```c
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>


struct BankAccount

{

    int account_number;

    char account_holder[50];

    float balance;

    int pin;

    long long int mob;

    char account_mail[50];

};


struct BankAccount createAccount()

{

    struct BankAccount new_account;

    new_account.account_number = rand() % 9000 + 18000000;


    printf("Name: ");

    scanf("%s", new_account.account_holder);


pincode:

    printf("Set your PIN (4 digits): ");

    scanf("%d", &new_account.pin);

    int r = new_account.pin, sum = 0;


    printf("%d\n", r);


    int rem = new_account.pin, str;
```

```c
r = r + 1000;

while (rem > 0)

{


    str = rem % 10;

    rem = rem / 10;

    sum++;

}

if (str == 9 && sum == 4)

{

    r = r - 1000;

}

sum = 0;

while (r > 0)

{

    {

        sum++;

        r = r / 10;

    }

}

if (sum == 4)

{

    goto mobile;

}

else

{

    printf("\n");

    printf("Invalid PIN configuration.");

    printf("\n");

    goto pincode;
```

```c
        }
mobile:
    printf("Mobile number (10 digits) :");
    scanf("%lld", &new_account.mob);
    long long int r2 = new_account.mob, sum2 = 0;
    // int rem2, str2;
    // rem2 = r2;


    printf("%lld\n", r2);


    // while (rem2 > 0)
    // {


    //    str2 = rem2 % 10;
    //    rem2 = rem2 / 10;
    //    sum2++;


    // }
    // printf("%d  %d  %d",str2,rem2,sum2);
    // if (sum2 == 10 && (str2 == 9 || str2 == 8 || str2 == 7 || str2 == 6))
    // {
    //    sum2=0;
    if (r2 > 5999999999)
    {
        while (r2 > 0)
        {
            sum2++;
            r2 = r2 / 10;
        }
        if (sum2 == 10)
        {
```

```c
            goto proceed;
        }
        else
        {
            printf("\n");
            printf("Invalid Mobile number configuration.");
            printf("\n");
            goto mobile;
        }
    }

    else
    {
        printf("\n");
        printf("Invalid Mobile number configuration.");
        printf("\n");
        goto mobile;
    }

proceed:

    printf("E-mail :");
    scanf("%s", &new_account.account_mail);


    new_account.balance = 0.0;


    printf("\nAccount created successfully. Your account number is: %d\n",
new_account.account_number);


    return new_account;
}
```

```c
void displayAccount(struct BankAccount account)
{
    printf("\nAccount Information\n");
    printf("Account Number: %d\n", account.account_number);
    printf("Account Holder's Name: %s\n", account.account_holder);
    printf("Account Holder's E-mail: %s\n", account.account_mail);
    printf("Current Balance: $%.2f\n", account.balance);


    FILE *ptr = fopen("display.txt", "w");


    char x[] = "\t\t\t\t\t\t\t\t\t\tAccount Information";
    fputs(x, ptr);


    char a[] = "\n\n\nAccount number : ";
    int size = strlen(a);
    sprintf(a + size, "%d", account.account_number);
    fputs(a, ptr);


    char b[] = "\nAccount Holder's name : ";
    int size2 = strlen(b);
    sprintf(b + size2, "%s", account.account_holder);
    fputs(b, ptr);


    char c[] = "\nAccount Holder's E-mail : ";
    int size3 = strlen(c);
    sprintf(c + size3, "%s", account.account_mail);
    fputs(c, ptr);


    char str[] = "\nCurrent Balance: ";
    int size4 = strlen(str);
    sprintf(str + size4, "%.2f", account.balance);
```

```c
        fputs(str, ptr);


        fclose(ptr);
}
void deposit(struct BankAccount *account, float amount, int account_number)
{
        time_t currentTime;
        time(&currentTime);


        // Convert the current time to the local time
        struct tm *localTime = localtime(&currentTime);


        if (amount > 0)
        {
                account->balance += amount;
                printf("Deposit successful. Your new balance is: $%.2f\n", account->balance);


                FILE *ptr = fopen("deposit.txt", "w");


                char x[] = "\t\t\t\t\t\t\t\t\t\t\tDeposit  Slip";
                fputs(x, ptr);


                char y[] = "\n\nTime -> ";
                int size0 = strlen(y);
                sprintf(y + size0, "%02d:%02d:%02d", localTime->tm_hour, localTime->tm_min, localTime->tm_sec);
                fputs(y, ptr);


                char a[] = "\n\nAccount number : ";
                int size = strlen(a);
                sprintf(a + size, "%d", account_number);
```

```c
        fputs(a, ptr);


        char b[] = "\nAccount Holder's name : ";
        int size2 = strlen(b);
        sprintf(b + size2, "%s", account->account_holder);
        fputs(b, ptr);


        char c[] = "\nDeposited amount : ";
        int size3 = strlen(c);
        sprintf(c + size3, "%.2f", amount);
        fputs(c, ptr);


        char str[] = "\nWithdrawal successful. Your new balance is: ";
        int size4 = strlen(str);
        sprintf(str + size4, "%.2f", account->balance);
        fputs(str, ptr);


        fclose(ptr);
    }
    else
    {
        printf("Invalid deposit amount. Please enter a positive value.\n");
    }
}
void withdraw(struct BankAccount *account, float amount, int account_number)
{
    time_t currentTime;
    time(&currentTime);


    // Convert the current time to the local time
    struct tm *localTime = localtime(&currentTime);
```

```c
    if (amount > 0)
    {
        if (account->balance >= amount)
        {
            account->balance -= amount;
            printf("Withdrawal successful. Your new balance is: $%.2f\n", account->balance);


            FILE *ptr = fopen("withdraw.txt", "w");


            char x[] = "\t\t\t\t\t\t\t\t\t\t\tWithdraw  Slip";
            fputs(x, ptr);


            char y[] = "\n\nTime -> ";
            int size0 = strlen(y);
            sprintf(y + size0, "%02d:%02d:%02d", localTime->tm_hour, localTime->tm_min, localTime->tm_sec);
            fputs(y, ptr);


            char a[] = "\n\nAccount number : ";
            int size = strlen(a);
            sprintf(a + size, "%d", account_number);
            fputs(a, ptr);


            char b[] = "\nAccount Holder's name : ";
            int size2 = strlen(b);
            sprintf(b + size2, "%s", account->account_holder);
            fputs(b, ptr);


            char c[] = "\nWithdrawal amount : ";
            int size3 = strlen(c);
```

```c
            sprintf(c + size3, "%.2f", amount);

            fputs(c, ptr);


            char str[] = "\nWithdrawal successful. Your new balance is: ";

            int size4 = strlen(str);

            sprintf(str + size4, "%.2f", account->balance);

            fputs(str, ptr);


            fclose(ptr);

        }

        else

        {

            printf("Insufficient balance. Withdrawal failed.\n");

        }

    }

    else

    {

        printf("Invalid withdrawal amount. Please enter a positive value.\n");

    }

}

int validatePIN(struct BankAccount account, int enteredPIN)

{

    return (account.pin == enteredPIN);

}


int main()

{


    time_t currentTime;

    time(&currentTime);
```

```c
// Convert the current time to the local time
struct tm *localTime = localtime(&currentTime);

// Print the formatted local time
printf("\nCurrent time: %02d:%02d:%02d\n",
    localTime->tm_hour, localTime->tm_min, localTime->tm_sec);

printf("\n\n");
printf("\t\t\t\t\t\t\t\t\t\tWelcome to DHANTERAS Bank!\n\n");
struct BankAccount accounts[100];
int num_accounts = 0;

while (1)
{
    printf("\nBanking System Menu\n");
    printf("1. Create Account\n");
    printf("2. Display Account Information\n");
    printf("3. Deposit Money\n");
    printf("4. Withdraw Money\n");
    printf("5. Check Account Balance\n");
    printf("6. Exit\n");

    int choice;
    printf("\n");
    printf("Enter your choice (1-6): ");
    scanf("%d", &choice);
    printf("\n");

    switch (choice)
    {
    case 1:
```

```c
        // creating account

        if (num_accounts < 100)

        {

            accounts[num_accounts++] = createAccount();

        }

        else

        {

            printf("Maximum number of accounts reached.\n");

        }

        break;

    case 2:

        // displaying account

        if (num_accounts > 0)

        {

            int account_number, enteredPIN;

        acc:

            printf("Enter your account number: ");

            scanf("%d", &account_number);

            int found = 0;

            for (int i = 0; i < num_accounts; i++)

            {

                if (accounts[i].account_number == account_number)

                {

                pin:

                    printf("Enter your PIN: ");

                    scanf("%d", &enteredPIN);

                    if (validatePIN(accounts[i], enteredPIN))

                    {

                        displayAccount(accounts[i]);

                    }

                    else
```

```c
                {
                    printf("Invalid PIN. Access denied.\n");

                    printf("\n");

                    goto pin;
                }

                found = 1;
            }
        }
        if (found == 0)
        {
            printf("Account not found.\n");

            goto acc;

            printf("\n\n");
        }
    }
    else
    {
        printf("No accounts available.\n");
    }
    break;
case 3:
    // Depositing money into the account
    if (num_accounts > 0)
    {
        int account_number, enteredPIN;
        float amount;
    acc2:
        printf("Enter your account number: ");
        scanf("%d", &account_number);
        int found = 0;
        for (int i = 0; i < num_accounts; i++)
```

```c
        {
            if (accounts[i].account_number == account_number)
            {
            pin2:
                printf("Enter your PIN: ");
                scanf("%d", &enteredPIN);
                if (validatePIN(accounts[i], enteredPIN))
                {
                    printf("Enter the amount to deposit: $");
                    scanf("%f", &amount);
                    deposit(&accounts[i], amount, account_number);
                }
                else
                {
                    printf("Invalid PIN. Deposit failed.\n");
                    printf("\n");
                    goto pin2;
                }
                found = 1;
            }
        }
        if (found == 0)
        {
            printf("Account not found.\n");
            goto acc2;
            printf("\n\n");
        }
    }
    else
    {
        printf("No accounts available.\n");
```

```c
            }
        break;
    case 4:
        // Withdrawing money from the account
        if (num_accounts > 0)
        {
            int account_number, enteredPIN;
            float amount;
        acc3:
            printf("Enter your account number: ");
            scanf("%d", &account_number);
            int found = 0;
            for (int i = 0; i < num_accounts; i++)
            {
                if (accounts[i].account_number == account_number)
                {
                pin3:
                    printf("Enter your PIN: ");
                    scanf("%d", &enteredPIN);
                    if (validatePIN(accounts[i], enteredPIN))
                    {
                        printf("Enter the amount to withdraw: $");
                        scanf("%f", &amount);
                        withdraw(&accounts[i], amount, account_number);
                    }
                    else
                    {
                        printf("Invalid PIN. Withdrawal failed.\n");
                        printf("\n");
                        goto pin3;
                    }
```

```c
                found = 1;
            }
        }
        if (found == 0)
        {
            printf("Account not found.\n");
            goto acc3;
            printf("\n\n");
        }
    }
    else
    {
        printf("No accounts available.\n");
    }
    break;
case 5:
    // Checking account balance
    if (num_accounts > 0)
    {
        int account_number, enteredPIN;
    acc4:
        printf("Enter your account number: ");
        scanf("%d", &account_number);
        int found = 0;
        for (int i = 0; i < num_accounts; i++)
        {
            if (accounts[i].account_number == account_number)
            {
            pin4:
                printf("Enter your PIN: ");
                scanf("%d", &enteredPIN);
```

```c
            if (validatePIN(accounts[i], enteredPIN))

            {

                displayAccount(accounts[i]);

            }

            else

            {

                printf("Invalid PIN. Access denied.\n");

                printf("\n");

                goto pin4;

            }

            found = 1;

        }

    }

    if (found == 0)

    {

        printf("Account not found.\n");

        goto acc4;

        printf("\n\n");

    }

}

else

{

    printf("No accounts available.\n");

}

break;

case 6:

    // Exiting the banking system

    printf("Exiting the banking system. Goodbye!\n Thank you for your visit!\n");

    exit(0);

default:

    printf("Invalid choice. Please enter a valid option.\n");
```

```
        }
    }


    return 0;
}
```