95. }
96. }

97. fclose(fp);
98. return 0;
99. }

100.  void updateAccount(struct Account a) {
101.  FILE *fp = fopen("bankdata.dat", "rb");
102.  FILE *temp = fopen("temp.dat", "wb");
103.  struct Account t;

104.  while (fread(&t, sizeof(t), 1, fp) == 1) {
105.  if (t.acc_no == a.acc_no)
106.  fwrite(&a, sizeof(a), 1, temp);
107.  else
108.  fwrite(&t, sizeof(t), 1, temp);
109.  }

110.  fclose(fp);
111.  fclose(temp);
112.  remove("bankdata.dat");
113.  rename("temp.dat", "bankdata.dat");
114.  }

115.  void menu(struct Account a) {
116.  int choice;
117.  float amt;

118.  do {
119.  printf("\n===== MENU =====\n");
120.  printf("1. Check Balance\n");
121.  printf("2. Deposit\n");
122.  printf("3. Withdraw\n");
123.  printf("4. Transfer\n");
124.  printf("5. History (Simple)\n");
125.  printf("6. Logout\n");
126.  printf("Enter choice: ");
127.  scanf("%d", &choice);

128.  switch(choice) {
129.  case 1:
a.  printf("\nBalance: %.2f\n", a.balance);
```

```
b. break;

130. case 2:
a. printf("\nEnter amount to deposit: ");
b. scanf("%f", &amt);
c. a.balance += amt;
d. updateAccount(a);
e. printf("Amount Deposited\n");
f. break;

131. case 3:
a. printf("\nEnter amount to withdraw: ");
b. scanf("%f", &amt);
c. if (amt > a.balance)
d. printf("Insufficient Balance\n");
e. else {
f. a.balance -= amt;
g. updateAccount(a);
h. printf("Amount Withdrawn\n");
i. }
j. break;

132. case 4: {
a. int tacc;
b. float tamt;
c. printf("\nEnter Receiver Account No: ");
d. scanf("%d", &tacc);
e. printf("Enter Amount: ");
f. scanf("%f", &tamt);

g. if (tamt > a.balance) {
h. printf("Not enough balance.\n");
i. break;
j. }

k. FILE *fp = fopen("bankdata.dat", "rb");
l. struct Account r;
m. int ok = 0;

n. while (fread(&r, sizeof(r), 1, fp) == 1) {
o. if (r.acc_no == tacc) {
 i. ok = 1;
ii. break;
p. }
q. }
r. fclose(fp);

s. if (!ok) {
t. printf("Receiver not found.\n");
u. break;
v. }

w. r.balance += tamt;
```

```
x.  updateAccount(r);

y.  a.balance -= tamt;
z.  updateAccount(a);

aa. printf("%.2f Transferred Successfully!\n", tamt);
bb. break;
133. }

134. case 5:
a.  printf("\nHistory: (Simple)\n");
b.  printf("Current Balance: %.2f\n", a.balance);
c.  break;

135. case 6:
a.  printf("\nLogged Out\n");
b.  return;

136. default:
a.  printf("Invalid choice!\n");
137. }
138. } while (1);
139. }

140. int main() {
141. int choice;
142. struct Account a;

143. while (1) {
144. printf("\n===== HOME =====\n");
145. printf("1. Register\n");
146. printf("2. Login\n");
147. printf("3. Exit\n");
148. printf("Choice: ");
149. scanf("%d", &choice);
150. getchar();

151. if (choice == 1)
152. registerAccount();
153. else if (choice == 2) {
154. if (login(&a))
a.  menu(a);
155. else
a.  printf("\nInvalid Login!\n");
156. }
157. else if (choice == 3) {
158. printf("\nThank you!\n");
159. break;
160. }
161. else
162. printf("Invalid Option!\n");
163. }
```

```
164.  return 0;
165.  }
166.
```