**Mar Athanasios College For Advanced Studies Tiruvalla**

(Affiliated to Mahatma Gandhi University, Kottayam)



# MCA CP 107-SOFTWARE DEVELOPMENT LAB-I

**(C Programming)**

**C PROJECT**

# (PROJECT NAME)

**NAME:**

**REG. NO:**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**MAR ATHANASIOS COLLEGE FOR ADVANCED STUDIES TIRUVALLA**

**Tiruvalla, Pathanamthitta, Kerala - 689101**

[**www.macfast.org**](http://www.macfast.org/) **|** [**www.macfast.in**](http://www.macfast.in/)

**2023-2025**

**Mar Athanasios College For Advanced Studies Tiruvalla**

(Affiliated to Mahatma Gandhi University, Kottayam)



# MCA CP 107-SOFTWARE DEVELOPMENT LAB-I

**(C Programming)**

**C PROJECT**

# (PROJECT NAME)

**Name :**

**Register Number :**

**Semester :**

**Subject : MCA CP 107- Software Development Lab-I (C programming)**

*Certified that this is a bonafide record of the practical work done by Mr****.<NAME>*** *Reg. No.* ***<REG NO>*** *in the Department of Computer Applications during 2023- 2025 at Mar Athanasios College For Advanced Studies, Tiruvalla (MACFAST).*

**Faculty-in-Charge Head of Department**

Examiners 1……………………………………

2…………………………………….



**MAR ATHANASIOS COLLEGE FOR ADVANCED STUDIES TIRUVALLA**

# DEPARTMENT OF COMPUTER APPLICATIONS

**MCA 2023-2025: BATCH B**

**C PROJECT**

**Faculty Coordinator: Kripa Thankachan**

**Project Title: Bank Management System**

|  |  |
| --- | --- |
| **Serial No:** | **Name** |
| 1 | Aaron Biji |
| 2 | Adithya S Nath |
| 3 | Alen P Y |
| 4 | Arjun S Pillai |
| 5 | Delna Joy |
| 6 | Gopikrishnan V |
| 7 | Mihir Mani |
| 8 | Prajil Prasad |
| 9 | Santish V Pradeep |
| 10 | Sreelakshmi D Nair |
| 11 | Tincy Treesa Joseph |

**Operations**

1. **Insert a record**
2. **Edit a record**
3. **Delete a record**
4. **View**
5. **ACCOUNT MASTER FILE**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Description** | **Attributes** | **Type** |
| 1 | Account Number | a\_no | int |
| 2 | Account Holder Name | a\_name | char[20] |
| 3 | Address | a\_addr | char[40] |
| 4 | Balance | a\_bal | int |

1. **TRANSACTION FILE**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Description** | **Attributes** | **Type** |
| 1 | Transaction ID | t\_id | int |
| 2 | Account Number | a\_no | char[20] |
| 3 | Date | t\_date | char[10] |
| 4 | Credit/Debit | t\_type | int |
| 5 | Amount | t\_amt | int |

1. **LOAN FILE**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Description** | **Attributes** | **Type** |
| 1 | Loan ID | l\_id | int |
| 2 | Account Number | a\_no | int |
| 3 | Loan Date | l\_date | char[10] |
| 4 | Loan Type | l\_type | char[20] |
| 5 | Loan Amount | l\_amt | int |
| 6 | Duration | l\_dur | int |

**SOURCE CODE**

#include <stdio.h>

#include <conio.h>

void fpcheck(FILE\*);

void add();

void addAcc();

void addTra();

void addLoan();

void edit();

void editAcc();

void editTra();

void editLoan();

void dlt();

void dltAcc();

void dltTra();

void dltLoan();

void view();

void viewAcc();

void viewTra();

void viewLoan();

typedef struct Account

{

int a\_no;

char a\_name[20];

char a\_addr[40];

int a\_bal;

} Account;

typedef struct Transaction

{

int t\_id;

char a\_no[20];

char t\_date[10];

int t\_type;

int t\_amt;

} Transaction;

typedef struct Loan

{

int l\_id;

int a\_no;

char l\_date[10];

char l\_type[20];

int l\_amt;

int l\_dur;

} Loan;

void main()

{

int choice;

do

{

clrscr();

printf("\t\t======== BANK MANAGEMENT SYSTEM ========");

printf("\n\n\n\t\t 1.INSERT \n");

printf("\t\t 2.EDIT \n");

printf("\t\t 3.DELETE \n");

printf("\t\t 4.VIEW \n");

printf("\t\t 5.EXIT \n");

printf("\t\t----------------------------");

printf("\t\t\n");

fflush(stdin);

scanf("%d", &choice);

switch (choice)

{

case 1:

clrscr();

add();

break;

case 2:

clrscr();

edit();

break;

case 3:

clrscr();

dlt();

break;

case 4:

clrscr();

view();

break;

case 5:

exit(0);

break;

default:

printf("//invalid input//\n");

}

getch();

} while (choice != 5);

}

void fpcheck(FILE \*fp)

{

if(fp==NULL)

{

printf("\nUnable to open File!");

getch();

exit(0);

}

}

void add()

{

int choice;

do

{

clrscr();

printf("\n\n\n\t\t 1.Add Account \n");

printf("\t\t 2.Add TRANSACTION \n");

printf("\t\t 3.Add LOAN \n");

printf("\t\t 4.Main Menu \n");

printf("--------------------------------------");

printf("\t\t\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

clrscr();

addAcc();

break;

case 2:

clrscr();

addTra();

break;

case 3:

clrscr();

addLoan();

break;

case 4:

main();

break;

default:

printf("invalid input//\n");

}

getch();

} while (choice != 0);

}

void addAcc()

{

FILE \*fp = fopen("acmaster.bin","ab");

Account a;

int n, i;

fpcheck(fp);

printf("\n\t\t |======== ADD ACCOUNT DETAILS =============|");

printf("\nEnter the no of records to be added: ");

fflush(stdin);

scanf("%d", &n);

for (i = 1; i <= n; i++)

{

printf("\nRecord %i\n",i);

printf("Enter Account Number: ");

fflush(stdin);

scanf("%d", &a.a\_no);

printf("Enter ACCOUNT Holder Name: ");

fflush(stdin);

scanf("%[^\n]%\*c", a.a\_name);

printf("Enter ADDRESS: ");

fflush(stdin);

scanf("%[^\n]%\*c", a.a\_addr);

printf("Enter Balance: ");

fflush(stdin);

scanf("%d", &a.a\_bal);

fwrite(&a, sizeof(Account), 1, fp);

}

printf("Records added successfully!");

fclose(fp);

}

void addTra()

{

FILE \*fp = fopen("transact.bin","ab");

Transaction t;

int n, i;

fpcheck(fp);

printf("\n\t\t |======== ADD TRANSACTION DETAILS =============|");

printf("\nEnter the no of records to be added: ");

fflush(stdin);

scanf("%d", &n);

for (i = 1; i <= n; i++)

{

printf("\nRecord %i\n",i);

printf("Enter Transaction ID: ");

fflush(stdin);

scanf("%d", &t.t\_id);

printf("Enter Account Number: ");

fflush(stdin);

scanf("%[^\n]%\*c", t.a\_no);

printf("Enter Transaction Date: ");

fflush(stdin);

scanf("%[^\n]%\*c", t.t\_date);

printf("Enter Transaction Type(Credit/Debit): ");

fflush(stdin);

scanf("%d", &t.t\_type);

printf("Enter Amount: ");

fflush(stdin);

scanf("%d", &t.t\_amt);

fwrite(&t, sizeof(Transaction), 1, fp);

}

printf("Records added successfully!");

fclose(fp);

}

void addLoan()

{

FILE \*fp = fopen("loan.bin","ab");

Loan l;

int n, i;

fpcheck(fp);

printf("\n\t\t |======== ADD LOAN DETAILS =============|");

printf("\nEnter the no of records to be added: ");

fflush(stdin);

scanf("%d", &n);

for (i = 1; i <= n; i++)

{

printf("\nRecord %i\n",i);

printf("Enter Loan ID: ");

fflush(stdin);

scanf("%d", &l.l\_id);

printf("Enter Account Number: ");

fflush(stdin);

scanf("%d", &l.a\_no);

printf("Enter Loan Date: ");

fflush(stdin);

scanf("%[^\n]%\*c", l.l\_date);

printf("Enter Loan Type: ");

fflush(stdin);

scanf("%[^\n]%\*c", l.l\_type);

printf("Enter Loan Amount: ");

fflush(stdin);

scanf("%d", &l.l\_amt);

printf("Enter Loan Duration: ");

fflush(stdin);

scanf("%d", &l.l\_dur);

fwrite(&l, sizeof(Loan), 1, fp);

}

printf("Records added successfully!");

fclose(fp);

}

void edit()

{

int choice;

do

{

clrscr();

printf("\n\n\n\t\t 1.Edit Account \n");

printf("\t\t 2.Edit TRANSACTION \n");

printf("\t\t 3.Edit LOAN \n");

printf("\t\t 4.Main Menu \n");

printf("--------------------------------------");

printf("\t\t\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

clrscr();

editAcc();

break;

case 2:

clrscr();

editTra();

break;

case 3:

clrscr();

editLoan();

break;

case 4:

main();

break;

default:

printf("invalid input//\n");

}

getch();

} while (choice != 0);

}

void editAcc()

{

FILE \*fp = fopen("acmaster.bin","rb+");

Account a;

int f,flag=0;

printf("Enter ACCOUNT Number to edit: ");

fflush(stdin);

scanf("%d",&f);

while(fread(&a, sizeof(Account), 1, fp))

{

if (a.a\_no == f)

{

flag = 1;

printf("\nEnter new details of ACCOUNT\n");

printf("Enter ACCOUNT Number: ");

fflush(stdin);

scanf("%d", &a.a\_no);

printf("Enter ACCOUNT Holder Name: ");

fflush(stdin);

scanf("%[^\n]%\*c", a.a\_name);

printf("Enter ADDRESS: ");

fflush(stdin);

scanf("%[^\n]%\*c", a.a\_addr);

printf("Enter Balance: ");

fflush(stdin);

scanf("%d", &a.a\_bal);

fseek(fp, -(int)sizeof(Account), SEEK\_CUR);

fwrite(&a, sizeof(Account), 1, fp);

printf("Account details edited successfully!");

break;

}

}

if(flag==0)

printf("Account not found!");

fclose(fp);

}

void editTra()

{

FILE \*fp = fopen("transact.bin","rb+");

Transaction t;

int f,flag=0;

printf("Enter TRANSACTION ID to edit: ");

fflush(stdin);

scanf("%d",&f);

while(fread(&t, sizeof(Transaction), 1, fp))

{

if (t.t\_id == f)

{

flag = 1;

printf("\nEnter new details of TRANSACTION\n");

printf("Enter Transaction ID: ");

fflush(stdin);

scanf("%d", &t.t\_id);

printf("Enter Account Number: ");

fflush(stdin);

scanf("%[^\n]%\*c", t.a\_no);

printf("Enter Transaction Date: ");

fflush(stdin);

scanf("%[^\n]%\*c", t.t\_date);

printf("Enter Transaction Type(Credit/Debit): ");

fflush(stdin);

scanf("%d", &t.t\_type);

printf("Enter Amount: ");

fflush(stdin);

scanf("%d", &t.t\_amt);

fseek(fp, -(int)sizeof(Transaction), SEEK\_CUR);

fwrite(&t, sizeof(Transaction), 1, fp);

printf("Transaction details edited successfully!");

break;

}

}

if(flag==0)

printf("Transaction details not found!");

fclose(fp);

}

void editLoan()

{

FILE \*fp = fopen("loan.bin","rb+");

Loan l;

int f,flag=0;

printf("Enter LOAN ID to edit: ");

fflush(stdin);

scanf("%d",&f);

while (fread(&l, sizeof(Loan), 1, fp))

{

if (l.l\_id == f)

{

flag = 1;

printf("\nEnter new details of LOAN\n");

printf("Enter Loan ID: ");

fflush(stdin);

scanf("%d", &l.l\_id);

printf("Enter Account Number: ");

fflush(stdin);

scanf("%d", &l.a\_no);

printf("Enter Loan Date: ");

fflush(stdin);

scanf("%[^\n]%\*c", l.l\_date);

printf("Enter Loan Type: ");

fflush(stdin);

scanf("%[^\n]%\*c", l.l\_type);

printf("Enter Loan Amount: ");

fflush(stdin);

scanf("%d", &l.l\_amt);

printf("Enter Loan Duration: ");

fflush(stdin);

scanf("%d", &l.l\_dur);

fseek(fp, -(int)sizeof(Loan), SEEK\_CUR);

fwrite(&l, sizeof(Loan), 1, fp);

printf("Loan details edited successfully!");

break;

}

}

if(flag==0)

printf("Loan details not found!");

fclose(fp);

}

void dlt()

{

int choice;

do

{

clrscr();

printf("\n\n\n\t\t 1.Delete Account \n");

printf("\t\t 2.Delete TRANSACTION \n");

printf("\t\t 3.Delete LOAN \n");

printf("\t\t 4.Main Menu \n");

printf("--------------------------------------");

printf("\t\t\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

clrscr();

dltAcc();

break;

case 2:

clrscr();

dltTra();

break;

case 3:

clrscr();

dltLoan();

break;

case 4:

main();

break;

default:

printf("invalid input//\n");

}

getch();

} while (choice != 0);

}

void dltAcc()

{

FILE \*fp1 = fopen("acmaster.bin", "rb");

FILE \*fp2 = fopen("temp.bin", "wb");

int f, flag = 0;

Account a;

fpcheck(fp1);

fpcheck(fp2);

printf("Enter ACCOUNT Number to delete: ");

scanf("%d", &f);

while (fread(&a, sizeof(Account), 1, fp1))

{

if (a.a\_no != f)

fwrite(&a, sizeof(Account), 1, fp2);

else

flag = 1;

}

if (flag == 0)

printf("Account not found!");

else

printf("Account deleted successfully!");

fclose(fp1);

fclose(fp2);

remove("acmaster.bin");

rename("temp.bin", "acmaster.bin");

}

void dltTra()

{

FILE \*fp1 = fopen("transact.bin", "rb");

FILE \*fp2 = fopen("temp.bin", "wb");

int f, flag = 0;

Transaction t;

fpcheck(fp1);

fpcheck(fp2);

printf("Enter TRANSACTION ID to delete: ");

scanf("%d", &f);

while (fread(&t, sizeof(Transaction), 1, fp1))

{

if (t.t\_id != f)

fwrite(&t, sizeof(Transaction), 1, fp2);

else

flag = 1;

}

if (flag == 0)

printf("Transaction record not found!");

else

printf("Transaction record deleted successfully!");

fclose(fp1);

fclose(fp2);

remove("transact.bin");

rename("temp.bin", "transact.bin");

}

void dltLoan()

{

FILE \*fp1 = fopen("loan.bin", "rb");

FILE \*fp2 = fopen("temp.bin", "wb");

int f, flag = 0;

Loan l;

fpcheck(fp1);

fpcheck(fp2);

printf("Enter LOAN ID to delete: ");

scanf("%d", &f);

while (fread(&l, sizeof(Loan), 1, fp1))

{

if (l.l\_id != f)

fwrite(&l, sizeof(Loan), 1, fp2);

else

flag = 1;

}

if (flag == 0)

printf("Loan record not found!");

else

printf("Loan record deleted successfully!");

fclose(fp1);

fclose(fp2);

remove("loan.bin");

rename("temp.bin", "loan.bin");

}

void view()

{

int choice;

do

{

clrscr();

printf("\n\t\t 1.View Account \n");

printf("\t\t 2.View TRANSACTION \n");

printf("\t\t 3.View LOAN \n");

printf("\t\t 4.Main Menu \n");

printf("--------------------------------------");

printf("\t\t\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

clrscr();

viewAcc();

break;

case 2:

clrscr();

viewTra();

break;

case 3:

clrscr();

viewLoan();

break;

case 4:

main();

break;

default:

printf("invalid input//\n");

}

getch();

} while (choice != 0);

}

void viewAcc()

{

FILE \*fp = fopen("acmaster.bin","rb");

Account a;

fpcheck(fp);

printf("\n\t\t |========ACCOUNT DETAILS =========|");

while (fread(&a, sizeof(Account), 1, fp))

{

printf("\n%-5d\t%-25s\t%-20s\t%-5d\n", a.a\_no, a.a\_name, a.a\_addr, a.a\_bal);

}

fclose(fp);

}

void viewTra()

{

FILE \*fp = fopen("transact.bin","rb");

Transaction t;

fpcheck(fp);

printf("\n\t\t |========TRANSACTION DETAILS =========|");

while (fread(&t, sizeof(Transaction), 1, fp))

{

printf("\n%-5d\t%-25s\t%-20s\t%-5d\t%-5d\n", t.t\_id, t.a\_no, t.t\_date, t.t\_type, t.t\_amt);

}

fclose(fp);

}

void viewLoan()

{

FILE \*fp = fopen("loan.bin","rb");

Loan l;

fpcheck(fp);

printf("\n\t\t |========LOAN DETAILS =========|");

while (fread(&l, sizeof(Loan), 1, fp))

{

printf("\n%-5d\t%-5d\t%-10s\t%-20s\t%-5d\t%-5d\n", l.l\_id, l.a\_no, l.l\_date, l.l\_type, l.l\_amt, l.l\_dur);

}

fclose(fp);

}

**OUTPUT**

**1.MAIN MENU**



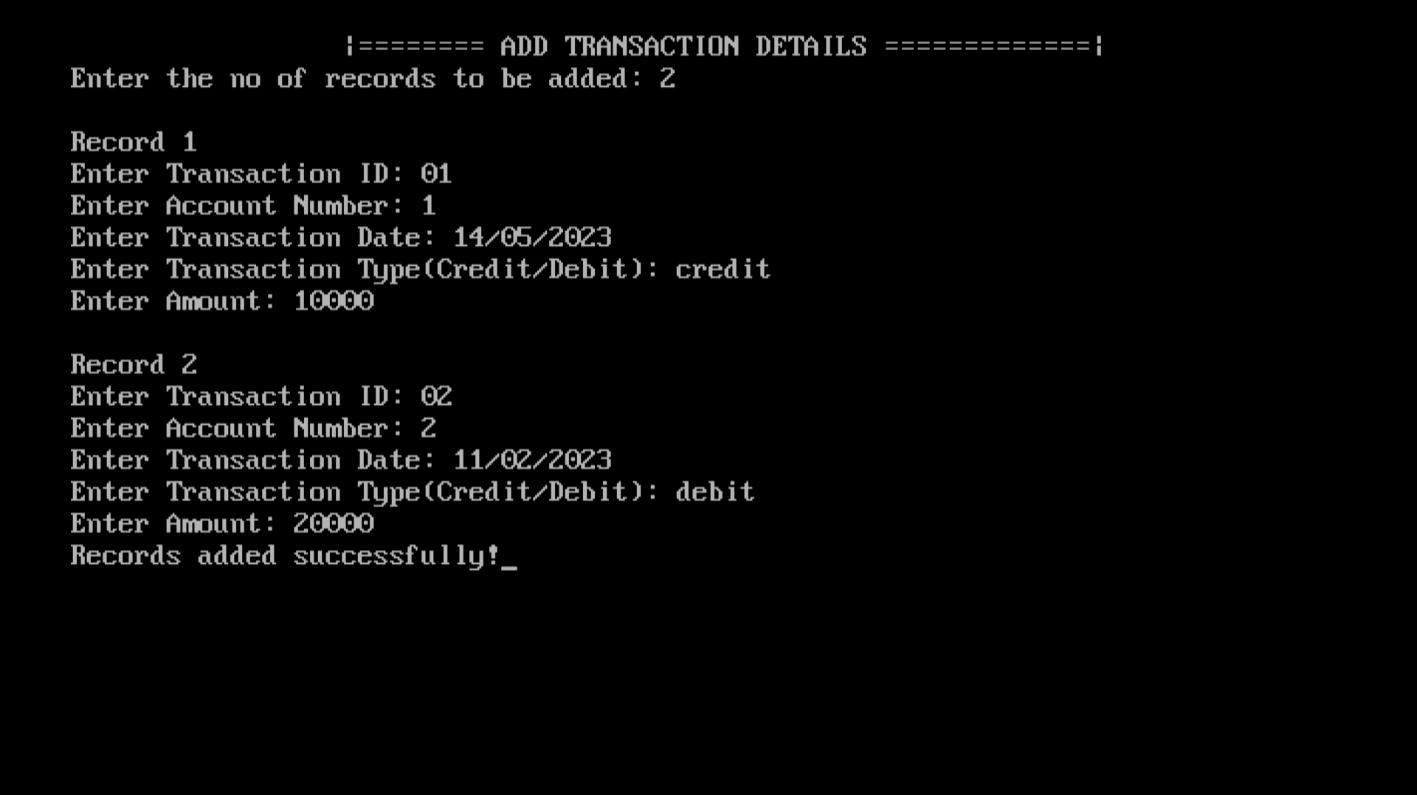
**2.INSERT MENU**

****

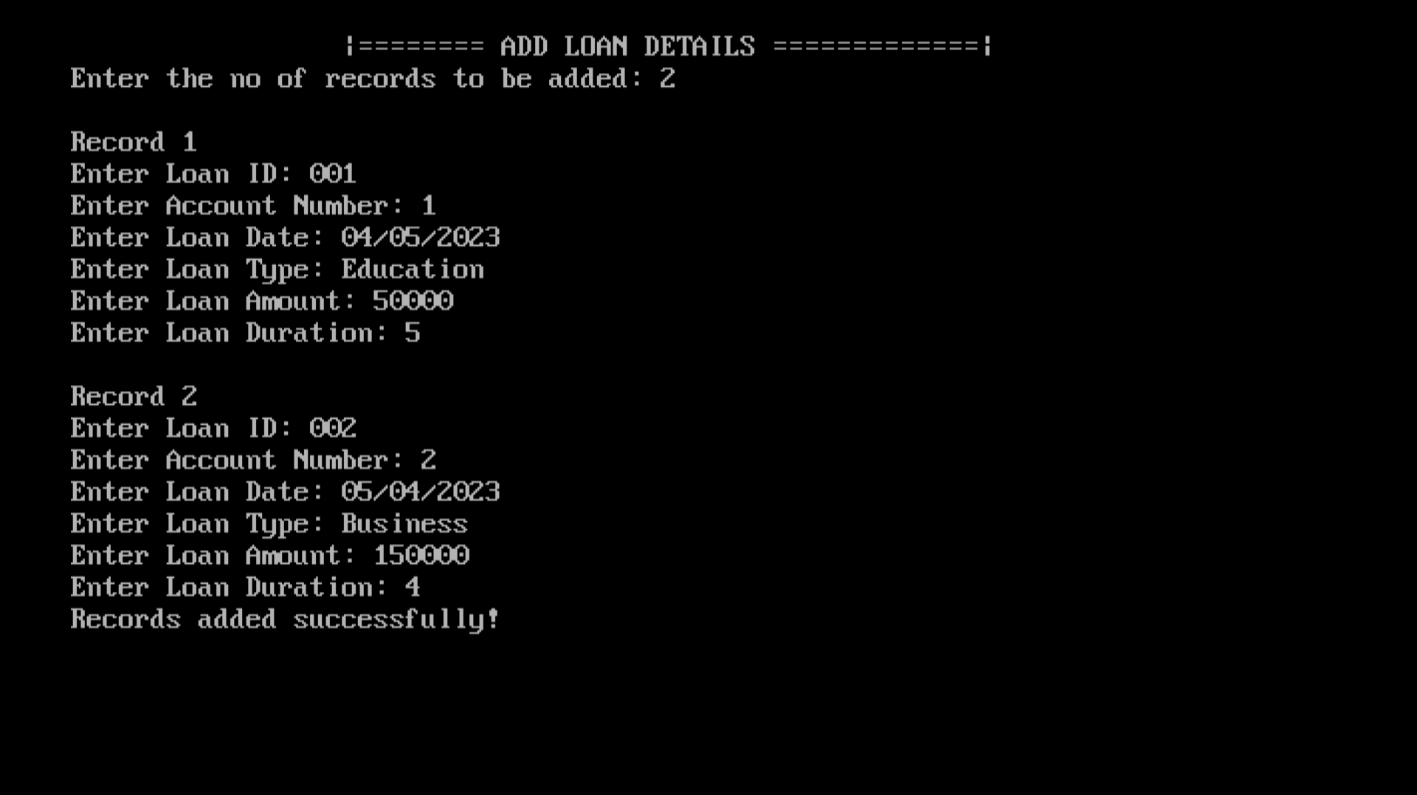
**3.ADDING ACCOUNT**

****

**4.ADDING TRANSACTION**

****

**3.ADDING LOAN DETAILS**

****

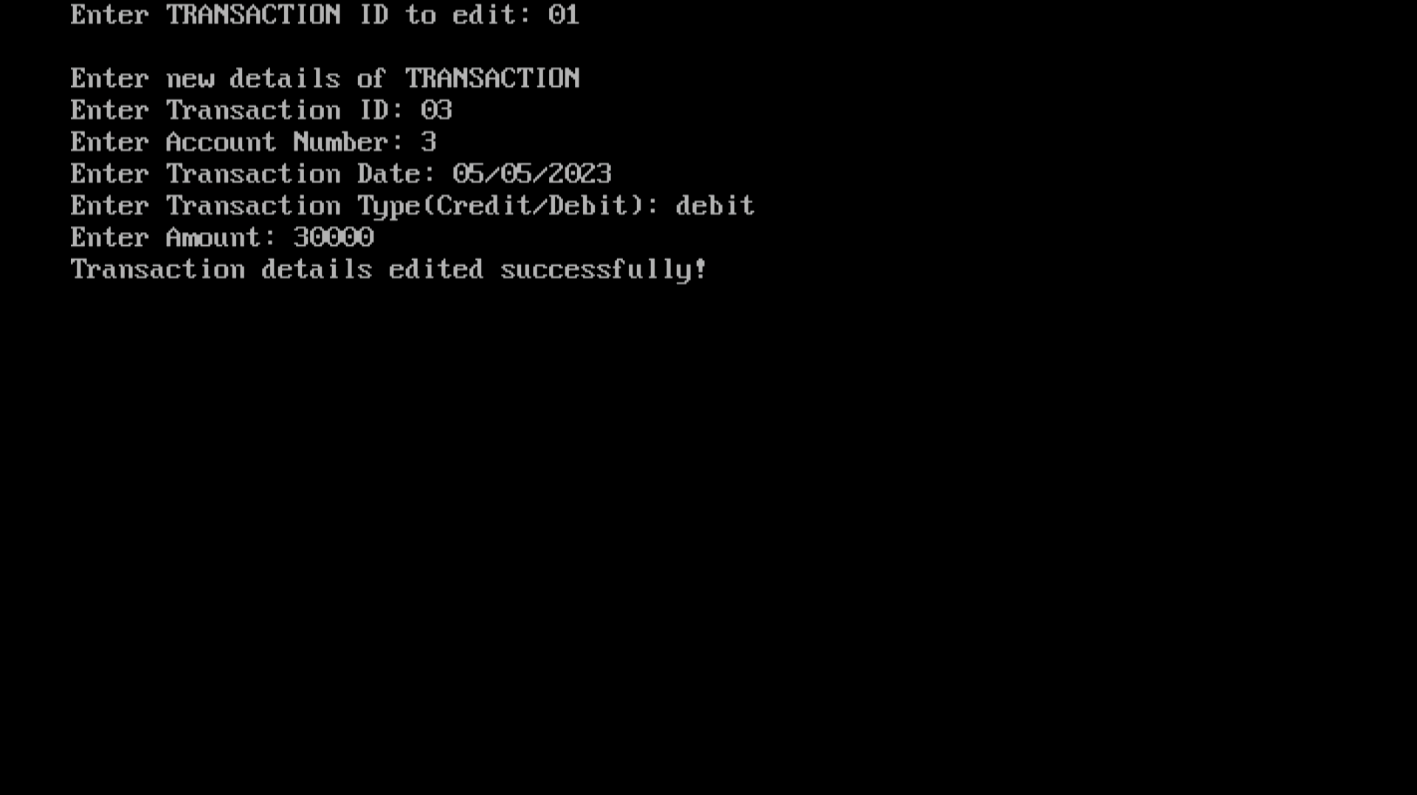
**3.EDIT MENU**

****

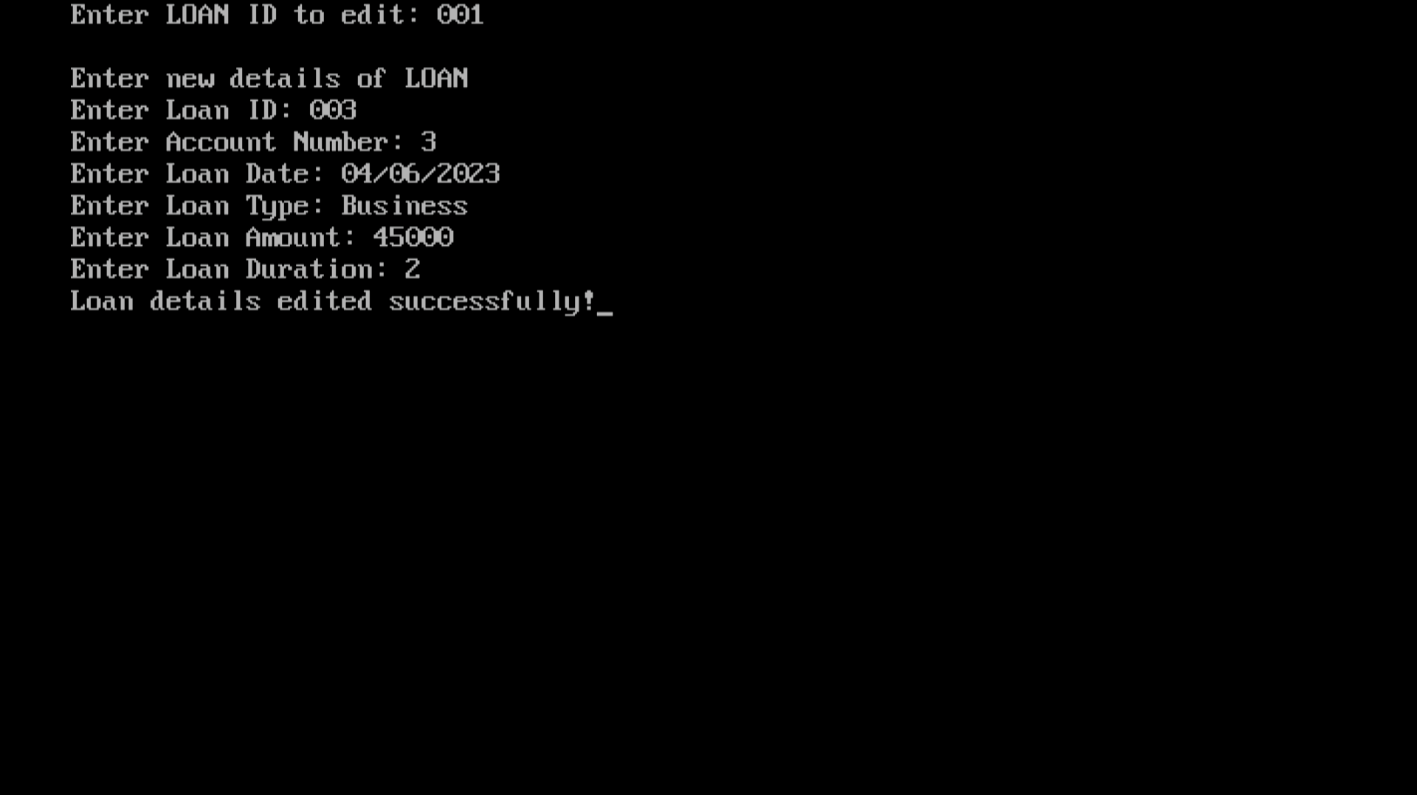
**4.EDITING ACCOUNTS**

****

**5.EDITING TRANSACTIONS**

****

**6.EDITING LOAN DETAILS**

****

**7.DELETE MENU**

****

**8.ACCOUNTS DELETION**

****

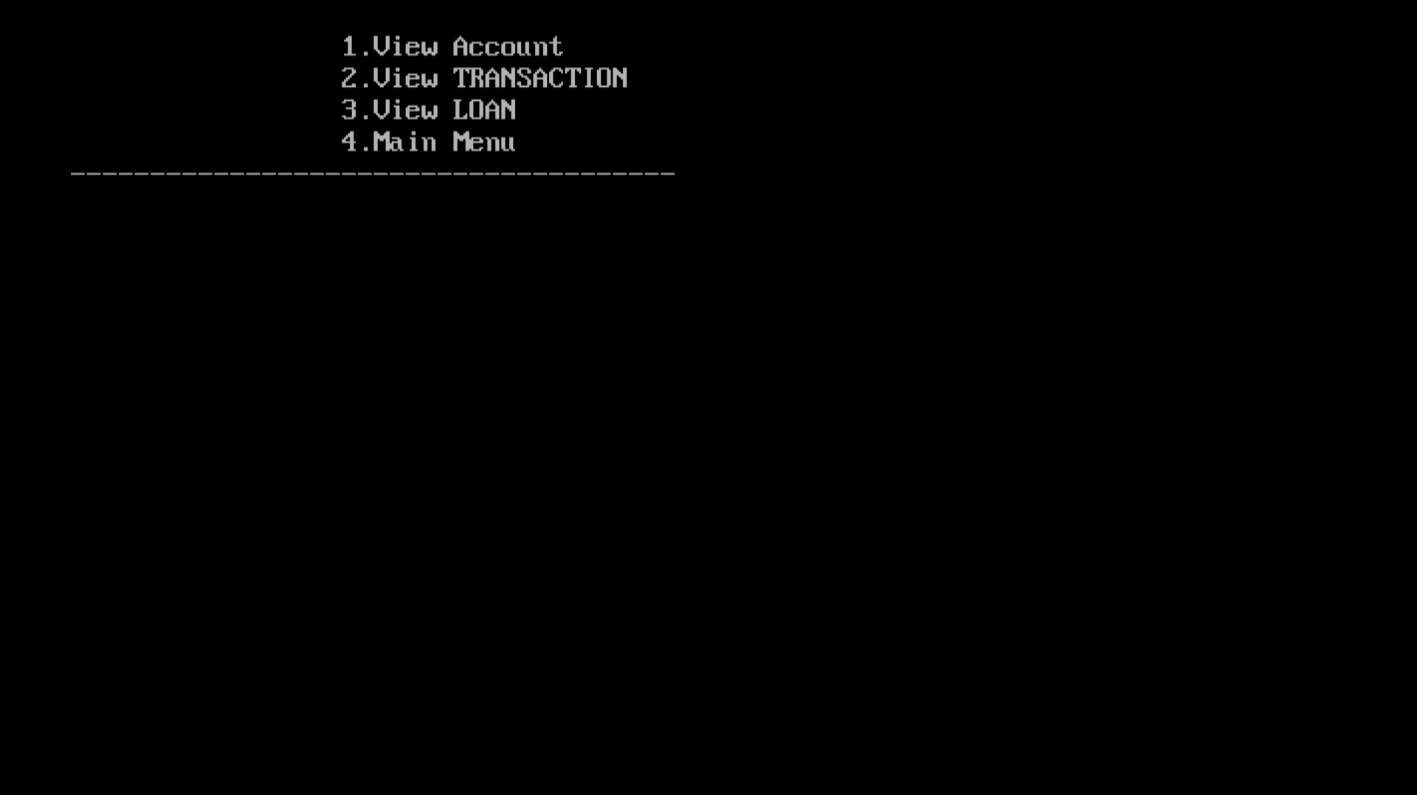
**9.TRANSACTION DELETION**

****

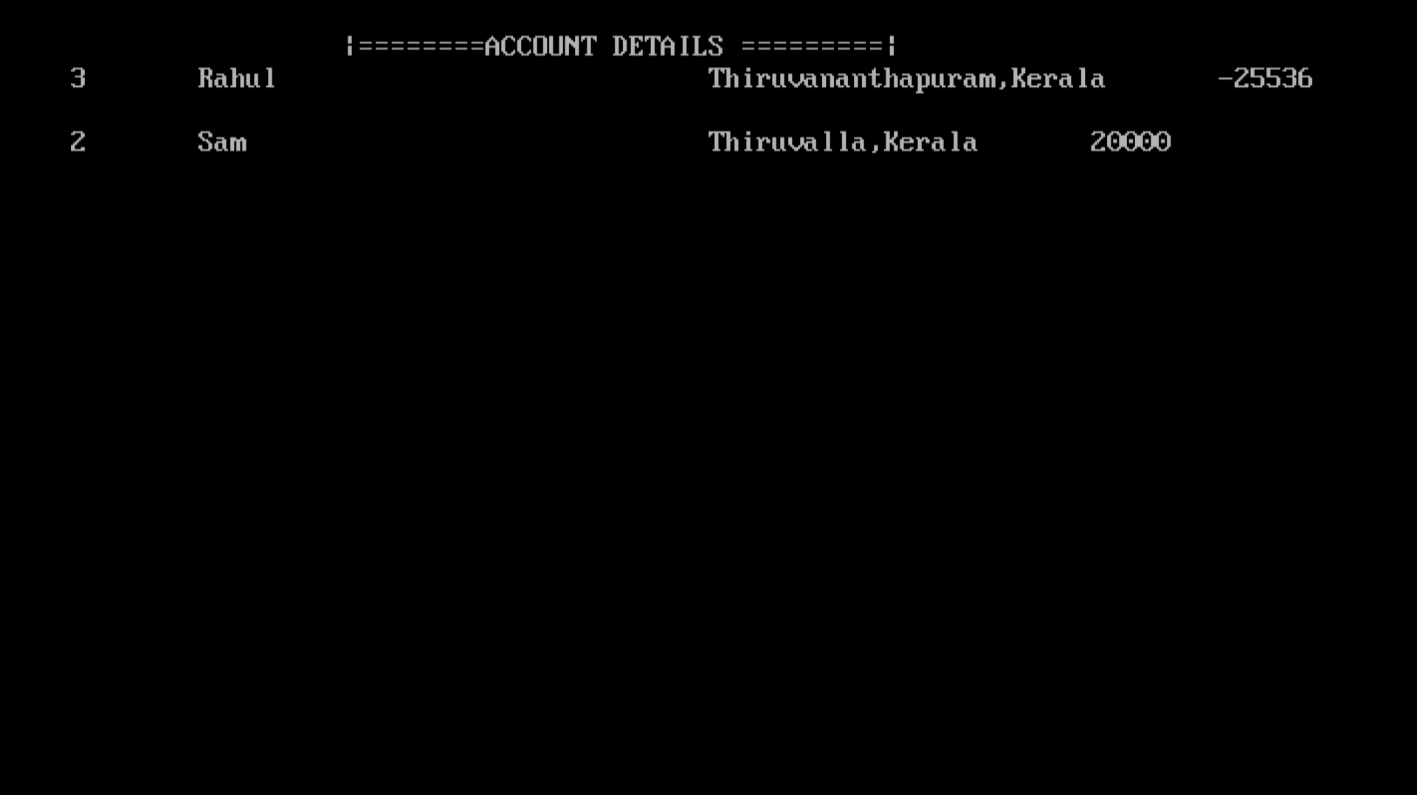
**10.LOAN DELETION**

****

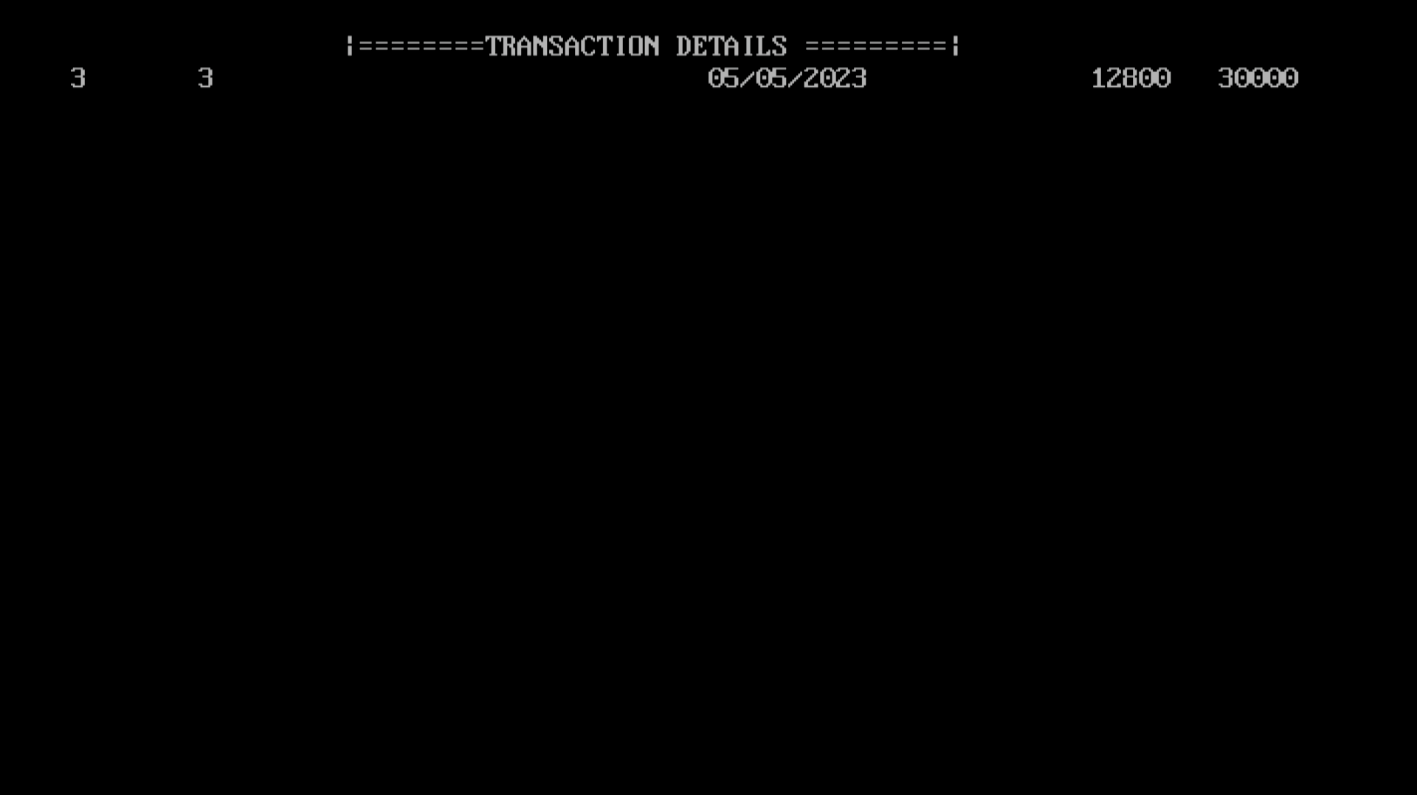
**11.VIEW MENU**

****

**12.VIEWING ACCOUNTS**

****

**13.VIEWING TRANSACTIONS**

****

**14.VIEWING LOANS**

****