Banking

System

**Project Member:**

**1823-**ImtiyazKhan

**1812-**EashanDessai

**1828-**ShubhamMishra

**ACKNOWLEDGEMENT**

We owe our gratitude to the Goa University for giving us the opportunity to prove our abilities and to the Department of Computer Science and Technology for providing us with necessary facilities.

We also would like to thank and express our gratitude to all those who helped to make this mini-project a success. We are deeply thankful to our project guide Prof. Dr. Jyoti Pawar and Prof. Preeti Khorjuvenkar for all the guidance and assistance rendered to us in the successful completion of this project.

Also a special thanks to the Junior Programmers of the Computer Science and Technology Department for all the help rendered in their capacities.

Special thanks to all our batch-mates for giving their inputs for our project.

**INTRODUCTION**

* A bank is a commercial or state institution that provides financial services. Bank is the place where customers feel the sense of safety for their property.
* In history, the primary purpose of a bank was to provide liquidity to trading companies. Banks have now advanced to many more services.
* To keep the belief and trust of customers, we have the ‘**Banking System’**.
* The banking system is an application which will maintain a person’s account in a bank.
* With this system, you can perform banking activities like in a REAL bank.
* This Banking System will provide the transactions going inside the bank without much manual processing.
* Here, For a customer you can create a new account, update information of an existing account, view and manage transactions, check the details of an existing account, remove existing account and view customers’ list.

**Objective of Project**

* To automate day to day banking processes.
* To have a faster, easier and efficient banking service.
* Maintain Information in files.

**Problem Statement**

* The Banking System is an application for maintaining a person’s account in a bank.
* The system provides services related to the maintenance of a customers account and transactions.
* The system provides the user with faster , easier and efficient banking services.

**FEATURES**

STRUCTURE:

Structure is a user-defined data type in C which allows you to combine different data types to store a particular type of record. Structure helps to construct a complex data type in more meaningful way. It is somewhat similar to an Array. The only difference is that array is used to store collection of similar data types while structure can store collection of any type of data.

FILE HANDLING:

File is a collection of bytes that is stored on secondary storage devices like disk. There are two kinds of files in a system. They are,

1. Text files (ASCII)
2. Binary files
   * Text files contain ASCII codes of digits, alphabetic and symbols.
   * Binary file contains collection of bytes (0’s and 1’s). Binary files are compiled version of text file.

## **Advantage of File:**

It will contain the data even after program exit. Normally we use variable or array to store data, but data is lost after program exit. Variables and arrays are non-permanent storage medium whereas file is permanent storage medium.

**PROGRAM DESIGN**

**Hierarchy** **Chart**:

Hierarchy chart

**DATA DECLARATION**

|  |  |
| --- | --- |
| **Variables** | **Data Types** |
| Customers, DoB | Structure |
| Cust\_name, address | Char array |
| Phone no. | Double int |
| Age, acct\_no. | Int |
| Amount | Float |

**PSEUDOCODE**

* Main\_module()
  + 1. Start
  + 2. Start\_module
    - Display intro
    - Accept the userid from user
    - Validate userid
    - Accept password from user
    - Validate password
      * + If userid and password correct than Goto main menu
      * 3. Display\_menu
        + Declare variable ‘choice’
        + Display the entire Main menu
        + Read input from user (Main Menu)
        + 4. Process\_module

If (choice==1)

Create new account

If (choice==2)

Update existing account

Update Address

Update phone

* + - * + If (choice==3)

Perform transactions

Deposit the amount

Withdraw the amount

Displays error message if balance is zero

Displays Account summary

* + - * + If (choice==4)

Display existing customer details

* + - * + If (choice==5)

View customer List

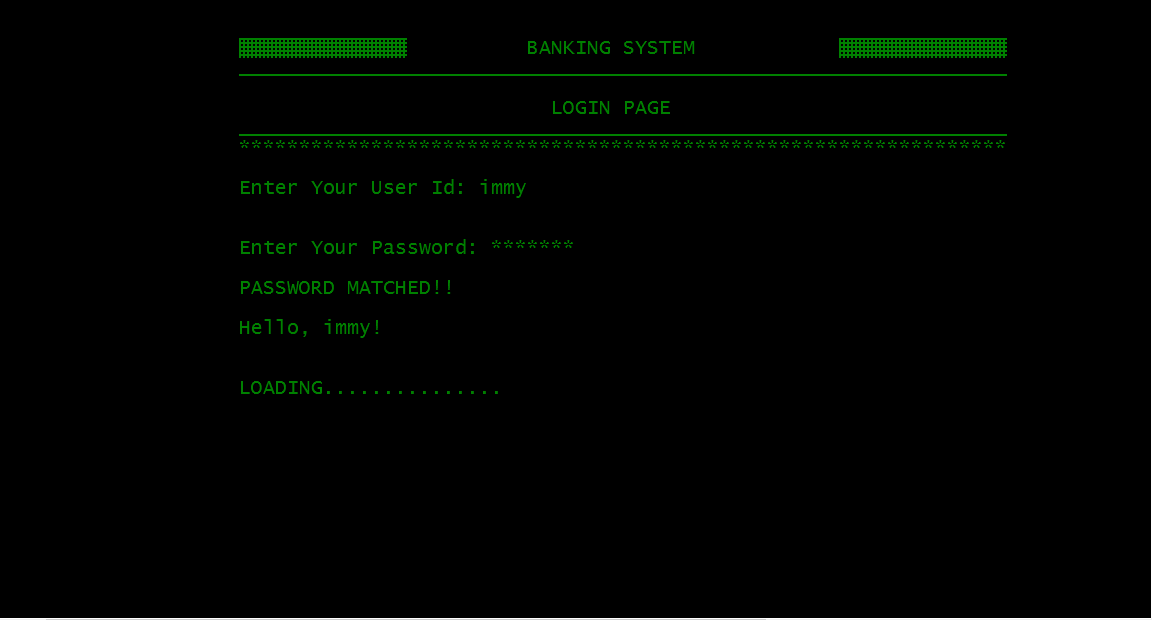
* + - * + If (choice==6)
        + Deactivating existing account
        + Displays error message if account doesn’t exist
        + If (choice==7)
        + View deactivated account
        + If (choice==8)

Exit from program

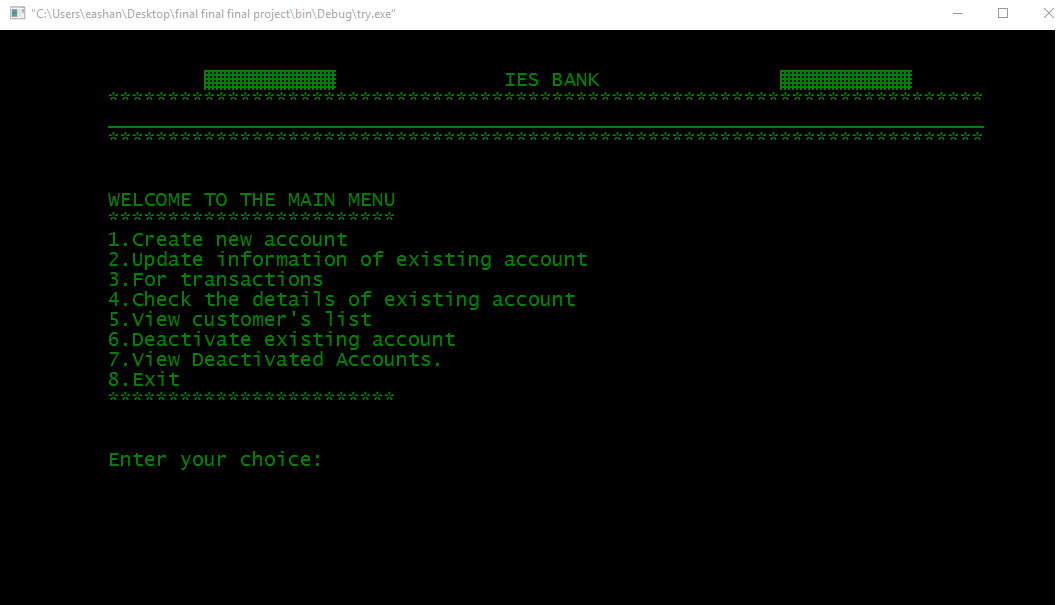
* + - * + 5.Terminate Module

**SCREENSHOTS**

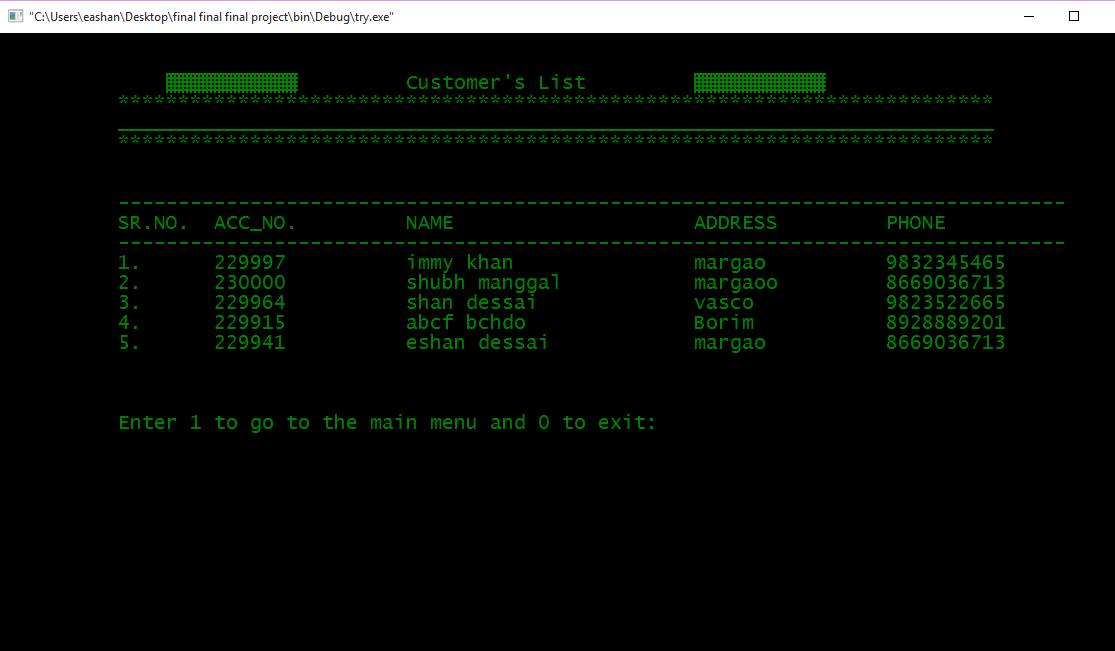
**Login page**



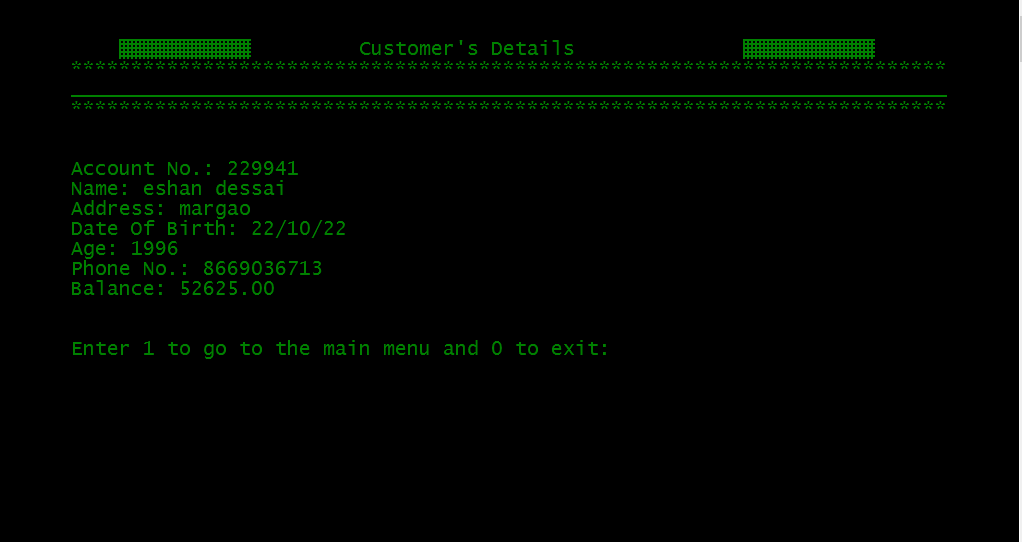
**Main Menu**



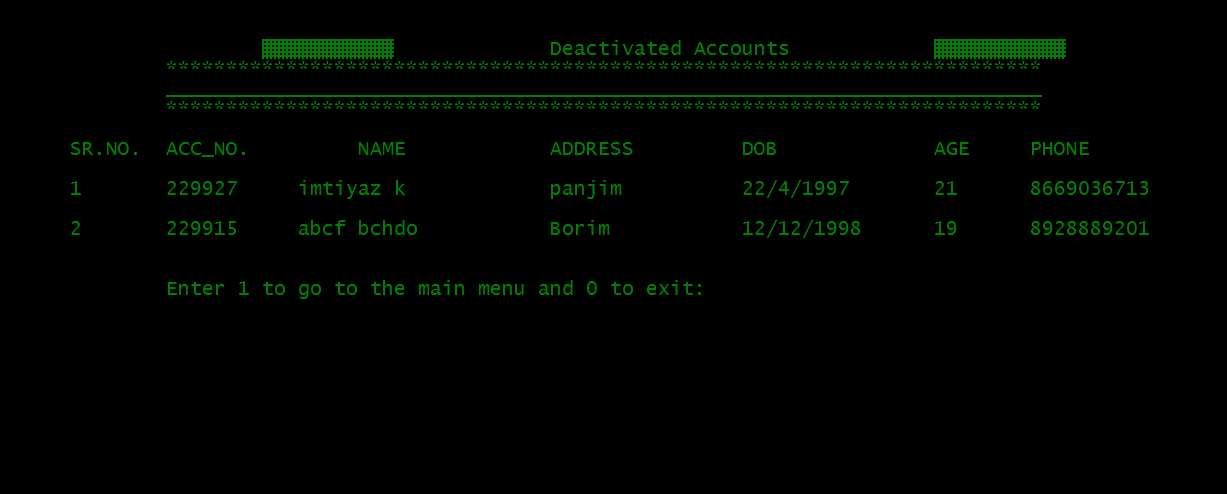
**List of Customers Account**



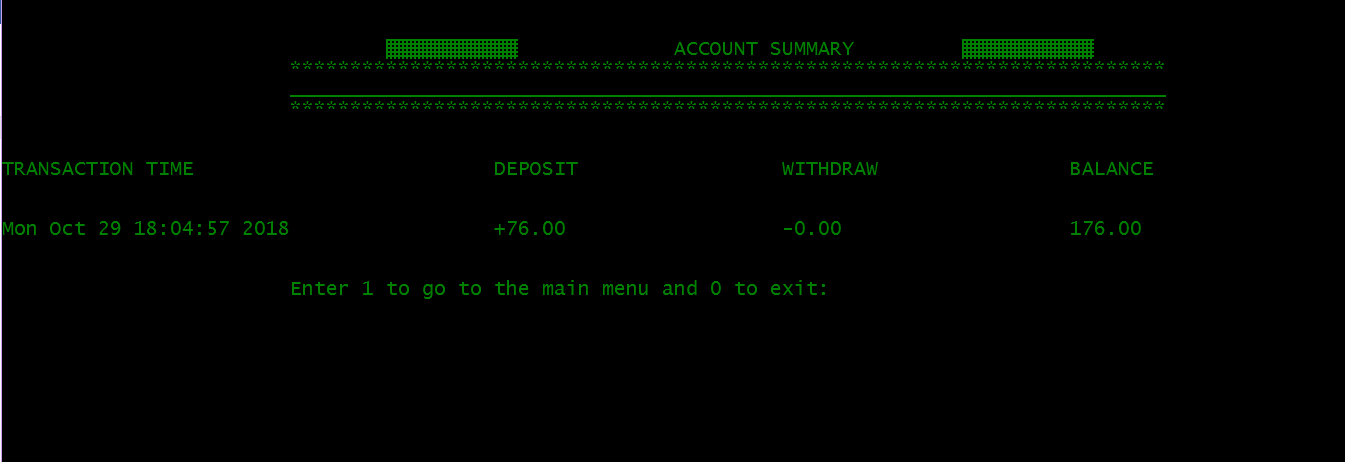
**Details of customer**



**Deactivated Accounts**



**Account Summary (passbook)**



**SOURCE CODE**

**main.c**

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

#include<string.h>

#include<conio.h>

//#include<dos.h>

#include "header.h"

//#define colo

void close1()

{

system("cls");

printf("\n\n\n\nThis C Mini Project is developed by Team Beta(Imtiyaz,Shubham,Eashan)!");

printf("\n\n\nThank You!!\n\n\n");

return 0;

}

int main()

{

int flag=0;

char password[20]="cproject",pass[50];

char userid1[20]="eashan";

char userid2[20]="immy";

char userid3[20]="shubham";

char pass1[20]="eashan123";

char pass2[20]="immy123";

char pass3[20]="shubham123";

char username[20];

int i,j,ans=0;

system("color 2");

printf("\n\n\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\t");

printf("BANKING SYSTEM");

printf("\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

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

printf("\n\n\t\t\tEnter Your User Id: ");

scanf("%s",username);

printf("\n\n\t\t\tEnter Your Password: ");

int p=0;

do{

pass[p]=getch();

if(pass[p]!='\r'){

printf("\*");

}

p++;

}while(pass[p-1]!='\r');

pass[p-1]='\0';

if(strcmp(userid1,username)==0)

{

if(strcmp(pass1,pass)==0)

{

printf("\n\n\t\t\tPASSWORD MATCHED!!");

printf("\n\n\t\t\tHello, %s! \n",userid1);

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

for(i=0;i<=15;i++)

{

Sleep(100);

printf(".");

}

system("cls");

menu();

}

else

{

printf("\n\n\t\t\tWrong Password!!\n\n");

login\_try2:

printf("\n\t\t\tEnter 1 to try again and 0 to exit: ");

scanf("%d",&ans);

if(ans==1)

{

system("cls");

main();

}

else if(ans==0)

{

system("cls");

close1();

}

else

{

printf("\n\n\t\t\tInvalid Input!!!");

Sleep(2000);

system("cls");

goto login\_try2;

}

}

}

else if(strcmp(userid2,username)==0)

{

if(strcmp(pass2,pass)==0)

{

printf("\n\n\t\t\tPASSWORD MATCHED!!");

printf("\n\n\t\t\tHello, %s! \n",userid2);

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

for(i=0;i<=15;i++)

{

Sleep(100);

printf(".");

}

system("cls");

menu();

}

else

{

printf("\n\n\t\t\tWrong Password!!\n\n");

login\_try3:

printf("\n\t\t\tEnter 1 to try again and 0 to exit: ");

scanf("%d",&ans);

if(ans==1)

{

system("cls");

main();

}

else if(ans==0)

{

system("cls");

close1();

}

else

{

printf("\n\n\t\t\tInvalid Input!!!");

Sleep(2000);

system("cls");

goto login\_try3;

}

}

}

else if(strcmp(userid3,username)==0)

{

if(strcmp(pass3,pass)==0)

{

printf("\n\n\t\t\tPASSWORD MATCHED!!");

printf("\n\n\t\t\tHello, %s! \n",userid3);

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

for(i=0;i<=15;i++)

{

Sleep(100);

printf(".");

}

system("cls");

menu();

}

else

{

printf("\n\n\t\t\tWrong Password!!\n\n");

login\_try4:

printf("\n\t\t\tEnter 1 to try again and 0 to exit: ");

scanf("%d",&ans);

if(ans==1)

{

system("cls");

main();

}

else if(ans==0)

{

system("cls");

close1();

}

else

{

printf("\n\n\t\t\tInvalid Input!!!");

Sleep(2000);

system("cls");

goto login\_try4;

}

}

}

else{

printf("\n\n\n\t\t\tYou have entered invalid User Id!!!\n\n");

login\_try5:

printf("\n\t\t\tEnter 1 to try again and 0 to exit: ");

scanf("%d",&ans);

if(ans==1)

{

system("cls");

main();

}

else if(ans==0)

{

system("cls");

close1();

}

else

{

printf("\n\n\t\t\tInvalid Input!!!");

Sleep(2000);

system("cls");

goto login\_try5;

}

}

}

**menu.c**

#include<stdio.h>

#include "services.h"

void menu()

{

int choice;

int c=1;

system("cls");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\t IES BANK\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\n\n\t\t\tWELCOME TO THE MAIN MENU ");

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

printf("\n\t\t\t1.Create new account\n\t\t\t2.Update information of existing account\n\t\t\t3.For transactions\n\t\t\t4.Check the details of existing account\n\t\t\t5.View customer's list\n\t\t\t6.Deactivate existing account\n\t\t\t7.View Deactivated Accounts.\n\t\t\t8.Exit");

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

choice:

printf("\n\n\n\t\t\tEnter your choice: ");

choice=getValidInteger();

switch(choice)

{

case 1:new\_acc();

break;

case 2:update\_acc();

break;

case 3:transact();

break;

case 4: view\_details();

break;

case 5:view\_list();

break;

case 6:erase\_acc();

break;

case 7:view\_deact();

break;

case 8:close1();

break;

default:printf("\n\t\t\tInvalid Input! TRY AGAIN");

goto choice;

break;

}

// }while(c==1);

}

**func.c**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include "services.h"

#include<time.h>

#include<string.h>

int getValidInteger()

{

char str[10];

int l,i;

short int flag=0;

choice:

do{

scanf("%s",str);

l=strlen(str);

if(str[0]=='+' || str[0]=='-' || (isdigit(str[0])))

{

flag=1;

}

//End If

if(flag==1)

{

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

{

if(str[i]<'0' || str[i]>'9')

{

flag=0;

break;

}

//End if

}

//End For Loop

}

//End If

if(flag==1)

{

return(atoi(str));

}

//End If

else

{

printf("\n\t\t\t\aInappropriate Input(Digit expected)");

printf("\n\t\t\tPlease Enter a Valid Input:\a ");

goto choice;

}

//End Else

}

while(flag==0);

//End Do While Loop

}

char getvalidname(char str[50])

{

int i,flag=0;

//char fname[50],lname[50];

for(i=0;i<strlen(str);i++)

{

if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))

{

flag=flag+0;

//return flag ;

}

else{

flag=flag+1;

//return flag;

}

}

if(flag>=1)

{

return flag;

}

else{

flag=0;

return flag;

}

}

int isLeapYear(int year, int mon)

{

int flag = 0;

if (year % 100 == 0)

{

if (year % 400 == 0)

{

if (mon == 2)

{

flag = 1;

}

}

}

else if (year % 4 == 0)

{

if (mon == 2)

{

flag = 1;

}

}

return (flag);

}

int age(int day,int months,int years)

{

int DaysInMon[] = {31, 28, 31, 30, 31, 30,

31, 31, 30, 31, 30, 31};

int days=day, month=months, year=years;

char dob[100];

time\_t ts;

struct tm \*ct;

/\* enter date of birth \*/

// printf("Enter your date of birth (DD/MM/YYYY): ");

//scanf("\n\t\t\t%d/%d/%d",&days,&month, &year);

/\*get current date.\*/

ts = time(NULL);

ct = localtime(&ts);

//printf("Current Date: %d/%d/%d\n",

// ct->tm\_mday, ct->tm\_mon + 1, ct->tm\_year + 1900);

days = DaysInMon[month - 1] - days + 1;

/\* leap year checking\*/

if (isLeapYear(year, month))

{

days = days + 1;

}

/\* calculating age in no of days, years and months \*/

days = days + ct->tm\_mday;

month = (12 - month) + (ct->tm\_mon);

year = (ct->tm\_year + 1900) - year - 1;

/\* checking for leap year feb - 29 days \*/

if (isLeapYear((ct->tm\_year + 1900), (ct->tm\_mon + 1)))

{

if (days >= (DaysInMon[ct->tm\_mon] + 1))

{

days = days - (DaysInMon[ct->tm\_mon] + 1);

month = month + 1;

}

}

else if (days >= DaysInMon[ct->tm\_mon])

{

days = days - (DaysInMon[ct->tm\_mon]);

month = month + 1;

}

if (month >= 12)

{

year = year + 1;

month = month - 12;

}

/\* print age \*/

//printf("\n## Hey you are %d years %d months and %d days old. ##\n", year, month, days);

return year;

}

int validdate(int days\_check, int months\_check, int years\_check)

{

int day=days\_check,month=months\_check,year=years\_check,count=0,flag=0;

int currentyear,difference;

struct tm \*current;

int dayflag=1,monthflag=2,yearflag=3;

int temp\_day=day;

while(temp\_day!=0)

{

temp\_day/=10;

count ++;

}

if(count>2)

{

flag=1;

// return flag;

}

else {

if(day==31)

{

if(month==2||month==4 || month==5 || month==6 || month==9|| month==11)

{

flag=1;

//return flag;

}

else{

flag=0;

}

}

else if(day>28 && month==2)

{

flag=1;

//return flag;

}

else if(day<1 || day>31)

{

flag=1;

// return flag;

}

else{

flag=0;

//return flag;

}

}

if(flag==1)

{

return dayflag;

}

else{

count=0;

int temp\_month=month;

while(temp\_month!=0)

{

temp\_month/=10;

count ++;

}

if(count>2)

{

flag=1;

//return flag;

}

else {

if(month<1 || month>12)

{

flag=1;

//return flag;

}

else{

flag=0;

//return flag;

}

}

if(flag==1)

{

return monthflag;

}

else{

count=0;

int temp\_year=year;

while(temp\_year!=0)

{

temp\_year/=10;

count ++;

}

if(count<4 || count>4)

{

flag=1;

//return flag;

if(flag==1)

{

return yearflag;

}

else{

flag=0;

}

}

else{

time\_t timenow;

time(&timenow);

current = localtime(&timenow);

currentyear = current->tm\_year+1900;

difference=currentyear-year;

if(year>currentyear || difference>100|| difference<10)

{

flag=1;

//return flag;

}

else{

flag=0;

//return flag;

}

if(flag==1)

{

return yearflag;

}

else

{

flag=0;

return flag;

}

}

}

}

}

int getValidintdate(int date)

{

char str[10];

int l,i;

short int flag=0;

choice:

do{

sprintf(str,"%d",date);

l=strlen(str);

if(str[0]=='+' || str[0]=='-' || (isdigit(str[0])))

{

flag=1;

}

//End If

if(flag==1)

{

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

{

if(str[i]<'0' || str[i]>'9')

{

flag=0;

return flag;

break;

}

//End if

}

//End For Loop

}

//End If

if(flag==1)

{

return flag;

}

//End If

else

{

return flag;

}

//End Else

}

while(flag==0);

//End Do While Loop

}

int getvalidfloat()

{

char str[10];

int l,i;

short int flag=0;

choice:

do{

scanf("%s",str);

l=strlen(str);

if(str[0]=='+' || str[0]=='-' || (isdigit(str[0])))

{

flag=1;

}

//End If

if(flag==1)

{

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

{

if(str[i]<'0' || str[i]>'9')

{

flag=0;

break;

}

//End if

}

//End For Loop

}

//End If

if(flag==1)

{

return(atof(str));

}

//End If

else

{

printf("\n\t\t\t\aInappropriate Input(Digit expected)");

printf("\n\t\t\tPlease Enter a Valid Input:\a ");

goto choice;

}

//End Else

}

while(flag==0);

//End Do While Loop

}

double validphone()

{

char str[10];

int l,i;

short int flag=0;

choice:

do{

scanf("%s",str);

l=strlen(str);

if(l==10 )

{

if(str[0]=='+' || str[0]=='-' || (isdigit(str[0])))

{

flag=1;

}

//End If

if(flag==1)

{

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

{

if(str[i]<'0' || str[i]>'9')

{

flag=0;

break;

}

//End if

}

//End For Loop

}

//End If

if(flag==1)

{

return((strtod(str,NULL)));

}

//End If

else

{

printf("\n\t\t\t\aInappropriate Input(Digit expected)");

printf("\n\t\t\tPlease Enter a Valid Input:\a ");

goto choice;

}

}

else{

printf("\n\t\t\t\aPhone Number Should be of only of 10 digits");

printf("\n\t\t\tEnter Again: ");

goto choice;

}

//End Else

}

while(flag==0);

//End Do While Loop }

**Services.h**

void new\_acc();

void view\_list();

void view\_details();

void erase\_acc();

int getValidInteger();

void transact();

void view\_deact();

char getvalidname(char str[50]);

int age(int day,int months,int years);

int validdate(int days\_check, int months\_check, int years\_check);

int getvalidfloat();

double validphone();

**account.c**

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

#include<ctype.h>

#include<time.h>

#include "services.h"

struct date{

int day,month,year;

};

struct {

char name[60];

char lname[60];

int acc\_no,age;

char address[60];

double phone;

float amt;

struct date dob;

struct date deposit;

struct date join;

struct date withdraw;

float credit;

float debit;

float balance;

char time[256];

}add,upd,check,rem,transaction,pass;

void calcTime(char \*str) {

time\_t ts;

ts = time(NULL);

strcpy(str, ctime(&ts));

str[strlen(str) - 1] = '\0';

return;

}

void new\_acc()

{

system("cls");

int ans=0,flag=0,i;

char temp\_fname[60],temp\_lname[60];

FILE \*ptr;

ptr=fopen("record.dat", "a+");

system("cls");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tADD RECORD\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

srand(time(NULL));

add.acc\_no=229000 + rand()%999 + 100;

name:

flag=0;

printf("\n\n\t\t\tEnter first name: ");

scanf("%s",add.name);

flag=getvalidname(add.name);

if(flag>=1)

{

printf("\n\t\t\tInvalid input, Name should contain only characters");

goto name;

}

else

{

flag=0;

printf("\n\t\t\tEnter last name: ");

scanf("%s",add.lname);

flag=getvalidname(add.lname);

if(flag>=1)

{

printf("\n\t\t\tInvalid input, Name should contain only characters");

goto name;

}

else{

dob:

printf("\n\t\t\tEnter the date of birth(dd/mm/yyyy): ");

scanf("%d/%d/%d",&add.dob.day,&add.dob.month,&add.dob.year);

int valid=0;

/\*

valid=getValidintdate(add.dob.day);

int valid\_sum=valid;

valid=getValidintdate(add.dob.month);

valid\_sum=valid\_sum+valid;

valid=getValidintdate(add.dob.year);

valid\_sum=valid\_sum+valid;

if(valid\_sum>=1)

{

printf("\n\t\t\tInvalid Input,Date should be of type integer");

}

else{\*/

valid=0;

valid=validdate(add.dob.day,add.dob.month,add.dob.year);

if(valid==1)

{

printf("\n\t\t\tInvalid Day!");

goto dob;

}

else if(valid==2)

{

printf("\n\t\t\tInvalid Month!");

goto dob;

}

else if(valid==3)

{

printf("\n\t\t\tInvalid Year!");

goto dob;

}

else{

add.age=age(add.dob.day,add.dob.month,add.dob.year);

address:

flag=0;

printf("\n\t\t\tEnter the address: ");

scanf("%s",add.address);

flag=getvalidname(add.address);

if(flag>=1)

{

printf("\n\t\t\tInvalid! Characters Expected!");

goto address;

}

else{

phone:

printf("\n\t\t\tEnter the phone number: ");

add.phone=validphone();

amount:

printf("\n\t\t\tEnter the amount to deposit:$");

add.amt=getValidInteger();

if (add.amt>0)

{

FILE \*passbk;

passbk=fopen("pass.dat","ab+");

pass.acc\_no=add.acc\_no;

pass.credit=add.amt;

pass.balance=add.amt;

pass.debit=0;

calcTime(pass.time);

//count++;

fwrite(&pass,sizeof(pass),1,passbk);

fclose(passbk);

}

else

{

printf("\n\t\t\tAmmount cannot be less than or equal to zero");

Sleep(200);

}

fprintf(ptr,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

system("cls");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tAccount Details\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\t\t\tAccount No:%d",add.acc\_no);

printf("\n\t\t\tName:\t%s %s",add.name,add.lname);

printf("\n\t\t\tD.O.B:\t%d/%d/%d",add.dob.day,add.dob.month,add.dob.year);

printf("\n\t\t\tAge:\t%d",add.age);

printf("\n\t\t\tAddress:%s",add.address);

printf("\n\t\t\tContact No:%.0lf",add.phone);

printf("\n\t\t\tAmount:%.2f",add.amt);

fclose(ptr);

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

for(int j=0;j<=15;j++)

{

Sleep(300);

printf(".");

}

}

}

}

}

system("cls");

printf("\n\n\t\t\tACCOUNT CREATED SUCCESSFULLY!!");

add\_invalid:

printf("\n\n\t\t\tPress 1 to add more records");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

new\_acc();

else if(ans==2)

menu();

else if(ans==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

void view\_list()

{

system("cls");

FILE \*view;

int ans=0,i=1;

view=fopen("record.dat", "r");

int count=0;

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

for(int j=0;j<=15;j++)

{

Sleep(100);

printf(".");

}

system("cls");

printf("\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tCustomer's List\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

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

printf("\n\t\t\tSR.NO.\tACC\_NO.\t\tNAME\t\t\tADDRESS\t\tPHONE");

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

while(fscanf(view,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

printf("\n\t\t\t%d.\t%d\t\t%s %s\t\t%s\t\t%.0lf ",i++,add.acc\_no,add.name,add.lname,add.address,add.phone);

count++;

}

fclose(view);

if(count==0)

{

system("cls");

printf("\n\t\t\tNO RECORDS!!\n");

}

add\_invalid:

printf("\n\n\n\n\t\t\tEnter 1 to go to the main menu and 0 to exit:");

ans=getValidInteger();

system("cls");

if (ans==1)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

void erase\_acc(void)

{

system("cls");

FILE \*oldrec,\*newrec,\*deactrec;

int ans=0;

oldrec=fopen("record.dat","r");

newrec=fopen("new.dat","w");

deactrec=fopen("deactivate.dat","a+");

int count=0;

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tDEACTIVATE ACCOUNT\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\n\t\t\tEnter the account no. of the customer to deactivate the account:");

rem.acc\_no=getValidInteger();

while(fscanf(oldrec,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

if(add.acc\_no!=rem.acc\_no)

{

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

}

else

{

fprintf(deactrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

count++;

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

for(int j=0;j<=15;j++)

{

Sleep(100);

printf(".");

}

printf("\n\n\t\t\tRECORD DEACTIVATED SUCCESSFULLY!!!\n");

}

}

fclose(deactrec);

fclose(oldrec);

fclose(newrec);

remove("record.dat");

rename("new.dat","record.dat");

if(count==0)

{

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

printf("\n\n\t\t\tRECORD NOT FOUND!!");

add\_invalid:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

erase\_acc();

else if(ans==2)

menu();

else if(ans==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

else

{

add\_invalid1:

printf("\n\t\t\tPress 1 to deactivate more accounts");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

erase\_acc();

else if(ans==2)

menu();

else if(ans==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid1;

}

}

}

void update\_acc()

{

system("cls");

int ans=0,choice,test=0,flag=0,flag1=0;

FILE \*oldrec,\*newrec;

oldrec=fopen("record.dat","r");

newrec=fopen("new.dat","w");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tUPDATE RECORD\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\t\t\tEnter the account no. of the customer whose info you want to change\n\t\t\t");

upd.acc\_no=getValidInteger();

while(fscanf(oldrec,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

if (add.acc\_no==upd.acc\_no)

{ test=1;

do

{

printf("\n\t\t\tWhich information do you want to change?\n\t\t\t1.Address\n\t\t\t2.Phone\n\n\t\t\tEnter your choice(1 for address and 2 for phone):");

choice=getValidInteger();

if(choice!=1 && choice!=2)

printf("\n\t\t\tInvalid Choice...! Try Again\a\n");

else

flag=1;

}while(flag==0);

//system("cls");

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

if(choice==1)

{

upd\_address:

printf("\n\t\t\tEnter the new address:");

scanf("%s",upd.address);

flag1=getvalidname(upd.address);

if(flag1>=1)

{

printf("\n\t\t\tInavlid Input!,Only Characters Expected");

goto upd\_address;

}

else{

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,upd.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

system("cls");

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

printf("\n\t\t\tChanges saved!");

}

}

else if(choice==2)

{

new\_phone:

printf("\n\t\t\tEnter the new phone number:");

upd.phone=validphone();

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,upd.phone,add.amt,add.join.day,add.join.month,add.join.year);

system("cls");

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

printf("\n\t\t\tChanges saved!");

}

}

else

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

}

fclose(oldrec);

fclose(newrec);

remove("record.dat");

rename("new.dat","record.dat");

if(test!=1)

{

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

system("cls");

printf("\n\t\t\tRECORD NOT FOUND!!\a\a\a");

add\_invalid:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to return to main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

update\_acc();

else if(ans==2)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

else

{

add\_inavalid:

printf("\n\n\n\t\t\tEnter 1 to go to the main menu and 0 to exit:");

ans=getValidInteger();

system("cls");

if(ans==1)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_inavalid;

}

}

}

//}

void transact()

{

system("cls");

int choice,test=0,ans,ans2;

//static int count=0;

int temp;

int i;

FILE \*old,\*newrec,\*passbk;

old=fopen("record.dat","r");

newrec=fopen("new.dat","w");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tTRANSCATION\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\t\t\tEnter the account no. of the customer: \n\t\t\t");

transaction.acc\_no=getValidInteger();

while(fscanf(old,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

if(add.acc\_no==transaction.acc\_no)

{

temp=transaction.acc\_no;

pass.acc\_no=transaction.acc\_no;

invalid:

test=1;

printf("\n\n\t\t\tDo you want to\n\t\t\t1.Deposit?\n\t\t\t2.Withdraw?\n\t\t\t3.Know The Account Summary\n\n\t\t\tEnter your choice: ");

choice=getValidInteger();

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

system("cls");

if (choice==1)

{

add\_invalidd:

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tDEPOSIT\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

passbk=fopen("pass.dat","ab+");

printf("\n\t\t\tEnter the amount you want to deposit:$ \n\t\t\t");

transaction.amt=getvalidfloat();

if(transaction.amt<=0){

printf("\n\n\t\t\tAmount cannot be less than or equal to zero!!\n");

add\_invalid5:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans2=getValidInteger();

system("cls");

if (ans2==1)

goto add\_invalidd;

else if(ans2==2)

menu();

else if(ans2==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid5;

}

}

else{

add.amt+=transaction.amt;

pass.credit=transaction.amt;

pass.balance=add.amt;

pass.debit=0;

calcTime(pass.time);

//count++;

fwrite(&pass,sizeof(pass),1,passbk);

fclose(passbk);

}

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

printf("\n\n\t\t\tDEPOSITED SUCCESSFULLY!!");

}

else if(choice==2)

{

add\_invalid2:

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tWITHDRAW\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

passbk=fopen("pass.dat","ab+");

printf("\n\t\t\tEnter the amount you want to withdraw:$ \n\t\t\t");

// scanf("%f",&transaction.amt);

transaction.amt=getvalidfloat();

if(transaction.amt>add.amt)

{

printf("\n\n\t\t\tSorry you have insufficient balance to conduct this transaction!!\n");

add\_invalid3:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans2=getValidInteger();

system("cls");

if (ans2==1)

goto add\_invalid2;

else if(ans2==2)

menu();

else if(ans2==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid3;

}

}

else if(transaction.amt<=0){

printf("\n\n\t\t\tWithdrawn Amount cannot be less than or equal to zeo!!\n");

add\_invalid4:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans2=getValidInteger();

system("cls");

if (ans2==1)

goto add\_invalid2;

else if(ans2==2)

menu();

else if(ans2==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid4;

}

}

else{

passbk=fopen("pass.dat","ab+");

add.amt-=transaction.amt;

pass.debit=transaction.amt;

pass.balance=add.amt;

pass.credit=0;

calcTime(pass.time);

//count++;

// printf("count: %d",count);

fwrite(&pass,sizeof(pass),1,passbk);

fclose(passbk);

}

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

printf("\n\n\t\t\tWITHDRAWN SUCCESSFULLY!!");

}

else if(choice ==3)

{

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tACCOUNT SUMMARY\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

passbk=fopen("pass.dat","rb");

printf("\n\n\nTRANSACTION TIME\t\t\t DEPOSIT\t\t WITHDRAW\t\t BALANCE\n\n");

while( fread(&pass, sizeof(pass), 1, passbk) == 1)

{

if(temp==pass.acc\_no)

printf("\n%s\t\t +%0.2f\t\t\t -%0.2f\t\t\t %0.2f", pass.time,pass.credit, pass.debit, pass.balance);

}

fclose(passbk);

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

}

else {

printf( "\nInvalid choice!!!\a");

goto invalid;

}

}

else{

fprintf(newrec,"\n\t\t\t%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",add.acc\_no,add.name,add.lname,add.dob.day,add.dob.month,add.dob.year,add.age,add.address,add.phone,add.amt,add.join.day,add.join.month,add.join.year);

}

}

fclose(old);

fclose(newrec);

remove("record.dat");

rename("new.dat","record.dat");

if(test!=1)

{

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

for(int j=0;j<=11;j++)

{

Sleep(100);

printf(".");

}

system("cls");

printf("\n\n\t\t\tRECORD NOT FOUND\a\a\a!!");

add\_invalid:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to go to the main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

transact();

else if(ans==2)

menu();

else if(ans==3)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

else

{

add\_inavalid:

printf("\n\n\n\t\t\tEnter 1 to go to the main menu and 0 to exit:");

ans=getValidInteger();

system("cls");

if(ans==1)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_inavalid;

}

}

}

void view\_deact()

{

system("cls");

FILE \*view;

int ans=0,i=1;

view=fopen("deactivate.dat", "r");

int count=0;

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

for(int j=0;j<=15;j++)

{

Sleep(100);

printf(".");

}

system("cls");

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tDeactivated Accounts\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\n\t\tSR.NO.\tACC\_NO.\t\tNAME\t\tADDRESS\t\tDOB\t\tAGE\tPHONE");

while(fscanf(view,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

printf("\n\n\t\t%d\t%d\t %s %s\t\t%s\t\t%d/%d/%d\t%d \t%.0lf ",i++,add.acc\_no,add.name,add.lname,add.address,add.dob.day,add.dob.month,add.dob.year,add.age,add.phone);

count++;

}

fclose(view);

if(count==0)

{

system("cls");

printf("\n\t\t\tNO RECORDS!!\n");

}

add\_invalid:

printf("\n\n\n\t\t\tEnter 1 to go to the main menu and 0 to exit:");

ans=getValidInteger();

system("cls");

if (ans==1)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

void view\_details()

{

system("cls");

FILE \*ptr;

int ans=0,choice,test=0,flag=0,flag1=0;

printf("\n\n\t\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tCustomer's Details\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\t\t\tEnter the account no. of the customer whose details you want to see: ");

check.acc\_no=getValidInteger();

ptr=fopen("record.dat","r");

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

for(int j=0;j<=15;j++)

{

Sleep(100);

printf(".");

}

system("cls");

printf("\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\t\tCustomer's Details\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

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

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

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

printf("\n\n");

while(fscanf(ptr,"%d %s %s %d/%d/%d %d %s %lf %f %d/%d/%d ",&add.acc\_no,add.name,add.lname,&add.dob.day,&add.dob.month,&add.dob.year,&add.age,add.address,&add.phone,&add.amt,&add.join.day,&add.join.month,&add.join.year)!=EOF)

{

if(add.acc\_no==check.acc\_no)

{

printf("\n\t\t\tAccount No.: %d\n\t\t\tName: %s %s\n\t\t\tAddress: %s\n\t\t\tDate Of Birth: %d/%d/%d\n\t\t\tAge: %d \n\t\t\tPhone No.: %.0lf\n\t\t\tBalance: %.2f ",add.acc\_no,add.name,add.lname,add.address,add.dob.day,add.dob.month,add.dob.year,add.age,add.phone,add.amt);

test=1;

}

}

fclose(ptr);

if(test!=1)

{

system("cls");

printf("\n\t\t\tRECORD NOT FOUND!!\a\a\a");

add\_invalid:

printf("\n\n\t\t\tPress 1 to try again");

printf("\n\t\t\tPress 2 to return to main menu");

printf("\n\t\t\tPress 3 to exit");

printf("\n\t\t\tEnter your choice: ");

ans=getValidInteger();

system("cls");

if (ans==1)

view\_details();

else if(ans==2)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_invalid;

}

}

else

{

add\_inavalid:

printf("\n\n\n\t\t\tEnter 1 to go to the main menu and 0 to exit:");

ans=getValidInteger();

system("cls");

if(ans==1)

menu();

else if(ans==0)

close1();

else

{

printf("\n\t\t\tInvalid!\a");

goto add\_inavalid;

}

}

}

**Any challenges faced**

* 1. Challenge was solving and displaying date on which account was created.
  2. Type of account (Saving and Current),monthly interest calculation for saving account and privileges given to that of the current account